

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN **WICC 2002**

*16 y 17 de mayo de 2002*



**Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur – Av. Alem 1253  
Bahía Blanca – Pcia. de Buenos Aires  
República Argentina**

---

*Organizado por la Red de Universidades Nacionales con carreras en Informática  
16 y 17 de Mayo de 2002*



## IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Mayo de 2002

## INDICE

### Aspectos Teóricos de Inteligencia Artificial (ATIA)

---

<b>Métodos de razonamiento aproximado en investigación socio económica (MERAIS)</b> <i>BIOUL, Gery - XODO, Daniel- NIGRO, Oscar - ZELASCO, José - TRIPODI, Gustavo - ILLESCAS, Gustavo</i>	4
<b>Semántica Declarativa Trivaluada para la Programación en Lógica Rebatible</b> <i>CECCHI, Laura - SIMARI, Guillermo</i>	9
<b>Introducing Probabilistic Reasoning in Defeasible Argumentation Using Labeled Deductive Systems</b> <i>CHESÑEVAR, Carlos - SIMARI, Guillermo</i>	14
<b>ILP en Aprendizaje Grupal</b> <i>DELLADIO FEDI, Telma</i>	19
<b>Embedding Abduction in Monmonotomic Theories</b> <i>DELRIEUX, Claudio</i>	24
<b>Modelos Formales del Razonamiento Científico</b> <i>DELRIEUX, Claudio</i>	29
<b>On the Logic of Theory Change: Incision Functions from Selection Functions</b> <i>FALAPPA, Marcelo A. - SIMARI, Guillermo R.</i>	34
<b>Planificación con Aprendizaje de Reglas Rebatibles para elección de Acciones utilizando Argumentación</b> <i>GARCÍA, Diego - GARCÍA, Alejandro</i>	38
<b>Una Arquitectura Multiagentes para Argumentación Distribuida</b> <i>GARCÍA, Diego R. - GARCÍA, Alejandro J. - SIMARI, Guillermo R.</i>	43
<b>Social interaction and argumentation in MAS</b> <i>MARTÍNEZ, Diego - GARCÍA, Alejandro</i>	48
<b>Reelaboración de Planes en Agentes Inteligentes. Operadores Básicos</b> <i>PARRA, Gerardo - SIMARI, Guillermo</i>	53
<b>La negociación como un proceso de búsqueda usando dialéctica</b> <i>RUEDA, Sonia - SIMARI, Guillermo</i>	56
<b>Logical Properties in Defeasible Logic Programming - a preliminary report -</b> <i>STANKEVICIUS, A. G. CAPOBIANCO, M. CHESÑEVAR, C. I.</i>	61

### Computación Gráfica. Visualización

---

<b>Procesamiento Digital de Imágenes Utilizando PDICalc</b> <i>ARLENGHI, L. - VITALE, C. - DELRIEUX, C. - RAMOSCELLI, G.</i>	67
<b>Estructuras Espaciales para la Representación de Mapas</b> <i>BACIC, Leandro - DELLADIO, Mauricio - MIGNANI, Mariano</i>	72
<b>Modelamiento de Objetos 3D</b> <i>CASTRO, Silvia - MARTIG, Sergio - CASTRO, Liliana - BOSCARDÍN, Liliana - SALGADO, Diana</i>	77
<b>Reconocimiento de Rostros</b> <i>CHICHIZOLA, Franco - DE GIUSTI, Armando - NAIOUF, Marcelo</i>	80
<b>Visualización de Información de Bases de Datos Heterogéneas</b> <i>COBO, Ma. Laura - MARTIG, Sergio - CASTRO, Silvia - FILLOTRANI, Pablo</i>	84
<b>Visualización Interactiva de Diagramas de Fase</b> <i>DELRIEUX, Claudio - DOMINGUEZ, Julián - REPETTO, Andrés</i>	89
<b>Interfaces en la Visualización de Información</b> <i>DI LUCA, Sandra - MARTIG, Sergio - CASTRO, Silvia - TRUTNER, Guillermo - VITTURINI, Mercedes</i>	94



## IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Mayo de 2002

<b>Visualización de Reglas de Asociación. Un acercamiento comparativo</b> <i>FILOCAMO, G. - GRANDINETTI, W. - CASTRO, S. - CHESÑEVAR, C</i>	98
<b>Generación de Objetos Flexibles 3D</b> <i>GONZÁLEZ, Adela</i>	103
<b>Reconocimiento de Formas en Imágenes Médicas</b> <i>GÓNZALEZ, Adela - GUERRERO, Roberto - ZAVALA, Alfredo</i>	107
<b>Un Ambiente Gráfico para Testear Estrategias Evolutivas en el Problema Inverso de las IFS</b> <i>IVANISSEVICH, M. Laura - MILLADO, Paula - SIERPE, Luis - DELRIEUX, Claudio</i>	111
<b>Sistemas Híbridos para el Reconocimiento de Patrones</b> <i>KATZ, Román - DELRIEUX, Claudio</i>	116
<b>Visualización de Datos Multivariados</b> <i>LÓPEZ, Lidia M.</i>	121
<b>Interacción en la Visualización de Información</b> <i>MARTIG, Sergio CASTRO, Silvia - DI LUCA, Sandra</i>	126
<b>Implementación de un Ambiente para el Estudio Experimental de la Dinámica de Poblaciones</b> <i>PADÍN, Mirta - DELRIEUX, Claudio - DOMINGUEZ, Julián</i>	130
<b>Estimación y Compensación de Movimientos en Video Codificado</b> <i>RUSSO, Claudia - RAMÓN, Hugo</i>	135
<b>Clasificación de imágenes digitales utilizando patrones N-dimensionales</b> <i>SANZ, Cecilia - DE GIUSTI, Armando</i>	140
<b>Simulación de Medios Artísticos Tradicionales</b> <i>ZAVALA, Alfredo</i>	145

### Tecnología Informática Aplicada en la Educación

<b>Una propuesta para Ambientes Colaborativos Inteligentes</b> <i>AGUIRRE, G. CHIARANI, M. GARCÍA, B. LEGUIZAMÓN, G. LUCERO, M. MANZUR, L. PIANUCCI, e I.</i>	150
<b>Experiencia Interdisciplinaria En La Practica Docente Del Profesora En Computación</b> <i>CAMILETTI, P. PIZARRO, R. A.</i>	155
<b>Specifying Quality Characteristics and Attributes for E-Learning Sites</b> <i>COVELLA, Guillermo OLSINA, Luis</i>	160
<b>Tecnología informática en Software Educativo: Herramientas para el diseño, administración y evaluación de cursos no presenciales</b> <i>DE GIUSTI, A. SANZ, C. MADOZ, C. GORGA, G. CHAMPREDONDE, R. PESADO, P. PASINI, A. BARBIERI, A. RODRIGUEZ, I. FEIERHERD, G. DEPETRIS, B.</i>	165
<b>www.lea.com.ar: "Un portal de accesibilidad para disminuidos visuales y ciegos"</b> <i>DÍAZ, Javier - HARARI, Ivana - FERREYRA, José</i>	168
<b>Hacia una experiencia de aprendizaje inicial de Ciencias de la Computación basada en la web</b> <i>FEIERHERD, Guillermo DEPETRIS, Beatriz</i>	173
<b>Aprendizaje Basado En Problemas Como Apoyo A Las Clases Presenciales A Través De Software Para Groupware</b> <i>FIGUEROA, Nancy - SALGUIERO, Fernando - CÁNEPA, Sabrina - LAGE, Fernando CATALDI, Zulma</i>	177
<b>Incorporación de las nuevas tecnologías de la Comunicación y de la Información al quehacer docente universitario</b> <i>GODOY GUGLIELMONE, Maria V. - MARIÑO, Sonia I. - DAPOZO, Gladys N. PETRIS, Raquel H.</i>	182
<b>Nuevos Entornos de aprendizaje. Los Ambientes Colaborativos Inteligentes</b> <i>LEGUIZAMÓN, Guillermo LUCERO, Margarita AGUIRRE, Guillermo CHIARANI, Marcela PIANUCCI, Irma MANZUR, Lilian GARCÍA, Berta</i>	187
<b>Entorno Web para la enseñanza del Método de Programación Lineal</b> <i>LÓPEZ, Ma. Victoria GOLOBISKY, Ma. Fernanda MARIÑO, Sonia I.</i>	192
<b>Aplicación del Método de Montecarlo para el cálculo de integrales definidas</b> <i>LÓPEZ, Ma. Victoria MARIÑO, Sonia I.</i>	197



## IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Mayo de 2002

<b>Un paquete de Mathematica para el aprendizaje de métodos de muestras artificiales de variables aleatorias no uniformes</b> <i>MARIÑO, Sonia I.</i>	202
<b>Desarrollo de programas educativos para el modelado y la simulación de sistemas. Algunos estudios de casos.</b> <i>MARIÑO, Sonia I. LÓPEZ, Ma. Victoria</i>	205
<b>Herramientas y estrategias visuales para sistemas de aprendizaje</b> <i>MARTIG, Sergio SEÑAS, Perla</i>	210
<b>La visualización de Algoritmos como Recurso para la Enseñanza de Programación</b> <i>MORONI, Norma SEÑAS, Perla</i>	213
<b>A System For Control Of Use Of Didactic Material At The Distance</b> <i>ISERHARDT RITZEL, Marcelo VALDENI de LIMA, José</i>	218
<b>Construcción De Tutoriales Basados En Componentes Reusables</b> <i>ROSANIGO, Zulema B PAUR, Beatriz BRAMATTI, Pedro</i>	224
<b>Generalización Del Modelo De Hipermedia Para La Construcción De Redes Conceptuales</b> <i>ROSANIGO, Zulema B PAUR, Beatriz BRAMATTI, Pedro</i>	228
<b>Informática Y Educacion</b> <i>SEÑAS, Perla</i>	233
<b>Mapas Conceptuales: una herramienta para el aprendizaje de Estructuras de Datos</b> <i>UVIÑA, Patricia - BERTOLAMI, Mabel - CENTENO, Ma Elena - ORIANA, Gabriela</i>	237
<b>Mapas Conceptuales Hipermediales: Su Aplicación En Cursos De Lectura Compresiva</b> <i>VITTURINI, Mercedes BENEDETTI, Laura SEÑAS, Perla</i>	240

### Ingeniería de Software y Bases de Datos

<b>INCA RUCA QUIMN: Proyecto de automatización de oficinas</b> <i>ACOSTA, Nelsson SIMONELLI, Daniel TOSINI, Marcelo</i>	246
<b>Interoperabilidad de Componentes Software mediante Lenguajes de Coordinación</b> <i>AMARO, Silvia MARTINEZ CAROD, Nardina PIEMENTEL, Ernesto</i>	250
<b>Tool Support for Verifying Applications using object-Oriented Patterns</b> <i>ARANDA, Gabriela FLORES, Andrés BUCCELLA, Agustina REYNOSO, Luis</i>	253
<b>Métricas de Performance en Administración de BDD en redes LAN y WAN</b> <i>BERTONE, Rodolfo DE GIUSTI, Armando</i>	258
<b>Methods for Measurement-Based COTS Assessments and Selection</b> <i>CECHICH, Alejandra PIATTINI, Mario</i>	262
<b>An extension to EMTPL</b> <i>COBO, Ma. Laura FALAPPA, Marcelo</i>	267
<b>Sistemas de Software Distribuido. Aplicaciones</b> <i>DE GIUSTI, A. PESADO, P. BERTONE, R. BORACCHIA, M. THOMAS, P. MADDOZ, C.</i>	272
<b>Formal Specifications in Component-Based Development</b> <i>ESTÉVEZ, Elsa FILLOTRANI, Pablo</i>	276
<b>SOL: Un lenguaje para el programador</b> <i>FONTAO, Rafael DELRIEUX, Claudio KALOCAI, Guillermo GOÑI, Gustavo</i>	280
<b>A Framework for Defining and Checking Constraints in Requirement Gathering and Definition</b> <i>GÓMEZ, Rodolfo FILLOTRANI, Pablo</i>	286
<b>Búsquedas en Espacios Métricos</b> <i>HERRERA, Norma - REYES, Nora - BAEZA-YATES, Ricardo - NAVARRO, Gonzalo CHÁVEZ, Edgar</i>	290
<b>Bases de Datos distribuidas: Agentes Móviles en problemas de Replicación, migración y recuperación de fallas</b> <i>MIATON, Ivana PESADO, Patricia BERTONE, Rodolfo DE GIUSTI, Armando</i>	298
<b>Project Line: Formally Specifying the Domain of Balanced Scorecard</b> <i>MONTEJANO, G. UZAL, R. RIESCO, D. DASSO, A. FUNES, A.</i>	303



## IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Mayo de 2002

<b>Spectra in Taxonomic Evidence in Databases III, Application in Celestial Bodies. Asteroids Families</b>	
<i>PLASTINO, Angel ORELLANA, Rosa PERICHINSKY, Gregorio</i>	308
<b>Integración de los meta-modelos de Workflow y UML</b>	
<i>RIESCO, Daniel ACOSTA, Edgardo UVA, Marcelo GRANDO, Adela</i>	313
<b>Extensiones al Meta-modelo UML Desarrollando Nuevos Esterotipos</b>	
<i>RIESCO, Daniel GRUMELLI, Alicia MACCIÓ, Alcides MARTELLOTO, Paola</i>	317
<b>Modelling of Processes and Formal Methods</b>	
<i>Riesco, D. MONTEJANO, G. UZAL, R.</i>	321
<b>A Systematic Approach to Generate Test Cases based on Combinations of Information</b>	
<i>SANCHEZ, Marisa A. FELDER, Miguel A.</i>	326
<b>Integrated Environment Of Systems Automated Engineering</b>	
<i>SERVETTO, Arturo GARCÍA MARTÍNEZ, Ramón PERICHINSKY, Gregorio</i>	331
<b>Proceso y Métodos de Ingeniería Liviana de Sistemas</b>	
<i>SERVETTO, Arturo GARCÍA MARTÍNEZ, Ramón PERICHINSKY, Gregorio</i>	335
<b>Estudio de Técnicas apropiadas para modelar aplicaciones de Hidroinformática en el contexto de los Sistemas de Información Ambiental</b>	
<i>URCIUOLO, Adriana ITURRASPE, Rodolfo SANDOVAL, Sandra PARSON, Ariel</i>	339
<b>El Ciclo De Vida De Los Sistemas: Los Conceptos Y Los Algoritmos De Evolución Simple Y Ciclica</b>	
<i>ZUBENKO, Yurity D. LAGE, Fernando J. CATALDI, Zulma</i>	343
<b>Controles Semánticos en Modelamiento Orientado a Objetos</b>	
<i>KAHNERT, Susana FILLOTRANI, Pablo</i>	349

### Procesamiento Concurrente, Paralelo y Distribuido

<b>Construcción de un sistemas GNU para plataforma SPARC de 64 bits</b>	
<i>CURTI, Hugo</i>	355
<b>Modelo de predicción de performance en sistemas paralelos</b>	
<i>DE GIUSTI, Laura - NAIOUF, Marcelo - DE GIUSTI, Armando</i>	360
<b>Procesamiento Paralelo y Distribuido en tratamiento masivo de datos</b>	
<i>DE GIUSTI, Armando - NAIOUF, Marcelo - SANZ, Cecilia - TINETTI, Fernando DE GIUSTI, Laura - BERTONE, Rodolfo</i>	364
<b>Una solución estática al problema de Buffer Overflow</b>	
<i>ECHAIZ, Javier GARCÍA, Rafael ARDENGHI, Jorge</i>	368
<b>Un Modelo de Programación Paralelo para Memoria Distribuida</b>	
<i>FERNÁNDEZ, Jacqueline - FUENTES, Mónica - PICCOLI, Fabiana - PRINTISTA, Marcela</i>	373
<b>Consistencia De Memoria En Sistemas Dvsm</b>	
<i>GARCÍA, Rafael B. ECHAIZ, Javier ARDENGHI, Jorge R.</i>	377
<b>Perception and Enterprise Communication Networks to Improve the Requirements Elicitation Process</b>	
<i>LUZURIAGA, Juan M. MARTÍNEZ, Rodolfo CECHICH, Alejandra</i>	382
<b>Escalabilidad y Balance de Carga en Sistemas Paralelos</b>	
<i>NAIOUF, Marcelo DE GIUSTI, Armando</i>	387
<b>Distributed and Parallel Processing for the Embgrid Project</b>	
<i>SANZ, Cecilia TINETTI, Fernando RUSSO, Claudia DENHAM, Mónica DE GIUSTI, Armando GRAU, Oscar</i>	391
<b>Evaluación del rendimiento y tiempo de transferencia de la información en redes de datos mediante modelos analíticos y técnicas de simulación</b>	
<i>SEGRE, Oscar R. - FUSARIO, Ruben Jorge - SANTA MARIA, Cristobal</i>	395
<b>Cómputo Científico: Algoritmos Paralelos para Aprovechar Redes Locales Instaladas</b>	
<i>TINETTI, Fernando BARBIERI, Andrés DENHAM, Mónica</i>	399
<b>Procesamiento Paralelo Distribuido Heterogéneo En Aplicaciones Estructurales Y Numéricas</b>	
<i>VAZQUEZ, Gustavo BRIGNOLE, Nélida</i>	402



## IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Mayo de 2002

<b>Integración de bloques IP en diseños SOC</b>	
<i>VILAGARCÍA WANZA Horacio</i> <i>BRIA, Oscar</i>	406
<b>Modelando la Performance de Aplicaciones en un Clusters de PC's</b>	
<i>VILLALOBOS, M. FLORES, S. PICCOLI, F. PRINTISTA, M.</i>	409

### Sistemas Inteligentes

---

<b>Agentes y aprendizaje de máquina en optimización, control y entornos de enseñanza</b>	
<i>AGUIRRE, Guillermo ERRECALDE, Marcelo LEGUIZAMÓN, Guillermo</i>	414
<b>Evolución de Controladores Basados en Redes Neuronales</b>	
<i>APOLLONI, Javier KAVKA, Carlos ROGGERO, Patricia</i>	420
<b>Optimización Evolutiva En Ambientes Dinamicos</b>	
<i>ARAGÓN, V. ESQUIVEL, S.</i>	425
<b>Aplicación de una técnica de visión a la representación de una superficie de terreno reconstruido a partir de contornos planares</b>	
<i>CAGNINA, Leticia GUERRERO, Roberto</i>	429
<b>Introduciendo Mejoras En Los Algoritmos Evolutivos</b>	
<i>ESQUIVEL, S. GALLARD, R.</i>	434
<b>Enfoques Heuristicos Para Problemas De Scheduling Estáticos Y Dinámicos</b>	
<i>ESQUIVEL, S. GALLARD, R. GATICA, C. FERRERO, S. LEGUIZAMÓN, G. ZUPPA, F.</i>	438
<b>Control, Planning, Y Navegacion Basados En Inteligencia Computacional</b>	
<i>ESQUIVEL, Susana KAVKA, Carlos LEGUIZAMÓN, Guillermo</i>	443
<b>Evolutionary Algorithms To Solve The School Timetabling Problem</b>	
<i>FERNÁNDEZ, N. ALFONSO, H. GALLARD, R.</i>	448
<b>Un enfoque evolutivo como método alternativo para la técnica de halftoning</b>	
<i>GUERRERO, Roberto LEGUIZAMÓN, G. VIANO, Huggo</i>	452
<b>Redes Neuronales aplicadas al Reconocimiento de Partrones</b>	
<i>LANZARINI, Laura DE GIUSTI, Armando</i>	456
<b>Hybrid Evolutionary Algorithms To Solve Scheduling Problems</b>	
<i>MINETTI, G. SALTO, C. BERMÚDEZ, C. FERNÁNDEZ, N. ALFONSO, H. GALLARD, R.</i>	460
<b>Webexpert - Tool For Knowledge Acquisition By Web</b>	
<i>NASCIMENTO, Raphael L. da ROCHA FERNANDEZ, Anita Mag.</i>	463
<b>Inserting Problem-Specific Knowledge In Multirecombined Evolutionary Algorithms</b>	
<i>PANDOLFI, D. DE SAN PEDRO, M. VILLAGRA, A. VILANOVA, G. GALLARD, R.</i>	468
<b>Performance Evaluation Of Selection Methods To Solve The Job Shop Scheduling Problem</b>	
<i>STARK, N. SLATO, C. ALFONSO, H. GALLARD, R.</i>	473
<b>Multirecombining Random And Seed Immigrants In Evolutionary Algorithms To Face The Flow The Flow Shop Scheduling Problem</b>	
<i>VILANOVA, G. VILLAGRA, A. PANDOLFI, D. DE SAN PEDRO, M. GALLARD, R.</i>	478

### Sistemas de Tiempo Real

---

<b>Fuzzy Logic Controllers on Chip</b>	
<i>ACOSTA, Nelson SIMONELLI, Daniel</i>	484
<b>Travelling Source Estimation using Spatio-temporal Data</b>	
<i>BRIA, Oscar N.</i>	489
<b>Diseño Y Desarrollo De Un Sistema De Soporte De Decisión Para El Análisis De Instrumentación De Plantas Industriales</b>	
<i>PONZONI, Ignacio VAZQUEZ, Gustavo E. CARBALLIDO, Jessica BRIGNOLE, Nélida</i>	495
<b>Sistemas en Tiempo Real</b>	
<i>RAMÓN, Hugo ROMERO, Fernando</i>	498



## IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Mayo de 2002

### Teoría de Computación

---

#### Herramientas de generación de procesadores de lenguajes para código móvil seguro

*AGUIRRE, J. ARROYO, M. FLORIO, N. FELIPPA, J. GOMEZ, G. BAVERA, F.  
CAYMES SCUTARI, P. NORDIO, D.*

503

#### La Geometría Computacional a nuestro alrededor

*GAGLIARDI, Edilma - TARANILLA, Ma. Teresa -  
BERÓN, Mario - PEÑALVER, Gregorio*

508

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

## WICC 2002

*16 y 17 de mayo de 2002*

### AREAS

Aspectos teóricos de Inteligencia Artificial

Computación Gráfica – Visualización

Tecnología Informática Aplicada en la Educación

Ingeniería de Software y Base de Datos

Procesamiento Concurrente, Paralelo y distribuido

Sistemas Inteligentes. Metaheurísticas

Teoría de Computación

Sistemas de Tiempo Real. Procesamiento de Señales.



---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

## WICC 2002

*16 y 17 de mayo de 2002*

### Aspectos teóricos de Inteligencia Artificial



**Métodos de razonamiento aproximado en investigación socioeconómica (MERAIS)****Campo del conocimiento y disciplina**

Descubrimiento de Conocimiento en Base de Datos (Data Mining Application).

**Personal participante:**

Director: Gery Bioul. [gbioul@exa.unicen.edu.ar](mailto:gbioul@exa.unicen.edu.ar)

Daniel Xodo. [dxodo@exa.unicen.edu.ar](mailto:dxodo@exa.unicen.edu.ar)

Oscar Nigro. [onigro@exa.unicen.edu.ar](mailto:onigro@exa.unicen.edu.ar)

José Zelasco. [jfz@fecic.edu.ar](mailto:jfz@fecic.edu.ar)

Gustavo Tripodi. [gtripodi@exa.unicen.edu.ar](mailto:gtripodi@exa.unicen.edu.ar)

Gustavo Illescas. [gillescas@fch.unicen.edu.ar](mailto:gillescas@fch.unicen.edu.ar)

Facultad de Ciencias Exactas- Universidad Nacional del Centro de la Provincia de Buenos Aires.  
Grupo de Investigación en Informática de Gestión

Teléfono

: +54 2293 432466

Dirección postal

Campus Universitarios, Paraje Arroyo Seco, (7000) Tandil, ARGENTINA

**Resumen técnico del proyecto**

El proyecto propone llevar a cabo investigaciones sobre la influencia e impacto de la utilización de redes neuronales y sistemas borrosos, según los siguientes pasos:

1. Características generales de las redes neuronales<sup>1</sup> y sistemas borrosos<sup>2</sup>,
2. Comparación entre redes neuronales y la estadística tradicional
3. Impacto en la aplicación de estas tecnologías emergentes

- a) en el análisis sociológico y de mercado,
- b) sistemas de información con soporte tecnológico para la ayuda en toma de decisiones,
- c) Inteligencia comercial y de negocios (Business Intelligence), y
- d) Control de gestión y herramientas informáticas.

La elección entre los diferentes sistemas de ayuda a la toma de decisiones depende entre otros factores del tipo de tarea a realizar, [Laudon,1988].

1a) Tipo de decisión

*Se han clasificado las decisiones en estructuradas y no estructuradas [Simon;1960]: las decisiones estructuradas son repetitivas, rutinarias y existe un procedimiento definido para abordarlas; por el contrario, en las decisiones no estructuradas el decisor debe proporcionar juicios y aportar su propia evaluación.*

1b) Nivel al que se toman las decisiones

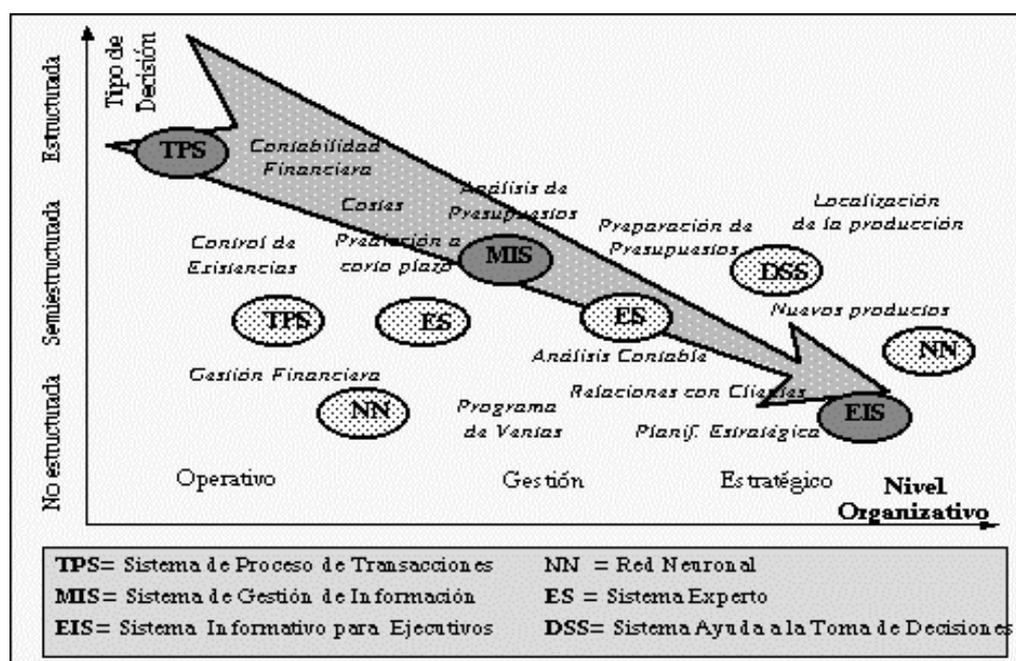
<sup>1</sup> Una definición de red neuronal hace uso del concepto matemático de grafo, objeto consistente en un conjunto de nodos (o vértices), más un conjunto de conexiones (o links) establecidas entre ellos. En este caso, el grafo describe la arquitectura del sistema y proporciona canales por los que puede discurrir su dinámica. Hay diferentes tipos de grafos, por ejemplo, grafos dirigidos (directed) y no dirigidos (undirected). En el primer tipo, las conexiones tienen asignado un sentido, mientras que en el otro son bidireccionales. Puede hablarse también de grafos densos (cuando casi todos los nodos están conectados con casi todos) y de grafos dispersos (cuando son pocas las conexiones entre los nodos). Un grafo puede componerse de diferentes tipos de nodos y diferentes tipos de conexiones.

<sup>2</sup> Los sistemas basados en lógica borrosas combinan unas variables de entrada (definidas en términos de conjuntos borrosos) por medio de grupos de reglas que producen uno o varios valores de salida. Este tipo de enfoque resulta especialmente interesante para problemas no lineales o no bien definidos. En efecto los sistemas borrosos permiten modelar cualquier proceso no lineal, y aprender de los datos haciendo uso de determinados algoritmos de aprendizaje. Además al contrario de los sistemas de redes neuronales, los basados en lógica borrosa permiten utilizar fácilmente el conocimiento de los expertos en un tema, bien directamente, bien como punto de partida para una optimización automática, al formalizar el conocimiento a veces ambiguo de un experto (o el sentido común)...Redes Neuronales y sistemas borrosos, B.Martín del Brío, A. Sanz Molina, Editorial Ra-Ma 1997.

También el nivel al que se toman las decisiones afecta la elección del tipo de sistema más apropiado. [Gorry;1971] han propuesto un esquema que relaciona el tipo de decisión (estructurada, semiestructurada y no estructurada) y el nivel organizacional (control operativo, de gestión y estratégico) con la herramienta a utilizar. Lógicamente en el nivel operativo dominan las decisiones estructuradas, en el nivel de gestión las semiestructuradas y en el estratégico las no estructuradas, formando una diagonal.

En la figura 1 hemos actualizado este esquema incorporando las herramientas informáticas más novedosas, e incluso anticipando lo que puede ser un futuro próximo.

Figura 1. Las herramientas a utilizar según el tipo de decisión y el nivel organizativo



### 1) Nivel operativo y decisiones estructuradas

El gráfico 1, muestra que en el nivel operativo y toma de decisiones estructurada dominan los sistemas informatizados convencionales. Entre ellos los programas de contabilidad financiera y de costos, los de elaboración de nóminas, y los que en general realizan tareas mecánicas. Son los llamados Sistemas de Proceso de Transacciones (TPS) basados en programación algorítmica convencional.

### 2) Nivel operativo y decisiones no estructuradas

El nivel operativo, pero con decisiones semiestructuradas, todavía está dominado por los programas convencionales, en lo que habría que incluir programas de control de tesorería, control de existencias y también las hojas de cálculo y sistemas gestores de bases de datos. Conforme las decisiones son menos estructuradas, empiezan a ser aplicables sistemas expertos y sistemas de ayuda a la toma de decisiones. Así, para la gestión financiera más básica, en la que además de realizar cálculos mecánicos: tipos de interés efectivo, cuotas de amortización de préstamos, etc., también hay que tomar decisiones que manejan información incompleta o precisan incorporar el conocimiento de un especialista humano. Finalmente, en el nivel de de toma de decisiones menos estructurada pueden incorporarse modelos neuronales: por ejemplo, asesorando en la concesión de las tarjetas de crédito de unos grandes almacenes o a qué clientes se les envía catálogos por correo.

### 3) Nivel de gestión y decisiones estructuradas

Avanzando en el eje de las x (nivel organizativo), las decisiones se hacen más complejas. Si las decisiones son estructuradas, como en el análisis de presupuestos y control, contabilidad

analítica, análisis contable, etc., dominan los programas convencionales, quizá incorporando algún módulo experto. Este tipo de programas se denominan Sistemas de Gestión de Información o Management Information Systems (MIS).

#### 4) Nivel de gestión y decisiones no estructuradas

A medida que las decisiones se hacen menos estructuradas se hacen más necesarios los sistemas expertos. En temas puntuales como la elaboración de presupuestos, la predicción de variables financieras como el beneficio, el cash-flow, podrían incorporarse modelos neuronales. Nótese como en este esquema descrito hemos situado el análisis de la información contable en las decisiones semiestructuradas. No hay una teoría general que pueda ser aplicada paso a paso pero tampoco encaja en las decisiones completamente intuitivas o no estructuradas.

#### 5) Nivel estratégico

En el nivel estratégico son barridos los programas convencionales manteniéndose únicamente las hojas de cálculo, por su capacidad de simulación. Los llamados Executive Information Systems (EIS), o Sistemas Informativos para Ejecutivos dominan las decisiones menos estructuradas, con tareas como la planificación estratégica y de contabilidad directiva. Las redes neuronales pueden cubrir un hueco importante en las decisiones no estructuradas, debido a esa capacidad de encontrar relaciones complejas entre los patrones de entrada.

## DESCRIPCION DEL PROYECTO

### Introducción

Las redes neuronales y los sistemas borrosos ofrecen nuevas alternativas, para estudiar los fenómenos cognitivos, pudiendo contribuir de una forma importante a su comprensión. Son capaces de descubrir relaciones entrada salida (o rasgos característicos) en función de datos empíricos, merced a su capacidad de aprendizaje a partir de ejemplos.

La Inteligencia Artificial opera de arriba hacia abajo, con reglas, conceptos y cálculos secuenciales, tal y como al parecer trabaja nuestra mitad izquierda del cerebro, mientras que las redes neuronales artificiales, operan de abajo a arriba (enfoque emergente), interpretando de una manera intuitiva y paralela las imágenes, sonidos y otros tipos de estímulos que llegan de forma masiva desde el exterior, como el lado derecho del cerebro.

No debe pensarse en ambas como tecnologías excluyentes, sino, complementarias. Los mejores resultados se obtendrán aplicando cada técnica al aspecto del problema donde resulte más conveniente. Precisamente con esta filosofía han nacido las redes expertas, combinación de ambas.

Los sistemas borrosos resultan un excelente complemento a las técnicas (redes neuronales y sistemas expertos), existiendo en la actualidad la clara tendencia de incorporar en un mismo sistema redes neuronales, lógica borrosa y sistemas expertos, en todas sus posibles combinaciones.

Las redes neuronales quizá tengan mucho más aspectos en común con la estadística que con la Inteligencia Artificial, de hecho, las redes neuronales artificiales han sido descritas en alguna ocasión como técnicas de ajuste estadístico inspiradas en la biología [Brío,1996].

Pueden beneficiarse de la estadística en numerosos aspectos, por ejemplo, el empleo de técnicas estadísticas para el análisis de la relevancia de las variables de entrada, y su utilización en una inicialización de los pesos relativos. En general resultan relativamente fáciles de emplear, y la interpretación de sus resultados resulta asequible a usuarios profanos.

La principal característica de las estadísticas tradicionales parte del hecho que existe un modelo conceptual previo de los datos, un supuesto teórico. No se refiere al campo de estudio que esto es irremplazable, sino a la distribución de los valores de las variables involucradas en el proceso de análisis estadístico. El enfoque con estadísticas tradicionales, tiene las características de un enfoque descendente.

Los modelos neuronales normalmente no parten de restricciones respecto de los datos de partida (tipo de dependencia funcional), ni suelen imponer presupuestos (como distribución gaussiana u otras).

Teniendo en cuenta que las estadísticas tradicionales en son muy buenas para el tipo de relaciones lineales, las redes neuronales lo son para relaciones no lineales. Ellos parecen adaptarse mejor en este sentido al análisis y visualización de los indicadores financieros y económicos, la evaluación de la gestión empresarial, los indicadores de satisfacción de clientes, productividad, personal, logística, costos por actividades, calidad, gestión de venta, ciclo de vida de los productos, velocidad de procesos, etc., son ejemplos donde raramente las relaciones son lineales.

Uno de los inconvenientes en la utilización de este tipo de redes consiste en que luego de todo un desarrollo no existe una ecuación que represente el proceso investigado y por consiguiente la interpretación del mismo.

Otra de las características diferentes que ofrece las redes neuronales es la posibilidad de un proceso de aprendizaje a partir de un entrenamiento dado y luego de este proceso, su reutilización para nuevos desarrollos y datos.

### Objetivos Generales

1.-Aplicación de distintos tipos de redes, Multicapa Perceptron, Función Radial Básica, Redes de Kohonen, y otras. Proceder a su interpretación comparándolas con la aplicación de las estadísticas tradicionales, su relación con cada aplicación y modo de interpretarlas, pronósticos de series temporales y capacidad de predicción con base de datos de gestión.

- a) en base de datos sociológicas y de mercado,
- b) para identificar y solucionar los problemas de información en la toma de decisiones
- c) para construir nuevos indicadores a los efectos de controlar y medir el desempeño de la empresa, y su relación con los clientes CRM.<sup>3</sup>

2.-Verificar la aplicación de redes neuronales en base de datos, originadas de trabajos cualitativos, y su comparación con los enfoques tradicionales. Tratamientos de Huber, Ragin, y otros. En este ítem conviene verificar la alternativa de enfoque con sistemas borrosos y su aplicación específica a tratamiento de valores provenientes de enfoques cualitativos.

- a) para utilizar las distintas herramientas informáticas disponibles en el mercado para la toma de decisiones y,
- b) Para construir sistema de apoyo en la toma de decisiones.<sup>4</sup>

---

<sup>3</sup> CRM (Customer Relationship Management) es el proceso que posibilita reorientar los mecanismos estratégicos empresariales desde la visión centrada en el producto hacia una perspectiva referida a la figura del cliente y su relación con la empresa, proceso por el cual la empresa maximiza la información de la que dispone acerca de sus clientes (información que obtiene fruto de la interacción a través de los distintos puntos de contacto empresa-cliente: servicios de atención, reclamaciones, fuerza de ventas, etc.) con el fin de incrementar su conocimiento acerca de ellos y construir a partir de tal conocimiento relaciones altamente rentables y duraderas con aquellos segmentos del censo de clientes que mayor rentabilidad puedan proporcionar a la empresa.

El concepto de CRM comprende la metodología, disciplina y tecnología (eminentemente materializada bajo la forma de sistemas software) que tiene por objeto automatizar y mejorar los procesos de negocio asociados a la gestión de la relación de la empresa con el cliente, principalmente en las áreas de venta, marketing, servicios de atención al cliente y soporte, con el fin último de incrementar los beneficios de la empresa mediante la optimización, personalización y diferenciación de dicha relación con el cliente.

La solución CRM combina una adquisición de información de los clientes de la empresa con la aplicación de una serie de tecnologías para la gestión de tal información y su conversión en conocimiento de negocio (datawarehousing, análisis estadístico, técnicas OLAP, DSS, EIS, minería de datos, etc.) Ello proporciona, al mismo tiempo, una plataforma común para la comunicación e interacción con el cliente que minimice las limitaciones en la integración asociadas a la tradicional estructura organizativa que distingue las visiones de los departamentos de ventas, marketing y atención al cliente, facilitando una perspectiva homogénea de la figura del cliente y la gestión uniforme de su relación con la empresa desde cada punto de contacto.

### Objetivos específicos

- Recopilación de las principales estructuras de redes neuronales, una descripción de las mismas y su aplicación a temas específicos de control de gestión e investigación de mercado.
- Interpretación del proceso para cada estructura neuronal, por ejemplo la aplicación de una función radial básica a un conjunto de variables. Esas variables pueden ser: género, edad, nivel socioeconómico, residencia, decisiones de compra. Como variable dependiente el grado de aceptación de un paquete de servicios. En este caso se debe analizar el período de aprendizaje, mantenimiento y validación de la red, verificar otros parámetros y utilizarlos, como así también el método de aprendizaje de la red. Luego analizar sus salidas ya en forma de gráficos, cruce de tablas y/o listados.
- Esta interpretación debe realizarse en un contexto teórico determinado y la comparación para cada uno de estos ejemplos con sus similares operaciones en la estadística tradicional.
- En lo que se refiere al CRM (Customer Relationship Management) se abarcarán todas las etapas correspondientes, a saber:
  - Integración: El desarrollo de una solución CRM parte de la información que la empresa dispone acerca de sus clientes. Típicamente, tal información vendrá dada bajo la estructura definida por un *datawarehouse* (DW) corporativo o departamental. Dicho DW aglutina los datos de negocio relevantes tras una labor de integración de los datos disponibles a través de las distintas fuentes, departamentos y canales a los que tenga acceso la empresa. Resultando en una fuente centralizada de la información relevante de cliente a partir de la cual ejecutar los procesos de análisis y de extracción de conocimiento de negocio.
  - Análisis: El análisis de la información del cliente disponible en el repositorio integrado de datos permite extraer el conocimiento de los clientes y mercado. Que posibilite diseñar y dirigir a partir de tal conocimiento acciones concretas de marketing a segmentos específicos de los clientes.

Cabe distinguir dos niveles distintos en la aplicación de los mecanismos y técnicas de análisis:

- a) Análisis de datos de negocio (técnicas OLAP, sistemas DSS, EIS, etc.)
- b) Análisis del conocimiento (extracción de conocimiento mediante análisis estadístico y técnicas de descubrimiento de conocimiento en bases de datos/minería de datos: árboles de clasificación, reglas de asociación, algoritmos genéticos, redes neuronales, *clustering*, etc.)

Esta fase de análisis resulta crítica desde la perspectiva de la obtención de beneficios a partir de la aplicación de una solución CRM. Únicamente mediante la adecuada ejecución de los procesos y técnicas de análisis puede alcanzarse un conocimiento válido y útil de las tendencias y patrones de comportamiento del cliente que permita establecer un modelo que posibilitará predecir su comportamiento y establecer una serie de métricas que soporten la toma de decisiones estratégicas.

Acción: La solución CRM es efectiva sólo si el conocimiento adquirido durante la etapa de análisis se materializa en acciones concretas sobre los procesos de negocio. La revisión y modificación de dichos procesos para dar cabida a las conclusiones extraídas del análisis de la información de cliente (lo aprendido sobre las preferencias, necesidades y comportamiento del cliente) constituye la etapa que cierra el ciclo de la solución CRM.

---

<sup>4</sup> Autores como Hartvigsen (1992), que ha elaborado un sistema experto para el análisis de la solvencia de las empresas reconoce las limitaciones y dificultades que plantea la construcción de los sistemas expertos para el análisis financiero. No hay una teoría formal y perfectamente estructurada para examinar la información contable de las empresas, aunque tampoco hay un desconocimiento total del procedimiento a seguir.

Para abordar este tipo de problemas difícilmente resolubles mediante los métodos convencionales o mediante sistemas expertos hemos propuesto combinarlos con otras herramientas de Inteligencia Artificial, como las redes neuronales.

# Semántica Declarativa Trivaluada para la Programación en Lógica Rebatible

Laura A. Cecchi

Guillermo R. Simari

Depto. de Informática y Estadística - Fa.E.A.    Depto. de Ciencias de la Computación  
UNIVERSIDAD NACIONAL DEL COMAHUE    UNIVERSIDAD NACIONAL DEL SUR  
e-mail:lcecchi@uncoma.edu.ar    e-mail:grs@cs.uns.edu.ar

PALABRAS CLAVES: Extensiones de la Programación en Lógica. Semántica Declarativa de Extensiones de la Programación en Lógica. Semántica de Juegos. Sistemas Argumentativos.

## 1 Introducción

La semántica operacional de los programas lógicos está basada en el método de resolución de Robinson, aplicado a cláusulas de Horn. Esto supone una limitación tanto sintáctica como semántica que, a pesar de las ventajas teóricas y prácticas que trae consigo, restringe en algunos casos la aplicabilidad de la Programación en Lógica (de ahora en más P.L.) en resolución de problemas. La superación de esta limitación puede atacarse de varias maneras. Un modo es extender la ejecución clásica a nuevos procedimientos junto con un cambio en la sintaxis, resultando en un incremento del poder expresivo de la P.L..

La Programación en Lógica Rebatible Básica [CS00a, GS99, Gar00] (de ahora en más P.L.R.B.) es una extensión de la P.L. en la que se permite representar conocimiento contradictorio, a través de la noción de negación fuerte, y conocimiento tentativo, incorporando a la sintaxis una nueva clase de reglas, las reglas rebatibles.

La semántica operacional de la P.L.R.B. está basada en un formalismo no monotónico: los Sistemas Argumentativos [PV00, CML99]. Las consecuencias de un programa lógico rebatible son determinadas a través de un análisis dialéctico de argumentos y contraargumentos. Si bien este mecanismo es suficiente para verificar si una consulta está justificada o no en un programa lógico rebatible, en los últimos años, la semántica operacional ha sido estudiada desde un punto de vista declarativo [Dun95, CDSS00, KT99], con el objeto de determinar el significado preciso de un programa lógico sin recurrir al control.

Con este espíritu se planteó en el marco del proyecto de investigación “Lógica, Revisión y Argumentación”, del Departamento de Informática y Estadística, Fa.E.A., U.N.C., un lineamiento cuyo objetivo es el estudio del sistema procedural de razonamiento no monótono de la P.L.R.B., en el que el criterio de decisión entre argumentos contradictorios es la especificidad [SL92].

El propósito de este trabajo es presentar la motivación de nuestra línea de investigación, nuestras metas, los resultados alcanzados y los desarrollos futuros. En la sección 2, se describen el problema a atacar y cuáles son nuestras metas. En la siguiente sección, se explica como se fueron desarrollando las primeras etapas planificadas y cumplimentadas

especificando la publicación que avala dicho resultado. Asimismo, se detalla el estudio en progreso. Finalmente, en la sección 4, se presentan las conclusiones y los trabajos futuros.

## 2 Descripción del Problema

De la filosofía propuesta por R. Kowalski en [Kow79], donde planteó la ecuación

$$\textit{Algoritmo} = \textit{Lógica} + \textit{Control}$$

resulta evidente que la semántica declarativa es necesaria para ayudar al programador a especificar el conocimiento y razonar a partir de él sin preocuparse por la parte de control del sistema. Esta es una de las características principales de la P.L.: *sólo expresa qué se debe hacer y no cómo hacerlo*.

Por otra parte, la definición de una semántica declarativa ayuda en el estudio de la Programación en Lógica Rebatible (P.L.R.) como sistema de razonamiento no monótono. Un conjunto de propiedades [BDK97], han sido presentadas con el objeto de clasificar y caracterizar el comportamiento de las semánticas de los programas lógicos con negación. Analizando el conjunto de propiedades que cumple es posible comparar al sistema con otros, mostrando sus ventajas y desventajas.

Diferentes frameworks declarativos [Dun95, Bil93, MG99] han sido desarrollados para modelar los sistemas argumentativos. Sin embargo, la mayoría ven a los argumentos como entidades abstractas cuyo rol está determinado por las relaciones de ataques con otros argumentos. Así, tanto el conjunto de argumentos como la preferencia entre contraargumentos deben ser explícitamente dados como entradas al framework.

Por otra parte, la mayoría hace un análisis declarativo estático del sistema argumentativo, a pesar de que la esencia de la argumentación es un diálogo interactivo. Finalmente, el comportamiento, que modelan las semánticas desarrolladas, no se adapta adecuadamente a la semántica procedural del sistema en estudio.

Todo lo expuesto anteriormente nos motivó a desarrollar una semántica que no sólo caracterice adecuadamente el comportamiento de la semántica operativa de la P.L.R. [SL92], i.e., que sea sensata y completa con respecto a dicho procedimiento, sino que, asimismo, mejore, en lo posible, los problemas encontrados en las semánticas declarativas ya existentes.

Si bien nuestra meta general es caracterizar la semántica operacional a través de una semántica declarativa adecuada para la P.L.R., nuestro interés particular se centrará en desarrollar una *semántica basada en juegos*. La elección de la categoría de los juegos está justificada en el sólido fundamento matemático con el que se ha modelado *la interacción entre los participantes del juego*.

## 3 Semántica Declarativa basada en Juegos

En la primera etapa de nuestro desarrollo, se circunscribió el estudio a la P.L.R. en donde la relación de preferencia entre argumentos contradictorios está restringida. No hemos permitido que la relación de preferencia sea un orden parcial, como tampoco aceptamos ciclos entre las preferencias. De este modo, evitamos ciertas falacias indeseables en el

Bajo las restricciones antes descriptas, se desarrolló una semántica declarativa trivaluada  $GS$  basada en juegos para la P.L.R.B., como lo demuestran los trabajos [CS99, CS00c, CS00a]. Dicha semántica es sensata y completa con respecto a la P.L.R.B. restringida.

La semántica trivaluada  $GS$  tiene como soporte matemático a las semánticas basadas en juegos introducidas a través de dos enfoques: HO-games [HO94] y AJM-games [Abr97, AM97]. En particular, se siguió el enfoque AJM-games por la similitud encontrada entre el árbol que especifica al conjunto de todas las posibles movidas legales en el juego y el árbol dialéctico que se debe realizar cada vez que queremos verificar si un literal pertenece a las consecuencias de un programa.

En  $GS$ , los movimientos permitidos del juego son todos los posibles argumentos a favor y en contra (contraargumentos) de todos los literales del lenguaje del programa. El juego determina cuáles son las jugadas legales en cada situación y de quién es el turno.

Con el objeto de verificar si un literal fijo<sup>1</sup>  $L$  pertenece al conjunto de las consecuencias de un programa  $\mathcal{P}$ , bajo  $GS$  deberemos *jugar* el juego siguiendo las instrucciones que se describen informalmente a continuación[CS99]:

- los turnos de los participantes alternan entre Proponente (P) y Oponente(O);
- juega primero P con un argumento a favor del literal;
- la siguiente jugada debe ser un contraargumento de la jugada anterior.

Con estas reglas generaremos la familia de todos los posibles juegos para  $L$ , tal que no haya dos juegos que comiencen con el mismo argumento que soporte a  $L$ . Si en alguno de los juegos de la familia O se quedó sin argumentos para derrotar a P, i.e., P ganó el juego, entonces  $L$  es consecuencia del programa. Si en la familia de juegos generada no existe ningún juego ganado por el proponente, entonces diremos que el valor de verdad de  $L$  es falso. Por último, contemplamos el caso en que la familia de juegos es vacía, cuya idea intuitiva es que no hay información sobre el literal y, por lo tanto, el valor de verdad de  $L$  es desconocido.

Aunque esta semántica modela el análisis dialéctico a través de un juego, la noción de argumento quedó indefinida. En otras palabras, se asume que el conjunto de argumentos para un literal es dado por algún oráculo. Esto motivó una nueva etapa en la investigación.

Una caracterización declarativa de la definición procedural de argumento ha sido introducida en [CS00b]. Dicha noción está basada en el conjunto estricto de consecuencias de un programa lógico rebatible básico. La unidad básica de los Sistemas Argumentativos, el argumento, es monotónica, luego se pudo establecer una analogía con la teoría desarrollada por Lifschitz en [Lif96] para la Programación en Lógica Básica. Así, el comportamiento de las consecuencias estrictas refleja el de la lógica clásica: ante información contradictoria genera el lenguaje. Ya que los argumentos no deben ser contradictorios, esto no será un problema. En dicho trabajo se demostró que la noción declarativa es equivalente al concepto procedural.

En este momento, nos encontramos estudiando la extensión de la semántica desarrollada a la P.L.R.B. sin restricciones sobre la relación de preferencia entre argumentos. Fundamentalmente, nuestro interés estará en poder capturar el análisis dialéctico sin falacias argumentativas, como reintroducción de argumentos en la dialéctica.

---

<sup>1</sup>En inglés ground [Fil01].

Ya que la relación de preferencia no ha sido instanciada aún con ningún método en particular, como prioridades entre reglas [PS96] o especificidad [SL92], se podrá comparar el comportamiento de la semántica con los diversos modos de resolver conflictos.

## 4 Conclusiones y Trabajos Futuros

Se ha presentado una de las líneas de investigación que se está desarrollando dentro del marco de un proyecto de investigación de la Universidad Nacional del Comahue. Se ha descrito el problema que motiva la definición de la semántica declarativa para la P.L.R.B. y se han precisado las metas que deseamos lograr. Asimismo, se detallaron los resultados alcanzados en las etapas planificadas, especificando, en cada caso, la publicación correspondiente.

Este resultado parcial motivó la etapa actual en la investigación que involucra la eliminación de la restricción sobre el criterio de decisión entre argumentos. En particular, es nuestro interés estudiar el criterio de preferencia entre argumentos basado en la especificidad.

Entre nuestros trabajos futuros se encuentran adaptar dicha semántica a otros sistemas, como el propuesto en [ABM98] y decidir si esta caracterización puede ayudar a dar significado a programas lógicos rebatibles que incluyan la negación por falla.

## Referencias

- [ABM98] G. Antoniou, D. Billington, y M.J. Maher. Normal forms for defeasible logic. En J. Jaffar, editor, *Proceedings International Joint Conference and Symposium on Logic Programming*, pages 160–174. MIT Press, 1998.
- [Abr97] Samson Abramsky. Semantics of Interaction. En A.Pitts y P. Dibyer, editors, *Semantics and Logic Computation*. Cambridge, 1997.
- [AM97] Samson Abramsky y Guy McCusker. Game Semantics. En H. Schwichtenberg y U. Berger, editors, *Logic and Computation: Proceedings of the 1997 Marktoberdorf Summer School*. Springer-Verlag, 1997.
- [BDK97] Gerhard Brewka, Jürgen Dix, y Kurt Konolige. *Nonmonotonic Reasoning. An Overview.*, volume 73 of *Lecture Notes Number* . CSLI Publications, Stanford - United States, 1997.
- [Bil93] David Billington. Defeasible Logic is Stable. *Journal of Logic Computation*, 3(4):379–400, 1993.
- [CDSS00] C. Chesñevar, J. Dix, F. Stolzenburg, y G. Simari. Relating defeasible and normal programming through transformation properties. En *VI CACiC*, Ushuaia, 2000.
- [CML99] Carlos Chesñevar, Ana Maguitman, y Ronald Loui. Logical models of arguments. En *ACM Computing Survey*. 1999.
- [CS99] F. Chesñevar, J. Dix, y G. Simari. Combining Defeasible and Normal Logic Programming. In *Proceedings of the 1999 Conference on Artificial Intelligence and Law*, pages 100–110. Cambridge, MA, 1999.

- [CS00a] Laura A. Cecchi y Guillermo R. Simari. Análisis de la semántica declarativa trivaluada GS para la programación en lógica rebatible básica. En *WICC - Aspectos Teóricos de Inteligencia Artificial*, 2000.
- [CS00b] Laura A. Cecchi y Guillermo R. Simari. Sobre la Relación entre la Definición Declarativa y Procedural de Argumento. Ushuaia, 2000. VI CACiC.
- [CS00c] Laura A. Cecchi y Guillermo R. Simari. Una semántica declarativa basada en juegos para la programación en lógica rebatible básica. En *Proceedings of ICIE*, 2000.
- [Dun95] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
- [Fil01] Pablo Rubén Fillottrani. *La negación en la Programación en Lógica*. PhD thesis, Universidad Nacional del Sur, 2001.
- [Gar00] Alejandro J. García. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional y Paralelismo*. PhD thesis, Universidad Nacional del Sur, 2000.
- [GS99] A. García y G. R. Simari. Strong and Default Negation in Defeasible Logic Programming. En *4th Dutch/German Workshop on Nonmonotonic Reasoning Techniques and their applications*, Amsterdam, 25 - 27, Marzo 1999.
- [HO94] Martin Hyland y Luke Ong. On full abstraction for PCF:I,II,III. A publicarse en *Information and Computation*. Technical report, Draft disponible a través de ftp en [theory.doc.ic.ac.uk](http://theory.doc.ic.ac.uk), 1994.
- [Kow79] Robert Kowalski. Algorithm = Logic + Control. *Communications of the ACM*, (7):424–435, July 1979.
- [KT99] A. Kakas y F. Toni. Computing argumentation in logic programming. *Journal of Logic and Computation*, (4):515–562, 1999.
- [Lif96] Vladimir Lifschitz. Foundations of logic programming. En G. Brewka, editor, *Principles of Knowledge Representation*, pages 1–57. CSLI Publications, 1996.
- [MG99] M. J. Maher y G. Governatori. A semantic decomposition of defeasible logics. *AAAI*, pages 299–305, 1999.
- [PS96] H. Prakken y G. Sartor. A dialectical model of assessing conflicting arguments in legal reasoning. *Artificial Intelligence and Law*, 4:331–368, 1996.
- [PV00] Henry Prakken y Gerard Vreeswijk. Logics for defeasible argumentation. En D. Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer Academic, second edition, 2000.
- [SL92] Guillermo R. Simari y R.P. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, pages 125–157, 1992.

# Introducing Probabilistic Reasoning in Defeasible Argumentation Using Labeled Deductive Systems

Carlos Iván Chesñevar

Guillermo Ricardo Simari

Departamento de Ciencias e Ingeniería de la Computación

UNIVERSIDAD NACIONAL DEL SUR

Av. Alem 1253 – B8000CPB Bahía Blanca – REPÚBLICA ARGENTINA

TEL/FAX: (+54) (291) 459 5135/5136 – EMAIL: {cic,grs}@cs.uns.edu.ar

KEY WORDS: Defeasible Argumentation, Probabilistic reasoning, Labelled Deductive Systems

## Abstract

Labeled Deductive Systems (LDS) were developed as a rigorous but flexible methodology to formalize complex logical systems, such as temporal logics, database query languages and defeasible reasoning systems.

$LDS_{AR}$  is a LDS-based framework for *defeasible argumentation* which subsumes different existing argumentation frameworks, providing a testbed for studying different relevant features (such as emerging logical properties, ontological aspects, semantic characterization, etc.)

This paper discusses some relevant issues concerning the introduction of *probabilistic reasoning* into defeasible argumentation. In particular, we consider a first approach for recasting the existing  $LDS_{AR}$  framework in order to incorporate numeric attributes (certainty factors) as part of the argumentation process.

## 1 Introduction and motivations

Labeled Deductive Systems (LDS) [Gab96] were developed as a rigorous but flexible methodology to formalize complex logical systems, such as temporal logics, database query languages and defeasible reasoning systems. In labeled deduction, the usual notion of formula is replaced by the notion of *labeled formula*, expressed as  $Label:f$ , where *Label* represents a label associated with the wff  $f$ . A labeling language  $\mathcal{L}_{Label}$  and knowledge-representation language  $\mathcal{L}_{kr}$  can be combined to provide a new, labeled language, in which labels convey additional information also encoded at object-language level. Formulas are labeled according to a family of *deduction rules*, and with agreed ways of propagating labels via the application of these rules.

The study of logical properties of *defeasible argumentation* [PV99, CML00] motivated the development of  $LDS_{AR}$  [Che01], an LDS-based argumentation formalism. In  $LDS_{AR}$ , labels provide information about the *epistemic status* of wffs (defeasible vs. non-defeasible) and the *name* of the wff involved.  $LDS_{AR}$  provides a useful formal framework for studying logical properties of defeasible argumentation in general, and of DeLP [Gar00] in particular. Equivalence results with other argumentative frameworks were also studied.

Labeled deduction has a number of features which make it suitable for characterizing new ontologies. Thus, different *variants* of defeasible argumentation can be explored from the original  $LDS_{AR}$  formulation by introducing changes in the object language. These changes can be introduced in a modular way, without affecting the whole framework.

This paper is motivated by extending the original notion of label in order to incorporate *probabilistic reasoning* in the  $LDS_{AR}$  framework. The success of argumentation-based approaches is partly due to the sound setting it provides *qualitative reasoning*. Numeric attributes, on the other hand, offer an useful source of information for *quantitative reasoning* in several knowledge domains. We think that combining both kinds of reasoning into an single argumentation framework would be highly desirable.

## 2 The $LDS_{AR}$ framework: a brief sketch<sup>1</sup>

In  $LDS_{ar}$ , the language  $\mathcal{L}_{kr}$  is the one of *extended logic programming*. Labels extend this language by distinguishing *defeasible* and *non-defeasible* information. A consequence relation  $\sim_{Arg}$  propagates labels, implementing the SLD resolution procedure along with a consistency check every time new defeasible information is introduced in a proof. This information is collected into a *set of support*, containing all defeasible information needed to conclude a given formula. Thus, arguments are modeled as labeled formulas  $\mathcal{A}:h$ , where  $\mathcal{A}$  stands for a set of (ground) clauses, and  $h$  for an extended literal.

Given a knowledge base  $\Gamma$  the consequence relation  $\sim_{Arg}$  allows to infer labeled formulas of the form *argument:literal*. Since arguments may be in conflict, a new, extended consequence relationship  $\sim_{\mathcal{T}}$  is defined. Those wffs derivable from  $\Gamma$  via  $\sim_{\mathcal{T}}$  will correspond to *dialectical trees*. These new labeled wffs will therefore have the form *dialectical tree:conclusion*.

## 3 Approaches to uncertainty based on probabilistic reasoning

As Judea Pearl points out [Pea88], commonsense reasoning involves *summarizing exceptions* at a given stage. In defeasible argumentation this is done by providing defeasible rules “ $p(X) \leftarrow q(X)$ ”, which provide a symbolic way of specifying “not every  $q(X)$  is  $p(X)$ ”. Another way of summarizing exceptions is to assign to each proposition a numerical measure of uncertainty, and then combine these measures according to uniform syntactic principles.

When introducing numerical values for modeling uncertainty, *extensional* and *intensional* approaches can be distinguished. Extensional approaches treat uncertainty as a generalized truth value attached to formulas. Computing the uncertainty of any formula is a function of the uncertainties of its subformulas. Intensional approaches, on the other hand, are model-based: uncertainty is attached to “states of affairs” or subsets of “possible worlds”. Typical examples of this extensional approaches are production systems and rule-based systems.

Extensional approaches are computationally attractive, but their semantics may be ‘sloppy’. Intensional approaches are semantically clear but computationally clumsy. Most research has been directed to find a trade-off between these two kinds of formalizing uncertainty. In order to recast defeasible argumentation in terms of probabilistic reasoning, we

---

<sup>1</sup>For space reasons we only give a brief summary of the main elements of the  $LDS_{AR}$  framework. We also assume that the reader has basic knowledge about the underlying concepts in defeasible argumentation formalisms. For an in-depth treatment the reader is referred to [Che01, CML00].

will adopt an extensional approach as it can be easily integrated in the existing ontology, as we will see in the next section. Semantical issues are not discussed in this paper. However, it must be noted that Gabbay's LDS provide a sound basis for defining formal semantics associated to arbitrary logical systems using labelled deduction.<sup>2</sup>

## 4 Handling probabilities in $LDS_{AR}$ : a first approach

Figure 1 shows the basic rules for building *generalized arguments* (*i.e.*, defeasible proofs which are non-contradictory wrt the strict knowledge  $\text{Strict}(\Gamma)$  for a given knowledge base  $\Gamma$ ). Rules 1 and 2 introduce non-defeasible and defeasible information, respectively. Rules 3 and 4 account for introducing conjunction and *modus ponens*. As discussed in previous sections, these natural deduction rules propagate labels when performing inference. Note that in our case object-language wffs have the form  $\text{Label} : f$ , where labels convey the following information: a) name and epistemic status of the wff (n=non-defeasible, d=defeasible); b) set of support  $\Phi$  (such that  $\Phi \vdash_{\text{SLD}} f$ ).

In order to introduce an uncertainty measure in object-language wffs, a natural approach would be just adding some certainty factor  $cf$ , such that  $cf(f) = 1$  whenever  $f$  corresponds to non-defeasible knowledge, and  $0 < cf(f) < 1$  whenever  $f$  stands for defeasible knowledge. A formula of the form  $[\alpha, cf(\alpha)]:\alpha$  in the knowledge base  $\Gamma$  would therefore stand for “ $\alpha$  is a defeasible formula which has the certainty factor  $cf(\alpha)$ ”.<sup>3</sup> Similarly, the formula  $[\emptyset, 1]:\alpha$  would stand for “ $\alpha$  is a non-defeasible formula”. Finally, performing an inference from  $\Gamma$  (*i.e.*, building a generalized argument) would result in inferring a formula  $[\Phi, cf(\Phi)]:\alpha$ , standing for “The set  $\Phi$  provides an argument for  $\alpha$  with a certainty factor  $cf(\Phi)$ ”.

As we can see, in this new setting every formula in  $\Gamma$  should be attached with a certainty factor, indicating whether the formula corresponds to non-defeasible or defeasible knowledge. Natural deduction rules propagate certainty factors as inferences are carried out. From the structure of the natural deduction rules in figure 1 we can distinguish two relevant aspects in such a new setting:

- **Propagating certainty factors:** In extensional systems, uncertainty can be treated as a generalized truth value, *i.e.* the certainty of a formula is defined as a unique function from the certainties of its subformulas. LDS allow us to proceed the same way: when performing an inference, a new label is defined in terms of existing (already inferred) labels.

Therefore propagating certainty factors turns out to be natural in our framework. As an example, consider two formulas  $[\Phi, cf(\Phi)]:\alpha$  and  $[\Psi, cf(\Psi)]:\beta$ . If  $\alpha, \beta$  could be derived (introducing conjunction), the resulting formula would have the form

$$[\Phi \cup \Psi, f_{\wedge}(cf(\Phi), cf(\Psi))]:\alpha, \beta$$

- **Handling consistency:** In the original formulation, consistency checking of a (defeasible) wff  $f$  wrt a set of arbitrary wffs  $\Phi$  stands for  $\Phi \cup \{f\} \not\vdash_{\text{SLD}} \perp$ . Note that

<sup>2</sup>See [Gab96] for details.

<sup>3</sup>Note that it is no longer necessary to distinguish between defeasible and non-defeasible formulas, as this qualification can be inferred from their associated certainty factors.

consistency checking in defeasible argumentation involves the set of strict knowledge  $\text{Strict}(\Gamma)$ , where all wffs are non-defeasible (*i.e.*  $cf(\gamma) = 1, \forall \gamma \in \text{Strict}(\Gamma)$ ). Using the approach discussed above, when a new defeasible formula  $f$  is inferred, it will have the form  $[\Phi, cf(\Phi)]:f$ . Consistency checking will involve assuming that  $f$  is “locally non-defeasible” (*i.e.*,  $[\Phi, cf(1)]:f$ . and non-contradictory wrt  $\text{Strict}(\Gamma)$ ).

Let us summarize some of the main issues which are relevant to consider when incorporating certainty factors into the argumentation process. Following [PV99], we can distinguish different levels in our analysis:

- **Argument construction:** As discussed before, the two basic aspects to be considered at this level are the propagation of certainty factors (cf) and handling consistency.
- **Counterargument / Defeat:** The counterargument relationship is defined in terms of *contradiction*. Therefore certainty factors should play no role in this case. However, defeat can be stated in terms of certainty factors as follows: an argument  $\mathcal{A}:h$  is a proper defeater  $\mathcal{B}:q$  if  $cf(\mathcal{A}:h) > (\geq)cf(\mathcal{B}:q)$ . Similarly, a blocking defeat situation would arise if  $cf(\mathcal{A}:h) = cf(\mathcal{B}:q)$ , or alternatively  $|cf(\mathcal{A}:h) - cf(\mathcal{B}:q)| \geq \epsilon$ , for  $\epsilon$  arbitrarily small.
- **Dialectical analysis:** Note that the whole process of determining whether a given argument is warranted or not only relies on binary relationships between arguments (blocking defeat / proper defeat). Therefore the introduction of certainty factors does not affect the construction of the dialectical tree. However, the *labeling* of the tree  $T$  might incorporate the cf’s in a natural way as follows:

- Given an argument  $\mathcal{A}:h$  which is a leaf node  $L$  in  $T$ , then  $cf(L)=cf(\mathcal{A})$ .
- Given an argument  $\mathcal{B}:q$  which is an inner node in  $T$ , then

$$cf(\mathcal{B}:q)=f_{tree}(cf(\mathcal{B}), cf(T_1), \dots cf(T_k))$$

where  $T_1 \dots T_k$  are immediate subtrees of  $\mathcal{B}:q$ . In other words, certainty factors are propagated bottom-up according to some function  $f_{tree}$  (*e.g.*, arguments with many defeaters could be deemed weaker as those which have only one defeater).

## 5 Conclusions and future work

Probabilistic reasoning has been mostly neglected in the defeasible argumentation community. This is maybe due to the historical origins of argumentative reasoning, which were more related to legal (qualitative) reasoning rather than to number-based attributes as those used in rule-based production systems.

In this paper we have suggested some basic ideas on how to extend the existing  $LDS_{AR}$  framework to incorporate probabilistic reasoning. As we have shown, labels provide a flexible tool for including numeric information which can be propagated using deduction rules.

The growing success of argumentation-based approaches has caused a rich cross-breeding with other disciplines, providing interesting results in different areas such as legal reasoning,

1.  $\frac{}{[\emptyset, \{n_i\}]:\alpha}$  for any  $[\emptyset, \{n_i\}]:\alpha \in \text{Strict}(\Gamma)$ .
2.  $\frac{\Pi(\Gamma) \cup \Phi \not\vdash_{\text{SLD}} \perp}{\Gamma, [\Phi, \{d_i\}]:\alpha}$  for any  $[\Phi, \{d_i\}]:\alpha \in \text{Defeasible}(\Gamma)$ .
3.  $\frac{\Gamma, [\Phi_1, L_1]:\alpha_1 \quad [\Phi_2, L_2]:\alpha_2 \quad \dots \quad [\Phi_k, L_k]:\alpha_k \quad \Pi(\Gamma) \cup \bigcup_{i=1..k} \Phi_i \not\vdash_{\text{SLD}} \perp}{\Gamma, [\bigcup_{i=1..k} \Phi_i, \bigcup_{i=1..k} L_i]:\alpha_1, \alpha_2, \dots, \alpha_k}$
4.  $\frac{\Gamma, [\Phi_1, L_1]:\beta \leftarrow \alpha_1, \dots, \alpha_k \quad [\Phi_2, L_2]:\alpha_1, \dots, \alpha_k \quad \Pi(\Gamma) \cup \Phi_1 \cup \Phi_2 \not\vdash_{\text{SLD}} \perp}{\Gamma, [\Phi_1 \cup \Phi_2, L_1 \cup L_2]:\beta}$

Figure 1: Rules for deriving generalized arguments in  $LDS_{AR}$ 

medical diagnosis and decision support systems. In this context, we contend that existing frameworks for defeasible argumentation (such as  $LDS_{AR}$ ) can be enriched by integrating numeric attributes (such as probabilities or certainty values), making them more attractive and suitable for other research and application areas. Part of our current research work is focused on these aspects.

## References

- [Che01] CHESÑEVAR, C. I. *Formalización de los Procesos de Argumentación Rebatible como Sistemas Deductivos Etiquetados*. PhD thesis, Departamento de Ciencias de la Computación - U.N.S. - Argentina, January 2001.
- [CML00] CHESÑEVAR, C. I., MAGUITMAN, A., AND LOUI, R. Logical Models of Argument. *ACM Computing Surveys* 32, 4 (December 2000), 337–383.
- [Gab96] GABBAY, D. *Labelling Deductive Systems (vol.1)*. Oxford University Press (Volume 33 of Oxford Logic Guides), 1996.
- [Gar00] GARCÍA, A. J. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional y Paralelismo*. PhD thesis, Dep. de Cs. de la Computación, Universidad Nacional del Sur, December 2000.
- [Pea88] PEARL, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [PV99] PRAKKEN, H., AND VREESWIJK, G. Logics for Defeasible Argumentation. In *Handbook of Philosophical Logic*, D. Gabbay, Ed. Kluwer Academic Publisher, 1999.
- [SCG01] SIMARI, G. R., CHESÑEVAR, C. I., AND GARCÍA, A. Modelling Argumentation in a Logic-Programming Setting: Formalization and Logical Properties. In *Proc. of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty – Workshop “Adventures in Argumentation”* (September 2001), Toulouse, France.
- [SL92] SIMARI, G. R., AND LOUI, R. P. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence* 53 (1992), 125–157.

# ILP en Aprendizaje Grupal

Telma Delladio

`td@cs.uns.edu.ar`

Departamento de Ciencias e Ingeniería de Computación

Universidad Nacional del Sur

Avda. Alem 1253 - Bahía Blanca - Bs. As.

Tel (0291) 459-5135 - Fax (0291) 459-5136

## 1 Introducción

La Programación en Lógica Inductiva (ILP) es una propuesta de aprendizaje que utiliza a la Programación en Lógica (PL) como herramienta de representación de conocimiento. Se han desarrollado muchos mecanismos de aprendizaje a través de la generación de programas lógicos que sintetizan una definición para un determinado concepto.

El tipo de aprendizaje desarrollado con ILP es un aprendizaje a partir de ejemplos. La clase de los ejemplos considerados (positivos y/o negativos) y el modo de procesamiento de los mismos varía de acuerdo a la técnica utilizada. El principal problema en este tipo de sistemas es la complejidad. Toda tarea de aprendizaje es compleja por naturaleza y por eso cualquier sistema que implemente algún mecanismo de aprendizaje deberá tratar con esta complejidad.

Se han desarrollado sistemas "generales" de aprendizaje a partir de ejemplos [Bra90, BG96] que son capaces de aprender conceptos diversos. El principal problema de estos sistemas es que la calidad de los conceptos aprendidos depende en gran medida de los ejemplos provistos (calidad y orden de procesamiento). Además para lograr cierto grado de eficiencia se suelen establecer restricciones en la generación de cláusulas junto con declaraciones de modos y/o tipos sobre los argumentos de los predicados que se desean generar en el proceso de aprendizaje. La generalidad de estos sistemas deriva en la ineficiencia de los mismos. Esto es, para lograr aprender cualquier concepto, hay que hacer un gran esfuerzo por brindar muy buenos ejemplos y gran cantidad de información adicional para que se pueda generar la definición del concepto en un tiempo y forma adecuados. Por otro lado, están los sistemas de aprendizaje, por ejemplo algunos clasificadores, que se han desarrollado para algún problema específico y que hacen uso de heurísticas para poder llevar adelante el aprendizaje en forma aceptable. Estos sistemas son de aplicación específica lo que hace que se puedan desarrollar mecanismos de aprendizaje puntuales al dominio de aplicación [LD94].

Cualquier sistema de aprendizaje tiene que tratar con el problema de la eficiencia, pues cualquier tarea de aprendizaje es muy compleja. Los sistemas desarrollados en el ámbito ILP basan su tarea de aprendizaje en la búsqueda de una definición para uno o más conceptos. En ILP las definiciones buscadas son programas lógicos, es decir un conjunto de cláusulas. Esta búsqueda se lleva a cabo en el Espacio de Hipótesis formado por todas aquellas cláusulas que son potenciales definiciones del concepto, o conceptos, que se están aprendiendo. Cuanto mayor sea este espacio, más difícil es la búsqueda. Todas las heurísticas o restricciones que puedan establecer este tipo de sistemas, tienen como objetivo reducir este espacio de hipótesis lo que hace más eficiente la búsqueda.

Por todo esto, uno de los principales objetivos en ILP es encontrar propuestas que ayuden de alguna forma a hacer este problema más tratable. Una propuesta considerable es encarar la tarea de aprendizaje en forma cooperativa. En muchas ocasiones es más efectivo o más fácil aprender algo en forma grupal. Diferentes personas pueden tener distintos puntos de vista, distinto conocimiento, distintas experiencias. Todo esto puede facilitar la tarea de aprendizaje.

## 2 Aprendizaje grupal

En el ámbito de ILP también es posible encarar la tarea de aprendizaje de forma cooperativa. La idea general es la misma; aprender un concepto, pero esta vez en forma grupal. Esto es, un grupo de entidades o agentes con distintos conocimientos, capacidades y observaciones trabajando en conjunto para aprender un mismo concepto. Esta propuesta no solo tiene que verse como una forma de llevar a cabo la tarea de aprendizaje en forma más eficiente, sino que puede presentarse en muchas situaciones como la única forma de lograr dicho aprendizaje. Por ejemplo, consideremos cuatro entidades  $E_1, E_2, E_3$  y  $E_4$ . Cada una de ellas con habilidades diferentes.  $E_1$  puede distinguir colores: rojo, amarillo, violeta, verde.  $E_2$  puede distinguir sabores: dulce, amargo, ácido.  $E_3$  puede distinguir formas: esférica, oblonga, achatada y  $E_4$  puede distinguir tamaños: grande, chico. Si el objetivo es aprender a identificar frutas como peras, manzanas y ciruelas es difícil que cada una de estas entidades por separado pueda hacerlo correctamente, pues ninguna de esas características (color, sabor, forma, tamaño) por separado es suficiente para distinguir una de estas frutas, pues la variedades de peras, manzanas y ciruelas es muy grande. Las manzanas ácidas pueden ser verdes o rojas. No todas las peras son verdes o amarillas, hay ciruelas rojas, violetas o amarillas, etc.

### 2.1 Esquema general

Esta propuesta de aprendizaje grupal consiste en definir un *grupo de aprendizaje*. Este grupo será un conjunto de  $n$  entidades,  $E_1, \dots, E_n$ , que compartirán sus conocimientos y capacidades para aprender un concepto, digamos  $C$ . Todas estas entidades tienen el mismo objetivo: “aprender  $C$ ”.

Cada entidad  $E_i$  tendrá su propio conocimiento previo (o base de conocimiento)  $B_i$  y tendrá acceso a un conjunto de observaciones  $O_i$ . La forma de acceso y el tipo de las observaciones  $O_i$

dependerá de las características de cada  $E_i$ . Las observaciones pueden ser positivas y/o negativas y procesarse en conjunto o incrementalmente, dependiendo del mecanismo de aprendizaje,  $M_i$ , que posea cada  $E_i$ . Cada entidad desarrollará una definición (subjetiva) parcial,  $D_i(C)$  del concepto  $C$  a partir de su conocimiento previo y sus observaciones. Luego a partir de  $D_1(C), \dots, D_n(C)$  se buscará sintetizar en forma conjunta una definición global  $D(C)$ . La calidad de esta definición, es decir; su correctitud y su completitud, dependerá de las definiciones parciales obtenidas y de la manera en que se lleve a cabo la composición de dicha definición.

La idea hasta aquí expuesta es muy general, un grupo de entidades con distintas características trabajando en conjunto para aprender un concepto determinado. Lo que se quiere estudiar es cómo llevar adelante esta propuesta en el marco de la Programación en Lógica Inductiva.

Una de las características de la propuesta ILP es la uniformidad de la representación. Esto es, en ILP *todo* son programas lógicos. Las observaciones pueden expresarse como hechos lógicos. El conocimiento previo se expresa a través de un programa lógico y los conceptos aprendidos también se expresan como programas lógicos que pasarán a formar parte del conocimiento previo. Por esto, el mapeo de este esquema general a la propuesta ILP es directo. Lo que no es directo en esta propuesta, ni en otras, es la solución de aquellos problemas propios de cualquier actividad grupal, como por ejemplo la comunicación, el consenso de soluciones, etc.

## 2.2 Esquema ILP

Como ya se dijo, todo en ILP son programas lógicos. Por lo tanto, el conocimiento previo  $B_i$  de cada entidad será un programa lógico que representará todo el conocimiento disponible que posee  $E_i$ . Este conocimiento puede ser conocimiento inicial o conocimiento adquirido a través del tiempo, por ejemplo como consecuencia de alguna actividad de aprendizaje. Las observaciones,  $O_i$ , de cada entidad son representadas por un conjunto de hechos lógicos. Por último las definiciones parciales  $D_i(C)$  estarán formadas por un conjunto de cláusulas obtenidas por  $E_i$  y que sirven como definición subjetiva del concepto  $C$ .

Lo importante aquí, es que cada definición  $D_i(C)$  es *parcial* desde el punto de vista de la definición general que se está buscando. Es de esperar que cada definición parcial sea lo más correcta y completa desde el punto de vista de la entidad que la genera. Esto es, posiblemente  $D_i(C)$  sea una definición completa y correcta para  $E_i$  considerando su conocimiento previo  $B_i$  y el conjunto de observaciones disponibles  $O_i$ . Por lo general, cuando se consideran observaciones positivas y negativas, en el marco ILP, se dice que una definición  $D_i(C)$  será completa si deriva o cubre todas las observaciones positivas y se dice que es correcta si no deriva o cubre ninguna de las observaciones negativas. Dependiendo del mecanismo de aprendizaje  $M_i$  que cada entidad implemente, es posible que cada una de las definiciones parciales  $D_i(C)$  puedan llegar a ser completas y correctas con respecto a su conjunto de observaciones  $O_i$ , pero no necesariamente lo serán con respecto a todas las observaciones disponibles en el grupo. Entonces, una definición parcial puede ser correcta desde el punto de vista de una entidad y puede no serlo si se considera el conocimiento

y las observaciones de *todo* el grupo de aprendizaje.

Cada entidad es capaz de definir al concepto  $C$  solo a partir de los recursos que tiene disponibles (conocimiento y observaciones). Si una entidad  $E_i$  solo tiene acceso a su base de conocimiento y a sus observaciones, es posible que la definición  $D_i(C)$  generada esté en contradicción, por ejemplo, con el conjunto de observaciones  $O_j$  de otra entidad  $E_j$ . Esto posiblemente, provoque una contradicción entre las definiciones parciales  $D_i(C)$  y  $D_j(C)$ .

Teniendo en cuenta todo esto, el objetivo del *grupo de aprendizaje* es sintetizar una definición general para  $D(C)$ . Indudablemente, dicha definición se conformará a partir de todas las definiciones parciales  $D_1(C), \dots, D_n(C)$ . Una propuesta ingenua sería considerar la unión de estas definiciones parciales como definición general. No obstante, un simple análisis muestra que esto no es factible. El argumento principal es que cada definición parcial cubre solo un aspecto del concepto que se quiere aprender y la definición general buscada deberá cubrir de la mejor manera posible todos los aspectos del concepto, es decir, ser lo más completa posible. Por otro lado, como ya se dijo, es posible que la simple unión entre dos definiciones parciales cree una contradicción.

Una de las cuestiones que hay que tener presente es que cada definición parcial es generada por una entidad en particular. Por lo tanto puede suceder que la definición parcial  $D_i(C)$  no sea “comprendida” por la entidad  $E_j$  ( $i \neq j$ ). Cada definición parcial  $D_i(C)$  es un conjunto de cláusulas lógicas, esto es, la definición de un predicado que modela el concepto  $C$  desde el *punto de vista* de  $E_i$ . En la definición de ese predicado es posible que  $E_i$  utilice conceptos (predicados) ya definidos en su base de conocimiento los cuales no necesariamente estarán disponibles para las demás entidades.

### 3 Trabajo futuro

Lo que se ha presentado es un esquema simple de aprendizaje grupal desde el punto de vista de la Programación en Lógica Inductiva y se señalan los principales inconvenientes que se deben sortear al momento de llevar adelante una propuesta de este estilo. Se presentaron tan solo algunas de las cuestiones que hay que analizar y solucionar al momento de *consensuar* una definición general para poder sacar el mayor provecho de las características de cada una de las entidades. Cualquier definición general debe tener un cierto *grado de aceptación* de cada una de las entidades pertenecientes al grupo. Se debe encontrar una manera equilibrada de valorar los aportes que realiza cada entidad, de acuerdo a su conocimiento y sus capacidades. Así como también, la manera más efectiva de transmitir o compartir entre las entidades ciertas piezas de conocimiento o información que sirvan para “comprender” la definición general.

Los *grupos de aprendizaje* pueden tener características muy variadas. Se puede considerar una amplia gama que va desde los grupos totalmente homogéneos hasta aquellos totalmente heterogéneos. Un grupo totalmente homogéneo sería aquel compuesto por un conjunto de entidades de iguales características (observaciones, bases de conocimiento y mecanismos de aprendizajes). El principal interés está en el estudio de grupos que no sean totalmente homogéneos, puesto que

esos son los grupos que ofrecen un marco de trabajo diferente al del aprendizaje individual.

En el futuro se definirán distintos *tipos* de grupos de aprendizaje y en base a sus características se estudiarán las mejores formas de solucionar los problemas que se presentan principalmente al momento de *consensuar* una definición general para el concepto que se está aprendiendo.

## Referencias

- [BG96] BERGADANO, F., AND GUNETTI, D. *Inductive Logic Programming, From Machine Learning to Software Engineering*. The MIT Press, 1996.
- [Bra90] BRATKO, I. *Prolog Programming for Artificial Intelligent*, second ed. Addison Wesley, 1990.
- [LD94] LAVRAC, N., AND DZEROSKI, S. *Inductive logic programming: Techniques and applications*, 1994.

# Embedding Abduction in Nonmonotonic Theories

Claudio Delrieux

Universidad Nacional del Sur, Bahia Blanca - ARGENTINA

claudio@acm.org

## 1 Introduction

An important ampliative inference schema that is commonly used is *abduction*. Abduction plays a central rôle in many applications, such as diagnosis, expert systems, and causal reasoning. In a very broad sense we can state that abduction is the inference process that goes from observations to explanations within a more general context or theoretical framework. That is to say, abductive inference looks for sentences (named *explanations*), which, added to the theory, enable deductions for the observations. Most of the times there are several such explanations for a given observation. For this reason, in a narrower sense, abduction is regarded as an inference to the best explanation. However, a problem that faces abduction is the explanation of *anomalous* observations, *i. e.*, observations that are contradictory with the current theory. It is perhaps impossible to do such inferences in monotonic theories. For this reason, in this work we will consider the problem of characterizing abduction in nonmonotonic theories. Our inference system is based on a natural deduction presentation of the implicational segment of a relevant logic, much similar to the  $R_{\rightarrow}$  system of Anderson and Belnap [1]. Then we will discuss some issues arising the pragmatic acceptance of abductive inferences in nonmonotonic theories.

## 2 Abduction of Anomalous Observations

Before discussing the worries of abduction in nonmonotonic theories, first we will give a formal characterization of a nonmonotonic reasoning system, and then include an explicit rule for abduction. In a nutshell, the system regards defeasible rules  $a(X) \succ\!\!\!-\ b(X)$ <sup>1</sup> as material implications *only* for the *modus ponens* inference rule (that is, contraposition, left strengthening, right weakening, and similar uses are explicitly left out). Defeasible rules can be activated in MP only when their antecedent is fully instantiated, *i. e.*, there is a ground substitution for  $X$  such that all the literals in  $a(X)$  have been inferred. The reasoning system, then, will chain inferences in a way very similar to (classical) deductions, with the addition of inferences in which a fully activated defeasible rule was used. This chains of inferences are named (*sub*)-*theories* [2, 6] or *arguments* [7, 8]. If a defeasible rule can be regarded as a *prima facie* material implication, then an argument for  $A$  is a *prima facie* proof or a *prediction* for  $A$ . We can then extend the (classical) consequence operator  $\vdash$  to the new operator  $\vdash\sim$ , where  $\mathcal{T}\vdash\sim A$  means that there is an argument for  $A$  in theory  $\mathcal{T}$ .

---

<sup>1</sup>both antecedent and consequent of defeasible rules are restricted to be *sets of literals* (interpreted as a conjunction), and  $X$  is a tuple of free variables.

Since we may reasonably expect that these inferences will eventually generate a pair of contradictory literals, and since we want to avoid (classical) trivialization, then our reasoning system must incorporate some kind of strengthening or restriction among the structural rules. For this reasoning, we adapted a presentation of the implicational segment of a relevant logic, much similar to the  $R_{\rightarrow}$  system of Anderson and Belnap [1]. Then, the reiteration rule is restricted to sentences that were inferred within the same subproof. If we need reiteration of a sentence  $S$  of a previous step outside the subproof, then we must either introduce  $S$  as a new assumption, or reproduce the inference steps that leads to the inference of  $S$ . To take due care of this, we establish an index schema that labels premise introduction and its ulterior discharge by means of  $\rightarrow_I$ . The use of a defeasible rule is regarded as a restricted *modus ponens* that also introduces a new hypothesis. The labeling schema obeys a simple set of cases (the subindices  $I$  and  $J$  denote sets of indices, and  $i$  and  $j$  denote individual indices).

- (*Premise*) An hypothetical premise  $a$  is introduced with an index  $i$  never used before (we will use the sequence of natural numbers).
- ( $\rightarrow_I$ ) From a (sub-)demonstration for  $b_I$  from premise  $a_j$  (with  $j \in I$ ) to infer  $(a \rightarrow b)_{I-\{j\}}$ .
- (*Reit.*) Reiteration of a sentence retains the indices.
- ( $\rightarrow_E$ ) In the *modus ponens* rule, the consequent retains the indices of the major and the minor premises: from  $a_I$  and  $(a \rightarrow b)_J$  to infer  $b_{I \cup J}$ .

We now add the case for defeasible rules.

- ( $\succ_E$ ) From a (sub-) demonstration of  $a(t)_I$  and  $a(X) \succ b(X)$  to infer  $a(t) \rightarrow b(t)_{I \cup \{k\}}$ , where  $k$  is an index never used before.<sup>2</sup>

**Example 2.1** Suppose that in our knowledge base we have

$$a, a \succ b, b \succ c, a \succ \neg c$$

In this situation, we may establish the following reasoning lines:

$$\begin{array}{ll} 1 & \left[ \begin{array}{l} a_{\{1\}} \\ (a \succ b)_{\{2\}} \\ b_{\{1,2\}} \\ (b \succ c)_{\{3\}} \end{array} \right. & \begin{array}{l} \textit{Premise} \\ \textit{Defeasible rule} \\ 1, 2, \succ_E \\ \textit{Defeasible rule} \end{array} \\ 5 & c_{\{1,2,3\}} & 3, 4, \succ_E \end{array}$$

$$\begin{array}{ll} 1 & \left[ \begin{array}{l} a_{\{1\}} \\ (a \succ \neg c)_{\{4\}} \end{array} \right. & \begin{array}{l} \textit{Premise} \\ \textit{Defeasible rule} \end{array} \\ 3 & \neg c_{\{1,4\}} & 1, 2, \succ_E \end{array}$$

Given a theory  $\mathcal{T}$  and an observed evidence  $e$ , then  $e$  is surprising if neither  $\mathcal{T} \vdash e$ , nor  $\mathcal{T} \vdash \neg e$ , and  $e$  is anomalous if  $\mathcal{T} \vdash \neg e$ . The first situation has received considerable interest since Peirce, who coined the word *abduction*, and characterized it as the third member of the triad of

<sup>2</sup>Both  $a(X)$  and  $a(t)$  denote sets of literals.  $X$  is a tuple of free variables,  $t$  is a tuple of ground terms such that  $X$  may be substituted for  $t$  in  $a(X)$ .

sylogistic reasoning (together with deduction and induction)<sup>3</sup>. The second situation (abduction of anomalous observations) has received only an occasional attention. It is a well established fact that monotonic theories cannot accommodate anomalous observations. For this reason, research in this direction must focus in abduction in nonmonotonic theories. For this reason, our final step is to propose a rule for abduction. This rule is based on several considerations (which cannot be discussed at length here because of space limitations). For this reason, we will give here only a short motivation. Given a nonmonotonic theory  $\mathcal{T}$  (i. e., a theory that may have defeasible rules), an abduction for an observation  $O$  should be a hypothetical explanation  $H$  that is compatible with  $\mathcal{T}$ , neither  $\mathcal{T}$  nor  $H$  should jointly (but not separately) explain  $O$ , and any other explanation  $H'$  should also explain  $H$  itself. Formally:

- (*Abd.*) From  $O_k$  to infer  $H_{S \cup \{k\}}$  iff
  1.  $H_{S \cup \{k\}} \cup \mathcal{T} \not\vdash \perp$ , ( $H$  is consistent with  $\mathcal{T}$ )
  2.  $\mathcal{T} \not\vdash O_k$ , ( $there is no argument for O in \mathcal{T}$ )
  3.  $H_{S \cup \{k\}} \not\vdash O_k$ , ( $there is no argument for O in H$ )
  4.  $\mathcal{T} \cup H_{S \cup \{k\}} \vdash O_k$ , ( $there an argument for O in \mathcal{T} \cup H$ )
  5. Any other set  $H'$  that satisfies the four conditions above is such that  $H' \cup \mathcal{T} \vdash H_{S \cup \{k\}}$  (i. e.,  $H$  is the most “shallow” explanation for  $O$ ).

**Example 2.2** Suppose that in a knowledge-based system we find the rules

$w(X) \succ i(X)$	<i>Normally if X has a work, then X receives an income.</i>
$w(X) \succ t(X)$	<i>Normally if X has a work, then X pays taxes.</i>
$w(X) \succ \neg s(X)$	<i>Normally if X has a work, then X does not study.</i>
$s(X) \succ w(X)$	<i>Normally if X studies, then X has a work.</i>
$c(X) \succ s(X)$	<i>Normally if X has a scholarship, then X studies.</i>
$c(X) \succ i(X)$	<i>Normally if X has a scholarship, then X receives an income.</i>
$c(X) \succ \neg t(X)$	<i>Normally if X has a scholarship, then X does not pay taxes.</i>

Given this, what can we expect about Scott, of whom we only know he pays taxes?

1	[	$t(\text{Scott})_{\{1\}}$	Premise
2	[	$(w(X) \succ t(X))_{\{2\}}$	Defeasible rule
3	]	$w(\text{Scott})_{\{1,2\}}$	1, 2, Abduction (Explanation)
4	[	$w(\text{Scott})_{\{1,2\}}$	3, Reit.
5	[	$(w(X) \succ i(X))_{\{3\}}$	Defeasible rule
6	]	$i(\text{Scott})_{\{1,2,3\}}$	4, 5, $\rightarrow_E$ (Prediction)
7	[	$w(\text{Scott})_{\{1,2\}}$	3, Reit.
8	[	$(w(X) \succ \neg s(X))_{\{4\}}$	Defeasible rule
9	]	$\neg s(\text{Scott})_{\{1,2,4\}}$	7, 8, $\rightarrow_E$ (Prediction)

<sup>3</sup>This shortest account of Peirce is surely unfair, since his purpose was much wider, for in his semiotic analysis of inference, abduction was central as the source of creativity and new knowledge [3].

By abduction, we can show that  $t(\text{Scott})$  because  $w(\text{Scott})$  (he pays taxes because he works), and from this inference, we can predict that he has an income, and that he does not study. It is a desirable feature here that further (iterated) abductions (for example,  $c(\text{Scott})$  because  $i(\text{Scott})$ ) are blocked for being inconsistent (see the next Section).

If we knew about another person, say *Kim*, of whom we know only that she receives an income, then we can generate two abductive explanations for her income. The first one,  $i(\text{Kim})$  because  $w(\text{Kim})$ , allow further predictions ( $t(\text{Kim})$  and  $\neg s(\text{Kim})$ ). The second one,  $i(\text{Kim})$  because  $c(\text{Kim})$ , allow other predictions ( $s(\text{Kim})$  and  $\neg t(\text{Kim})$ ). In this situation we have two unrelated explanations, of which we can not make any preference (again, see next Section). However, further knowing that, for instance,  $s(\text{Kim})$ , will block the first explanation in favor of the second.

### 3 Combining Defeaters

A final issue we wish to discuss is the defeat among various arguments in a nonmonotonic theory. This situation arises if we admit the possibility of *iterating* abductive explanations. In the Sect. 3, we introduced a “shallow” abductive operator, but it can be iterated to produce “deeper” explanations.

**Example 3.1** (After [4] and [5]). Suppose we have the following theory.

$$\mathcal{T} = \left\{ \begin{array}{ll} r(T) \succ\!\!-\! wr(T), & \text{(if it rains, the road is wet),} \\ r(T) \succ\!\!-\! wl(T), & \text{(if it rains, the lawn is wet),} \\ r(T) \succ\!\!-\! \neg s(T), & \text{(if it rains, it's not sunny),} \\ s(T) \succ\!\!-\! \neg r(T), & \text{(if it's sunny, it does not rain),} \\ so(T) \succ\!\!-\! wl(T), & \text{(if the sprinklers are on, the lawn is wet),} \\ s(T) \wedge h(T) \succ\!\!-\! so(T), & \text{(if it's sunny and hot, the sprinklers are on),} \\ wl(T) \succ\!\!-\! ws(T), & \text{(if the lawn is wet, the shoes are wet),} \\ wr(T) \succ\!\!-\! ws(T) \}. & \text{(if the road is wet, the shoes are wet).} \end{array} \right.$$

In this situation, suppose we observe that our shoes are wet ( $E = ws(\text{today})$ ). The possible (shallow) explanations for this are that either the road is wet, or that the lawn is wet, or both. However, none of these suffices to generate a “most specific” explanation.

To generate a more specific explanation we can iterate the abductive inference, that is, to generate a new “evidence” set  $E'$  that contains  $E$  plus any of the independently generated explanations, and then use this new context to try to generate a new abductive explanation. This procedure may be easy to formalize, but, as we will see, it may be that an argument is conflicting with some of these abductive hypotheses, and a criterion for combining defeat should be taken into account.

**Definition 3.1** Given a context  $\mathcal{T} \cup E$  with an underlying knowledge  $\mathcal{K}$ . Then

1. The set of argumentative supported conclusions  $A_c$  are generated from  $E \cup \mathcal{T}$ .
2. The set of abductive explanations  $B_c$  are generated from  $E \cup \mathcal{T}$ .
3. If there is a pair of contradictory literals  $a \in A_c$  and  $b \in B_c$ , then either
  - (I) Any argument  $A$  for  $a$  defeats any argument generated with  $b$ .

(II) Any argument generated with  $b$  defeats any argument generated with  $a$ .

(III) Any of the defeasible rules used in the arguments for  $a$  or in the explanations for  $b$  is defeated (syntactically blocked).

4. Firm conclusions  $C$  are the members of  $A_c$  and  $B_c$  that were not defeated in 3.

If we need to iterate abduction, then the firm conclusions  $C$  are added to the “evidence”  $E$ , and the process is repeated.

**Example 3.2** Suppose we are in the situation of Example 3.1, and we observe our shoes wet ( $ws(\text{today})$ ), and we remember that today it was sunny ( $s(\text{today})$ ). Then, what can we conclude? The most general abductive explanations are  $wr(\text{today})$  and  $wl(\text{today})$ . By the moment, any of these explanations is compatible with the observations and there is no defeat. If we iterate the abductive process, we find that  $r(\text{today})$  is explanation for  $wr(\text{today})$ , and  $r(\text{today})$  or  $so(\text{today})$  are explanations for  $wl(\text{today})$ .

Following strategy I, we assimilate  $so(\text{today})$  as the only tenable explanation for  $ws(\text{today})$ , that is, we conjecture that the sprinkler was on, it got the lawn wet, and then our shoes got wet. If we push this further, we can also conjecture that today it was hot in addition of being sunny.

Instead, if we follow strategy II, then the explanation  $r(\text{today})$  blocks our remembrance of being sunny. Then, our explanation now is that it rained, the rain got the road and the lawn wet, and then our shoes got also wet.

If we use strategy III, then both previous explanations are valid and compatible, and we reject the rules that mutually exclude  $r(\text{today})$  and  $s(\text{today})$ , that is, we suppose that today it may be hot and sunny at one time, and rainy at another.

## References

- [1] Alan R. Anderson and Nuel Belnap. *Entailment: The Logic of Necessity and Relevance*. Princeton University Press, 1975.
- [2] Gehrard Brewka. Cumulative Default Logic: in Defense of Nonmonotonic Default Rules. *Artificial Intelligence*, 50(2):183–205, 1991.
- [3] C. Hartshorne *et. al.* *Collected Papers of Charles S. Peirce*. Harvard University Press, Cambridge, MA, 1958.
- [4] Kurt Konolige. Abduction versus Closure in Causal Theories. *Artificial Intelligence*, 53(2-3):255–272, 1992.
- [5] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Mateo, California, 1988.
- [6] D. Poole, R. Goebel, and R. Aleliunas. Theorist: a Logical Reasoning System for Defaults and Diagnosis. In *The Knowledge Frontier*, pages 331–352, New York, 1987. Springer.
- [7] Guillermo R. Simari and Ronald P. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53(2-3):125–158, 1992.
- [8] G. A. W. Vreeswijk. Abstract Argumentation Systems. *Artificial Intelligence*, 90(2):225–279, 1997.

# Modelos Formales del Razonamiento Científico

Claudio Delrieux

Universidad Nacional del Sur, Bahía Blanca - ARGENTINA

claudio@acm.org

## 1 Introducción

El objetivo de esta línea de trabajo consiste en formalizar el razonamiento científico utilizando un lenguaje lógico. Dentro del método científico, especialmente dentro de las ciencias experimentales, se estableció un conjunto de procedimientos que permiten llevar adelante la explicación y predicción de fenómenos y la generación, corroboración y refutación de teorías. La formalización de estos procedimientos constituye uno de los principales objetos de estudio de la teoría de la ciencia. El conocimiento científico se ordena y configura en estructuras complejas. Las unidades de organización más destacadas dentro de esta estructura son las *teorías*. Las teorías científicas tienen la función de establecer conexiones sistemáticas dentro de un aspecto de la realidad. De ese modo es posible la inferencia de determinados hechos a partir de otros. Si el *tipo* de conocimiento que constituye una teoría científica fuese conocimiento verdadero justificado y el método científico fuese deductivamente válido, entonces no habría diferencia entre las teorías científicas y las lógicas. Sin embargo, las teorías científicas involucran tipos de conocimiento cuya justificación es problemática, y mecanismos de inferencia ampliativos. Por lo tanto estas teorías no tienen el *status* de ser deductivamente válidas.

Hempel [3] propone que la lógica de la predicción y de la explicación proceden según un mismo esquema  $\mathcal{T} \vdash e$ , donde  $\mathcal{T}$ , el *explanans* es un conjunto de leyes, y  $e$ , el *explanandum* es el fenómeno o hecho a explicar. La sistematización por medio de este esquema se denominó *paradigma hipotético-deductivo* o deductivo-nomológico, dado que el *explanans* constituye una pieza de conocimiento hipotético, del cual se debe deducir la evidencia. El procedimiento de inferir el *explanans* no puede ser deductivo, es decir,  $\mathcal{T}$  nunca puede ser *verdadera*. Una conclusión, señalada por Popper, es que las teorías científicas no se *verifican* sino que se *refutan*. Dicho de otra forma, no existe evidencia posible que garantice la verdad lógica de una teoría, pero una sola predicción o explicación incorrecta -aunque sea frente a una cantidad enorme de casos correctos- sirve para mostrar que una teoría es falsa.

Una propuesta pragmáticamente más adecuada que el paradigma H-D fue realizada por Lakatos en sus *programas de investigación* [4]. Lakatos observa que una teoría no puede ser absolutamente verdadera. Es más, una teoría puede ser falsa pero tener consecuencias verdaderas y operacionales. Por esta razón los programas de investigación reales permanecen abiertos y sujetos al cambio y la evolución. Esto determina que un programa pueda sobrevivir sin que por ello sea verdadero. Los programas de investigación son estructuras que incluyen a las teorías científicas, integrándolas con un conjunto de procedimientos de inferencia y que poseen un conjunto periférico de hipótesis auxiliares. El *núcleo* de un programa es un conjunto de conocimiento que se considera central, y que define la teoría como tal. Este núcleo es el conjunto de conocimientos (leyes, generalizaciones o postulados) que determina la identidad de la teoría y por consiguiente del programa mismo. El núcleo, por lo tanto, se considera definitivo, y el resto de la estructura del programa opera de modo tal de protegerlo

de la refutación. Esta protección consiste básicamente en implementar un *cinturón protector* de hipótesis auxiliares, que impiden que el núcleo sea refutado. De esa forma, existen dos procedimientos heurísticos para confrontar a la teoría con la evidencia de un resultado experimental  $e$ . Mientras no ocurran refutaciones, entonces se aplica la heurística positiva, en la cual se busca una sistematización del cinturón protector (como consecuencia del núcleo duro o por medio de nuevas leyes). Si el resultado  $e$  no es correctamente predicho o explicado por la teoría, entonces se aplica la heurística negativa, que consiste en encontrar una hipótesis  $c$  particular al caso  $e$  tal que de la teoría aumentada con  $c$  se siga  $e$ . Si dicha hipótesis no es compatible con el resto de la teoría, entonces algo en la misma deberá corregirse.

## 2 Formalizando los *programas* de Lakatos

En esta Sección propondremos una formalización de los *Programas*, en términos de los sistemas de razonamiento no monotónicos. También esbozaremos algunas ideas tendientes a comparar entre sí diferentes *programas* en función de algunas características ventajosas, como ser el progreso empírico y el éxito relativo, dejando planteadas otras posibilidades como por ejemplo el poder sistematizador, o la menor cantidad de hipótesis *ad hoc*. Esta formalización presupone que el conocimiento válido  $\mathcal{K}$  es conocido y universalmente aceptado por todos los programas, al igual que la evidencia  $E$ . Utilizaremos como mecanismo de inferencia subyacente a un sistema abstracto de razonamiento argumentativo [5, 6]. Es decir, las predicciones o explicaciones en los programas se obtienen construyendo argumentos.

**DEFINICIÓN 1** Dado un conjunto  $\mathcal{K}$  de conocimiento válido y un conjunto  $E$  de evidencia (literales de base), un programa de investigación  $P = \langle \mathcal{T}, C \rangle$  consta de una teoría no monotónica  $\mathcal{T}$  y un cinturón protector  $C$ . La teoría  $\mathcal{T}$  está representada por medio de un conjunto de condicionales derrotables  $a(X) \succ b(X)$ , y  $C$  por medio de literales de base que representan las hipótesis *ad hoc*. Un programa de investigación  $P = \langle \mathcal{T}, C \rangle$  predice (o explica) a un literal de base  $e$  cuando existe un argumento para  $e$  a partir de  $\mathcal{K}, \mathcal{T}, C$  y  $E$ .

Es importante recordar que la existencia de un argumento para un literal no es condición suficiente para asumir que la teoría argumentativa acepta a dicho literal como conclusión, dado que pueden existir argumentos en contra que lo derroten. Esto es análogo a lo que ocurre en un programa de investigación, donde el programa puede realizar predicciones para un observable, pero también predicciones contradictorias con el mismo observable. Por lo tanto deberíamos repasar cuáles son las “actitudes proposicionales” que la teoría argumentativa nos permite asumir respecto del literal  $e$ .

**DEFINICIÓN 2** Dado un programa  $P$  y un literal de base  $e$ , se pueden dar los casos:

	<i>Definitiva</i>	<i>Revisable</i>
<i>Aceptar <math>e</math></i>	<i>Argumentos solamente para <math>e</math></i>	<i>Argumentos tanto para <math>e</math> como para <math>\neg e</math> pero existe un argumento no derrotado para <math>e</math></i>
<i>Aceptar <math>\neg e</math></i>	<i>Argumentos solamente para <math>\neg e</math></i>	<i>Argumentos tanto para <math>e</math> como para <math>\neg e</math> pero existe un argumento no derrotado para <math>\neg e</math></i>
<i>Indeciso</i>	<i>No hay argumentos ni para <math>e</math> ni para <math>\neg e</math></i>	<i>Argumentos tanto para <math>e</math> como para <math>\neg e</math> pero no existen argumentos no derrotados</i>

Dado un programa  $P$  y un literal de base  $e$  el cual se confirma en la experiencia, se puede dar solamente uno de entre los siguientes casos:

Predicción de $P$	Status de $P$ luego de la confirmación de $e$
Aceptación definitiva de $e$	$P$ se confirma
Aceptación revisable de $e$	$P$ se confirma parcialmente
Aceptación definitiva de $\neg e$	Anomalía grave
Aceptación revisable de $\neg e$	Anomalía parcial
Indecisión definitiva	Hecho sorprendente
Indecisión revisable	"Laguna"

**EJEMPLO 1** Supongamos que nuestro conocimiento, evidencia, y suposiciones acerca de las propiedades de algunos animales es el siguiente:

$$\mathcal{K} = \{\text{pingüino}(X) \Rightarrow \text{ave}(X)\}.$$

$$\mathcal{T}_1 = \{\text{ave}(X) \succ\!\!-\text{vuela}(X), \text{pingüino}(X) \succ\!\!-\neg\text{vuela}(X), \text{en-avión}(X) \succ\!\!-\text{vuela}(X)\}.$$

$$E = \{\text{pingüino}(\text{Opus}), \text{ave}(\text{Pintín}), \text{ave}(\text{Tweety}), \text{pingüino}(\text{Schmuck}), \text{en-avión}(\text{Schmuck})\}.$$

Por el momento, nuestro programa no requiere hipótesis auxiliares  $C_1 = \{\}$ . En estas condiciones, las predicciones que hace nuestro programa  $P_1 = \langle \mathcal{T}_1, C_1 \rangle$  con el conocimiento subyacente  $\mathcal{K}$  y la evidencia  $E$  se basan en lo siguiente:

1. Existe un argumento para  $\text{vuela}(\text{Opus})$  (por ser ave), pero hay otro argumento para  $\neg\text{vuela}(\text{Opus})$  (por ser pingüino) que es más específico y que lo derrota. Por lo tanto  $P_1$  predice  $\neg\text{vuela}(\text{Opus})$ .
2. Existe un argumento para  $\text{vuela}(\text{Tweety})$ , por lo tanto  $P_1$  predice dicho literal.
3. Existe un argumento para  $\text{vuela}(\text{Pintín})$ , por lo tanto  $P_1$  predice dicho literal.
4. Existe un argumento para  $\text{vuela}(\text{Schmuck})$  (por estar en avión) y otro argumento para  $\neg\text{vuela}(\text{Schmuck})$  (por ser pingüino). Ambos argumentos se derrotan mutuamente, por lo que  $P_1$  no hace predicciones al respecto.

Supongamos ahora que realizamos observaciones acerca de los individuos voladores, encontrando la siguiente evidencia:

$$E_r = \{\neg\text{vuela}(\text{Opus}), \neg\text{vuela}(\text{Tweety}), \text{vuela}(\text{Gringa}), \text{vuela}(\text{Pintín}), \text{vuela}(\text{Schmuck})\}.$$

En dicho caso, podemos ver que nuestro programa  $P_1$  tiene el siguiente status frente a las nuevas observaciones: con  $\neg\text{vuela}(\text{Opus})$  tiene una confirmación parcial, con  $\neg\text{vuela}(\text{Tweety})$  tiene una anomalía grave, con  $\text{vuela}(\text{Gringa})$  tiene una observación sorprendente, con  $\text{vuela}(\text{Pintín})$  tiene una confirmación, y con  $\text{vuela}(\text{Schmuck})$  tiene una laguna.

Por el momento, no estableceremos diferencias entre confirmaciones y confirmaciones parciales, lagunas y hechos sorprendentes, y anomalías parciales o graves. La distinción será importante más adelante, cuando tratemos en la próxima Sección de formalizar algunos aspectos de la dinámica de los programas, es decir, cuándo y cómo el programa debe modificarse para adaptarse a las nuevas observaciones. De esa forma, denominaremos  $V$  al conjunto de casos donde la evidencia confirma total o parcialmente al programa  $P$ ,  $I$  las indeterminaciones o casos sorprendentes o lagunas para  $P$ , y  $F$  los casos refutatorios o anomalías parciales o graves. En el ejemplo anterior, podemos ver que  $V_1 = \{\neg\text{vuela}(\text{Opus}), \text{vuela}(\text{Pintín})\}$ ,  $I_1 = \{\text{vuela}(\text{Gringa}), \text{vuela}(\text{Schmuck})\}$ ,  $F_1 = \{\neg\text{vuela}(\text{Tweety})\}$ .

**DEFINICIÓN 3** Un programa  $P_1$  es empíricamente más progresivo que otro  $P_2$  siempre que tenga estrictamente mayor cantidad de casos confirmatorios. Es decir,

$$P_1 \prec_{prog} P_2 \Leftrightarrow V_2 \subset V_1.$$

$P_1$  es más exitoso que  $P_2$  si cada caso confirmatorio de  $P_2$  es también confirmatorio de  $P_1$ , cada caso refutatorio de  $P_1$  es también refutatorio de  $P_2$ , pero hay por lo menos un caso confirmatorio de  $P_1$  que no es confirmatorio de  $P_2$  o por lo menos un caso refutatorio de  $P_2$  que no es refutatorio de  $P_1$ . Es decir,

$$P_1 \prec_{ex} P_2 \Leftrightarrow V_2 \subseteq V_1, F_1 \subseteq F_2, \exists v \in V_1 - V_2 \vee \exists f \in F_2 - F_1$$

**EJEMPLO 2** Supongamos estar en una situación similar a la del Ejemplo 1. A partir de la evidencia previa  $E_d$  del ejemplo, un programa  $P_2$  edificado a partir una segunda teoría  $\mathcal{T}_2$  realizará las siguientes predicciones:

$$\mathcal{T}_2 = \{ave(X) \succ\text{---} vuela(X), pingüino(X) \succ\text{---} \neg vuela(X)\}.$$

1. Existe un argumento para  $vuela(Opus)$  (por ser ave), pero hay otro argumento para  $\neg vuela(Opus)$  (por ser pingüino) que es más específico y que lo derrota. Por lo tanto  $P_2$  predice  $\neg vuela(Opus)$ .
2. Existe un argumento para  $vuela(Tweety)$ , por lo tanto  $P_2$  predice dicho literal.
3. Existe un argumento para  $vuela(Pintín)$ , por lo tanto  $P_2$  predice dicho literal.
4. Existe un argumento para  $\neg vuela(Schmuck)$  (por ser pingüino), por lo que  $P_2$  predice dicho literal.

Si luego nos confrontamos a las mismas observaciones

$$E_r = \{\neg vuela(Opus), \neg vuela(Tweety), vuela(Gringa), vuela(Pintín), vuela(Schmuck)\},$$

podemos ver que el nuevo programa  $P_2$  tiene con  $\neg vuela(Opus)$  una confirmación parcial, con  $\neg vuela(Tweety)$  tiene una anomalía grave, con  $vuela(Gringa)$  tiene una observación sorprendente, con  $vuela(Pintín)$  tiene una confirmación, y con  $vuela(Schmuck)$  tiene otra anomalía grave. Entonces  $V_2 = \{\neg vuela(Opus), vuela(Pintín)\}$ ,  $I_2 = \{vuela(Gringa)\}$ ,  $F_2 = \{\neg vuela(Tweety), vuela(Schmuck)\}$ .

En estas condiciones, si bien  $P_1$  y  $P_2$  son igualmente progresivos, dado que coinciden sus casos confirmatorios,  $P_2$  es menos exitoso que  $P_1$  porque existe por lo menos una observación ( $vuela(Schmuck)$ ) que es anómala para  $P_2$  pero no para  $P_1$ .

### 3 La dinámica de los programas

En la Sección anterior identificamos seis posibles resultados de un programa  $P$  confrontado a un resultado experimental  $e_r$ . En esta Sección trataremos de dilucidar cuál debería ser la estrategia de un programa en función del status que asume frente a nuevos resultados experimentales. Si el programa se confronta con un hecho sorprendente, entonces debería recurrir a algún mecanismo de inferencia ampliativa para poder explicarlo (en este punto es donde se

requiere de un contexto de descubrimiento). En cambio, cuando los resultados son anómalos, entonces es necesario aplicar la heurística negativa, es decir, tratar de modificar las hipótesis periféricas del mismo para defender el núcleo de la refutación o para mantener progresivo al programa. Cuando existe confirmación total o parcial debería aplicarse la heurística positiva, es decir, tratar de aplicar el núcleo del programa para sistematizar también el cinturón protector, y de esa manera reducir la cantidad de hipótesis *ad hoc*. Finalmente, cuando el programa está indeterminado por tener una laguna, es decir, si el programa generó teorías o argumentos no definitivos tanto a favor como en contra de la nueva evidencia, entonces es discutible si alguna de las dos heurísticas debería ser aplicada. Una manera de asimilar la importancia de los mecanismos de inferencia que presentamos en los anteriores Capítulos consiste en verlos como posibles formalizaciones de estas estrategias heurísticas de un programa cuando se confronta con uno alguno de los casos mencionados.

**EJEMPLO 3** *Supongamos estar en la misma situación que en el Ejemplo 1 y deseamos desde el programa  $P_1$  encontrar explicación al hecho sorprendente  $vuela(Gringa)$ . Una estrategia para explicar un hecho sorprendente  $e$  consiste en encontrar por abducción [1, 2] algún hecho particular  $h$  el cual, junto con la teoría, permita inferir  $e$ . En este caso tenemos dos posibles explicaciones más específicas<sup>1</sup> encontradas por abducción:  $ave(Gringa)$  y  $en-avión(Gringa)$ , hipótesis que luego habrá que corroborar por la experiencia. Mientras que estas hipótesis no estén corroboradas, cualquiera de las dos (o ambas) pueden pasar a formar parte del cinturón protector  $C$ .*

**EJEMPLO 4** *Siguiendo nuevamente con el Ejemplo 1 desde el programa  $P_1$ . Tenemos en  $\neg vuela(Tweety)$  una anomalía grave. Sin embargo, podemos encontrar una explicación abductiva que corrija la anomalía. En efecto, si agregamos la hipótesis  $pingüino(Tweety)$  al programa, entonces el programa tendrá ahora la posibilidad de construir un argumento para  $\neg vuela(Tweety)$  que derrota al argumento para  $vuela(Tweety)$ , es decir, el programa se protege de la observación anómala.*

**EJEMPLO 5** *Siguiendo nuevamente con el Ejemplo 1 desde el programa  $P_1$ . Tenemos en  $vuela(Schmuck)$  una laguna. En esta situación es necesario o bien establecer un ranking entre los condicionales derrotados utilizados para generar los argumentos para  $vuela(Schmuck)$  y para  $\neg vuela(Schmuck)$ , o bien es necesario modificar alguno de los condicionales (por ejemplo,  $pingüino(X) \wedge en-avión(X) \succ vuela(X)$ ).*

## Referencias

- [1] Claudio Delrieux. Computational Theory of Science: Implementing Research Programmes. En *Proceedings of the IC-AI 2000 Conference*, págs. 775–783, CSREA Press, ISBN 1-892512-2, 2000.
- [2] Claudio Delrieux. The Rôle of Defeasible Reasoning in the Modelling of Scientific Research Programmes. En *IC-AI 2001 Conference*, págs. 861–868, ISBN 1-892512-81-5, 2001.
- [3] C. Hempel y P. Oppenheim. The Logic of Explanation. *Philosophy of Science*, 15:135–175, 1948.
- [4] Imre Lakatos. *Proofs and Refutations. The Logic of Mathematical Discovery*. Cambridge, 1976.
- [5] G. Simari y R. Loui. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence*, 53(2-3):125–158, 1992.
- [6] G. Vreeswijk. Abstract Argumentation Systems. *Artificial Intelligence*, 90(2):225–279, 1997.

---

<sup>1</sup>Es decir, la explicación que realiza la menor cantidad de suposiciones.

# On the Logic of Theory Change: Incision Functions from Selection Functions

Marcelo A. Falappa      Guillermo R. Simari

Artificial Intelligence Research and Development Laboratory  
Department of Computer Science and Engineering  
Universidad Nacional del Sur  
Av. Alem 1253 - (B8000CPB) Bahía Blanca - Argentina  
PHONE/FAX: (+54) 291 459 5136  
E-MAIL: [mfalappa,grs]@cs.uns.edu.ar

**Key Words:** Belief Revision, AGM Theory, Partial Meet Contractions, Kernel Contractions.

## Abstract

This work presents a connection between partial meet contractions (AGM: Alchourrón, Gärdenfors and Makinson) and kernel contractions (Hansson). We present a way to define incision functions (used in kernel contractions) from selection functions (used in partial meet contractions). We present some properties of both kinds of functions and their relations.

## 1 Introduction

Belief Revision systems are logical frameworks for modelling the dynamics of knowledge. That is, how we modify our beliefs when we receive new information. The main problem arises when that information is inconsistent with the beliefs that represent our epistemic state. In order to accept the new information we have to give up some information while preserving as much of the old information as possible.

There are a lot of frameworks to model the dynamic of knowledge but AGM is the most popular one. It defines three kinds of changes: expansion, contraction and revision. Contractions in AGM are called *partial meet contractions* and they are based on a selection among subsets of the original set that do not imply the information to be retracted. In order to make that, partial meet contractions use *selection functions*. On the other hand, *kernel contractions* [Han93] are based on a selection among the sentences that imply the information to be retracted. In order to make that, kernel contractions use *incision functions*. However, contraction is not the one kind of operator that can be defined in partial meet and kernel way. It is possible to define *prioritized revisions*, that make a kind of change in which the new information is always accepted. Also it is possible to define a *consolidation* [Han97] that makes a contraction of a set of sentences by a contradiction, a *semi-revision* [Han97] that makes a non-prioritized revision of a set of sentences by a single sentence, and a *revision by sets of sentences* [FKIS02] that makes a non-prioritized revision of a set of sentences by a set of sentences.

All these operators have a particular property: every partial meet operator is a kernel operator. Since every partial meet operator is defined in terms of a selection function and every kernel operator is defined in terms of an incision function we study the relation of these functions, giving a way to define incision functions from selection functions. Moreover, we present some properties for both kinds of functions and their relations.

## 2 Partial meet contractions

This kind of contraction is based on the concept of a remainder set.

**Definition 2.1** (Alchourrón, Gärdenfors & Makinson [AGM85]): Let  $K$  be a set of sentences and  $\alpha$  a sentence. Then  $K^\perp\alpha$  is the set of all  $K'$  such that  $K' \in K^\perp\alpha$  if and only if  $K' \subseteq K$ ,  $K' \not\vdash \alpha$  and if  $K' \subset K'' \subseteq K$  then  $K'' \vdash \alpha$ . The set  $K^\perp\alpha$  is called the *remainder set* of  $K$  with respect to  $\alpha$ , and its elements are called the  $\alpha$ -remainders of  $K$ . ■

For instance, if  $K = \{p, p \rightarrow q, r, r \rightarrow s, r \wedge s \rightarrow q, t \rightarrow u\}$  then the set of  $q$ -remainders is  $\{\{p \rightarrow q, r \rightarrow s, r \wedge s \rightarrow q, t \rightarrow u\}, \{p, r \rightarrow s, r \wedge s \rightarrow q, t \rightarrow u\}, \{p \rightarrow q, r, r \wedge s \rightarrow q, t \rightarrow u\}, \{p, r, r \wedge s \rightarrow q, t \rightarrow u\}\}$ . The set of  $v$ -remainders of  $K$  is equal to  $\{K\}$  since  $K \not\vdash v$ . The set of  $(p \rightarrow p)$ -remainders of  $K$  is  $\emptyset$  because  $p \rightarrow p \in \text{Cn}(\emptyset)$  and there is no subset of  $K$  failing to imply  $p \rightarrow p$ .

In order to define the a partial meet contraction operator, we need a selection function. This function selects sentences of  $K$  to be preserved and it is called selection function because it makes a selection among  $\alpha$ -remainders.

**Definition 2.2:** Let  $K$  be a set of sentences. A *selection function* for  $K$  is a function “ $\gamma$ ” ( $\gamma : \mathbf{2}^{2^{\mathcal{L}}} \Rightarrow \mathbf{2}^{2^{\mathcal{L}}}$ ) such that for any sentence  $\alpha \in \mathcal{L}$ , it holds that:

- 1)  $\gamma(K^\perp\alpha) \subseteq K^\perp\alpha$ .
- 2) If  $K^\perp\alpha = \emptyset$  then  $\gamma(K^\perp\alpha) = K$ . ■

For instance, given  $K = \{p, p \rightarrow q, r, r \rightarrow q, \neg s\}$  and  $\alpha = q$  then  $K^\perp\alpha = \{\{p, r, \neg s\}, \{p, r \rightarrow q, \neg s\}, \{p \rightarrow q, r, \neg s\}, \{p \rightarrow q, r \rightarrow q, \neg s\}\}$  and some possible results of  $\gamma(K^\perp\alpha)$  are  $\{\{p, r, \neg s\}\}$ ,  $\{\{p, r \rightarrow q, \neg s\}\}$ ,  $\{\{p, r, \neg s\}, \{p \rightarrow q, r \rightarrow q, \neg s\}\}$  and  $\{\{p, r \rightarrow q, \neg s\}, \{p \rightarrow q, r, \neg s\}, \{p \rightarrow q, r \rightarrow q, \neg s\}\}$ .

Partial meet contractions are defined as follows: given a set of sentences  $K$ , a sentence  $\alpha$  and a selection function  $\gamma$  for  $K$ , the *partial meet contraction* of  $K$  by  $\alpha$ , noted by  $K \div \gamma \alpha$ , is equal to  $\cap \gamma(K^\perp\alpha)$ . That is,  $K \div \gamma \alpha$  is equal to the intersection of the  $\alpha$ -remainders of  $K$  selected by  $\gamma$ .

## 3 Kernel contractions

This kind of contraction is based on the concept of a kernel set.

**Definition 3.1** (Hansson [Han93]): Let  $K$  be a set of sentences and  $\alpha$  a sentence. Then  $K^\perp\alpha$  is the set of all  $K'$  such that  $K' \in K^\perp\alpha$  if and only if  $K' \subseteq K$ ,  $K' \vdash \alpha$ , and if  $K'' \subset K'$  then  $K'' \not\vdash \alpha$ . The set  $K^\perp\alpha$  is called the *kernel set*, and its elements are called the  $\alpha$ -kernels of  $K$ . ■

For instance, given  $K = \{p, p \rightarrow q, r, r \rightarrow s, r \wedge s \rightarrow q, t \rightarrow u\}$  and  $\alpha = q$  then the set of  $\alpha$ -kernels is  $K^\perp \alpha = \{\{p, p \rightarrow q\}, \{r, r \rightarrow s, r \wedge s \rightarrow q\}\}$ . If  $K = \{p, p \rightarrow q\}$  then  $K^\perp(p \rightarrow p) = \{\emptyset\}$  because  $p \rightarrow p \in \mathcal{Cn}(\emptyset)$  and  $K^\perp \neg p = \emptyset$  since  $K \not\vdash \neg p$ .

In order to define the operator of kernel contraction we need to use an incision function. This function selects sentences of  $K$  to be removed and it is called incision function because it makes an incision in every  $\alpha$ -kernel.

**Definition 3.2:** Let  $K$  be a set of sentences. An *incision function* for  $K$  is a function “ $\sigma$ ” ( $\sigma : 2^{2^{\mathcal{L}}} \Rightarrow 2^{\mathcal{L}}$ ) such that for any sentence  $\alpha \in \mathcal{L}$ , the following hold:

- 1)  $\sigma(K^\perp \alpha) \subseteq \cup(K^\perp \alpha)$ .
- 2) If  $X \in K^\perp \alpha$  and  $X \neq \emptyset$  then  $(X \cap \sigma(K^\perp \alpha)) \neq \emptyset$ .

The limit case in which  $K^\perp \alpha = \emptyset$  then  $\sigma(K^\perp \alpha) = \emptyset$ . ■

For instance, taking  $K = \{p, p \rightarrow q, r, r \rightarrow s, r \wedge s \rightarrow q, t \rightarrow u\}$  and  $\alpha = q$  then  $K^\perp \alpha = \{\{p, p \rightarrow q\}, \{r, r \rightarrow s, r \wedge s \rightarrow q\}\}$  and some possible results of  $\sigma(K^\perp \alpha)$  are  $\{p, p \rightarrow q, r \rightarrow s\}$ ,  $\{p \rightarrow q, r \wedge s \rightarrow q\}$  and  $\{p, r\}$ .

Kernel contractions are defined as follows: given a set of sentences  $K$ , a sentence  $\alpha$  and a incision function  $\sigma$  for  $K$ , the *kernel contraction* of  $K$  by  $\alpha$ , noted by  $K \div \sigma \alpha$ , is equal to  $K \setminus (K^\perp \alpha)$ . That is,  $K \div \sigma \alpha$  can be obtained erasing from  $K$  the sentences cutted by  $\sigma$ .

## 4 Incision functions from selection functions

Now we will present a formal definition of a incision function from a selection function.

**Definition 4.1:** Let  $\gamma$  a selection function for a set  $K$ . We may define an incision function  $\sigma_\gamma$  as follows: if  $\beta \in K$  and  $\beta \notin \cap \gamma(K^\perp \alpha)$  then  $\beta \in \sigma(K^\perp \alpha)$ . ■

The intuition behind this definition is the following:  $\alpha$  is selected by a incision function  $\sigma$  (over the set of  $\alpha$ -kernels of  $K$ ) if  $\alpha$  does not belong to some  $\alpha$ -remainder of  $K$  selected by  $\gamma$ . For instance, if  $K = \{p, r, s, p \rightarrow q, r \rightarrow q, \neg t \rightarrow \neg q\}$  then  $K^\perp q = \{\{p, r, s, \neg t \rightarrow \neg q\}, \{p, s, r \rightarrow q, \neg t \rightarrow \neg q\}, \{s, p \rightarrow q, r \rightarrow q, \neg t \rightarrow \neg q\}, \{r, s, p \rightarrow q, \neg t \rightarrow \neg q\}\}$ . Suppose that  $\gamma(K^\perp q) = \{\{p, r, s, \neg t \rightarrow \neg q\}, \{r, s, p \rightarrow q, \neg t \rightarrow \neg q\}\}$ . Then the associated incision function is  $\sigma_\gamma(K^\perp q) = \sigma_\gamma(\{p, p \rightarrow q\}, \{r, r \rightarrow q\}) = \{p, p \rightarrow q, r \rightarrow q\}$ .

Now we present some plausible properties for selection functions in order to define incision functions. Some of them have been modified from [FKIS02] to be applied in our framework.

**Equitableness:** If  $K^\perp \alpha = H^\perp \alpha$  then  $\beta \in K \setminus \cap \gamma(K^\perp \alpha)$  if and only if  $\beta \in H \setminus \cap \gamma(H^\perp \alpha)$ .

**Weak Equitableness:** If  $X \in (K^\perp \alpha) \cap (H^\perp \alpha)$  and  $\beta \in X$  then  $\beta \in K \setminus \cap \gamma(K^\perp \alpha)$  if and only if  $\beta \in H \setminus \cap \gamma(H^\perp \alpha)$ .

The purpose of *equitableness* is that  $\sigma_\gamma$  be a well defined function. *Weak equitableness* tried to make “fair” to  $\sigma_\gamma$ , that is, if  $K$  and  $H$  share some  $\alpha$ -kernel, then  $\sigma_\gamma$  will make the same cut in both contractions (contraction of  $K$  by  $\alpha$  and contraction of  $H$  by  $\alpha$ ).

These two properties are needed when we are using *global selection functions* (called two-place selection function by Hansson [Han96]), that is, selection functions that can be applied to different sets of sentences. A global selection function  $\gamma$  has two arguments, one for the set

**Definition 4.2:** Let  $K$  and  $H$  be sets of sentences and  $\gamma$  a selection function for  $K$  and  $H$ . Then  $\gamma$  is an *equitable selection function* if  $K^\perp\alpha = H^\perp\alpha$  implies that  $K \setminus \cap\gamma(K^\perp\alpha) = H \setminus \cap\gamma(H^\perp\alpha)$ .

■

The intuition behind this definition is that, if the set of minimally subsets of  $K$  implying  $\alpha$  is equal to the set of minimally subsets of  $H$  implying  $\alpha$  then  $\beta$  is erased in the selection of  $\alpha$ -remainders of  $K$  if and only if  $\beta$  is erased in the selection of  $\alpha$ -remainders of  $H$ .

For example, given  $K = \{p, s, p \rightarrow q, s \rightarrow \neg t\}$ ,  $H = \{p, t, u, \neg v, p \rightarrow q\}$  and  $\alpha = q$ , then:

$$\begin{aligned} K^\perp\alpha &= \{\{s, p \rightarrow q, s \rightarrow \neg t\}, \{p, s, s \rightarrow \neg t\}\} \\ H^\perp\alpha &= \{\{t, u, \neg v, p \rightarrow q\}, \{p, t, u, \neg v\}\} \end{aligned}$$

We have that  $K^\perp\alpha = H^\perp\alpha = \{\{p, p \rightarrow q\}\}$ . Suppose that  $\gamma(K^\perp\alpha) = \{\{p, s, s \rightarrow \neg t\}\}$ . That is,  $\gamma$  selects only  $\alpha$ -remainders that does not contain to  $p \rightarrow q$ . If  $\gamma$  is an equitable selection function  $\gamma(H^\perp\alpha)$  must be equal to  $\{\{p, t, u, \neg v\}\}$ . Analogously, if  $\gamma(K^\perp\alpha) = \{\{s, p \rightarrow q, s \rightarrow \neg t\}\}$  and  $\gamma$  is an equitable selection function then  $\gamma(H^\perp\alpha)$  must be equal to  $\{\{t, u, \neg v, p \rightarrow q\}\}$ .

## 5 Conclusions and future work

We presented a way to define incision functions from selection functions. That means, if we have a partial meet contraction operator based on a selection function  $\gamma$ , we may define a kernel contraction operator based on a incision function  $\sigma_\gamma$  obtained from  $\gamma$ . The reverse process is not possible because there are some kernel contractions that are no partial meet contractions [Han93, Han96].

We defined some properties for these selection functions (equitableness, weak equitableness) and we hope to relate more properties such that transitivity, relationally [AGM85, Han96]) and maximality [Han96]. Moreover, we hope to define new properties to global selection/incision functions, that is, functions to be applied to more than one set of sentences.

## References

- [AGM85] Carlos Alchourrón, Peter Gärdenfors, and David Makinson. *On the Logic of Theory Change: Partial Meet Contraction and Revision Functions*. *The Journal of Symbolic Logic*, 50:510–530, 1985.
- [FKIS02] Marcelo A. Falappa, Gabriele Kern-Isberner, and Guillermo R. Simari. *Belief Revision, Explanations and Defeasible Reasoning*. Submitted to *Artificial Intelligence Journal*, 2002.
- [Han93] Sven Ove Hansson. *Kernel Contraction*. *The Journal of Symbolic Logic*, 1993.
- [Han96] Sven Ove Hansson. *A Textbook of Belief Dymanics: Theory Change and Database Updating*. Uppsala University, Department of Philosophy, Uppsala, Sweden, 1996.
- [Han97] Sven Ove Hansson. *Semi-Revision*. *Journal of Applied Non-Classical Logic*, in press, 1997.

# Planificación con Aprendizaje de Reglas Rebatibles para elección de Acciones utilizando Argumentación

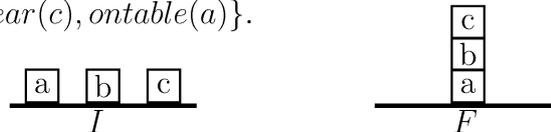
Diego García

Alejandro García

Departamento de Ciencias e Ingeniería de la Computación  
 Universidad Nacional del Sur. Av. Alem 1253, (8000) Bahía Blanca, Argentina  
 {dgarcia,agarcia}@cs.uns.edu.ar

En este trabajo se presenta un planificador, que durante la búsqueda de un plan, genera conocimiento en la forma de reglas rebatibles, para representar información de cuales acciones lo llevaron a encontrar un plan y cuales no. De esta forma, cuando se desea encontrar un plan para un nuevo problema, en lugar de explorar el árbol de búsqueda exhaustivamente, las reglas rebatibles generadas en planificaciones anteriores son utilizadas para argumentar sobre que acción aplicar en cada estado.

**Ejemplo 1 :** Supongamos que en el conocido “mundo de bloques” se quiere obtener un plan para hacer una pila con 3 bloques que están inicialmente sobre la mesa. En este ejemplo, la representación interna del estado inicial  $I$  esta dada por:  $I = \{clear(a), clear(b), clear(c), ontable(a), ontable(b), ontable(c)\}$  y el estado final o meta por:  $F = \{on(b, a), on(c, b), clear(c), ontable(a)\}$ .



Se asumirá por conveniencia un planificador que parte del estado inicial hacia el estado final, eligiendo en cada estado  $E$  de un conjunto de acciones posibles  $PA(E)$ , y explorando el árbol de búsqueda en profundidad. En el caso del Ejemplo 1 el conjunto de  $PA(I)$  es  $\{stack(b, a), stack(c, a), stack(a, b), stack(c, b), stack(a, c), stack(b, c)\}$ . Ante la ausencia de experiencia (reglas rebatibles), el planificador explorará todo el árbol de búsqueda. Aunque el planificador llegue a obtener un plan, seguirá explorando todo el árbol de búsqueda con el fin de obtener el plan óptimo y generar la mayor cantidad de experiencia posible (positiva y negativa) en la forma de reglas rebatibles.

En el proceso de exploración del árbol de búsqueda, cuando se llega a un nodo hoja  $N$ , se produce una recompensa positiva (+) o negativa (−) para premiar o penalizar las acciones que llevaron hasta ese nodo  $N$ . Esta recompensa se propaga luego hacia sus ancestros en el árbol, y se utiliza para generar reglas rebatibles que compilan la experiencia que obtuvo el planificador al ejecutar una acción en una situación particular. Cada nodo del árbol será etiquetado con un par  $(R, N)$ , donde  $R$  será la recompensa y  $N$  un número que representa un nivel del árbol. Cuando el nodo hoja corresponde al estado final  $F$ , se produce una recompensa positiva por haber encontrado un plan, y se etiqueta al nodo con  $(+, N)$ , donde  $N$  es el nivel de la hoja. Si la hoja, en cambio, es un nodo  $A$  ya visitado anteriormente (ancestro), se produce una recompensa negativa, ya que ese camino no conduce a un plan. En este caso se etiqueta con  $(-, N)$ , donde  $N$  es el nivel de dicho ancestro  $A$ .

Para cada nodo interno  $E$ , la etiqueta se calcula de la siguiente manera:

(a) Si existe al menos un hijo de  $E$  con recompensa positiva, entonces la etiqueta de  $E$  será  $(+, N)$  donde  $N$  es el mínimo nivel, entre las etiquetas de sus hijos con recompensa positiva. En este caso  $N$  corresponde a la longitud del plan mas corto que pasa por  $E$ .

(b) Si todos sus hijos tienen recompensa negativa, entonces la etiqueta de  $E$  será  $(-, N)$  donde  $N$  es el mínimo nivel entre las etiquetas de sus hijos.

En la Figura 1 puede verse el subárbol que se forma de elegir la acción  $stack(c, b)$  a partir del estado inicial  $I$  del Ejemplo 1. Cada nodo del árbol muestra la configuración de los bloques y está etiquetado con un nombre  $E_i$  : y el par  $(recompensa, nivel)$ . Los arcos están etiquetados con la acción que permite pasar del nodo padre al hijo.

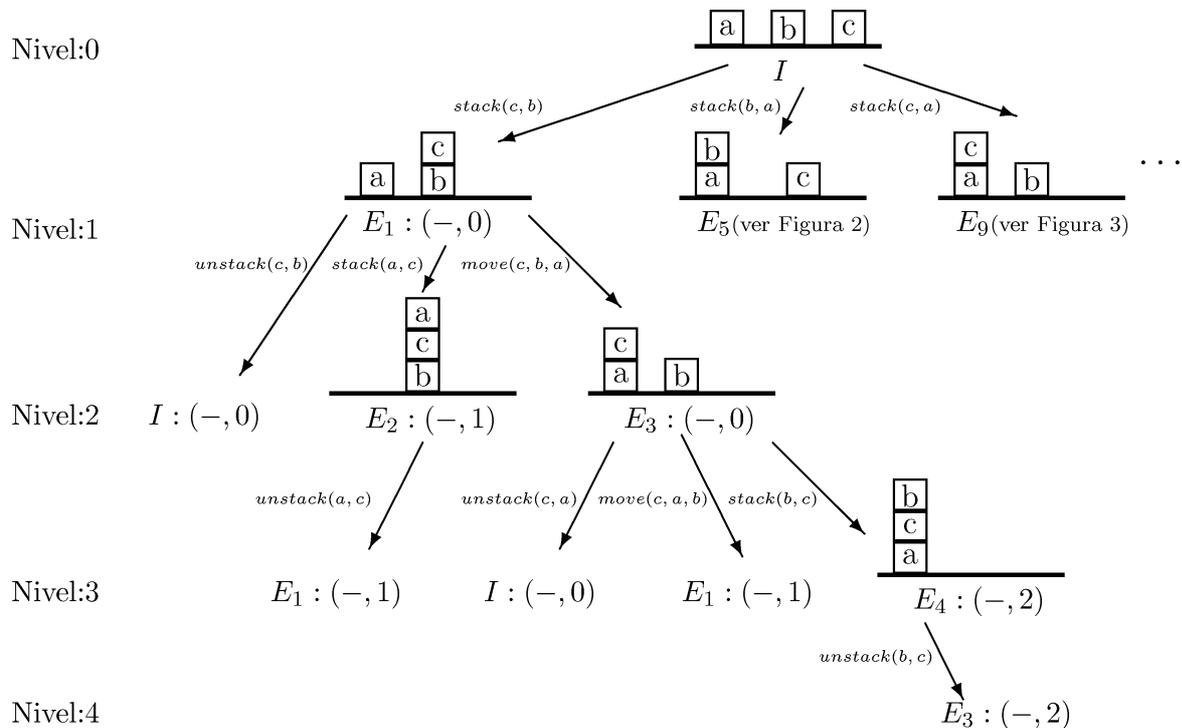


Figura 1: Parte del árbol de búsqueda del Ejemplo 1

En algunos nodos se generan reglas rebatibles que representan la experiencia de haber aplicado una acción  $Ac$ . Si la recompensa fue positiva se generará una regla de la forma “ $do(Ac) \prec G \cup E$ ”, y si fue negativa “ $\sim do(Ac) \prec G \cup E$ ”. Durante la exploración del subárbol de la figura 1, se generan las siguientes reglas:

- $$(R_1) \sim do(stack(a, c)) \prec goal(on(b, a)), ontable(a), ontable(b), clear(a), clear(c), on(c, b)$$
- $$(R_2) \sim do(stack(b, c)) \prec goal(on(b, a)), goal(on(c, b)), ontable(a), ontable(b), clear(a), clear(b), on(c, a)$$
- $$(R_3) \sim do(stack(c, b)) \prec goal(on(b, a)), goal(on(c, b)), ontable(a), ontable(b), ontable(c), clear(a), clear(b), clear(c)$$

Estas reglas se utilizarán para podar el árbol de búsqueda que se está explorando (Figura 3), y luego serán generalizadas como experiencia para construir planes en otros problemas similares (ver Ejemplo 2). La regla  $(R_1)$  se genera en el nodo  $E_1$ , y representa el conocimiento adquirido al explorar el subárbol con raíz  $E_2$ . La regla debe leerse de la siguiente forma: “si la búsqueda se encuentra en el estado  $E_1$ , definido por  $ontable(a), ontable(b), clear(a), clear(c), on(c, b)$ , y la meta restante es  $goal(on(b, a))$ , entonces hay razones para no apilar  $a$  sobre  $c$ ”. Esto se debe a que la ejecución de  $stack(a, c)$  en  $E_1$  lleva a la exploración de un subárbol cuyas hojas son todos nodos ya visitados, y descendientes de  $E_1$ . Esta situación se distingue por medio de la etiqueta  $(-, 1)$  del nodo  $E_2$ . El “-” en  $(-, 1)$  representa que todas las hojas del subarbol con

raíz  $E_2$  son nodos ya visitados, y el “1” representa que todos los nodos visitados se encuentran del nivel 1 hacia abajo, por lo tanto son todos descendientes de  $E_1$ , ya que  $E_1$  se encuentra en el nivel 1. La reglas  $R_2$  y  $R_3$  se generan en los estados  $E_3$  e  $I$  respectivamente, por razones análogas a las explicadas para  $R_1$ .

En general, si una acción  $Ac$  lleva de un estado  $S$  a un estado  $E$ , y  $E$  tiene una etiqueta  $(-, N)$ , entonces la acción  $Ac$  recibe una recompensa negativa ya que ningún descendiente de aplicar esa acción lleva a un plan. Si además  $N$  es menor o igual al nivel de  $S$ , entonces se generará una regla de la forma “ $\sim do(Ac) \prec Antecedente$ ”, donde *Antecedente* esta formado por las metas (goals) que restan por cumplir en  $S$ , unido a la representación de  $S$ . En la siguiente figura se muestra como se generan reglas rebatibles que representan experiencia *positiva*.

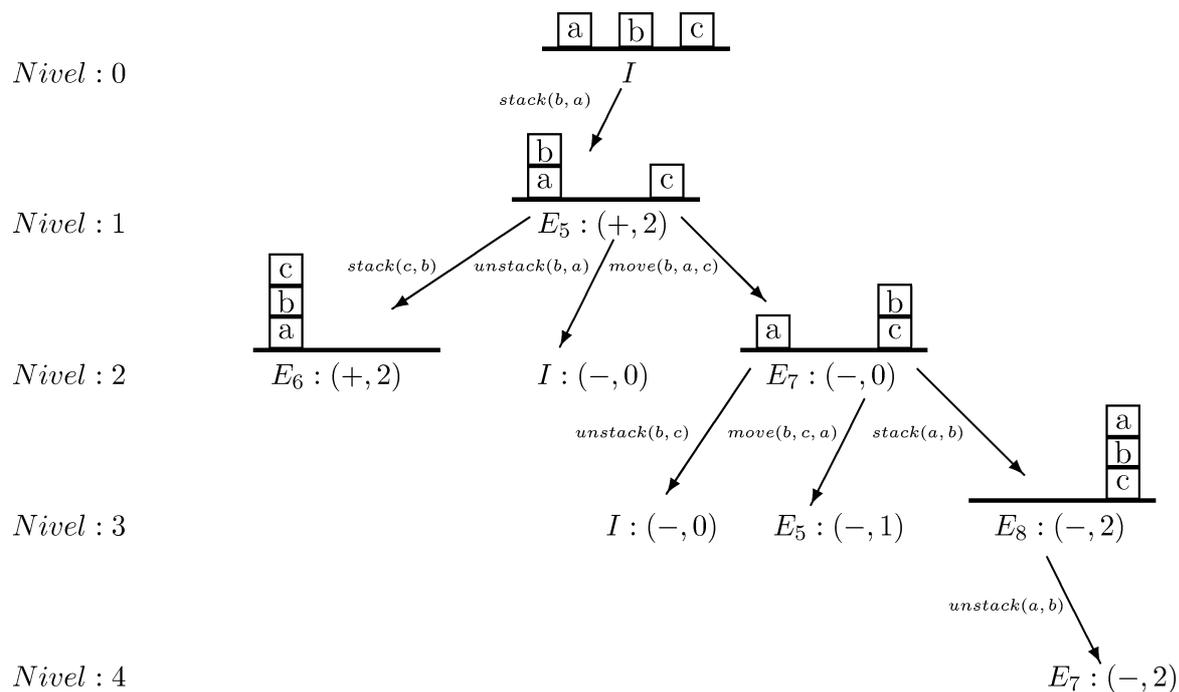


Figura 2: Parte del árbol de búsqueda del Ejemplo 1

A continuación se muestran las reglas generadas en el subárbol de la Figura 2:

$$(R_4) \text{ do}(\text{stack}(c, b)) \prec \text{goal}(\text{on}(c, b)), \text{ontable}(a), \text{ontable}(c), \text{clear}(b), \text{clear}(c), \text{on}(b, a).$$

$$(R_5) \sim \text{do}(\text{stack}(a, b)) \prec \text{goal}(\text{on}(b, a)), \text{goal}(\text{on}(c, b)), \\ \text{ontable}(a), \text{ontable}(c), \text{clear}(b), \text{clear}(c), \text{on}(b, c).$$

$$(R_6) \text{ do}(\text{stack}(b, a)) \prec \text{goal}(\text{on}(b, a)), \text{goal}(\text{on}(c, b)), \\ \text{ontable}(a), \text{ontable}(b), \text{ontable}(c), \text{clear}(a), \text{clear}(b), \text{clear}(c).$$

La regla  $(R_4)$  se genera en el nodo  $E_5$ , y representa el conocimiento adquirido al explorar el subárbol con raíz  $E_6$ , que es una hoja por ser el estado final  $F$ . A diferencia de las reglas generadas antes, esta regla representará conocimiento positivo, ya que este camino del árbol corresponde a un plan posible. La regla debe leerse de la siguiente forma: “si la búsqueda se encuentra en el estado  $E_5$ , definido por  $\text{ontable}(a), \text{ontable}(c), \text{clear}(b), \text{clear}(c), \text{on}(b, a)$ , y la meta restante es  $\text{goal}(\text{on}(c, b))$ , entonces hay razones para apilar  $c$  sobre  $b$  ( $\text{stack}(c, b)$ )”. Esto se debe a que la ejecución de  $\text{stack}(c, b)$  en  $E_5$  lleva a concretar un plan. La regla  $(R_5)$  se genera en el nodo  $E_7$  de forma análoga a lo explicado para  $(R_1)$ . La regla  $(R_6)$  en cambio, se genera en el nodo  $I$  de forma análoga a  $(R_4)$ .

La Figura 3 muestra otra parte del árbol de búsqueda del Ejemplo 1, donde se genera en el nodo  $I$  la regla  $(R_7) = \sim do(stack(c,a)) \prec goal(on(b,a), goal(on(c,b)), ontable(a), ontable(b), ontable(c), clear(a), clear(b), clear(c))$ . Además, el conocimiento generado hasta el momento (reglas  $R_1$  a  $R_6$ ) permiten podar el árbol de búsqueda. Observese que en el estado  $E_9$ , la acción  $stack(b,c)$  no es elegida porque tiene un argumento para no ejecutarla, generado con  $R_2$ .

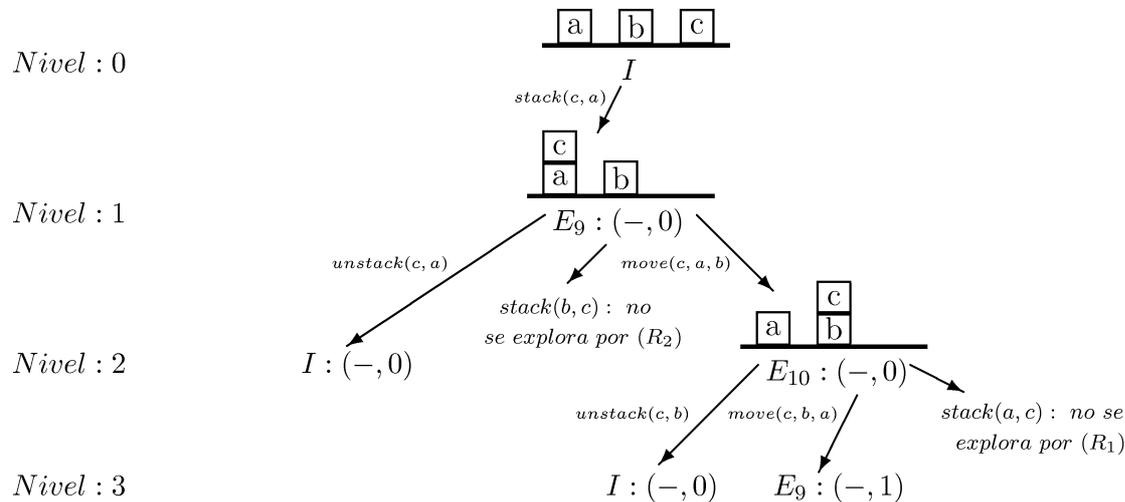


Figura 3: Parte del árbol de búsqueda del Ejemplo 1

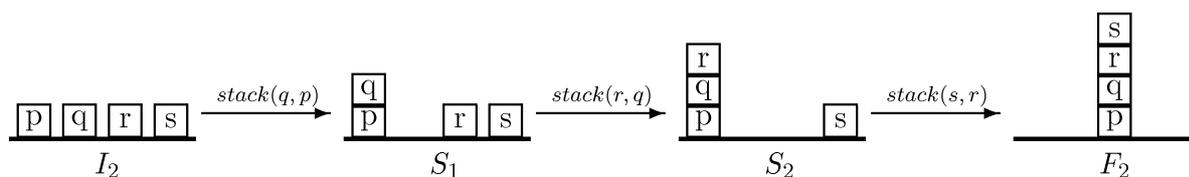
Una vez finalizada la exploración del árbol, las reglas rebatibles generadas son generalizadas y guardadas como experiencia para la búsqueda de futuros planes. Para generalizar una regla, se reemplaza cada átomo por una variable, y se elimina del cuerpo de la misma la descripción del estado, dejando solamente las metas. A continuación figuran las reglas obtenidas de generalizar las reglas  $R_1$  a  $R_7$ :

- $(G_1) \sim do(stack(X1, X3)) \prec goal(on(X2, X1))$
- $(G_2) \sim do(stack(X2, X3)) \prec goal(on(X2, X1), goal(on(X3, X2))$
- $(G_3) \sim do(stack(X3, X2)) \prec goal(on(X2, X1), goal(on(X3, X2))$
- $(G_4) do(stack(X3, X2)) \prec goal(on(X3, X2))$
- $(G_5) do(stack(X2, X1)) \prec goal(on(X2, X1), goal(on(X3, X2))$
- $(G_6) \sim do(stack(X1, X2)) \prec goal(on(X2, X1), goal(on(X3, X2))$
- $(G_7) \sim do(stack(X3, X1)) \prec goal(on(X2, X1), goal(on(X3, X2))$

## Utilización de la experiencia para encontrar planes

El conjunto de reglas rebatibles generadas por el planificador (como por ejemplo  $G_1$  a  $G_7$ ) define un programa lógico rebatible. Por lo tanto, puede utilizarse un intérprete de DeLP para razonar y generar argumentos a favor y en contra de ejecutar acciones. De esta forma, antes de explorar el árbol de búsqueda para un nuevo problema, el planificador utilizará el intérprete de DeLP para elegir, dentro de las acciones posibles a realizar en un estado, una acción  $Ac$  que esté garantizada a partir del conocimiento generado en planes anteriores. Esto es, se utilizará DeLP para buscar un literal de la forma  $do(Ac)$  que esté garantizado.

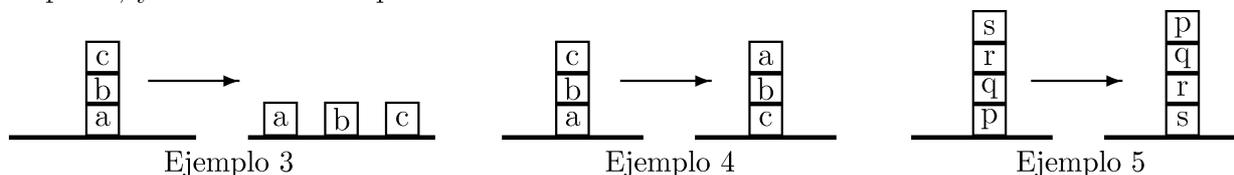
**Ejemplo 2 :** Supongamos que el planificador fue utilizado inicialmente para resolver el problema del Ejemplo 1, generando las reglas  $G_1$  a  $G_7$ , y se desea obtener ahora un plan para hacer una pila con cuatro bloques  $p, q, r$  y  $s$  que están inicialmente sobre la mesa. Utilizando la experiencia de las reglas rebatibles, el plan es generado directamente expandiendo únicamente cuatro nodos como muestra la siguiente figura:



La representación interna del estado inicial esta dada por:  $I_2 = \{clear(p), clear(q), clear(r), clear(s), ontable(p), ontable(q), ontable(r), ontable(s)\}$  y el estado final o meta por:  $F_2 = \{on(s, r), on(r, q), on(q, p), clear(s), ontable(p)\}$ . Las metas restantes a cumplir en el estado inicial, se obtienen como la diferencia entre el estado final y el inicial. Estas son:  $G = \{goal(on(s, r)), goal(on(r, q)), goal(on(q, p))\}$ . Sea  $\Delta$  el conjunto de reglas rebatibles generadas como experiencia de planes anteriores. A partir del programa lógico rebatible  $\Delta \cup G$ , el único literal garantizado por DeLP es  $do(stack(p, q))$  ya que el argumento que lo sustenta:  $\{do(stack(q, p)) \prec goal(on(q, p)), goal(on(r, q))\}$  que es generado usando la regla ( $G_5$ ), no está derrotado. Esto significa que dadas las metas  $G$  hay un argumento no derrotado, basado en la experiencia del planificador, para aplicar la acción  $stack(p, q)$ . Todas las demás acciones posibles de aplicar en el estado  $I_2$ , y que tienen algún argumento, son derrotados por un argumento en contra. Por ejemplo, el argumento  $\{do(stack(r, q)) \prec goal(on(r, q)), goal(on(s, r))\}$  generado con ( $G_5$ ) que sustenta  $do(stack(r, q))$  es derrotado por  $\{\sim do(stack(r, q)) \prec goal(on(q, p)), goal(on(r, q))\}$  generado utilizando la regla ( $G_3$ ). Una vez ejecutada la acción  $stack(q, p)$  se llega al estado  $S_1$  donde las metas restantes son  $\{goal(on(s, r)), goal(on(r, q))\}$ . Con estas nuevas metas como hechos el único literal garantizado por DeLP es ahora  $do(stack(r, q))$  sustentado por el argumento no derrotado  $\{do(stack(r, q)) \prec goal(on(r, q)), goal(on(s, r))\}$  generado también con ( $G_5$ ). Al aplicar esta acción se llega al estado  $S_2$  donde la meta restante es  $\{goal(on(s, r))\}$ . Ahora el único literal garantizado es  $do(stack(s, r))$  sustentado por el argumento  $\{do(stack(s, r)) \prec goal(on(s, r))\}$  generado con ( $G_4$ ). La acción  $do(stack(s, r))$  lleva directamente al estado final.

## Resultados Preliminares

A continuacion se muestran algunos ejemplos y el desempeño del planificador. Después de ejecutar el planificador para el Ejemplo 1 y el Ejemplo 3, si se lo ejecuta para el Ejemplo 4, el planificador encuentra en tres pasos el plan óptimo, utilizando sólo la experiencia y sin necesidad de explorar el árbol de búsqueda. Si se lo ejecuta para el Ejemplo 5, encuentra dos planes (de los cuales uno es el óptimo), solamente expandiendo ocho nodos del árbol de búsqueda, y utilizando la experiencia en seis de los mismos.



## Referencias

- [1] S. Džeroski, L. De Raedt, and H. Blockeel. *Relational Reinforcement Learning. The Eighth International Conference ILP'98*. Pages 11-22, Madison, Wisconsin, USA, 1998.
- [2] A. García. *Programación en Lógica Rebatible, Definición, Semántica Operacional y Paralelismo*. Tesis Doctoral. Universidad Nacional del Sur. 2000.
- [3] D. Kasakov and D. Kudenko. *Machine Learning and Inductive Logic Programming for Multi-Agent Systems Summer School of MAS*. Praga. 2001.

# Una Arquitectura Multiagentes para Argumentación Distribuida

Diego R. García   Alejandro J. García   Guillermo R. Simari

Departamento de Ciencias e Ingeniería de la Computación  
 Universidad Nacional del Sur  
 Av. Alem 1253, (8000) Bahía Blanca, Argentina  
 {drg, ajg, grs}@cs.uns.edu.ar

La finalidad de este proyecto es diseñar e implementar un sistema multiagentes que permita implementar en forma distribuida un sistema de Argumentación Rebatible [5, 1, 2, 3, 6, 4]. En la Argumentación Rebatible las conclusiones de un agente son sustentadas con *argumentos*. Una conclusión se dice *garantizada* cuando el argumento que la sustenta no posee *derrotadores* (contra-argumentos que lo derrotan) o todos sus derrotadores son a su vez derrotados. Dado un argumento  $\mathcal{A}$  pueden existir uno o varios derrotadores para  $\mathcal{A}$ . Cada derrotador ataca a un punto interno diferente del argumento el cual se denomina *punto de ataque*. La arquitectura aquí descripta puede utilizarse para implementar diferentes sistemas de argumentación rebatible que existen en la actualidad. En este trabajo se describe una implementación particular que utiliza DeLP (Defeasible Logic Programming) [4], un lenguaje que permite representar conocimiento con reglas rebatibles y estrictas, y las respuestas a las consultas son garantías construyendo un *árbol de dialéctica*.

El sistema multiagentes estará formado por un agente  $Ag$  que construye un argumento para sustentar una conclusión  $c$ , y un conjunto no determinado de *agentes derrotadores*  $AD_i$  cuyo objetivo es construir argumentos derrotadores para atacar argumentos de otros agentes. Los agentes compartirán cierto conocimiento  $\mathcal{K}$  que les permite construir argumentos y derrotadores. De esta forma, cuando se quiere decidir si una conclusión  $c$  está garantizada a partir del conocimiento  $\mathcal{K}$ , un agente  $Ag$  construirá un argumento  $\mathcal{A}$  para sustentar  $c$ , y luego, agentes derrotadores intentarán construir contra-argumentos para derrotar al argumento  $\mathcal{A}$  construido por  $Ag$ . Cada uno de estos agentes derrotadores se ejecuta independientemente del resto, y se concentrará en atacar al argumento  $\mathcal{A}$  en un punto diferente. Como para derrotar a un argumento, cada uno de los agentes derrotadores  $AD_i$  debe encontrar un contra-argumento  $\mathcal{C}$  que a su vez no sea derrotado, entonces el argumento  $\mathcal{C}$  puede a su vez ser atacado por otros agentes derrotadores recursivamente.

La siguiente definición extraída de [4] muestra el proceso dialéctico de argumentación de DeLP que permite decidir si una conclusión está garantizada. Un argumento  $\mathcal{A}$  para una conclusión  $h$  es denotado  $\langle \mathcal{A}, h \rangle$ . Un árbol de dialéctica para  $\langle \mathcal{A}_0, h_0 \rangle$ , se denota  $\mathcal{T}_{\langle \mathcal{A}_0, h_0 \rangle}$ , y se construye de la siguiente forma:

1. La raíz del árbol es etiquetada con  $\langle \mathcal{A}_0, h_0 \rangle$ .
2. Sea  $N$  un nodo del árbol etiquetado  $\langle \mathcal{A}_n, h_n \rangle$ , y  $[\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \dots, \langle \mathcal{A}_n, h_n \rangle]$  la secuencia de etiquetas del camino que va desde la raíz hasta el nodo  $N$ . Sean  $\langle \mathcal{B}_1, q_1 \rangle, \langle \mathcal{B}_2, q_2 \rangle, \dots, \langle \mathcal{B}_k, q_k \rangle$  todos los derrotadores de  $\langle \mathcal{A}_n, h_n \rangle$  (cada uno para un punto de ataque diferente). Para cada derrotador  $\langle \mathcal{B}_i, q_i \rangle$  ( $1 \leq i \leq k$ ), tal que, la línea de argumentación  $[\langle \mathcal{A}_0, h_0 \rangle, \langle \mathcal{A}_1, h_1 \rangle, \langle \mathcal{A}_2, h_2 \rangle, \dots, \langle \mathcal{A}_n, h_n \rangle, \langle \mathcal{B}_i, q_i \rangle]$  sea aceptable, existe un nodo hijo  $N_i$  de  $N$  etiquetado con  $\langle \mathcal{B}_i, q_i \rangle$ . Si no existe ningún derrotador  $\langle \mathcal{B}_i, q_i \rangle$  en tales condiciones, entonces el nodo  $N$  es una hoja.

Los nodos en un árbol de dialéctica se pueden marcar como “D” *derrotado*, o “U” *no derrotado*<sup>1</sup>. de la siguiente forma:

1. Todas las hojas de  $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$  se marcan con “U” en  $\mathcal{T}_{\langle \mathcal{A}, h \rangle}^*$ .
2. Sea  $N$  un nodo interno de  $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$ . El nodo  $N$  se marca con “U” si todo nodo hijo de  $N$  está marcado con “D”, y  $N$  se marca con “D” si existe al menos un nodo hijo de  $N$  marcado con “U”.

Una conclusión  $h$  sustentada por  $\mathcal{A}$  está garantizada si la raíz de  $\mathcal{T}_{\langle \mathcal{A}, h \rangle}$  es un nodo “U” (no derrotado).

La construcción del árbol de dialéctica y el proceso de marcado pueden realizarse en forma distribuida si, para un argumento dado, sus derrotadores son computados en paralelo. El modelo de argumentación distribuido que describimos aquí, está compuesto por un conjunto de sitios, que son esencialmente servidores que reciben consultas de otros sitios y las procesan concurrentemente. El sistema multiagentes propuesto se ejecuta sobre estos sitios. Estos sitios pueden encontrarse en una misma computadora (host) o en diferentes máquinas, y se comunican entre sí por medio de una red. Dentro de estos sitios se ejecutan concurrentemente los agentes derrotadores, que son creados como respuesta al pedido de un agente en otro sitio o en el mismo. De esta forma, los agentes estarán distribuidos en diferentes sitios, y cada uno de ellos se encargará de computar un nodo del árbol de dialéctica en forma concurrente.

## Generación distribuida del árbol de dialéctica

Para realizar el proceso de generación y marcado del árbol de dialéctica en forma distribuida, un agente  $Ag$  que ha construido un argumento  $\mathcal{A}$  para sustentar una conclusión  $h$ , calcula todos los puntos de ataque de  $\mathcal{A}$  (denotado  $PA(\mathcal{A})$ ). La búsqueda de derrotadores para los puntos de ataque la realizan los agentes derrotadores, los cuales se distribuyen entre todos los sitios del sistema, siguiendo alguna política de distribución. Esto es, para cada punto de ataque  $p_i \in PA(\mathcal{A})$ , se solicita, a través de un servicio especial, la creación de un agente derrotador  $AD_i$  en un sitio determinado. La función de este agente  $AD_i$  es atacar al argumento  $\mathcal{A}$  en el punto  $p_i$  que se le asigne, y luego informar al agente  $Ag$  si puede o no derrotar a  $\mathcal{A}$ . Una vez que se han distribuido todos los puntos de ataque a diferentes agentes derrotadores, el agente  $Ag$  debe esperar que los agentes derrotadores respondan. Pueden darse dos casos:

1. Si uno de los agentes  $AD_i$  responde afirmativamente, es decir,  $AD_i$  pudo construir un derrotador para  $\mathcal{A}$ , entonces:
  - (a) la conclusión  $h$  no se puede garantizar
  - (b) no es necesario seguir esperando la respuesta del resto de los agentes derrotadores, los cuales pueden ser eliminados para evitar cómputo innecesario (poda de un sub-árbol).
2. Si ninguno de los agentes  $AD_i$  responde afirmativamente, entonces  $h$  estará garantizado.

---

<sup>1</sup>En inglés *undefeated*

## Agentes Derrotadores

El objetivo de estos agentes es intentar derrotar un argumento  $\mathcal{A}$ , atacándolo en un punto dado  $p$ , como respuesta al pedido de otro agente. Para hacer esto un agente derrotador  $AD_i$  debe encontrar un derrotador  $\mathcal{D}$  para  $\mathcal{A}$ , que no sea derrotado. Para saber si  $\mathcal{D}$  es derrotado o no, el agente  $AD_i$  debe calcular los puntos de ataque de  $\mathcal{D}$ , distribuirlos entre los sitios del sistema, y esperar la respuesta. La distribución de estos puntos de ataque, crea nuevos agentes derrotadores, que pueden a su vez generar mas agentes derrotadores recursivamente. Si el derrotador  $\mathcal{D}$  encontrado no tiene puntos de ataque, entonces  $\mathcal{D}$  no puede ser derrotado y  $AD_i$  puede responder al agente que lo solicitó (i.e. a su padre) que derrota a  $\mathcal{A}$ . Si  $\mathcal{D}$  tiene puntos de ataque, una vez que  $AD_i$  los distribuyó, debe esperar que sus hijos (i.e. agentes solicitados por él) respondan. Pueden darse tres casos:

1. Si uno de estos agentes solicitados por  $AD_i$  responde que derrota a  $\mathcal{D}$ , entonces:
  - (a) el derrotador  $\mathcal{D}$  debe ser descartado,
  - (b) no es necesario seguir esperando la respuesta de los demás hijos que intentan derrotar a  $\mathcal{D}$ , y pueden ser terminados para evitar computo innecesario (poda de un sub-árbol).
  - (c)  $AD_i$  debe buscar para el punto de ataque  $p$  que le fue asignado, otro derrotador  $\mathcal{D}'$  para  $\mathcal{A}$  y verificar con este mismo proceso que  $\mathcal{D}'$  no sea derrotado generando agentes derrotadores para  $\mathcal{D}'$ .
2. Si todos los agentes solicitados por  $AD_i$  responden que no derrotan a  $\mathcal{D}$ , entonces  $AD_i$  puede responder, al agente que lo solicitó (padre), que **derrota** a  $\mathcal{A}$  en el punto  $p$ .
3. Si el agente  $AD_i$  no logra encontrar un derrotador para  $\mathcal{A}$  que no sea derrotado, entonces responde que **no derrota** a  $\mathcal{A}$  en el punto  $p$ .

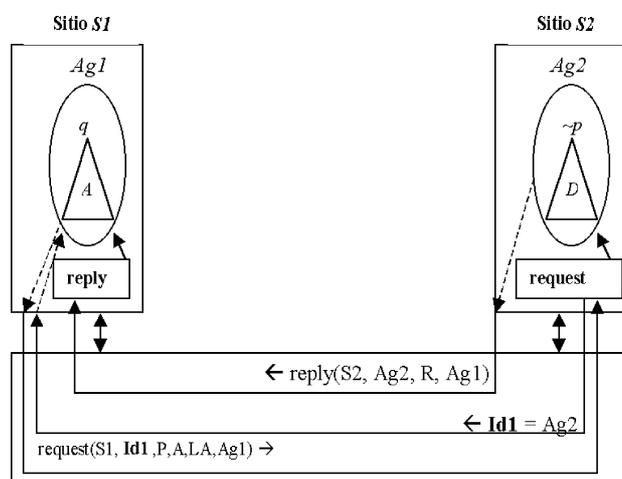
Dado que los agentes deben conocer quien los solicitó para poder responderles, entonces cada agente tendrá un identificador único dentro del sitio donde se esta ejecutando.

## Implementación

Para implementar el sistema multiagentes se utilizó BINPROLOG [7], un interprete de PROLOG con capacidad *multi-thread*. BINPROLOG permite ejecutar servidores, los cuales pueden atender concurrentemente consultas locales o remotas, y manejar un blackboard sincronizado. Los sitios donde se ejecutan los agentes fueron implementados como un proceso del interprete BINPROLOG en el cual se corre un servidor que escucha en un determinado port. A este se agrega un interprete de DeLP, y un conjunto de servicios especiales implementados en BINPROLOG para la comunicación y sincronización de los agentes. La dirección de un sitio está formada por la dirección de red del host donde se esta ejecutando mas el número de port donde escucha el servidor de BINPROLOG.

Los sitios pueden estar en una sola computadora, siempre y cuando escuchen en diferentes ports, o pueden correrse en diferentes maquinas que estén conectadas por una red. Si un sitio  $S_1$  necesita invocar algún servicio de otro sitio  $S_2$ , simplemente ejecuta remotamente en  $S_2$  el predicado que implementa el servicio. La información que se necesite enviar, como parte del pedido, viaja en los argumentos de dicho predicado

Si un agente  $Ag_1$  que se encuentra en el sitio  $S_1$  desea solicitar la creación de un agente derrotador  $Ag_2$  que intente derrotar a un argumento  $\mathcal{A}$  en un punto de ataque  $P$ , y  $Ag_2$  debe ejecutarse en el sitio  $S_2$ , entonces se ejecuta remotamente en  $S_2$  el predicado `request(D1, Id1, P, A, LA, Id2)`. El parámetro  $D1$  es la dirección del sitio  $S_1$ ,  $Id1$  el identificador del agente  $Ag_1$ ,  $P$  el punto de ataque,  $A$  es el argumento a derrotar,  $LA$  la línea de argumentación, y  $Id2$  el identificador del nuevo agente creado en  $S_2$  como respuesta al pedido. Al ejecutarse el predicado `request` se crea un nuevo agente derrotador  $Ag_2$  en un nuevo thread dentro del sitio  $S_2$ . Este nuevo agente intentará construir un derrotador para el punto  $P$  de  $A$ . Cuando el agente derrotador  $Ag_2$  finaliza su tarea en el sitio  $S_2$ , y quiere responder al agente  $Ag_1$  que lo creó desde el sitio  $S_1$  con dirección  $D1$ , entonces ejecuta remotamente en  $D1$  el predicado `reply(D2, Id2, R, Id1)`, donde  $D2$  es la dirección de el sitio  $S_2$ ,  $Id2$  es el identificador de  $Ag_2$ ,  $R$  es la respuesta (derrota o no derrota), y  $Id1$  es el identificador de  $Ag_1$ .



Como se explicó anteriormente, un argumento  $\mathcal{A}$  puede tener más de un punto de ataque, y para rotular a  $\mathcal{A}$  como derrotado o no derrotado, se debe esperar hasta que uno de sus agentes derrotadores responda que derrota a  $\mathcal{A}$  en uno de esos puntos, o hasta que todos sus agentes derrotadores respondan que no derrotan a  $\mathcal{A}$ . Por lo tanto, cuando un agente  $Ag_1$  construye un argumento  $\mathcal{A}$  se genera el conjunto de puntos de ataque y se distribuyen en los sitios disponibles creando los agentes derrotadores. El agente  $Ag_1$  almacena en su sitio la información sobre cada agente derrotador creado, y luego se suspende a la espera de las respuestas. Quien se encarga de despertar al agente cuando uno de los agentes derrotadores  $Ag_2$  responde es el predicado `reply(D2, Id2, R, Id1)` que realiza lo siguiente:

1. Recupera la información almacenada por  $Ag_1$  sobre  $Ag_2$ , utilizando  $D2, Id2, Id1$  y registra que  $Ag_2$  ha respondido  $R$ .
2. Si la respuesta es afirmativa  $R=$ yes entonces:
  - (a) despierta a  $Ag_1$  para informarle que ha sido derrotado
  - (b) para evitar computo innecesario, termina con todos los demás hijos (derrotadores) que aun no han respondido mediante el servicio `shutDownAgent`, y elimina del sitio  $S_1$  toda la información almacenada sobre los derrotadores generados para  $Ag_1$ .

3. Si la respuesta es negativa  $R=no$ , y ya respondieron negativamente todos los demás derrotadores entonces despierta a  $Ag_1$  para informale que no puede ser derrotado en ninguno de sus puntos de ataque.

Observe que el paso 2b corresponde a una poda del árbol de dialéctica que se realiza en forma distribuida. El servicio `shutDownAgent(Id)` se ejecuta en el sitio donde se encuentra el agente `Id` y antes de eliminar al agente `Id`, debe eliminar a todos los derrotadores que puedan haberse generado para `Id`, y así recursivamente. Con lo cual `shutDownAgent(Id)` elimina todo el posible sub-árbol de raíz `Id` que se pueda haber generado hasta ese momento, con el objetivo de rotular al argumento de `Id` como derrotado o no derrotado.

## Conclusiones

Las principales ventajas de esta arquitectura son eficiencia y robustez. Aunque la generación del árbol de dialéctica se realiza en profundidad, el paralelismo logrado por los agentes permite calcular cada rama del árbol en paralelo. Este paralelismo incorpora una componente de generación del árbol a lo ancho, que permite encontrar más rápidamente cualquier rama de pocos niveles que permita decidir si un nodo está derrotado. El sistema también resulta mas robusto, ya que si un sitio se cae, pueden reasignarse las tareas en otro sitio. La implementación realizada, ante la caída de un sitio, reasigna todos los agentes de ese sitio en otros que esten funcionando.

## Referencias

- [1] Grigoris Antoniou, Michael J. Maher, and David Billington. Defeasible logic versus logic programming without negation as failure. *Journal of Logic Programming*, 42:47–57, 2000.
- [2] A. Bondarenko, P.M. Dung, R.A. Kowalski, and F. Toni. An abstract, argumentation-theoretic approach to default reasoning. *Artificial Intelligence*, 93:63–101, 1997.
- [3] Phan M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and  $n$ -person games. *Artificial Intelligence*, 77:321–357, 1995.
- [4] Alejandro J. García. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional, y Paralelismo*. (<http://cs.uns.edu.ar/~ajg>). Universidad Nacional del Sur, Bahía Blanca, Argentina, 2000.
- [5] John Pollock. Implementing defeasible reasoning. *workshop on Computation Dialectics*, 1996.
- [6] Henry Prakken and Giovanni Sartor. Argument-based logic programming with defeasible priorities. *J. of Applied Non-classical Logics*, 7(25-75), 1997.
- [7] Paul Tarau. Intelligent Mobile Agent Programming at the Intersection of Java and Prolog. In *Proceedings of The Fourth International Conference on The Practical Application of Intelligent Agents and Multi-Agents*, pages 109–123, London, U.K., 1999.

## Social interaction and argumentation in MAS

Diego C. Martínez \* Alejandro J. García

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 - REPUBLICA ARGENTINA

Tel/Fax: (+54)(291)4595135/5136 - E-mail: {**dcm,ajg**}@cs.uns.edu.ar

A multiagent system (MAS) is made up of multiple interacting autonomous agents. It can be viewed as a society in which each agent performs its activity, cooperating to achieve common goals, or competing for them. Thus, every agent has the ability to do social interactions with other agents establishing dialogues via some kind of agent-communication language, under some communication protocol.

We think argumentation is suitable to model several kind of dialogues in multi-agents systems. It's being used by now as a form of negotiation between agents. Our current research activities are related to the use of argumentation in agent's interaction, such as negotiation among several participants, persuasion, acquisition of knowledge and other forms of social dialogue. Usually, argumentation appears as a mechanism to deal with disagreement between agents, for example when some conflict of interest is present. Argumentation can be used, not only to argue about something, but to know more about other agents: it is enough powerful to play an important role in general social interaction in multi-agents systems. For this reason, it is also important to define an *agent communication language* with special messages to use argumentation in several types of dialogues. According to Walton [DW99] dialogues can be classified in *negotiation*, where there is a conflict of interests, *persuasion* where there is a conflict of opinion or beliefs, *indagation* where there is a need for an explanation or proof of some proposition, *deliberation or coordination* where there is a need to coordinate goals and actions, and one special kind of dialogue called *eristic* based on personal conflicts. Except the last one, all this dialogues may exist in multi-agents systems as part of social activities among agents. Sometimes negotiation implies persuasion, sometimes persuasion requires indagation and all of them can take part in any coordination process. We think these dialogues are the basis of agent's social interaction, and what determines its social abilities: how much the agent knows about other agents in order to improve its behaviour in the society. All dialogues mentioned here need a strong, well defined protocols. Our aim is to establish several mechanisms to deal with all this kind of interactions through an argumentation formalism.

---

\*Becario de la Comisión de Investigaciones Científicas (CIC)

# 1 On dialogues

The process by which a group of agents communicate to one another to try to come to a mutually acceptable agreement on some matter is called *negotiation* [Weiss]. Negotiation can be used to resolve conflicts in a wide variety of multi-agents domains. [JFLPSW]. There are different techniques to reaching agreement in MAS, like game-theory and economic based techniques: strategic negotiation, auctions, coalition formation, contracting, etc [Kraus01]. The process of negotiation involves making proposals, trading options and making concessions to culminate, if possible, in an agreement [MS].

Several researchers in automated negotiation have proposed that argumentation is suitable to model negotiation between intelligent agents [AS99, MS]. Argumentation can be considered as a process in which the parties (in this case, agents) exchange arguments for or against some proposition, usually in order to persuade each other. An argument is a subjective explanation of some statement being alleged in the process.

There are a lot of argumentation models that have been developed inside Artificial Intelligence. Among these models, different formal systems of defeasible argumentation are defined, where arguments for and against a proposition are produced and evaluated to verify the acceptability of that proposition. In this manner, defeasible argumentation allows reasoning with incomplete and uncertain information. The development of this kind of systems has grown in the last years. Despite the variety, the main idea here is that any proposition will be accepted as true if there exist an argument  $A$  that support it, and this argument is acceptable according to an analysis between  $A$  and its counterarguments. Therefore, in the set of arguments of the system, some of them will be “acceptable” or justified arguments, while others not [?, GCS94, Pol94].

Note that argumentation is being used as a mechanism to achieve nonmonotonic reasoning, so we can also use argumentation to model the agent’s internal process of deliberation, for example to choose the right action to perform.

In this report, we will focus on the use of argumentation as a form of social interaction between agents. We show the most important aspects in multiagents systems in which argumentation can take place, such as negotiation and persuasion dialogues.

## 1.1 Negotiation scenario

We will base this brief discussion about social activities in MAS on a simple example describing a conflictive situation.

Suppose there are two agents  $A$  and  $B$  trying to access to some public resource  $R$ . This resource can be used only by one agent at a time and both agents claim for that resource at the same moment.

We can see this situation as a conflict between  $A$  and  $B$ , just because we know that  $R$  cannot be used by more than one agent at a time. Conflict is generally defined as a situation in which two agents hold different and incompatible goals. The agents then need to start some kind of dialogue to try to obtain this resource. The objective is to reach some reasonable settlement. We will call a *negotiation dialogue among arguments* to a sequence of pairs  $(Ag, Arg)$  where  $Arg$  is the argument announced by the agent  $Ag$ . The order of the pairs

Suppose agent  $A$  starts the negotiation. Then  $A$  will show the argument  $a_1$  supporting its desire of accessing  $R$ . Agent  $B$  will examine  $a_1$  and will produce the argument  $b_1$ , supporting its desire of accessing  $R$  and attacking some parts of  $a_1$ . Agent  $A$  will analyze  $b_1$  and then produce  $a_2$ , which is a new argument attacking  $b_1$ , and so on.

If agent  $B$  cannot produce a new argument on its own defense in the negotiation process, then  $B$  must end the dialogue accepting the initial request of  $A$  to access  $R$ . Formally, if the sequence of arguments presented in the negotiation dialogue is  $\{a_1, b_1, a_2, b_2, \dots, a_n\}$ , and  $B$  is not able to produce an argument against  $a_n$ , then  $A$  wins the negotiation, because its last supporting argument cannot be defeated by  $B$ . The sequence of arguments presented in the dialogue is called *argumentation line*. The set of all argumentation lines that can be produced starting with  $a_1$  is called the *dialectical tree* of  $a_1$ . It is possible to say that one of the goals of any agent involved in negotiation is to block the opponent's capability to produce better arguments. However, an agent is also allowed to attack other arguments in the argumentation line, if desired, and not necessarily the last one. This is the starting point of the exploration of a new argumentation line.

In nonmonotonic reasoning systems using an argumentation formalism, for any argument  $a$ , the entire dialectical tree must be analyzed in order to accept or reject the argument  $a$ . We said before that agent's defeasible reasoning can be achieved by argumentation. The main difference between an internal process of argumentation and the use of argumentation as a mechanism for implementing negotiation is that the former is concerned with the acceptance of any given proposition, defining a complete analysis between all the arguments for and against that proposition, while the latter is only concerned with the result of a given sequence of arguments. The notion of *justification of arguments* is only related to the internal process of reasoning of the agent.

What happens when there are more than two agents in the dialogue? Suppose we have  $n$  agents  $A_1, A_2, \dots, A_n$  trying to access the same resource  $R$ . Every agent  $A_i$  produces an argument  $a_i$  supporting its desire of accessing  $R$ . Each  $a_i$  is then in conflict with every  $a_j$ , ( $j \neq i, 1 \leq j, i \leq n$ ), so only one of the arguments can be accepted, say  $a_k$  and the access to  $R$  is granted for  $A_k$ . Some negotiation protocol is needed, for example, to establish the order in which the arguments are presented in the dialogue.

When negotiating, agents need to choose the best argument according to its goals and the current state of negotiation. This can also be done through internal argumentation processes. Choosing the wrong argument can lead to the lost of the debate. Obviously, this raw form of negotiation may not produce a *Pareto Optimal*<sup>1</sup> agreement for each agent. It depends on the intelligence of the participants to select better arguments to present to the dialogue. In order to do this, it is good for the agent to know something about other agents, their desires, goals and intentions. This is known as the *social knowledge* of the agent. The agents can use this information to guide its moves in the dialogue.

## 2 On agent's internal argumentation

An agent needs additional information in order to select an argument for the dialogue. Usually, arguments in negotiation are used to add or retract an intention or to change the preferences

---

<sup>1</sup>An agreement is Pareto Optimal if there is no other agreement that dominates it.

Suppose agent  $A$  starts the negotiation. Then  $A$  will show the argument  $a_1$  supporting its desire of accessing  $R$ . Agent  $B$  will examine  $a_1$  and will produce the argument  $b_1$ , supporting its desire of accessing  $R$  and attacking some parts of  $a_1$ . Agent  $A$  will analyze  $b_1$  and then produce  $a_2$ , which is a new argument attacking  $b_1$ , and so on.

If agent  $B$  cannot produce a new argument on its own defense in the negotiation process, then  $B$  must end the dialogue accepting the initial request of  $A$  to access  $R$ . Formally, if the sequence of arguments presented in the negotiation dialogue is  $\{a_1, b_1, a_2, b_2, \dots, a_n\}$ , and  $B$  is not able to produce an argument against  $a_n$ , then  $A$  wins the negotiation, because its last supporting argument cannot be defeated by  $B$ . The sequence of arguments presented in the dialogue is called *argumentation line*. The set of all argumentation lines that can be produced starting with  $a_1$  is called the *dialectical tree* of  $a_1$ . It is possible to say that one of the goals of any agent involved in negotiation is to block the opponent's capability to produce better arguments. However, an agent is also allowed to attack other arguments in the argumentation line, if desired, and not necessarily the last one. This is the starting point of the exploration of a new argumentation line.

In nonmonotonic reasoning systems using an argumentation formalism, for any argument  $a$ , the entire dialectical tree must be analyzed in order to accept or reject the argument  $a$ . We said before that agent's defeasible reasoning can be achieved by argumentation. The main difference between an internal process of argumentation and the use of argumentation as a mechanism for implementing negotiation is that the former is concerned with the acceptance of any given proposition, defining a complete analysis between all the arguments for and against that proposition, while the latter is only concerned with the result of a given sequence of arguments. The notion of *justification of arguments* is only related to the internal process of reasoning of the agent.

What happens when there are more than two agents in the dialogue? Suppose we have  $n$  agents  $A_1, A_2, \dots, A_n$  trying to access the same resource  $R$ . Every agent  $A_i$  produces an argument  $a_i$  supporting its desire of accessing  $R$ . Each  $a_i$  is then in conflict with every  $a_j$ , ( $j \neq i, 1 \leq j, i \leq n$ ), so only one of the arguments can be accepted, say  $a_k$  and the access to  $R$  is granted for  $A_k$ . Some negotiation protocol is needed, for example, to establish the order in which the arguments are presented in the dialogue.

When negotiating, agents need to choose the best argument according to its goals and the current state of negotiation. This can also be done through internal argumentation processes. Choosing the wrong argument can lead to the lost of the debate. Obviously, this raw form of negotiation may not produce a *Pareto Optimal*<sup>1</sup> agreement for each agent. It depends on the intelligence of the participants to select better arguments to present to the dialogue. In order to do this, it is good for the agent to know something about other agents, their desires, goals and intentions. This is known as the *social knowledge* of the agent. The agents can use this information to guide its moves in the dialogue.

## 2 On agent's internal argumentation

An agent needs additional information in order to select an argument for the dialogue. Usually, arguments in negotiation are used to add or retract an intention or to change the preferences

---

<sup>1</sup>An agreement is Pareto Optimal if there is no other agreement that dominates it.

of the opponent<sup>2</sup>. Basically, there are two types of information that contribute to this task:

- Opponent's mental state.
- Actual state of the dialogue.

The actual state of the dialogue is given by the set of arguments in the argumentation line. The agent must evaluate the last argument in the debate, and try to produce a counterargument for it. If the agent can not do this, then it's possible to evaluate an earlier argument in the argumentation line, and produce a counterargument as a refutation of the basis of the debate, because what is not refuted during the negotiation process is not necessarily accepted. Of course, this kind of actions depends on the intelligence of the agent and the negotiation protocol being used.

The agent may not have a total knowledge of the opponent's mental state. What is known by the agent about other agents depends on its *social ability*. Obviously, it is not easy to argue with a totally unknown agent, because the uncertainty of common goals, interests and beliefs makes the persuasion in the dialogue very difficult: agreement is always the basis of negotiation. Learning about others is a very important collateral effect of agents interaction. Every "encounter" between agents is an opportunity to know something about each other. This can also be done through argumentation, because every argument that an agent receives from another agent is an example of what this agent believes in. Every argument can be used then, not only to find disagreements, but to find common beliefs, which is useful to this and future dialogues.

## References

- [AG97] Alejandro García. *La programación en lógica rebatible. Su definición teórica y computacional* Tesis de Magister en Cs. de la Computación. Universidad Nacional del Sur. 1997.
- [AS99] A. Stankevicius. *Modelling Negotiation Protocols in a Dialectical Framework*. Proceedings del Primer Workshop de Investigadores en Ciencias de la Computacion (WICC 99). San Juan, 1999.
- [Kraus01] S. Kraus. *Automated Negotiations and Decision Making*. Course notes. Advanced Course on Artificial Intelligence ACAI-01, Prague, Czech Republic, July, 2-13, 2001.
- [Weiss] G. Weiss. *Multiagents Systems: A Modern Approach to Distributed Artificial Intelligence*. G. Weiss, editor. MIT Press. Cambridge, Massachusetts. 1999.
- [JFLPSW] Jennings, Faratín, Lomuscio, Parsons, Sierra and Wooldridge. *Automated haggling: Building artificial negotiators*. Pacific Rim International Conference on Artificial Intelligence, 2000.
- [MS] Motshegwa, T and Shroeder, M. *Negotiation with argumentation: how to argue efficiently*, ACAI 2001 - EASSS 2001 Proceedings. Czech Technical University. 2001.
- [GCS94] Alejandro García, Carlos Chesñevar, Guillermo Simari. *The Role of Dialectics in Defeasible Argumentation* Proceedings of the XIV International Conference of the Chilean Computer Science Society. Concepción, Chile, Noviembre de 1994.
- [Pol94] Poole, D.L. *Justification and defeat*. Artificial Intelligence, Vol. 67 N. 2 pp 377-407
- [DW99] Douglas Walton. The New Dialectic: A Method for Evaluating an Argument used Purpose in a Given Case. *Proto Sociology: An International Journal of Interdisciplinary Research*, Vol. 13, 1999. pp 70-91.

---

<sup>2</sup>This is called *persuasion*, which according to Walton [?] is another kind of dialogue, different from negoti-

## Reelaboración de Planes en Agentes Inteligentes. Operadores Básicos

**Gerardo PARRA**

Departamento de Informática y Estadística  
UNIVERSIDAD NACIONAL DEL COMAHUE  
e-mail: gparra@uncoma.edu.ar

**Guillermo R. SIMARI**

Departamento de Ciencias de la Computación  
UNIVERSIDAD NACIONAL DEL SUR  
e-mail: grs@cs.uns.edu.ar

**Palabras Clave:** INTELIGENCIA ARTIFICIAL, PLANEAMIENTO, DINÁMICA DE CREENCIAS

### Introducción

Los agentes inteligentes autónomos, por su proactividad, se ven obligados a considerar la satisfacción de sus metas a través de un conjunto estructurado de acciones que conforman un plan. El modelo BDI (*Belief, Desires and Intentions*)[9] para representar el conjunto cognitivo de un agente es una posibilidad interesante que permite estudiar el problema que introduce el dinamismo natural del entorno en el que un plan particular se desenvuelve.

El dinamismo del entorno provoca que algunos de los planes deban ser modificados para poder alcanzar las metas finales. Esta actividad de replaneamiento es, en realidad, una revisión del mismo. Ciertas partes pueden ser conservadas, pero otras deben ser removidas y reemplazadas por subplanes convenientes que ofrezcan la posibilidad de éxito para el plan global.

Este trabajo de investigación postula la conveniencia de adoptar el punto de vista del área de Dinámica de Creencias[5,6,8] al considerar la actividad de replaneamiento de un agente inteligente. El área de planeamiento en Inteligencia Artificial contiene una riqueza de propuestas que en general resulta confusa. El objetivo es obtener un modelo abstracto de plan que se avenga al tratamiento del replaneamiento mediante operadores de cambio. Esto brindará la posibilidad de estudiar de forma esquemática este problema.

### Reelaboración de Planes. Operadores Básicos

El objetivo central del área de *planning* en el contexto de Inteligencia Artificial es construir algoritmos que hagan posible a un agente elaborar un curso de acción para lograr sus metas. El resultado producido por un dispositivo de *planning* (*planner*) es una secuencia de acciones las cuales, cuando son ejecutadas en un mundo que satisface la descripción del estado inicial, lograrán

la obtención de la meta. En general, existe una amplia variedad de lenguajes para representar el mundo, las metas del agente y las acciones posibles. En este trabajo de investigación, adoptamos, en primera instancia, la representación STRIPS[1] como lenguaje de representación.

Uno de los más recientes dispositivos de planning es Graphplan[2,3,4]. El funcionamiento de Graphplan alterna entre dos fases: la *construcción del grafo de planning* y la *extracción de la solución*. La primera fase construye un *grafo de planning*, estructurado por niveles, hacia adelante en el tiempo hasta que se logra una condición necesaria (pero que puede ser insuficiente) para la existencia de un plan. Luego, la fase de extracción de solución realiza un recorrido hacia atrás sobre el grafo, buscando un plan que resuelva el problema. Si no es hallada una solución, el ciclo se repite mediante la construcción de un nuevo nivel del grafo de planning.

Graphplan determina un plan para un problema de planning de la siguiente manera. En primer lugar, construye el grafo de planning hasta que las metas del problemas aparezcan como nodos del grafo. Luego, realiza un recorrido hacia atrás sobre los nodos del grafo con el fin de encontrar un conjunto de acciones, no mutuamente excluyentes entre sí, que permitan lograr las metas a partir de las condiciones iniciales.

El objetivo de este trabajo de investigación es obtener un modelo abstracto de plan que se avenga al tratamiento del replaneamiento mediante operadores de cambio. Con tal motivación, hemos considerado las siguientes operaciones de cambio:

- **Expansión.** Este fenómeno tiene lugar cuando es necesario incorporar nuevas acciones al plan original para poder cumplir las metas.
- **Contracción.** Se produce cuando es necesario remover algunas piezas del plan global debido, por ejemplo, a la imposibilidad de llevarlas a cabo.
- **Revisión.** Tiene lugar cuando algunas partes del plan deben ser removidas y reemplazadas por subplanes convenientes que ofrezcan la posibilidad de éxito para el plan global.

En [10] se estableció un modelo para representar expansiones en grafos de planning. Se define un *operador de expansión a nivel  $i$* , que se comporta de la siguiente manera. Dado un grafo de planning y un esquema de acción, esta función devuelve un nuevo grafo de planning en el cual, la nueva acción  $a$  se halla presente en el nivel de acción  $i$ , las precondiciones y las poscondiciones de  $a$  se encuentran en los niveles  $i - 1$  e  $i + 1$ , respectivamente y, además, existen los arcos que vinculan a los nodos involucrados.

El operador de expansión de grafos de planning es caracterizado mediante un conjunto de postulados de racionalidad y es definido constructivamente[10]. La característica fundamental de este operador es que permite reutilizar, en gran medida, el grafo de planning original.

Con el fin de modelar la operación de contracción, hemos definido un *operador de contracción a nivel  $i$* [11] que toma un grafo de planning y un esquema de acción, y devuelve un nuevo grafo de planning en el cual, la acción  $a$  deja de estar presente en el nivel de acción  $i$  y, además, dejan de existir los arcos que vinculan a la acción  $a$  con sus precondiciones y sus poscondiciones.

En forma similar al caso de expansiones, el operador de contracción de grafos de planning es caracterizado mediante un conjunto de postulados de racionalidad y es definido de manera constructiva[11]. La definición de este operador de contracción hace posible la reutilización de gran parte del grafo de planning original.

## Comentarios Finales

La contribución principal de este trabajo de investigación es la introducción de un modelo para representar *operaciones de cambio* en grafos de planning. Hemos desarrollado un operador de expansión de grafos de planning, caracterizándolo mediante un conjunto de propiedades deseables[10]. Además, se ha introducido un operador de contracción para grafos de planning y hemos ofrecido un conjunto de postulados para tal operador[11].

Mediante los operadores de expansión y contracción es posible reutilizar gran parte del grafo de planning original. En trabajos futuros, se explorará la definición de operadores de revisión de grafos de planning, su caracterización mediante postulados de racionalidad y su definición constructiva.

## Referencias

- [1] R. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *J. Artificial Intelligence*, 2(3/4), 1971.
- [2] A. Blum and M. Furst. Fast planning through planning graph analysis. In *Proceedings of the XIV International Joint Conference of AI*, pages 1636-1642, 1995.
- [3] A. Blum and M. Furst. Fast planning through planning graph analysis. *J. Artificial Intelligence*, 90(1-2):281-300, 1997.
- [4] Daniel S. Weld. Recent Advances in AI Planning. *AI Magazine*, 1999.
- [5] Carlos Alchourrón, Peter Gärdenfors and David Makinson. On the Logic Of Theory Change: Partial Meet Contraction and Revision Functions. *The Journal of Symbolic Logic*, 50:510-530, 1985.
- [6] Peter Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. The MIT Press, Bradford Books, Cambridge, Massachusetts, 1988.
- [7] Gerardo Parra. *Semi Revisión Plausible en Bases de Creencias*. Tesis de Magister, Dep. de Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, 1998.
- [8] Sven O. Hansson. *A Textbook of Belief Dynamics*. Kluwer Academic Press, 1996.
- [9] M. Georgeff, B. Pell, M. Pollack, M. Tambe, and M. Wooldridge. The Belief-Desire-Intention Model of Agency. In J.P.Müller, M.P.Singh, and A.S. Rao, editors, *Intelligent Agents V* (LNAI Volume 1555), pages 1-10. Springer-Verlag: Berlin, Germany, 1999.
- [10] G. Parra, M. Falappa y G. Simari. Replaneamiento en Agentes Inteligentes. Parte I: Expansión. Enviado a *VII International Congress of Information Engineering (ICIE 2001)*. Buenos Aires. Abril de 2001.
- [11] G. Parra y G. Simari. *Replaneamiento en Agentes Inteligentes. Contracción de Grafos de Planning*. VII Congreso Argentino de Ciencias de la Computación. Volumen II, páginas 1081-1093. Octubre de 2001 - Universidad Nacional de la Patagonia Austral - El Calafate - Santa Cruz.

## La negociación como un proceso de búsqueda usando dialéctica

**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

BAHIA BLANCA - ARGENTINA

*La negociación es una actividad fundamental en un sistema multiagentes. Sus integrantes negocian para coordinar actividades y distribuir recursos y tareas, intentando alcanzar un estado aceptable para todos.*

*Esta línea de investigación se funda en percibir a la negociación como un proceso de búsqueda en un espacio de estados. La heurística de la búsqueda está basada en una dialéctica argumentativa. Los agentes realizan propuestas y contrapropuestas elaborando argumentos que intentan persuadir al resto de los participantes del proceso sobre la conveniencia de aceptar un estado particular. La actitud de los agentes frente a la negociación está ligada a su rol y a las relaciones que mantiene dentro de la organización.*

Los modelos de negociación están ligados a las características del sistema. Si todos los miembros forman parte de una organización y comparten objetivos comunes, la relación es colaborativa, aún cuando con frecuencia va a ser necesario interactuar para conciliar intereses. Alternativamente, en el sistema pueden interactuar agentes antagónicos con objetivos opuestos y la relación entonces será de competitividad. En cualquier caso, la negociación parte de la necesidad de coordinar demandas enfrentadas, en un contexto en el que ninguno de los participantes controla a los demás, pero sí puede influir en su comportamiento.

Cualquiera sea el contexto de la negociación, en el proceso intervienen tres elementos fundamentales: el protocolo, el objeto de negociación y el modelo de decisión. El protocolo de negociación es el conjunto de reglas que gobiernan la interacción. El objeto de negociación va a estar caracterizado por atributos y por las condiciones de aceptación para cada uno de ellos. El modelo de decisión es el mecanismo interno de toma de decisiones con el cual cada participante razona para intervenir de acuerdo a sus propios objetivos.

Los tres elementos están fuertemente ligados. La complejidad del modelo de decisión está vinculada al protocolo, a la naturaleza del objeto de negociación y a las operaciones que pueden realizarse sobre él. Las características del entorno determinan el peso relativo de cada uno de estos elementos y provocan a su vez que algunas formas de negociación puedan resultar más adecuadas que otras, de acuerdo a la situación particular en que se encuentren los agentes.

## La negociación como un proceso de búsqueda

La negociación puede pensarse como un proceso de búsqueda distribuida sobre un espacio de acuerdos potenciales. Este espacio de búsqueda no es accesible en su totalidad y cada agente colabora agregando nuevos estados que representan acuerdos, parciales o completos. La búsqueda no se realiza siguiendo un esquema convencional sino a través de un mecanismo dialéctico, basado en argumentación. El proceso no es lineal de modo que el espacio no se reduce hasta alcanzar la solución, sino que puede “moverse”, contrayéndose o ampliándose, al incorporar nuevos estados.

La estructura del objeto de negociación determina las dimensiones y la topología del espacio de búsqueda potencial. Cada atributo del objeto puede ligarse a una dimensión, de modo que agregar un atributo implicar agregar una dimensión a la búsqueda. Cambiar los valores de los atributos implica un movimiento dentro de la misma dimensión.

En la mayoría de los casos cada agente conoce sólo una parte del espacio de búsqueda y dentro de ella sólo hay una porción que satisface sus expectativas. Cada participante propone un conjunto específico de puntos, los posibles acuerdos, que le resultan aceptables. El conjunto puede contener un único punto, varios puntos distribuidos dentro del espacio, todos los puntos de una cierta región, varias regiones de puntos, *etc.*

Dentro del conjunto de puntos aceptables, no necesariamente todos tienen el mismo “valor relativo” para el agente, algunos valores son preferibles a otros. Los conjuntos pueden además, crecer o contraerse debido a cambios en el entorno o por la “persuación” ejercida por otro agente acerca de su visión. La negociación termina cuando los participantes encuentran un punto de acuerdo compartido o cuando el protocolo establece que la búsqueda terminó sin éxito.

Bajo esta metáfora la negociación requiere que los agentes tengan ciertas capacidades mínimas:

- Proponer un conjunto de puntos dentro de la región aceptable.
- Decidir si una respuesta le resulta aceptable

La experiencia de los agentes agregaría una capacidad adicional interesante que permitiría que los agentes realicen propuestas, al menos las iniciales, en base a las negociaciones anteriores.

El protocolo “tómalo o déjalo” es la forma de negociación más simple. Uno de los participantes ofrece un objeto con un valor fijo para uno de los atributos. El conjunto de puntos de acuerdo tiene entonces un único elemento. Los participantes oponentes pueden aceptar o rechazar la propuesta.

El protocolo de la “subasta” es algo más elaborado, en la alternativa en que varios participantes cumplen roles simétricos y existe uno que realiza una propuesta inicial para fijar el valor de un atributo. Los demás agentes compiten entre sí efectuando ofertas que

modifican el valor del atributo, hasta quedar fijo al no existir más contrapropuestas. Una situación más compleja, dentro del mismo protocolo, permite acordar los valores de dos atributos simultáneamente, dentro de intervalos especificados.

En el caso más simple de negociación la estructura y el contenido de los atributos que caracterizan al objeto puede ser fija y los participantes sólo pueden aceptar o rechazar la propuesta de acuerdo. Estos protocolos no resultan adecuados cuando la estructura del objeto de negociación es compleja o está caracterizado por diferentes atributos a balancear. El motivo es que el proceso de negociación basado en propuesta-aceptación o propuesta-rechazo puede resultar largo e ineficiente si no se logra un acuerdo inmediato. Asimismo, ninguna de estas formas de interacción es apropiada si durante el proceso se modifican las preferencias de los agentes debido a cambios en el entorno y/o en sus creencias.

### **Propuestas, críticas, contrapropuestas y argumentos**

Las aplicaciones en las que deben conciliarse varios atributos requieren de la especificación de objetos multidimensionales, protocolos de negociación flexibles y modelos de decisión complejos. Las capacidades mínimas de los agentes deben ampliarse e incluir cierta habilidad “social” para percibir las preferencias de los otros individuos y elaborar propuestas y contrapropuestas que las consideren.

El oferente debería iniciar la negociación proponiendo un punto que supone aceptable no sólo para sí mismo sino también para los demás. Si uno de los participantes no acepta la propuesta, debería especificar que parte de la misma no le resulta adecuada, o realizar una contrapropuesta que a su parecer pueda ser aceptada por el oferente.

Una oferta puede no ser adecuada porque algunos de los atributos no es negociable o algunos de los valores no resultan aceptables. En el caso más simple el receptor de la oferta realiza una “crítica” en la que especifica que parte de la propuesta rechaza de plano o debe modificarse.

En general, cuanto más información brinde cada participante, más definido y restringido queda el espacio de acuerdos, y más eficiente resulta el proceso. En particular, el receptor puede ofrecer una “contrapropuesta” que le resulte favorable. En la contrapropuesta puede no estar explícito que atributo no tiene el valor adecuado, de modo que el receptor tiene que inferirlo. Como contrapartida, el ofrecimiento es detallado y el espacio de acuerdos queda más restringido.

El proceso de negociación puede abreviarse considerablemente si cada individuo considera, además de sus propias necesidades y objetivos, su conocimiento acerca de las necesidades y objetivos de los demás. De este modo puede evitar hacer propuestas, ajustes o contrapropuestas que no van a ser aceptadas. En este sentido, cada agente actúa de acuerdo a “qué” es lo que está negociando y con “quién” lo está haciendo.

En cualquiera de los casos mencionados, las propuestas, críticas y contrapropuestas se refieren a los valores de los atributos de los objetos de negociación, sin indicar las razones que los motivan. En un proceso de negociación, la habilidad de uno de los participantes para “fundamentar” su actitud puede provocar un cambio en el comportamiento de los demás.

En una negociación más elaborada, uno de los participantes tiene capacidad para “persuadir” a los demás. Esto modifica el espacio de acuerdos, agregando oportunidades que supone favorables para todos. El espacio de acuerdo no se restringe entonces con la contrapropuesta, sino que cambia incluyendo nuevas alternativas.

El agente que realiza la contrapropuesta construye “argumentos” intentando justificar su posición o intentando persuadir a los demás con una alternativa que puede resultarles atractiva y que no habían contemplado. En todos los casos, la atención está focalizada en los objetos de negociación y sus atributos.

En un modelo de negociación aún más compleja, intervendrán aspectos vinculados al contexto social de los participantes. La actitud de cada agente frente a la negociación, su forma de justificar sus propuestas o su capacidad para persuadir a los demás, están ligadas a su rol, a sus relaciones, a ciertas convenciones y al conocimiento compartido.

### **El Contexto Social**

La interacción entre los agentes de un sistema puede mejorar considerablemente si el contexto brinda algunas “convenciones” que establecen acuerdos generales acerca del lenguaje a utilizar, su semántica y el comportamiento de los agentes. Estas convenciones permiten simplificar el proceso de decisión de cada agente porque reducen la incertidumbre respecto al comportamiento de los demás y eliminan conflictos de significado.

Algunas convenciones toman el carácter de “normas” que estructuran las interacciones y obligan a los agentes a actuar de acuerdo a ellas. El sistema se transforma entonces en una “organización”, en la cual cada agente tiene un comportamiento autónomo pero restringido.

Las normas establecen “compromisos” que fuerzan a los individuos a actuar de determinada manera estructurando su interacción. La organización impone penalidades a los agentes que no cumplen los compromisos y obligaciones durante su interacción.

Las actividades de una organización pueden dividirse en múltiples “escenarios” perfectamente separados y posiblemente concurrentes. Cada escenario involucra a un grupo de individuos que cumplen diferentes “roles”. Cada agente puede estar ligado a su vez a varios escenarios y en cada uno de ellos cumplir un rol diferente. En este modelo, toda interacción se desarrolla en el marco de un escenario particular, que brinda el conjunto de convenciones que las articulan.

Aunque los escenarios están separados, con frecuencia las acciones de uno de ellos afecta a los demás. El modelo debe vincular entonces a los diferentes escenarios en una estructura a través de la cual los agentes pueden navegar respetando la estructura. La estructura de escenarios permite otorgar significado a los argumentos intercambiados durante el proceso de negociación restringiendo el comportamiento de los agentes en dos niveles:

- El rol de cada agente en un escenario lo compromete a actuar de determinada manera respecto a lo que puede decir, a quién y cuando.
- Las conexiones entre los diferentes escenarios determinan los caminos que cada agente puede seguir para moverse.

Las acciones de un agente pueden tener consecuencias no locales que provocan un efecto sobre otros escenarios y condicionan sus acciones futuras. Un agente puede establecer un acuerdo dentro de un escenario que lo compromete para actuar de determinada manera en otro. Además, sus acciones pueden modificar la estructura de escenarios creando o bloqueando algunos caminos.

Durante el proceso de negociación los agentes establecen diferentes tipos de “relaciones sociales”, que pueden afectar considerablemente el nivel de persuasión y argumentación. Un individuo particular actuará de manera diferente según esté negociando con un par dentro de la misma empresa, un subalterno, un jefe o un competidor. Las relaciones sociales se establecen en función de los roles que cumplen los diferentes miembros de la organización.

### **Bibliografía Relacionada**

**García A., Chesñevar C., Simari G.,** *An argumentative framework for reasoning with inconsistent and incomplete information.* Proceedings of the ECAI 98. Workshop on Practical Reasoning and Rationality, 1998.

**Jennings N.R., Faratin P., Lomuscio A. R., Parsons M., Wooldridge M., Sierra C.** *Automated Negotiation: Prospects, methods and challenges.* Group Decision and Negotiation, 10:199-215, 2001

**Jennings N.R., Parsons S., Noriega P., Sierra C.** *A framework for argumentation-based negotiation.* In Intelligent Agents IV, pages 177-192, 1997.

**Kraus S., Nirkhe M., Sycara K.,** *Reaching agreements through argumentation: a logical model.* In DAY Workshop '93, pages 233-247, 1993

**Martínez D.** *Formalización del Conflicto entre Agentes: Argumentación y Derrota.* Tesis de Magister en Ciencias de la Computación, 2001.

**Simari G., Loui R. P.,** *A Mathematical Treatment of Defeasible Reasoning and its Implementation.* Artificial Intelligence 53, 125-157, 1992.

# Logical Properties in Defeasible Logic Programming

## – a preliminary report –

A. G. Stankevicius      M. Capobianco      C. I. Chesñevar

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial  
 Departamento de Ciencias e Ingeniería de la Computación  
 Universidad Nacional del Sur  
 Bahía Blanca - Buenos Aires - ARGENTINA  
 e-mail: {ags, mc, cic}@cs.uns.edu.ar

## 1 Introduction

Logics for nonmonotonic reasoning have often been described by the property they lack—that is, monotonicity—instead of by those they do enjoy. These theories flourished in the early ‘80s in response to the inconveniences incomplete and changing information posed to classic, monotonic approaches. Several nonmonotonic formalisms were introduced in the literature: inheritance networks, default logic, preferential entailment, autoepistemic logic, and defeasible argumentation among others. The introduction of these proposals in a short span of time made it difficult to decide which approach is best suited for a given context.

In a landmark paper, Dov Gabbay [3] pioneered the comparison of nonmonotonic theories with respect to a set of desirable properties. He endorsed focusing our attention on the properties of the inference relation induced by each system, that is, the relation between a conclusion and the set of premises supporting it. Further pursuing this approach, first Kraus *et al.* [5], and later Makinson [6], studied which set of core properties every nonmonotonic theory must have. These properties can be roughly divided in the so-called *pure* conditions, that solely depend on the inference relation, and those that also interact with the logical connectives. For any inference relation  $\vdash$  its corresponding *inference operator*, noted  $C$ , is defined as  $C(\Phi) = \{\phi \mid \Phi \vdash \phi\}$ . Assuming that  $\Phi$  and  $\Psi$  are sets of premises, the pure conditions can be defined in terms of this operator as follows:

**Inclusion:**  $\Phi \subseteq C(\Phi)$

**Idempotence:**  $C(\Phi) = C(C(\Phi))$

**Cut:**  $\Phi \subseteq \Psi \subseteq C(\Phi)$  implies that  $C(\Phi) \subseteq C(\Psi)$

**Cautious monotonicity:**  $\Phi \subseteq \Psi \subseteq C(\Phi)$  implies that  $C(\Psi) \subseteq C(\Phi)$

Inclusion is clearly desirable in the context of any sensible inference relation, as it stands for accepting the premises upon which we reason. Idempotence shows we have inferred as much as possible, in the sense that no new conclusions can be obtained from  $\Phi$  through the operator  $C$ . Cut ensures that adding known consequences into our set of premises does not give us new conclusions. Cautious monotonicity, the dual of cut, states the opposite: adding known

consequences into our set of premises does not make us loose conclusions. Those inference relations satisfying all these properties are called *cumulative*.<sup>1</sup>

Some authors have suggested that every nonmonotonic theories should be engineered around a cumulative inference relation. For instance, Kraus *et al.* [5] pushed this stance further claiming that “... [these properties] are rock-bottom properties without which a system should not be considered a logical system.” In what follows we study whether the entailment relations induced by a particular nonmonotonic theory are cumulative.

## 2 Our Study Case

*Defeasible Logic Programming* (DeLP, henceforth), combines a language similar to the one of logic programming with an argumentative inference engine [10, 9, 4]. DeLP structures knowledge using a set of facts, denoting the evidence at hand, also complemented with two meta-linguistic relations written as rules. On the one hand, *strict rules* represent uncontroversible information (*e.g.*, being a penguin is an uncontroversible reason for being a bird). On the other hand, *defeasible rules* represent disputable information (*e.g.*, being a bird is a disputable reason for being able to fly). Thus, a defeasible logic program  $\mathcal{P}$  is just a tuple  $(\Pi, \Delta)$  where  $\Pi$  is a set of facts and strict rules, and  $\Delta$  a set of defeasible rules. The conclusions endorsed by a given a program  $\mathcal{P}$  are obtained through a dialectical process based on three auxiliary inference relations. In the remainder of this section we consider what properties are met by each of these relations.

### 2.1 Strict and defeasible inference

The first inference relation we consider is called *strict inference* as it only concerns strict rules. In this relation, a conclusion  $\phi$  is inferred from a program  $\mathcal{P}$ , noted  $\mathcal{P} \vdash \phi$ , when it can be derived from  $\mathcal{P}$  simply by interpreting its strict rules as inference rules. The second relation, called *defeasible inference*, allows the use of both strict and defeasible rules in order to form conclusions. We write  $\mathcal{P} \rightsquigarrow \phi$  when a conclusion  $\phi$  is defeasibly inferred from a program  $\mathcal{P}$ . Even though these two relations look rather simple-minded, they play a fundamental role within DeLP: they are the building blocks of the more elaborate notions of argument and warrant.

Let  $C_{\vdash}$  and  $C_{\rightsquigarrow}$  be the consequence operators associated respectively to strict and defeasible inference, that is to say,  $C_{\vdash}(\Phi) = \{\phi \mid \Phi \vdash \phi\}$  and  $C_{\rightsquigarrow}(\Phi) = \{\phi \mid \Phi \rightsquigarrow \phi\}$ . The following proposition summarizes our findings about these relations:

**Proposition 2.1** *The consequence operators  $C_{\rightsquigarrow}$  and  $C_{\vdash}$  satisfy inclusion, idempotence, cut, and cautious monotonicity.*

It is straightforward to see that  $C_{\rightsquigarrow}$  and  $C_{\vdash}$  are both monotonic, and therefore cumulative. Any standard proof of monotonicity from classical logic also applies here.

It should be mentioned that the strict part of any defeasible logic program is by definition bound to be consistent with respect to strict inference, *i.e.*, no pair of contradictory literals can be derived.

---

<sup>1</sup>for an excellent survey of these and other abstract properties we refer the interested reader to [1].

## 2.2 Argument-based inference

The last auxiliary inference relation concerns the construction of *arguments*. Simply put, an argument is a tentative piece of reasoning supporting a certain conclusion. In this system, an argument is the subset of defeasible rules that allows to defeasibly derive a given conclusion. We say that  $\phi$  is an *argument-based inference* from a program  $\mathcal{P}$ , noted as  $\mathcal{P} \rightsquigarrow_{\mathcal{A}} \phi$ , if it is possible to build an argument for  $\phi$  using the defeasible rules in  $\mathcal{P}$ . The corresponding consequence operator for this relation is defined as  $C_{\mathcal{A}}(\Phi) = \{\phi \mid \Phi \rightsquigarrow_{\mathcal{A}} \phi\}$ . This inference relation greatly differs from the previous one.

**Proposition 2.2** *The consequence operator  $C_{\mathcal{A}}$  satisfies inclusion, but fails idempotence, cut, and cautious monotonicity.*

Inclusion trivially holds, given that every fact is supported by the empty argument (note that facts can be strictly derived). The failure of idempotence is shown next:<sup>2</sup>

**Example 2.1** Let  $\mathcal{P} = (\Pi, \Delta)$  be a DeLP program, where  $\Pi = \{a\}$  and  $\Delta = \{b \prec a; \sim b \prec a\}$ . Note that in this case  $C_{\mathcal{A}}(\{a\}) = \{b; \sim b\} \neq \emptyset = C_{\mathcal{A}}(C_{\mathcal{A}}(\{a\}))$ .

Finally, the following counterexamples show that argument-based inference does not satisfy cut nor cautious monotonicity.

**Example 2.2** Consider the DeLP program  $\mathcal{P} = (\Pi, \Delta)$ , where  $\Pi = \{b; d; \sim c \leftarrow a\}$  and  $\Delta = \{a \prec b, d; b \prec c; c \prec d\}$ . In this setting, it is possible to build the argument  $\mathcal{A}_1 = \{b \prec c; c \prec d\}$  for  $b$ . However, should  $b$  be added to  $\mathcal{P}$  as a fact, we now can obtain a new argument  $\mathcal{A}_2 = \{a \prec b, d\}$  for  $a$ , previously unavailable.

**Example 2.3** Let  $\mathcal{P} = (\Pi, \Delta)$  be a DeLP program, where  $\Pi = \{a\}$  and  $\Delta = \{b \prec a; \sim b \prec a\}$ . In this context, the arguments  $\mathcal{A}_1 = \{b \prec a\}$  for  $b$  and  $\mathcal{A}_2 = \{\sim b \prec a\}$  for  $\sim b$  can be constructed from  $\mathcal{P}$ . However, should  $b$  be added to  $\mathcal{P}$  as a fact, we can no longer argue  $\mathcal{A}_2$  (it now contradicts our strict knowledge). Thus, since  $\mathcal{A}_1$  was the only argument supporting  $\sim b$ , this conclusion is lost.

## 2.3 Warrant

The whole purpose of a DeLP program is obviously to establish what set of conclusions holds. This purpose is filled by the notion of *warrant*. Given certain program  $\mathcal{P}$ , a conclusion  $\phi$  is said to be *warranted on the basis of  $\mathcal{P}$* , noted  $\mathcal{P} \rightsquigarrow_{\mathcal{W}} \phi$ , if it is supported by a non-defeated argument constructed from  $\mathcal{P}$ . It should be stressed that in order to establish whether an argument is non-defeated, we must take into account all its potential counter-arguments—one of them might defeat our initial argument. Since these counter-arguments are in fact arguments, there may exist defeaters for the defeaters, and so on, thus requiring a complete recursive analysis. The set of warranted conclusions of a program  $\mathcal{P}$  represents its semantics.

In a like manner as before, the consequence operator for  $\rightsquigarrow_{\mathcal{W}}$  can be defined as  $C_{\mathcal{W}}(\Phi) = \{\phi \mid \Phi \rightsquigarrow_{\mathcal{W}} \phi\}$ . The next proposition summarizes our results in regard with this inference relation.

**Proposition 2.3** *The consequence operator  $C_{\mathcal{W}}$  satisfies inclusion but fails idempotence, cut, and cautious monotonicity.*

---

<sup>2</sup>for simplicity sake, we shall adopt propositional defeasible rules, despite the fact that these rules represent general knowledge and should contain variables.

It is easy to see why  $C_{\mathcal{W}}$  complies inclusion: just note that every fact in a given program is trivially supported by the empty argument, and that the empty argument cannot be defeated since it is entirely based on strict information. To show the failure of cut and cautious monotonicity we have found the following counter-examples:

**Example 2.4** We can use the same counter-example as before (example 2.2) to show that  $C_{\mathcal{W}}$  does not satisfy cut. Note that the argument  $\mathcal{A}_1 = \{b \multimap c; c \multimap d\}$  for  $b$  also warrants it on the basis of  $\mathcal{P}$ . However, should  $b$  be added to  $\mathcal{P}$  as a fact, then the new argument  $\mathcal{A}_2 = \{a \multimap b, d\}$  for  $a$  can be built, which now warrants  $a$  on the basis of the new program.

**Example 2.5** Consider a DeLP program  $\mathcal{P} = (\Pi, \Delta)$ , where  $\Pi = \{a\}$  and  $\Delta = \{c \multimap b; b \multimap a; \sim c \multimap a\}$ . The arguments  $\mathcal{A}_1 = \{c \multimap b; b \multimap a\}$ ,  $\mathcal{A}_2 = \{\sim c \multimap a\}$ , and  $\mathcal{A}_3 = \{b \multimap a\}$ , supporting respectively  $c$ ,  $\sim c$ , and  $b$ , can be all built from  $\mathcal{P}$ . Yet, only  $\sim c$  and  $b$  are warranted on the basis of  $\mathcal{P}$ , since  $\mathcal{A}_2$  defeats  $\mathcal{A}_1$ , but both  $\mathcal{A}_2$  and  $\mathcal{A}_3$  remain undefeated. Finally, should  $b$  be added to  $\mathcal{P}$  as a fact, the new argument  $\mathcal{A}_4 = \{c \multimap b\}$  for  $c$  now defeats  $\mathcal{A}_2$ . Thus, since  $\mathcal{A}_2$  was the only argument for  $\sim c$ , this previously warranted conclusion is now lost.

### 3 Conclusions

In this work we have studied the different inference relations defined within defeasible logic programming with respect to a set of desirable properties for any nonmonotonic reasoner. The results we have obtained are summarized in the following table:

	Inclusion	Idempotence	Cut	Cautious M.
$\vdash$	✓	✓	✓	✓
$\vdash$	✓	✓	✓	✓
$\vdash_{\mathcal{A}}$	✓	×	×	×
$\vdash_{\mathcal{W}}$	✓	×	×	×

Note that the enhanced inference relations satisfy less properties than the simpler ones. This stems from the fact that each inference relation is included in the previous one, that is to say, for a given DeLP program,  $C_{\vdash} \subseteq C_{\vdash} \subseteq C_{\mathcal{A}} \subseteq C_{\mathcal{W}}$ . Based on these properties, the notion of warrant does not seem to improve argument-based inference. However, the consequence operator  $C_{\mathcal{A}}$  fails to preserve consistency. That is to say, for a set of premises  $\Phi$  non contradictory with respect to strict inference,  $C_{\mathcal{A}}(\Phi)$  may become contradictory. On the contrary,  $C_{\mathcal{W}}$  does preserve consistency (a fundamental property for any logical system).

It is worth remarking that DeLP semantics (*i.e.*, the set of warranted conclusions) does not verify cumulativity. As we mentioned before, to some authors DeLP would not be deemed as a logical system. Nevertheless, DeLP has been successfully applied to a number of challenging scenarios with encouraging results. In addition, several others nonmonotonic theories do not hold cumulativity as well. For instance, Pollock's OSCAR [7], Prakken and Sartor's prioritized argumentation theory [8], or Vreeswijk's abstract argumentation framework [11]. These results show that cumulativity appears to be too restrictive for any sufficiently expressive entailment relation.

### References

- [1] ANTONIU, G. *Nonmonotonic Reasoning*. MIT Press, 1996.

- [2] CHESÑEVAR, C. I. *Formalización de los Procesos de Argumentación Rebatible como Sistemas Deductivos Etiquetados*. PhD thesis, Departamento de Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, Mar. 2001.
- [3] GABBAY, D. Theoretical foundations for nonmonotonic reasoning in expert systems. In *Logics and Models of Concurrent Systems*, K. Apt, Ed. Springer-Verlag, Berlin, Germany, 1985.
- [4] GARCÍA, A. J. *Programación en Lógica Rebatible: Lenguaje, Semántica Operacional, y Paralelismo*. PhD thesis, Departamento de Ciencias de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina, Dec. 2000.
- [5] KRAUS, S., LEHMANN, D., AND MAGIDOR, M. Non-monotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence* 44 (1990), 167–207.
- [6] MAKINSON, D. General patterns in nonmonotonic reasoning. In *Handbook of Logic in Artificial Intelligence and Logic Programming*, H. Gabbay and Robinson, Eds., vol. 3. Oxford University Press, 1994, pp. 35–110.
- [7] POLLOCK, J. L. *Cognitive Carpentry: A Blueprint for How to Build a Person*. The MIT Press, 1995.
- [8] PRAKKEN, H., AND SARTOR, G. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics* 7 (1997), 25–75.
- [9] SIMARI, G. R., CHESÑEVAR, C. I., AND GARCÍA, A. J. The Role of Dialectics in Defeasible Argumentation. In *Proceedings of the XIV Conferencia Internacional de la Sociedad Chilena para Ciencias de la Computación* (Concepción, Chile, Nov. 1994), Universidad de Concepción, pp. 111–121.
- [10] SIMARI, G. R., AND LOUI, R. P. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence* 53, 2–3 (1992), 125–157.
- [11] VREESWIJK, G. A. W. Abstract Argumentation Systems. *Artificial Intelligence* 90, 1–2 (1997), 225–279.

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

## WICC 2002

*16 y 17 de mayo de 2002*

### Computación Gráfica – Visualización



# Procesamiento Digital de Imágenes Utilizando PDICalc

L. Arlenghi, A. Vitale, C. Delrieux y G. Ramoscelli

*Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.  
Voice: (54)(291)5710098 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org*

**Palabras Clave:** PROCESAMIENTO DE IMÁGENES, PLANILLAS DE CÁLCULO.

## 1 Introducción

El Procesamiento Digital de Imágenes (PDI) es un área de creciente importancia tecnológica [3, 4]. Su objetivo general consiste en manipular señales bidimensionales (imágenes) para mejorar alguna de sus características. Podemos decir que aplicar PDI consiste en diseñar en forma iterativa una secuencia de procesos, eligiendo los mismos de acuerdo a su efecto sobre las imágenes intermedias, y manipulando los parámetros en forma conveniente [2, 6]. Muchas veces, sin embargo, estas complejas cadenas de procesos deben aplicarse a más de una imagen, lo cual resulta tedioso pues es necesario aplicar cada uno de los pasos con los mismos parámetros. Esto es especialmente notorio en el procesamiento de imágenes satelitales, por ejemplo de satélites temáticos multibanda, dado que estas operaciones deben realizarse para varias bandas en paralelo, lo cual hace que la tarea sea tediosa y sujeta a errores [7]. Todas estas dificultades llevan a pensar en buscar una manera flexible e intuitiva de plantear el diseño de un procesamiento de imágenes, especialmente para aquellos usuarios que utilizan el PDI de imágenes satelitales. En este trabajo presentamos el sistema PDICalc, el cual es una planilla de cálculo que permite manejar imágenes como si fuesen celdas de una planilla común. Esto permite acceder a cada paso del procesamiento y modificar los parámetros de cada paso individualmente, sin tener que repetir todo el proceso. También permite la reutilización de una secuencia de procesos o de sus partes a cualquier otra imagen, aplicar la misma secuencia en paralelo a varias celdas, y guardar las sesiones intermedias.

## 2 El sistema PDICalc

La idea principal de este sistema es sencilla pero efectiva: concebir al PDI como una planilla de cálculo, cuyas celdas están ocupadas por imágenes. Esto da origen al sistema PDICalc, el cual es, efectivamente, una planilla de cálculo para PDI [1]. El contenido de cada celda puede provenir de un archivo, o bien obtenerse por medio del cómputo de algún tipo de operación sobre otras celdas. La interfase gráfica de PDICalc es semejante en su operación a una planilla común, de manera que cualquier usuario se sienta cómodo con su uso. El editor de expresiones se activa cada vez que se “pincha” una celda, y permite programar el contenido de la misma. Por ejemplo `celda3=celda1+celda2` expresa que la imagen en la

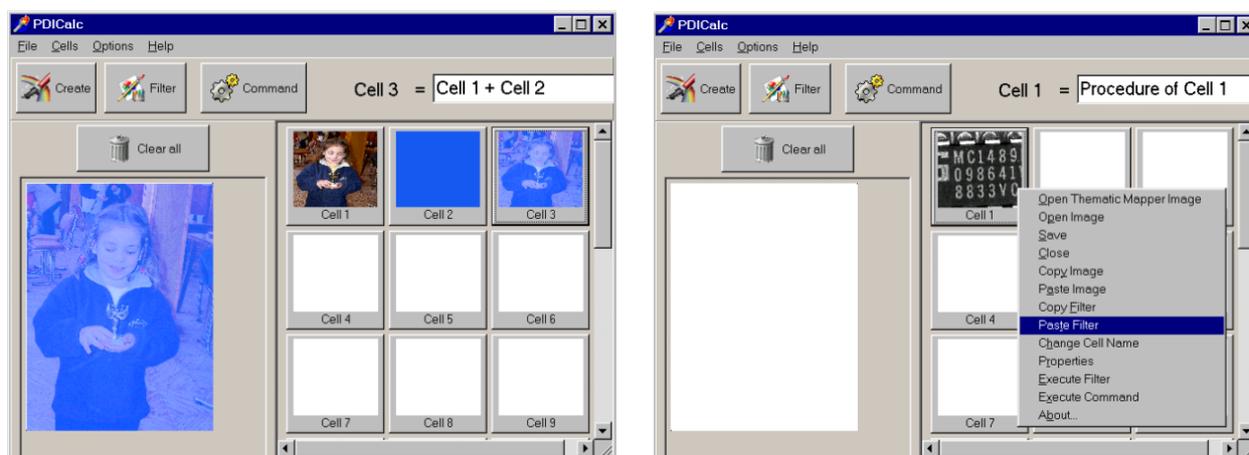


Figura 1: (a) Cómputo de una imagen en una celda como suma de otras dos celdas, y (b) opciones de edición.

celda 3 se obtiene computando pixel por pixel, en el espacio cromático RGB, la suma de los pixels de las celdas 1 y 2, ver Fig. 1(a).

Esta expresión también puede obtenerse “pinchando” la celda con el botón derecho, opción que despliega un menú en el cual se permite modificar el origen de una celda (archivo, ecuación computada a partir de otras celdas, valor constante, etc.), editar los parámetros de la imagen, los filtros, y varias otras que serán descritas a continuación (ver Fig. 1(b)). La gramática que permite formar expresiones para computar el contenido de una celda tiene como símbolos terminales  $Celda_i$ , y algunas constantes de imagen (imagen blanca, con un valor dado de gris, con un valor dado de RGB, etc.). Las operaciones aritméticas se interpretan como operaciones pixel a pixel en el intervalo entero  $[0..255]$  (el usuario tiene opción de decidir en cada caso de qué manera se cierra el álgebra, por ejemplo cuando un pixel se hace negativo, mayor que 255, etc.). La gramática incluye también la operación unaria *filtrar* que denota un filtro genérico (ecualización, convolución, morfológico, etc.), de manera de poder aplicar dicho procesamiento a una celda arbitraria. Muchos de estos filtros implican una conversión previa de la imagen al espacio cromático YIQ (por ejemplo para aplicar ecualización). La gramática permite combinar varias evaluaciones en una sola expresión, por ejemplo  $celda3=filtrar(celda1+filtrar(celda2-g127))$  denota la operación por la cual se le resta a la imagen en la celda 2 el valor de gris 127, al resultado se le aplica un filtrado (a elegir por medio del menú del botón derecho). A esta imagen se le suma la imagen en la celda 1, y al nuevo resultado se le aplica otro filtrado. Esta forma de proceder es posible, pero siempre es aconsejable descomponer estos procesamientos complejos en varias etapas simples (en nuestro caso con la secuencia  $celda4=celda2-g127$ ,  $celda5=filtrar(celda4)$ ,  $celda6=celda1+celda5$ ,  $celda7=filtrar(celda6)$ ).

La gramática permite expresiones circulares, como por ejemplo  $celda1=filtrar(celda1)$ . Este tipo de casos son útiles y necesarios cuando es necesario aplicar un procesamiento iterativo una cantidad indeterminada de veces (varias erosiones o varias dilataciones, por ejemplo). Por dicha razón hemos elegido que la evaluación de las expresiones sea *explícita* (por medio del botón *comando* en la GUI de la planilla) al contrario de lo que ocurre en las planillas de cálculo convencionales. De esa forma el usuario puede observar no solo el resultado de ir aplicando iterativamente estos filtros, sino el efecto que se produce cuando a la imagen obtenida se le aplica un

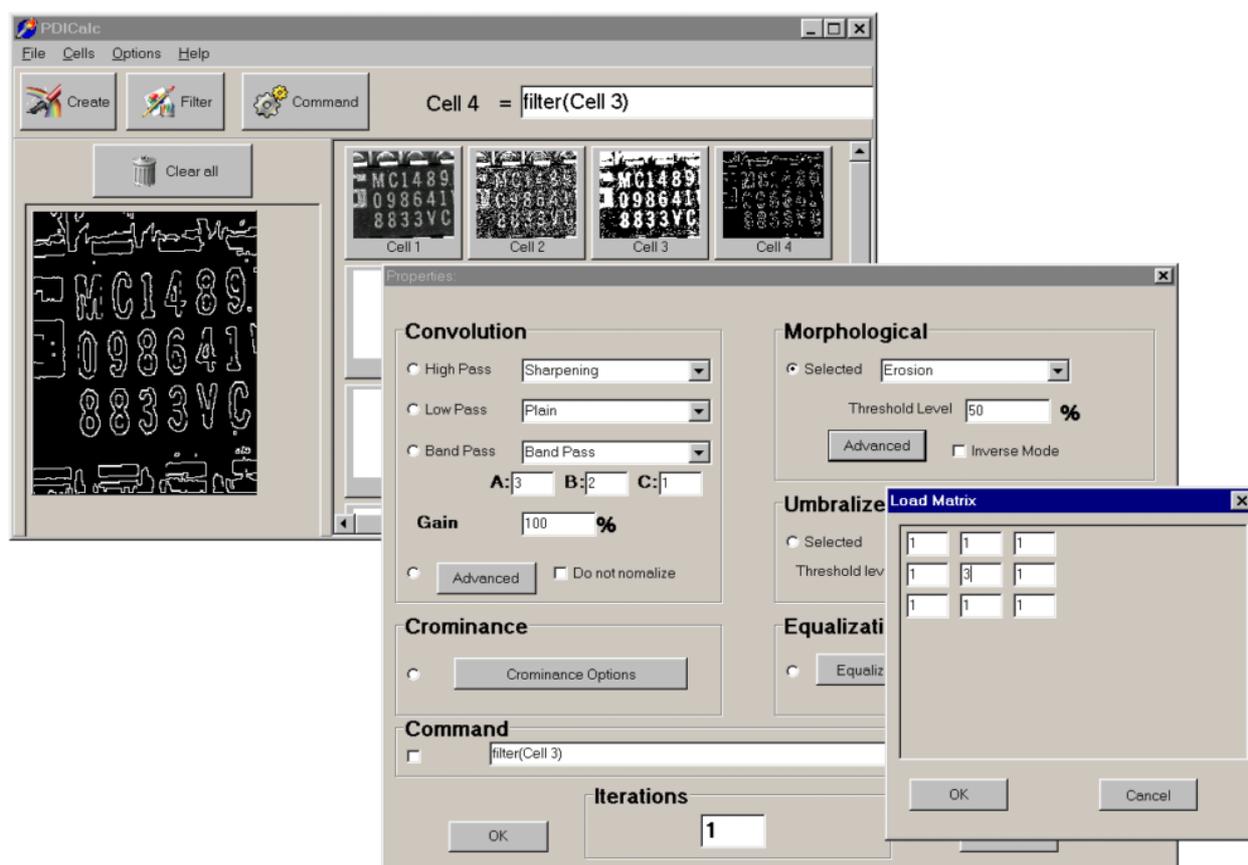


Figura 2: Menú de programación de los filtros.

procesamiento ulterior. Al pinchar una celda con el botón derecho del mouse se descuelga un menú que permite, manipular el contenido de la celda de varias maneras (abrir una imagen desde un archivo, o bien guardarla, copiar o pegar una imagen desde el clipboard, editar las propiedades del filtrado a aplicar, cambiar el nombre de la celda, eliminar la celda, etc.) Entre las propiedades del filtrado a aplicar es posible elegir la aplicación de filtrado por convolución, ecualización, filtrado morfológico, etc., donde cada uno de los parámetros específicos (kernel de los filtros, matrices morfológicas, etc.) se pueden elegir (ver Fig. 2). La programación de estos filtros también se puede copiar y pegar, lo cual es evidentemente de gran utilidad. Al pinchar la celda con el botón izquierdo, el contenido de la misma se despliega en el preview. Pinchando sobre este lugar se despliega una ventana de view donde la imagen se aprecia en su tamaño real. La ventana de view tiene una cantidad programable de imágenes previas almacenadas, de manera de poder comparar en tamaño real varias imágenes al mismo tiempo. Por último, podemos mencionar que en el menú descolgable de archivo es posible guardar en disco como “proyecto” el estado completo de la planilla, incluyendo las imágenes contenidas en las celdas, la programación de los filtros, etc. La imagen de cada celda se guarda como archivo .bmp, y el estado de los filtros se guarda en un formato propietario de pequeño tamaño.

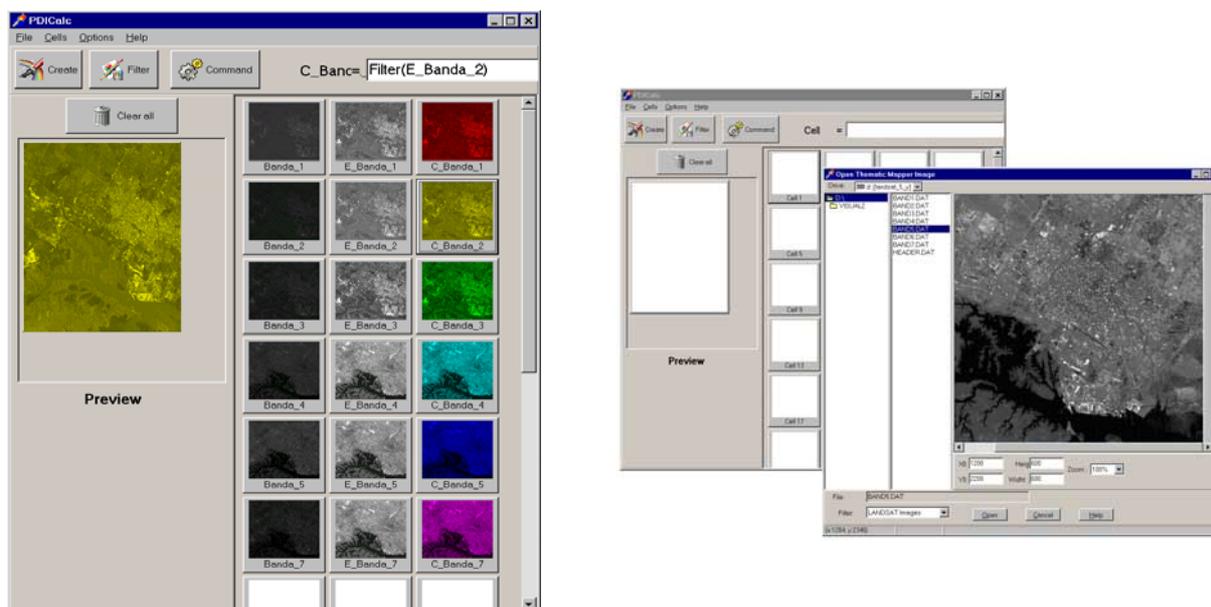


Figura 3: (a) Ecualización y *pseudocoloring* simultáneos de las 6 bandas, y (b) selección de una zona dentro de la imagen (Imagen cortesía de CONAE).

### 3 Procesamiento de imágenes temáticas

Una imagen satelital de tipo pasivo está normalmente compuesta por varias bandas, cada una de las cuales captura el sensado de un segmento específico del espectro electromagnético. Usualmente el usuario de estas imágenes debe seleccionar un área determinada, lo cual produce un mapa de bits en escala de grises. Sin embargo, en la mayoría de las aplicaciones, es necesario realizar procesamientos más complejos. Por ejemplo, el usuario puede requerir un *pseudocoloring* para visualizar la imagen con cierto realismo, asignando el nivel de tres bandas específicas a los primarios RGB. La mayor parte de las veces este proceder genera imágenes excesivamente oscuras y de bajo rango dinámico, por lo que es preferible antes ecualizar los mapas de bits respectivos. Este paso debe estar sincronizado entre las bandas intervinientes (la misma corrección de histograma) para evitar una distorsión en el color, y llegar a una visualización satisfactoria normalmente lleva varios pasos de prueba y error. En el sistema PDICalc, como hemos visto, este procedimiento se programa de una manera sencilla y puede reutilizarse para distintas porciones de la imagen sin reprogramar. En la Fig. 3(a) podemos observar una imagen de Landsat TM-5 procesada en el PDICalc. En la columna de la izquierda se observan los mapas de bits de las 6 bandas (excluyendo la 6), en la columna del medio se aplicó la misma ecualización a cada celda de la izquierda, y en la columna de la derecha se modificó la crominancia para asignar un color diferente a cada banda. PDICalc ofrece la opción de abrir una imagen satelital directamente del CD y en el formato provisto por la CONAE, donde se puede elegir la banda en particular, la zona dentro de la imagen, y se puede previsualizar para observar si la selección es adecuada (ver Fig. 3(b)). Los valores de posición y tamaño de la selección quedan almacenados en forma persistente, para facilitar la apertura secuencial de todas las bandas necesarias en forma idéntica.

Una vez que se ha elegido la zona de trabajo, entonces es muy sencillo aplicar los procedimientos usuales de procesamiento y detección. Por ejemplo, la segmentación del índice de vegetación puede computarse por medio del cociente entre las bandas 3 y 4 [5]. Esto

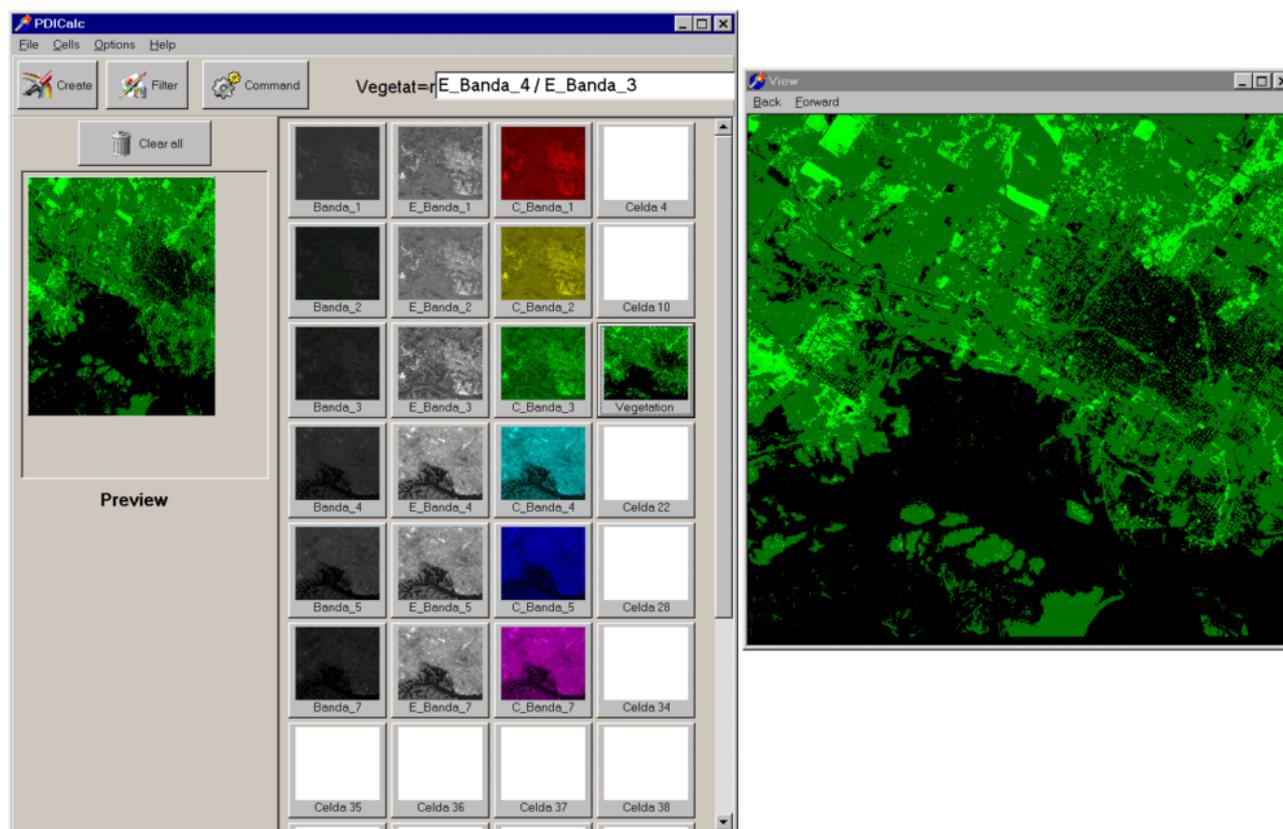


Figura 4: Segmentación del índice de vegetación.

puede lograrse de una manera sencilla simplemente programando una celda para realizar tal operación. En la Fig. 4 se muestra la citada operación, y la imagen en la celda resultante se muestra en tamaño verdadero en la ventana de *view*. Esta última ventana permite almacenar una secuencia de imágenes para poder comparar rápidamente entre resultados alternativos y elegir el más adecuado.

## Referencias

- [1] L. Arlenghi, A. Vitale, C. Delrieux, y G. Ramoscelli. PDICalc: Una Planilla de Cálculo para el Procesamiento Digital de Imágenes. *Congreso Argentino de Ciencias de la Computación*, , págs. 909–920. CACIC 2001.
- [2] K. Castleman. *Digital Image Processing*. Prentice-Hall, New York, 1989.
- [3] R. González y R. Woods. *Digital Image Processing*. Addison-Wesley, 1996.
- [4] Anil Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Cambridge, 1996.
- [5] T. Lillesand y R. Kiefer. *Remote Sensing and Image Interpretation (4th. ed.)*. Willey & Sons, New York, 2000.
- [6] Jae Lin. *2D Signal and Image Processing*. Prentice-Hall, Cambridge, 1991.
- [7] J. C. Russ. *The Image Processing Handbook*. CRC Press, Boca Raton, FL, 1989.

# Estructuras Espaciales para la Representación de Mapas

Leandro A. Bacic    Mauricio Delladio    Mariano R. Mignani  
Departamento de Ciencias e Ingeniería de la Computación  
Laboratorio de Visualización y Computación Gráfica (VyGLab)  
Universidad Nacional del Sur  
Av. Alem 1253, (8000) Bahía Blanca, Argentina  
Tel (0291) 459-5135 – Fax (0291) 459-5136

## 1 Introducción

Unos pocos años atrás la información geográfica se representaba en mapas de papel y la manipulación de esta información estaba limitada a un proceso manual no interactivo. El rápido desarrollo de la digitalización de la información geográfica junto con una demanda creciente de manipulación y análisis de estos datos ha generado la necesidad de software dedicado. La información geográfica está constituida por grandes volúmenes de información espacial y se hace necesario contar con métodos eficientes para almacenarla y recuperar datos relevantes.

Los datos espaciales consisten de objetos espaciales basados en puntos, líneas, superficies, volúmenes y otros datos de mayor dimensión. Ejemplo de datos espaciales son ciudades, rutas, ríos, etc. En un mapa, estos objetos espaciales combinados con ciertos atributos no espaciales tales como nombres de ciudades, nombres de calles, numeración de rutas, etc. conforman la información geográfica. Las bases de datos espaciales facilitan el almacenamiento y el procesamiento eficiente de información espacial y no espacial, idealmente sin favorecer una sobre otra.

Una propuesta para la representación de datos espaciales es separar estructuralmente los datos no espaciales de los espaciales manteniendo apropiadamente la relación entre ambos. Esto nos permite agilizar las consultas sobre los datos espaciales (las operaciones espaciales son realizadas directamente sobre la estructura de datos espaciales) y nos da la libertad de elegir una estructura de datos espacial más apropiada de las que nos impone la estructura no espacial.

Existen muchos estudios teóricos dedicados al tratamiento de información espacial; no obstante, la mayoría de los trabajos de alcance público se basan en la utilización de bases de datos relacionales las cuales no resultan adecuadas debido a la naturaleza de la información subyacente.

En este contexto, se están estudiando distintas técnicas que permitan representar y manejar adecuadamente datos espaciales, restringiendo la atención a objetos cuyas componentes estén definidas en el plano 2D. Como parte de este estudio se está desarrollando un prototipo para la recuperación y visualización de información espacial que sirva de base para el diseño de distintas aplicaciones de planeamiento urbano.

## 2 Estructura de Datos Espacial

Existen diversas representaciones que se presentan adecuadas para modelar datos espaciales definidos en el plano 2D con el objetivo de representar información geográfica en 2D.

La elección de una estructura para la representación de esta información espacial depende de los objetos sobre los que se va trabajar y del tipo de operaciones que se desean aplicar sobre los mismos. Los objetos deben poseer una forma geométrica bien definida y pueden contar con ciertos

atributos que los caractericen. Las operaciones a utilizar sobre la estructura de datos espacial deben permitir realizar consultas sobre los objetos o sobre sus atributos, tales como consultas de posicionamiento, proximidad o basadas en atributos.

Entre las distintas formas de representación de datos espaciales en 2D se pueden mencionar las basadas en:

- a) Ocupación del plano, las cuales asumen la existencia de un arreglo de *pixels* que representan implícitamente los objetos geométricos; así se genera una imagen 2D en la que los objetos están representados como subconjuntos de *pixels* del mismo color y son distinguidos unos de otros por el color de los *pixels* adyacentes.
- b) Información explícita de la geometría, donde los objetos están representados por figuras geométricas (líneas, puntos, curvas, etc) ubicadas sobre el plano 2D.

Debido a que un mapa poligonal se puede pensar como un conjunto de segmentos de línea que representan calles, ríos, vías ferroviarias, tendidos eléctricos, etc., la segunda propuesta de representación aparece como más natural que la primera.

Las estructuras utilizadas en la representación de datos espaciales pueden ser jerárquicas o no jerárquicas. Entre las no jerárquicas se pueden mencionar listas secuenciales, listas inversas, etc [Sam90] las cuales presentan ciertas deficiencias en el manejo de esta clase de datos.

Por su lado, la mayoría de las estructuras utilizadas en la representación de datos espaciales son jerárquicas, estando basadas en el principio de descomposición recursiva. Las estructuras de datos jerárquicas (EDJ) son útiles porque permiten tener distintos niveles de abstracción, enfocando la atención sobre un subconjunto del total de datos, lo que permite construir un conjunto de operaciones mucho más eficientes. Las EDJ son de gran interés por su claridad conceptual y su facilidad de implementación. Entre las distintas representaciones jerárquicas se encuentra el quadtree.

## 2.1 Quadrees

Una forma de representación de datos espaciales es hacer uso de estructuras de datos basados en ocupación espacial. Los métodos de ocupación espacial descomponen el espacio de datos en regiones llamadas Buckets. Estos métodos son conocidos como métodos *Bucketing*. Dentro de las distintas propuestas de descomposición basadas en Buckets se pueden mencionar [Sam90]:

- Propuestas basadas en el concepto de rectángulo abarcador mínimo (ej. R-Tree).
- Propuestas basadas en la descomposición del espacio en celdas disjuntas las cuales son mapeadas dentro de los buckets (ej:  $R^+$ -tree y Cell tree).
- Propuestas basadas en la descomposición del espacio en bloques de tamaño uniforme (ej: uniform grid).
- Propuestas basadas en la descomposición regular disjunta del espacio en bloques (o Celdas) cuyo ancho está restringido a ser potencia de dos, y sus posiciones están también restringidas (ej: quadtrees).

El término *quadtree* es usado, en el sentido más general, para describir una clase de EDJ cuya propiedad común es la de basarse en el principio de descomposición recursiva del espacio. Los quadtrees se diferencian de acuerdo a las siguientes características:

- *El tipo de datos que representan*: por ejemplo puntos, rectángulos, regiones, curvas, líneas, superficies, y volúmenes.
- *El proceso de descomposición*: la descomposición puede ser en partes iguales en cada nivel (llamada *descomposición regular*) o puede depender de la entrada.

- *La resolución de la descomposición*: es el número de veces que se aplica en proceso de descomposición y puede ser fija o depender de ciertas propiedades de los datos de entrada.

En general, un *quadtree* puede ser visto como un árbol donde cada nodo interno tiene cuatro nodos hijos también conocidos como cuadrantes (Nw, Ne, Sw, Se), mientras que cada hoja contiene una lista de objetos. La información de la base de datos está distribuida en las hojas. Es decir, cada uno de los nodos hoja, contiene parte de esta información.

Actualmente los *quadtree* son usados para puntos (Point quadtree), líneas y curvas (PM quadtree y sus variaciones), rectángulos, regiones (PR quadtree), superficies y volúmenes, lo que permite cubrir un amplio espectro de necesidades.

## 2.2 PM Quadrees

PM Quadtree es el término utilizado para describir una estructura de datos utilizada en la representación de mapas poligonales. Existen distintas variaciones del PM quadtree, entre las que se pueden mencionar PM1, y sus posteriores mejoras PM2 y PM3 [Sam90] las cuales se basan en vértices (puntos) o en lados (segmentos de líneas), y PMR que se basa solo en segmentos de línea. Sus implementaciones hacen uso de la misma estructura de datos. Todas son construidas aplicando el principio de “partir” repetidamente el conjunto de vértices o lados hasta obtener subconjuntos que son suficientemente simples para ser organizados en alguna otra estructura de datos y asociados a hojas del PM quadtree.

## 2.3 PMR Quadtree

El PMR quadtree es una variante del PM quadtree basada en segmentos de línea, el cual hace uso de una regla de división probabilística (probabilistic splitting). A un bloque se le permite contener un número variable de segmentos de línea. El PRM quadtree es construido insertando los segmentos uno a uno en una estructura inicialmente vacía, la cual consiste de un único bloque. Cada segmento de línea es insertado en todos aquellos bloques que intersecta u ocupa. Durante este proceso, cada bloque afectado es controlado para ver si la inserción causa el exceso de un *umbral de división* (splitting threshold). Si se excede este umbral, entonces el bloque es dividido *solo una vez* en cuatro bloques de igual tamaño. Esto evita dividir un nodo del árbol muchas veces cuando hay unas pocas líneas muy cercanas dentro de un bloque, lo que aumenta la velocidad de procesamiento y reduce el tamaño de la estructura.

Durante el proceso de borrado un segmento de línea es removido de todos los bloques que intersecta u ocupa y luego estos bloques se controlan para ver si es necesario mezclarlos (e.i las suma de los segmentos de línea de estos bloques es menor que el *umbral de división*).

## 2.4 Recuperación de la información a través de consultas - Ventana de consulta

Una operación importante (relacionada a una base de datos espacial) que necesita cierto grado de eficiencia es la recuperación de información contenida en la base de datos espacial.

Si bien existen distintos métodos de consulta para recuperar información de una base de datos espacial tales como consultas basadas en coordenadas espaciales, atributos no espaciales o selección de regiones poligonales, uno de los más adecuados es el conocido como ventana de consulta.

Una ventana de consulta recupera todos los objetos pertenecientes a un área rectangular determinada por un rango de valores de coordenadas  $x$  e  $y$ , que define la ventana. El método de consulta recupera de la base de datos subyacente todos aquellos bloques de información que son total o parcialmente cubiertos por la ventana de consulta. Luego, se procesan estos bloques para extraer los objetos relevantes.

Cabe destacar que en algunos casos, un mismo objeto, el cual es indivisible, puede estar representado en más de un bloque. Por ejemplo, una línea L puede estar contenida en más de un bloque a la vez. Esto introduce redundancia en la base de datos y en consecuencia, ineficiencia. En estos casos se debe encontrar una manera de recuperar cada objeto sólo una vez.

### 3 Prototipo en Desarrollo

En base a lo dicho y con el objetivo de contar con un modelo que sirva de base para el diseño de distintas aplicaciones que sean útiles para el planeamiento urbano, se desarrolló un prototipo que permite la generación de bases de datos espaciales, la recuperación de información de estas bases de datos y la visualización de la información recuperada.

#### 3.1 Generación de la Base de Datos Espacial

La base de datos espacial es generada usando un método que combina el mecanismo de división recursivo de PM quater y el concepto de umbral de división de PMR quater. El valor del umbral de división es ingresado por el usuario al comienzo de la generación y define una cota superior a la cantidad de segmentos de líneas almacenados en cada hoja.

#### 3.2 Recuperación de la información

La recuperación de información se realiza usando el método de ventana de consulta el cual permite extraer información de un subconjunto de la base de datos subyacente mediante la selección de un área rectangular.

La selección de la ventana se lleva a cabo utilizando el mouse, que permite una forma sencilla y práctica de selección rectangular. Después de la selección se procede a la recuperación de la información espacial mediante la intersección de la ventana de consulta con la base de datos subyacente.

#### 3.3 Visualización de la información

El prototipo presenta dos vistas del mapa, una vista pequeña del mapa completo (*mapa de referencia*) y otra vista donde se muestra la información recuperada del mapa después de una consulta (*mapa de navegación*); también incluye el control del desplazamiento dentro del mapa de navegación, la visualización de las coordenadas reales del mapa, y la posibilidad de ver los bloques (buckets) en que fue descompuesto el espacio de datos.

Por último cabe destacar que debido a que AutoCAD es una herramienta muy utilizada en la creación de mapas, surge la necesidad de realizar conversiones desde estos formatos a algún otro más conveniente. Por tal motivo se incluye en el prototipo un conversor de formato DXF (AutoCAD) [Rul96] a un formato intermedio que sea útil en la generación de la BD espacial.

### 4 Trabajos Futuros y Posibles Extensiones

Como ya se ha mencionado, el prototipo de recuperación y visualización de información espacial se está desarrollando con la finalidad que sirva de base para el diseño y construcción de aplicaciones que sean útiles para el planeamiento urbano.

El desarrollo de una aplicación debe integrar información espacial y no espacial. Hasta ahora se desarrolló lo relativo a geometría y topología. En lo relacionado con la generación de la BD

espacial, como trabajo futuro, se plantea la necesidad de estudiar la manera de incorporar esta información no espacial a la espacial ya existente. Por su parte, se plantea la incorporación de nuevos métodos de consulta basados en atributos y la generalización del método de la ventana de consulta que permita la selección de regiones de forma poligonal. Con respecto a la visualización, se plantea incorporar mayor funcionalidad y flexibilidad. Todo esto constituye, sin duda, la base para el desarrollo de un amplio espectro de aplicaciones.

## Referencias

[AS97] Walid G. Aref and Hanan Samet. *Efficient Window Block Retrieval in Quatree-Based Spatial Databases*. *GeoInformatica*, 1(1):59-91, April 1997.

[BK2001] Richard K. Brail and Richard E. Klosterman, editors. *Planning Support Systems: Integrating Geographic Information Systems, Models, and Visualization Tools*. Esri Press, first edition, June 2001.

[HS91] Erik G. Hoel and Hanan Samet. *Efficient Processing of Spatial Queries in Line Segment Databases*. In *Proceeding of the 2nd Symp. on Large Spatial databases (SSD'91)*, pages 238-255, Zürich, Aug. 1991.

[HSS97] Gísli R. Hjaltason, Hanan Samet, and Yoram J. Sussmann. *Speeding up Bulk-Loading of Quadtrees*. In *Proceedings of 5th Intl. ACM Workshop on Advances in GIS*, pages 50-53, Las Vegas, NV, November 1997.

[Rul96] Keith Rule. *3D Graphics File Format: A Programmer's Reference*, chapter 7: DXF AutoCAD R13, pages 195-240. Addison-Wesley Developers Press, first edition, Sept 1996.

[Sam90] Hanan Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

[Sam95] Hanan Samet. *Spatial Data Structures*. Appears in *Morden Database Systems: The Object Model, Interoperability, and Beyond*, pages 361-385. Addison-Wesley/ACM Press, Reading, MA, 1995.

[Tim98] Sabine Timpf. *Hierarchical Structures in Map Series*. PhD, Technical University Vienna, 1998.

# Modelamiento de Objetos 3D

Mg. Silvia Castro<sup>(1,3)</sup> Lic. Sergio Martig<sup>(1,3)</sup>  
Dra. Liliana Castro<sup>(2,3)</sup> Mg. Liliana Boscardín<sup>(2,3)</sup> Lic. Diana Salgado<sup>(2,3)</sup>  
<sup>(1)</sup> Departamento de Ciencias e Ingeniería de la Computación  
<sup>(2)</sup> Departamento de Matemática  
<sup>(3)</sup> Laboratorio de Investigación en Visualización y Computación Gráfica  
Instituto de Investigación en Ciencia y Tecnología Informática (IICyTI)  
Universidad Nacional del Sur  
Bahía Blanca

## Resumen

Las escenas tridimensionales generadas hoy en día contienen modelos geométricos altamente detallados; éstas están emergiendo rápidamente como la próxima generación de aplicaciones tales como las basadas en Internet, los complejos ambientes virtuales tridimensionales, el CAD colaborativo, la visualización interactiva y los videojuegos multiusuario, entre otras. Así, se espera que la futura generación de sistemas gráficos computacionales trate con complejos modelos 3D renderizados de manera interactiva y se espera un crecimiento de tales modelos 3D en la red.

Esta situación motiva el desarrollo de modelos de volúmenes, representados a través de su superficie y/o de su interior, que puedan satisfacer los requerimientos de uso efectivo de espacio en disco y de ancho de banda de red, así como también una reducción sustancial del tiempo de transferencia sobre la red. Indudablemente, estos modelos plantean rigurosas demandas en lo referente a ancho de banda, capacidad de almacenamiento y tiempo de transmisión.

Tradicionalmente, los modelos geométricos de gran complejidad han sido construidos de subdivisiones de modelos de objetos aproximados poligonalmente denominadas redes, o de sistemas de escaneo que pueden producir modelos de gran complejidad con cientos de miles o millones de vértices e información de atributos adicionales a cada uno de ellos tales como el color y las normales. En todos los casos, la representación resultante requiere grandes cantidades de memoria de almacenamiento y grandes tiempos de transmisión y renderizado. Por lo dicho, requerimientos mínimos de un sistema gráfico de software de hoy en día incluyen compresión, multirresolución, simplificación aproximada de redes y transmisión progresiva.

El principal objetivo de nuestro trabajo es la construcción de buenos modelos de volúmenes, representados tanto a través de su superficie como de su interior, de modo tal que sea posible una aproximación de alta calidad de grandes cantidades de datos. Estamos trabajando en la obtención de representaciones 3D que soporten multirresolución, transmisión progresiva, compresión, refinamiento electivo y renderizado rápido y eficiente.

## Introducción

Hoy en día, los dispositivos de escaneo permiten la adquisición finamente detallada de objetos 3D; además, los costos del hardware gráfico caen aceleradamente, la World Wide Web en el campo de la realidad virtual, especialmente para acceso remoto, hace necesario un acceso muy rápido a los datos; estos son algunos de los factores que demandan una generación y un uso siempre creciente de modelos de volúmenes que puedan enfrentarse con esta complejidad.

Las grandes demandas que plantean tales modelos se refieren a los recursos de cálculo, almacenamiento, tiempo de transmisión y renderizado. Con la finalidad de manejar los gigantescos datasets volumétricos, es necesario tener diversas alternativas que satisfagan los requerimientos mencionados. Estos datasets se definen típicamente mediante muestras discretas, bien alineadas

sobre grillas regulares o bien aleatoriamente dispersos en el espacio describiendo una forma 3D y/o un conjunto de datos adicionales en determinados puntos del dominio.

A partir de las muestras discretas, se construye un modelo geométrico que incluya los atributos y que aproxime lo mejor posible la forma subyacente. Multirresolución, compresión y progresividad son esenciales para almacenar, mostrar y transmitir complejas bases de datos 3D dentro de tiempos mínimos, satisfaciendo cotas de error y/o cotas de complejidad.

Los conjuntos de datos volumétricos utilizados actualmente en diversas aplicaciones en distintas áreas del conocimiento tienen características muy variadas pero un problema en común: un tamaño tan grande que afecta tanto los requerimientos de almacenamiento y transmisión como su tiempo de visualización. En los últimos años se han hecho esfuerzos con el objeto de mejorar la performance de rendering de estos conjuntos de datos volumétricos, pero hace muy poco que se ha comenzado a hacer propuestas basadas en un modelamiento adecuado. Este trabajo está direccionado a lograr una construcción eficiente de modelos 3D, tanto volumétricos como de superficie, que sean aproximaciones de alta calidad de datos geométricos de gran complejidad permitiendo además una manipulación interactiva de los mismos.

Es decir que si bien se ha dedicado mucho esfuerzo para encontrar buenos modelos de representación volumétrica aún resta mucho por hacer .

### **Líneas de investigación actual en Modelamiento**

Teniendo en cuenta la necesidad de mejores métodos de modelamiento de volúmenes que permitan buenas performances en cuanto a espacio de almacenamiento y a tiempo de transmisión, orientados tanto a volúmenes representados mediante su superficie como a volúmenes representados con su interior, hemos direccionado nuestro trabajo en dos líneas:

- *Manejo de superficies, incluyendo compresión, subdivisión, simplificación y multirresolución.* Los modelos 3D que representan los objetos por su superficie están manejando millones de polígonos interactivamente y esto está muy lejos de ser renderizado actualmente en las PC's en tiempos interactivos. Las técnicas actuales de aceleración gráfica mejoran significativamente la performance pero no ayudan cuando en una escena son visibles a la vez una gran cantidad de objetos. Si bien actualmente se han desarrollado muchas técnicas de simplificación que permiten representar escenas complejas efectivamente, aún no se han logrado representaciones para los casos en los que los modelos tienen detalles de alta frecuencia. Es por ello que se está trabajando en representaciones de volúmenes a través de su superficie que, dados modelos de gran complejidad en cuanto a los detalles de alta frecuencia, permitan una representación simplificada de alta calidad.
- *Manejo de volúmenes incluyendo compresión de datos, subdivisión, simplificación y multirresolución.* Las wavelets han sido descubiertas por las comunidades de gráfica y de visualización para una aproximación eficiente de grandes conjuntos de datos, habiéndose usado en diversas aplicaciones. Dada una pérdida de información predefinida, el principal objetivo de las wavelets es aproximar los datos con tan pocos coeficientes distintos de cero como sea posible. Las propiedades de localización de las wavelets permiten controlar la calidad de la aproximación tanto general como localmente; obviamente esta pérdida de información es función de la posición espacial y la calidad es controlada por la representación en wavelets. Si bien aún se aplican restricciones al conjunto de datasets que se representan con wavelets, lo expuesto habla de la importancia de un desarrollo del tema en lo referente a generalizar las técnicas de wavelets a conjuntos de datos que no pueden atacarse con las representaciones actuales de las mismas. Dentro de este contexto se está trabajando en la extensión de la definición de wavelets sobre tetraedros a wavelets suaves por medio del esquema de

lifting con el objetivo de modelar objetos complejos de manera compacta dando además una medida de la calidad de dicha aproximación

Es realmente un desafío encontrar métodos y estructuras de datos que sean compactas, eficientes, confiables y que permitan una buena calidad visual de los modelos.

## Conclusiones

El objetivo del grupo es la obtención de algoritmos y estructuras de datos eficientes para la representación, manipulación y transmisión de objetos 3D gigantes y complejos. En ambos casos, se pretende obtener nuevos modelos 3D que permitan la visualización y la exploración interactiva de grandes conjuntos de datos sobre PCs y en la red, permitiendo así hacer accesible la tecnología de visualización a un mayor rango de usuarios.

## Bibliografía

- [1] Chiueh, T., Yang, C., He T., Pfister, H. & Kaufman, A., *Integrated Volume Compression and Visualization*, Proceedings of the IEEE Visualization '97, pp. 329-336, 1997.
- [2] Cignoni, P., Montani, C., Scopigno, R., *A comparison of mesh simplification algorithms*, Computers and Graphics, 22(1):37-54, 1998.
- [3] Deering, M., *Geometric Compression*, Computer Graphics, SIGGRAPH '95 Proceedings, pp. 13-20, Agosto 1995.
- [4] Eck, M., DeRose, T., et al., *Multiresolution Analysis of arbitrary Meshes*, SIGGRAPH'95 Proceedings, pp. 173-182, Agosto 1995.
- [5] Gross, M., Staadt, O. & Gatti, R., *Efficient triangular surface approximation using wavelets and quadtree data structures*, IEEE Transactions on Visualization and Computer Graphics, 2(2):130-143, Junio 1996.
- [6] Heckbert, P. & Garland, M., *Survey of polygonal surface simplification*, Technical Report, CS Dept., Carnegie Mellon University.
- [7] Heckbert, P., Rossignac, J. et al., *Course N° 25: Multiresolution surface modeling*, En Course Notes for SIGGRAPH'97, 1997.
- [8] Hoppe, H., *Progressive Meshes*, Computer Graphics, SIGGRAPH'96 Proceedings, pp. 98-108, Agosto 1996.
- [9] Lounsbery, J., *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*, PhD Thesis, University of Washington, Seattle, 1994.
- [10] Popovic, J. & Hoppe, H., *Progressive Simplicial Complexes*, SIGGRAPH'97 Proceedings, pp. 108-116, Agosto 1997.
- [11] Stollnitz, E., DeRose, T. & Salesin, D., *Wavelets for Computer Graphics: Theory and Applications*, Morgan Kaufmann Publishers, Inc, 1996.

## Reconocimiento de Rostros

Franco Chichizola<sup>1</sup>, Armando De Giusti<sup>2</sup>, Marcelo Naiouf<sup>3</sup>

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>4</sup>.  
Facultad de Informática. Universidad Nacional de La Plata.*

### Resumen

Se investigan y analizan los sistemas de reconocimiento de rostros, con el fin de lograr uno que sea tolerante a diferentes variaciones como posición, características de los rostros, e iluminación.

El aporte de la línea de investigación es obtener un sistema de reconocimiento de rostros completamente automático que permita utilizar imágenes complejas que contengan un rostro de cualquier tamaño y en cualquier posición, tratando de incrementar el número de personas (rostros) reconocidos por el sistema, y de reducir el tiempo necesario para el aprendizaje.

### Palabras Claves

Eigenfaces – Normalización – Reconocimiento – Paralelización – Distribución de datos.

### Introducción

El reconocimiento de rostros automatizado, involucra alguna de las siguientes etapas:

- Adquisición de la imagen.
- Normalización de los rostros: en esta etapa se realiza un preprocesamiento de la imagen para lograr que todas las muestras tengan el mismo formato. Por ejemplo, se localiza o se extrae el rostro de una imagen más compleja, se modifica su tamaño, se solucionan problemas como la variación de iluminación, expresión, posición, etc.
- Alguna forma de entrenamiento: consiste en utilizar algún mecanismo que le permita al sistema “aprender” los rostros que constituyen el conjunto de entrenamiento. El tipo de entrenamiento que sea utilizado para el aprendizaje dependerá, en gran medida, de la metodología que se esté utilizando para el reconocimiento. Por ejemplo, al utilizar redes neuronales, el entrenamiento consiste en obtener los valores correspondientes a cada una de las conexiones (pesos) que constituyen a la red neuronal.

---

<sup>1</sup> Ayudante Diplomado Semi Dedicación. [francoch@lidi.info.unlp.edu.ar](mailto:francoch@lidi.info.unlp.edu.ar)

<sup>2</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. [degiusti@lidi.info.unlp.edu.ar](mailto:degiusti@lidi.info.unlp.edu.ar)

<sup>3</sup> Profesor Titular. [mnaiouf@lidi.info.unlp.edu.ar](mailto:mnaiouf@lidi.info.unlp.edu.ar)

<sup>4</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.  
TE/Fax +(54) (221) 422-7707. <http://lidi.info.unlp.edu.ar>

- La etapa de reconocimiento: básicamente consiste en alimentar al sistema con diferentes imágenes de personas, esperando obtener como resultado, una forma de codificación unívoca que nos permita identificar de qué persona se trata, o bien determinar que el rostro no está en la base de conocimiento.

Para la implementación de este tipo de sistemas, independientemente de la técnica o metodología que sea implementada, hay involucrados generalmente dos conjuntos de datos. El primero es utilizado durante la etapa de aprendizaje, el cual es comúnmente llamado *conjunto de entrenamiento*. Se debe de tratar que los patrones que integran este conjunto, sean lo más diferentes posible entre sí, y que además, representen al problema, para poder obtener un buen porcentaje de generalización.

El segundo conjunto de patrones, es el que se utiliza durante la etapa de reconocimiento y es llamado *conjunto de prueba*.

Una de las técnicas que se utiliza para el reconocimiento de rostros es el de “Análisis de Componente Principal”, también llamado “modelo de Eigenfaces”. Este método es tolerante a variaciones de posición y características (como bigotes, barba, anteojos) de los rostros. Mientras que soporta una cierta variación en la iluminación, pero no diferencia de tamaño en las imágenes de la cara.

Este modelo consta de las etapas de entrenamiento y de reconocimiento, necesitándose que las imágenes que se utilicen estén normalizadas en cuanto al tamaño, y que en ellas aparezca solo el rostro.

Por lo tanto, debe existir una etapa previa de normalización, en la cual se debe extraer el rostro de la imagen completa, y luego se modifica el tamaño de éste. Esta tarea requiere un cómputo considerable, lo que justifica el objetivo de la paralelización.

## **Temas de Investigación y desarrollo**

Es de hacer notar que un primer prototipo está actualmente operativo, trabajando off-line y en procesamiento secuencial.

- Estudio de la etapa de reconocimiento a fin de mejorarla y lograr un mayor porcentaje de aciertos.
- Desarrollo de un sub-sistema completamente automático para la etapa de normalización de rostros. De esta forma la base de datos puede componerse de cualquier imagen que posea al menos un rostro. Esta etapa incluye la localización del rostro dentro de la imagen, escalamiento de la cara, eliminación de diferencias en la iluminación, y resolución de otros problemas que puedan existir en la imagen.
- Comparación del sistema completo con otros modelos automáticos de reconocimiento de rostros.
- Ampliaciones del modelo para reconocer imágenes de rostros en colores y para el reconocimiento sobre video.

- Creación de un sistema inteligente que permita agregar a la base los rostros que se intentaron reconocer y que no estaban en ella.
- Paralelización del sistema para lograr resultados en tiempo real, y para lograr distribuir los datos y el procesamiento de manera que se pueda utilizar bases de mayor tamaño.

## Equipamiento de experimentación

En el LIDI se dispone de un cluster de PCs con 32 equipos homogéneos. Por otra parte, se tiene acceso a la computadora Clementina (SGI Origin 2000) que tiene 40 procesadores con memoria compartida distribuida.

Naturalmente también se puede experimentar con arquitecturas pseudo-paralelas tal como redes heterogéneas con un soporte de comunicaciones tipo PVM o MPI.

## BIBLIOGRAFIA BASICA

- [1] S. Lin, S. kung, y L. Lin, "Face Recognition/Detection by Probabilistic Decision-Based Neural Network", *IEEE Trans. Neural Networks*, vol. 8, pp. 114-132, 1997.
- [2] Steve Lawrence, C. Lee Giles, Ah Chung Tsoi, Andrew D. Back "Face Recognition: A Convolutional Neural Network Approach". *IEEE Transactions on Neural Networks*, *Special Issue on Neural Networks and Pattern Recognition*, Volume 8, Number 1, pp. 98-113, 1997.
- [3] F. Samaria and S. Young, "HMM based architectures for face identification", *Image and Computer Vision*, vol. 12, pp 537-583, October 1994
- [4] F. Samaria, "Face Recognition Using Hidden Markov Models", PhD thesis, University of Cambridge, Cambridge, U.K. 1994.
- [5] Laurenz Wiskott, Jean-Marc Fellous, Norbert Krüger and Christoph von der Malsburg *Face Recognition by Elastic Bunch Graph Matching*, in *Intelligent Biometric Techniques in Fingerprint and Face Recognition*, eds. L.C. Jain et al., publ. CRC Press, ISBN 0-8493-2055-0, Chapter 11, pp. 355-396, (1999)
- [6] M. Turk and A. Pentland, "Face recognition using eigenfaces", in *Proceedings of International Conference on Pattern Recognition*, pp. 586-591, 1991
- [7] Jose Gerardo Gonzalez, *Face Recognition and Detection Using Eigenfaces*, 1997
- [8] MIT Media Laboratory Vision and Modeling Group (VISMODO) Face Recognition Demo Page  
<http://www-white.media.mit.edu/vismod/demos/facerec/>
- [9] AT&T Laboratories Cambridge, "The ORL Database of Faces",  
[ftp://ftp.uk.research.att.com/pub/data/att\\_faces.zip](ftp://ftp.uk.research.att.com/pub/data/att_faces.zip)  
<http://www.cam-orl.co.uk/facedatabase.html>
- [10] Champredonde Raúl y Chichizola Franco, "Paralelización de algoritmos en ADA sobre Clementina2. Aplicación a un sistema de reconocimiento de rostros". En Prensa.
- [11] Correa Martín Sebastián y Chichizola Franco. Trabajo de grado de Licenciatura en Informática "Sistema de Reconocimiento de Rostros". 2001.



# Visualización de Información de Bases de Datos Heterogéneas

María Laura Cobo\* Sergio R. Martig Silvia Castro Pablo R. Fillottrani  
Departamento de Ciencias e Ingeniería de la Computación<sup>†</sup>  
Universidad Nacional del Sur  
Bahía Blanca - Argentina  
fax: +54 291 4595136  
[mlcobo, srm, smc, prf] @cs.uns.edu.ar

**Palabras claves:** Visualización, Bases de datos, Programación Orientada a Objetos

## 1 Introducción

Debido al aumento de la cantidad de datos con los que nos tenemos que enfrentar, está ganando en importancia la exploración de espacios de información complejos. Estos espacios de información y su exploración han sido estudiados en los últimos años por una gran variedad de autores [Str98, BYRN99, CMS99].

Proveer ayudas de orientación intuitivas y técnicas de navegación es esencial para permitir la exploración de los mismos. Por espacios de información complejos nos referimos a espacios de información que no sólo son muy grandes sino además, heterogéneos en su estructura. Los espacios de información complejos se caracterizan, por ejemplo, por información que relaciona distintas modalidades (por ejemplo, tablas, gráficos y video) y que difiere en su dimensión.

La exploración de este tipo de información debe manejar una interacción con los datos que permita extraer información útil del conjunto total de datos disponibles. La exploración de información comprende varios aspectos:

- Las facilidades de interacción para moverse (browse) en el espacio de información y buscar información
- La visualización de información que vuelca gráficamente un espacio de información o resultado de la búsqueda
- Las técnicas de análisis para facilitar la interpretación de los datos retornados como resultado de lo pedido en el proceso de búsqueda.

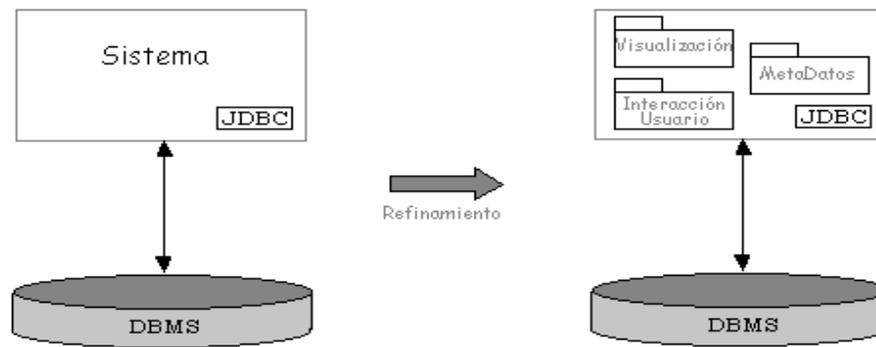
La recuperación de la información se refiere a cómo obtener información con ciertas características. El tipo de consulta que el usuario puede formular es altamente dependiente de modelo de recuperación de información subyacente, en este caso los datos una base de datos. Los usuarios en busca de respuestas necesidades específicas deben interactuar con los datos para precisar qué es importante y qué no desde su punto de vista. Para que sea útil, esta interacción debe poder realizarse en todos los niveles del proceso de visualización.

El área de los lenguajes de consulta para bases de datos relacionales deben moverse hacia una mayor flexibilidad, permitiendo más potencia en la especificación de la consulta. Actualmente, un tópico de investigación importante lo constituyen los lenguajes de consulta visuales. Las

---

\*Becaria de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

<sup>†</sup>Parcialmente financiado por la Secretaría de Ciencia y Tecnología (Universidad Nacional del Sur).

Figura 1: Arquitectura *two-tier*

metáforas visuales pueden ayudar a los usuarios no expertos a formular consultas booleanas complejas.

Cuando una persona desea obtener determinada información de una base de datos, tiene una o más metas en mente y usa un sistema de búsqueda como la herramienta que le brinde la posibilidad de alcanzar esas metas. Estas metas pueden requerir información de muy amplio espectro: el resultado de una consulta puede mostrarse en forma *precisa* o *imprecisa*, de modo tal que al visualizarlos posteriormente de dicha búsqueda se pueda tener la posibilidad de observar diferencias. Así, en este proceso interactivo de especificación de la consulta, visualización de los resultados obtenidos de la búsqueda y examen de los mismos, la información es presentada al usuario ya sea al finalizar el proceso o en la reformulación de la consulta para alcanzar la meta deseada.

Las búsquedas por palabras clave, los lenguajes de consulta y las consultas guiadas por ejemplos son parte de la exploración en las bases de datos. En particular, últimamente han adquirido relevancia las consultas dinámicas debido a su efectividad; estas consultas son especificadas mediante barras de desplazamiento que definen intervalos de interés en los atributos. Este tipo de consulta permite una integración natural de la búsqueda y la visualización de los resultados. El proceso de visualización tendiente a realizar una exploración visual de los datos necesita integrarse con los sistemas utilizados para manipular grandes volúmenes de información relacional y estructurada, incluyendo la conexión con sistemas de bases de datos relacionales. El primer paso de este proceso lo constituye la extracción de los datos a visualizar desde de la base de datos.

En este trabajo se presenta un proyecto cuyo objetivo es diseñar un sistema que realice la transformación de los datos tabulares de una base de datos relacional en un tabla de datos general, de modo que permita su posterior visualización. El sistema debe posibilitar al usuario que seleccione, según su criterio, algunos de los atributos presentes en una base de datos para generar una nueva tabla a partir de la cual se pueda visualizar sólo la información que le resulta relevante. De esta manera el sistema en cuestión deberá establecer una conexión con cualquier base de datos relacional, no importando su estructura ni sus contenidos, obtener información sobre su conformación, y a partir de ella solicitar al usuario el grupo información que le resulta interesante, a fin de que el sistema elabore una tabla interna, o *tabla de metadatos*, con la información concreta de la base de datos para su posterior visualización.

## 2 Arquitectura del sistema

El esquema general del sistema se puede observar en la figura 2. Las componentes intervinientes son la base de datos con los datos a ser visualizados, y la aplicación de visualización en sí. Esta aplicación esta formada por un módulo de interacción con el usuario, que se encarga de mostrarle al usuario la estructura de la base de datos y le permite a este elegir cuales son las tablas de su

interés e inclusive las columnas que considera relevantes para el proceso de visualización; otro módulo con el proceso de visualización propiamente dicho, que toma la tabla con los metadatos que fueron escogidos por el usuario y se encarga de mostrarselos gráficamente; y un tercer módulo encargado de recuperar la estructura original de la base de datos y genera la tabla de metadatos según los requerimientos del usuario generado por el módulo de interface.

Para la implementación de este sistema se escogió el lenguaje de programación Java, con el objetivo de aprovechar su portabilidad y la disposición de librerías de acceso a diversas bases de datos. Por esta razón el último de los módulos descritos debe usar la API JDBC [WFC<sup>+</sup>99]. La implementación que se está llevando a cabo actualmente sólo utiliza la conexión ODBC de Windows, pero la arquitectura está diseñada para poder extenderse a otros protocolos de acceso a bases de datos.

## 2.1 JDBC

[WFC<sup>+</sup>99]

JDBC es una API Java que permite el acceso virtual a cualquier tipo de dato tabular. Esta API consiste en un grupo de clases e interfaces escritas en el lenguaje de programación Java y que a su vez proveen una API estándar para desarrollar herramientas para bases de datos y hace posible aplicaciones sobre estas bases de datos. La manera en la que se logra este objetivo se reduce básicamente al envío de consultas SQL a la base de datos relacional concreta; para que esto sea posible debe soportar todos los posibles dialectos de SQL. Algunas versiones de JDBC también brindan la posibilidad de interactuar con diferentes fuentes de datos. De esta manera una aplicación puede acceder virtualmente a cualquier fuente de datos y correr bajo cualquier plataforma que contenga una máquina virtual Java; es decir que utilizando esta API, el programador no se ve en la necesidad de escribir un programa diferente para el acceso a datos en diferentes plataformas, sino que simplemente escribe un único programa utilizando API JDBC; el programa tiene entonces la capacidad de enviar las consultas de manera apropiada a la fuente de datos. La combinación de la plataforma Java y la API permite al programador abstraerse de la arquitectura subyacente, dándole el lema de programación “escriba una vez, ejecute donde quiera”. La funcionalidad de la API, puede resumirse en los siguientes tres pasos:

1. Establecer una conexión con una fuente de datos.
2. Enviarle consultas y sentencias de actualización a esa fuente de datos.
3. Procesar el resultado.

Esta API Java que se está describiendo se adecuó perfectamente a las condiciones establecidas por el sistema que se pretende desarrollar. Esto implica que la API logra establecer el proceso de comunicación y obtención de datos vía ODBC de manera mucho más adecuada y segura en el sistema concreto.

Se podría estar tentado en utilizar ODBC; el problema de esta alternativa es que la interface de ODBC está implementada en el lenguaje C y las llamadas desde Java a código C provocan un decremento de algunas propiedades importantes como la seguridad y la portabilidad de las aplicaciones. La traducción literal del código fuente del ODBC a un API Java tampoco es una buena solución ya que la traducción de algunas instrucciones C a Java no son triviales. Por ejemplo, aquellas que utilicen punteros, estructura de datos que no existe como tal en Java. El otro punto a favor del uso de un API puramente Java radica en el hecho de que la aplicación completa es totalmente portable y segura ya que puede pasar a ejecutarse en cualquier máquina que tenga una máquina virtual Java.

Un aspecto que es de suma importancia, es la capacidad de manipular metadatos, *i.e.* datos sobre la estructura de la base de datos como los tipos de las columnas de las tablas; esta información es imprescindible para el desarrollo del sistema, más específicamente en la construcción

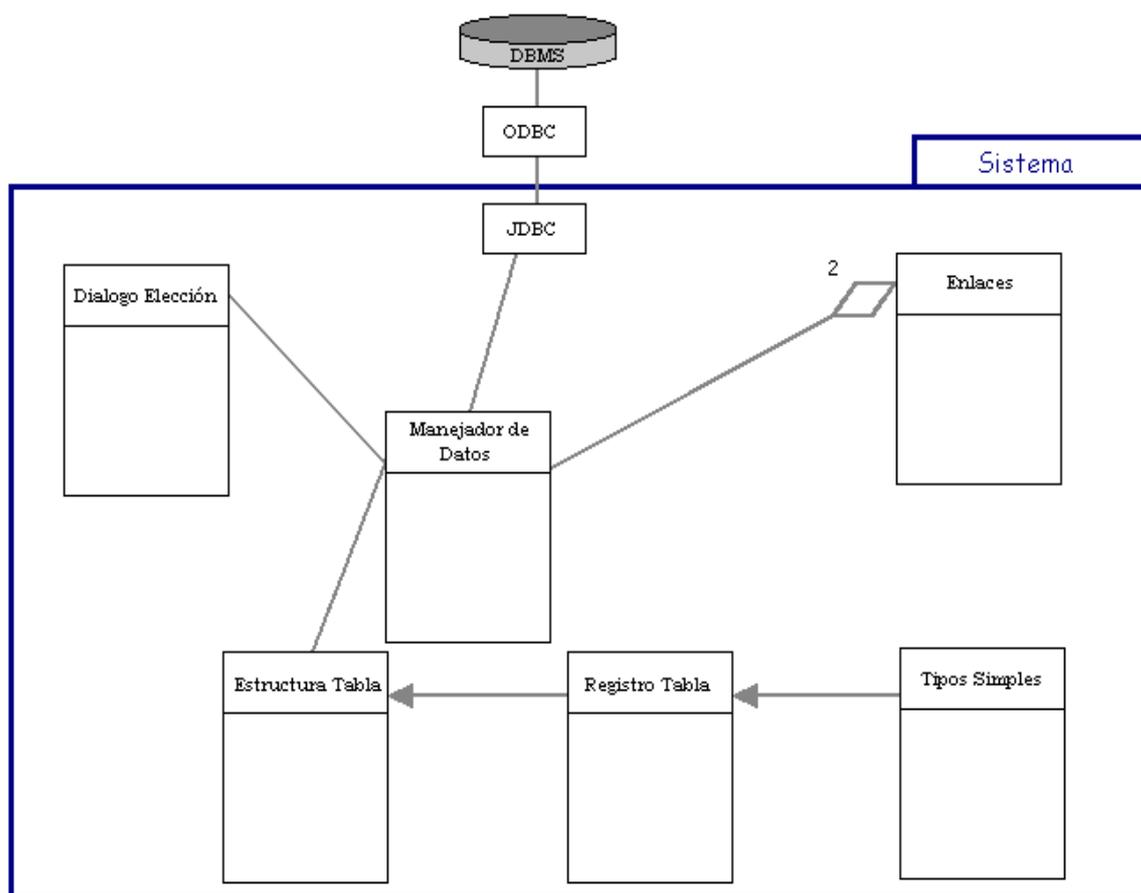


Figura 2: Diagrama de clases del sistema

de la tabla interna previamente mencionada. Esto no sólo brinda una manera de manipular la información de los datos almacenados en la base de datos sino también una manera de conocer datos sobre la estructura de la base. La mayoría de las bases de datos están organizadas en tablas; para este tipo de aplicaciones es muy importante poder recuperar información sobre la base de datos, como la cantidad de tablas de la base de datos, las columnas de cada tabla, el tipo de datos y nombre de cada columna, las claves primarias, etc. Cada tipo de manejador de base de datos, DBMS, tiene su propia funcionalidad para otorgar este tipo de información. El API JDBC provee una interface que es implementada mediante un driver de manera que los métodos devuelvan información sobre el driver y/o el manejador de datos para el cual el driver fue escrito. Esta interface brinda a los usuarios y a las herramientas una forma estándar de obtener metadatos.

## 2.2 Estructura de la aplicación servidora

En este caso, la idea es empezar realizando una aplicación que se adapte a la forma *two-tier* propuesta. A pesar de que puede resultar claro la utilidad de transformar más adelante el sistema en *three-tier*, a fin de que resulte más general y modular.

En principio el sistema tendría la estructura que se observa en la figura 2.2.

De acuerdo al diagrama de clases, el sistema cuenta con las siguientes clases:

**API JDBC:** Es el API Java que fue mencionado en la sección 2.1 y que llevará a cabo la conexión con la base de datos a través de ODBC.

**Manejador de Datos:** Es quien mantendrá la interacción con JDBC, indicándole que datos pedirle a la base de datos, estos datos pueden ser datos concretos o bien metadatos. A

su vez en su funcionalidad incluye el pasar la información primero a la Clase Diálogo Elección y mas adelante a la clase Estructura Tabla para formar la estructura que haya sido indicada por el usuario.

**Diálogo Elección:** Esta interacción con el usuario, indicará al manejador de datos. cuales datos de cada tabla disponible guardar en la estructura interna dada por las clases siguientes.

**Estructura Tabla:** Esta clase guarda la forma de una tabla particular, como la tabla esta formada por registros, conoce a la clase Registro Tabla a pesar de que esta última no conoce la existencia de esta clase.

**Registro Tabla:** Esta clase representa a cada uno de los registros de una tabla particular. Requiere del conocimiento de la clase Tipos Simples ya que necesita los métodos necesarios para su manipulación.

**Tipos Simples:** Son los tipos básicos que puede contener la tabla. Esta clase no tiene conciencia de la existencia de las otras clases a pesar de que es conocida desde la clase Registro Tabla.

**Enlaces:** Esta relacionada a travesee de una agregación con la clase Estructura Tabla. Su función es guardar la relación que pueda existir entre dos tablas diferentes de la base de datos, guardando a su vez el campo que las mantiene enlazadas.

La base de datos o DBMS, pude variar de una base Access hasta una Oracle, pasando por cualquier otro tipo existente. Por la naturaleza de la implementación esta funcionará sólo sobre el sistema operativo Windows, ya que la interacción está desarrollada para motores que se comprendan con ODBC.

La elección de Java como lenguaje de implementación está fundamentada en las características del lenguaje, es decir, en el hecho de que Java es un lenguaje de programación robusto, seguro, fácil de utilizar y entender y downlodable de los entornos de red; estas características lo hacen un buen lenguaje base para aplicaciones sobre bases de datos.

Una posible futura extensión de este sistema sería pasar el modelo de una arquitectura *two-tier* a una *three-tier*, obteniendo así una estructura en la que se cuenta con una aplicación servidora que establece el contacto con la base de datos y construye la tabla de metadatos, solicitada por los usuarios a través de navegadores o aplicaciones remotas. Estas aplicaciones remotas cumplirían el papel de clientes, y conforman la tercera parte de la arquitectura. Las implicias de esta decisión serían que la aplicación en la nueva arquitectura no contendría ninguna interacción con el usuario de manera directa; como así tampoco estaría involucrada en el proceso de visualización de manera concreta, solo provee los datos de la base solicitados en forma de una estructura Java.

## Referencias

- [BYRN99] Ricardo Baeza-Yate and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
- [CMS99] Stuart Card, John Mackinlay, and Ben Shneiderman. *Readings in Information Visualization*. Morgan Kaufmann, 1999.
- [Str98] Thomas Strothotte. *Computational Visualization*. Springer Verlag, 1998.
- [WFC<sup>+</sup>99] Seth White, Maydene Fisher, Rick Cattell, Graham Hamilton, and Mark Hapner. *JDBC API Tutorial and Reference, Second Edition*. Addison Wesley, 1999.

# Visualización Interactiva de Diagramas de Fase

Claudio Delrieux, Julián Dominguez y Andrés Repetto

*Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.*  
Voice: (54)(291)4591501 ext. 3381 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org

**Palabras Clave:** VISUALIZACIÓN CIENTÍFICA — SISTEMAS NO LINEALES Y CAÓTICOS — STREAMLINES — (LIC) LINE INTEGRAL CONVOLUTION.

## 1 Introducción

En la literatura se han presentado varios métodos para la visualización de campos vectoriales (glifos, trayectorias de partículas, íconos, Streamlines, Hyperstreamlines, Spot Noise, LIC, etc.) [1, 2, 3, 12, 15, 16], de las cuales Streamlines y LIC son las más importantes y utilizadas. En flujos estacionarios, el método de las Streamlines o líneas de flujo coincide con la integración de trayectorias en sistemas dinámicos [9, 14]: se posiciona una condición inicial (o *semilla*) en el espacio de fases, y se integra la ecuación diferencial por medio de diferenciales finitos. Para obtener un cubrimiento uniforme del diagrama de fases o para ubicar todos los puntos críticos del sistema, se requiere encontrar un conjunto representativo de trayectorias, lo cual implica encontrar las respectivas semillas, y el paso y la longitud de integración. La elección de un valor para el diferencial finito o factor de integración debe ser realizada cuidadosamente para no producir excesivo oversampling o undersampling. La única manera de asegurar un paso óptimo en toda circunstancia consiste en utilizar un paso adaptativo, como se sugiere en [4]. La cantidad de iteraciones, junto con el paso de integración y la velocidad media en la trayectoria, son todos factores que inciden en la longitud final que tendrá la misma en el diagrama de fases. En muchas circunstancias la correcta apreciación de un campo de velocidades se logra visualizando un gran conjunto de trayectorias relativamente cortas (“colas de cometa” [6, 7]), y para brindar una buena idea de la dirección del flujo se pueden representar las trayectorias modificando la saturación [8]. El posicionamiento uniforme de las semillas no garantiza un cubrimiento satisfactorio. Ni siquiera garantiza que todos los puntos críticos del sistema dinámico se representen adecuadamente. Una solución para estos problemas consiste en realizar la dinámica inversa del sistema a partir de la semilla, una cierta cantidad de iteraciones, para determinar si el punto está cerca de un repeledor. Esta técnica de “path-search” [5] permite lograr un cubrimiento más parejo del diagrama de fases, y además acelera la generación del diagrama de fases porque requiere integrar una menor cantidad de trayectorias. Un problema inevitable es que muchas de estas técnicas son directamente dependientes del sistema en particular, por lo que una visualización adecuada requiere de la asistencia del usuario.

Por su parte, la LIC (Line Integral Convolution) [1] realiza una integración más corta (típicamente se realizan menos de 100 iteraciones). La trayectoria generada no se grafica, sino que se utiliza para recorrer los texels correspondientes, integrando el color de los mismos para

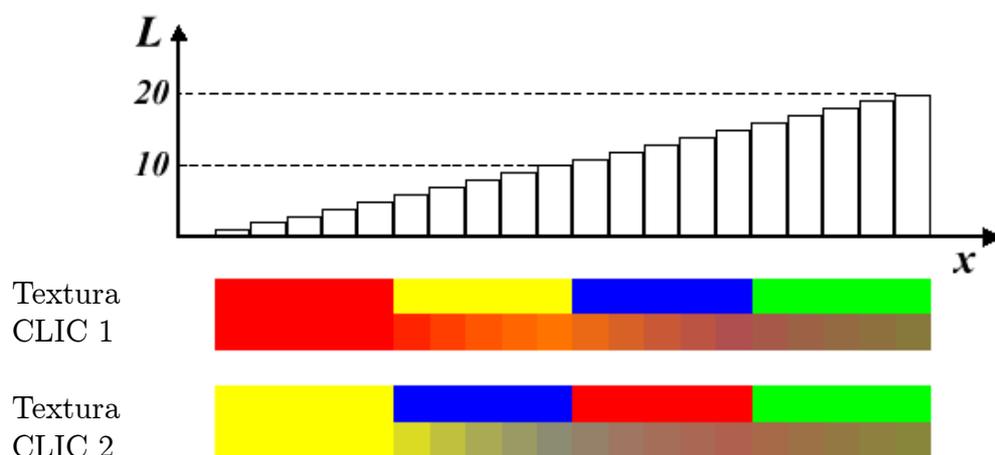


Figura 1: Representación gráfica (1D) del comportamiento del CLIC, aplicado a dos texturas. El sentido del flujo es de izquierda a derecha.

encontrar el color final del pixel. Por dicha razón, la LIC produce un cubrimiento uniforme del diagrama de fases, por lo que todos los puntos críticos del sistema son satisfactoriamente representados. Una mejora inmediata consiste en multiplicar, para cada pixel, el aporte de cada texel por medio de una misma función ponderadora o *kernel* de convolución de longitud  $L$  igual a la cantidad de pasos de integración. Sin embargo muchas veces resulta difícil identificarlos adecuadamente. No siempre se pueden distinguir atractores de repelidores, y resulta casi imposible determinar la existencia de trayectorias cerradas (focos, ciclos límite, trayectorias homoclínicas). La mejora de las posibilidades de la LIC constituye un activo campo de investigación [8, 10, 11, 13, 17], especialmente en lo que concierne a mejorar su costo computacional y la representación del sentido del flujo y las trayectorias cerradas. Muchas de las mejoras mencionadas más arriba para las streamlines son directamente aplicables en la LIC. En particular podemos mencionar que la LIC adaptativa es aproximadamente un orden de magnitud más veloz [4]. Por último, podemos mencionar un aspecto de importancia: la textura utilizada para la generación de la LIC. En efecto, la textura constituye un medio para introducir determinadas características geométricas y cromáticas en la imagen final, las cuales interactúan con la distribución geométrica de las trayectorias en el diagrama de fases. En [5, 6] se sugiere un conjunto de técnicas para generar texturas más adecuadas, las cuales luego pueden ser utilizadas para resaltar diversos aspectos de un determinado campo vectorial.

## 2 LIC acumulativo

El LIC acumulativo (*Cumulative LIC* o *CLIC*) [5, 6] surge de combinar las técnicas de desaturación de las trayectorias, y de posicionamiento no uniforme de las semillas en las streamlines. La idea consiste en acumular el color de los texels visitados en una LIC, en vez de computar una convolución (ver Fig. 1), e ir graficando el color acumulado en la imagen final. De esa forma, el cómputo del color final de un pixel es mucho más sencillo que con la LIC, pueden reutilizarse los cálculos parciales para encontrar los colores de los pixels vecinos, y cada trayectoria computada genera varios pixels en la imagen final. En general esta técnica permite bajar en promedio un orden de magnitud el tiempo de cómputo con calidades similares a la LIC original. Además el CLIC puede utilizarse con todas las demás

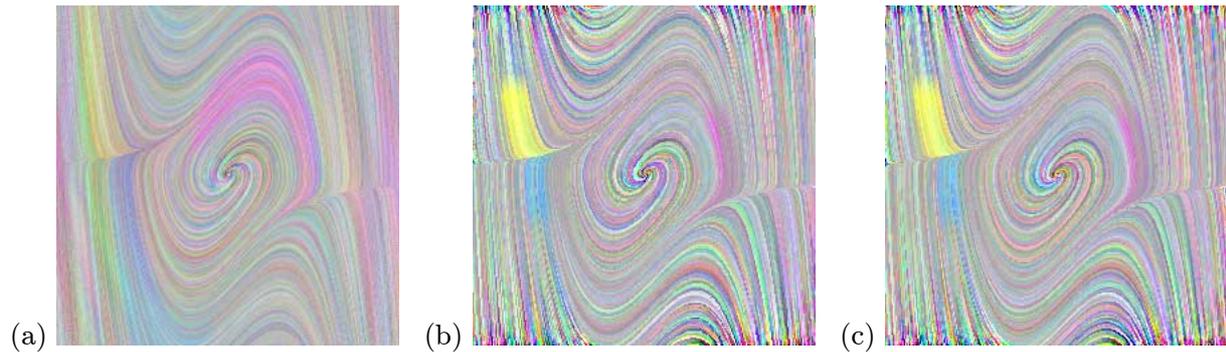


Figura 2: (a) LIC original,  $L=300$ ,  $t=212$  seg. (b) CLIC adaptativo sembrado en cada pixel,  $L=300$ ,  $t=2.5$  seg. (c) CLIC adaptativo con *path search*, sembrado al 10% y con segundo paso,  $L=300$ ,  $t=0.65$  seg.

técnicas mencionadas en la sección anterior (paso adaptativo, *path search*), con lo cual se obtiene un método cuyo tiempo de cómputo es muy cercano a los streamlines, pero con la calidad de la LIC. Un detalle importante del CLIC es que –manejado cuidadosamente– permite utilizar kernels mucho mayores que la LIC convencional ( $> 500$ ) sin un incremento dramático en el tiempo de cómputo. Aplicado en forma directa, sobre cada pixel de la figura final, produce resultados similares a una LIC de mismo kernel  $L$ , con tiempos sensiblemente menores (ver Fig. 2). El mejor aprovechamiento de la técnica consiste en efectuar un “sembrado” no uniforme en un determinado porcentaje de los pixels de la imagen de salida. Experimentalmente podemos mencionar que generando una trayectoria CLIC cada 5 o 10 pixels, se produce un cubrimiento casi total de la imagen de salida (dependiendo de cómo sea el diagrama de fases y del valor de  $L$  elegido). El cubrimiento de la imagen final puede mejorarse sensiblemente utilizando *path search*, y eventualmente se puede computar un segundo paso de CLIC para generar una trayectoria en cada pixel no visitado, obteniéndose de esa manera un cubrimiento total. Es fácil ver, entonces, que aumentando la longitud  $L$  del kernel no aumenta necesariamente el tiempo de cómputo del CLIC. Un  $L$  mayor genera trayectorias más largas, con más pixels visitados, lo cual permite disminuir tanto la cantidad de trayectorias a recorrer como la cantidad de cómputo en el segundo paso.

Una observación detallada realizada a las imágenes resultantes puede mostrar que las trayectorias con  $L$  grandes van perdiendo saturación muy rápidamente, lo cual dificulta la visualización de las características geométricas del diagrama de fases. Una forma de solucionar este problema consiste en utilizar una función ponderadora lineal (de muy bajo costo computacional) que permita acumular el color a lo largo de la trayectoria con menor intensidad. De esa forma el factor  $k$  o *carry* permite controlar la desaturación de las trayectorias. En la Fig. 3 podemos ver el efecto del factor  $k$  sobre el CLIC. Se observa que con menor  $k$  la influencia de la textura original es mayor.

### 3 Doble CLIC

Como pudimos ver en la Sección anterior, el CLIC y sus mejoras asociadas nos permiten visualizar diagramas de fase en tiempos semejantes a las streamlines, con calidades semejantes a la LIC. Sin embargo, el método es nuevamente dependiente del sistema dinámico en particular y requiere la elección cuidadosa de los parámetros por parte del usuario para ser correctamente utilizado. Según se mencionara más arriba, la textura de entrada juega un rol

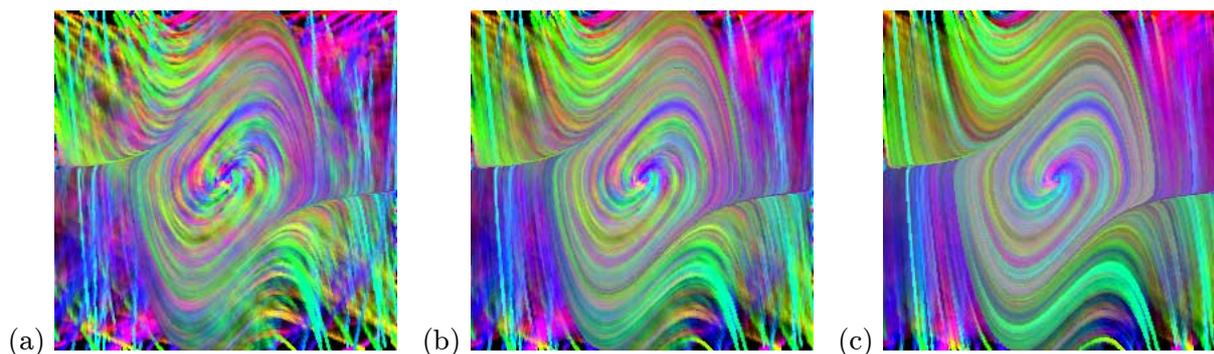


Figura 3: Visualizaciones del diagrama de fases del oscilador de Van der Pol (a)  $k = 0.2$ , (b)  $k = 0.5$ , y (c)  $k = 1$ .

preponderante en la calidad del resultado, y dado que el CLIC es notablemente económico para computar, la idea para lograr un método autónomo consiste en utilizar un CLIC (probablemente de baja calidad pero poco costo computacional) *como textura de entrada* para el cómputo del CLIC. Es decir, aplicamos dos veces el método. La primera transforma la textura original en una textura intermedia de mejor calidad, y la segunda produce la visualización final. De esa forma, el “doble CLIC” logra combinar la rapidez y autonomía de los streamlines con la calidad de la LIC. En la Fig. 4 podemos comparar, para un campo vectorial y una textura especialmente difícil, el efecto de aplicar dos veces el CLIC. De esa forma se logra un efecto visual de mayor semejanza al LIC original. Esta forma de trabajar permite elegir diferentes valores de  $k$  para realizar un CLIC inverso y luego hacia adelante (utilizando el primero como textura de entrada para el segundo). Se observa que la visualización producida por los nuevos métodos (CLIC y doble CLIC) son de calidad semejante a la LIC, con una eficiencia computacional mucho mayor.

## Referencias

- [1] B. Cabral y L. Leedom. Imaging Vector Fields Using Line Integral Convolution. *ACM Computer Graphics (SIGGRAPH Proceedings)*, 25(3):263–270, 1993.
- [2] Roger Crawfis, Nelson Max, y Barry Becker. Vector Field Visualization. *IEEE Computer Graphics and Applications*, 14(5):50–56, 1994.
- [3] Thierry Delmarcelle and Lambertus Hesselink. Visualizing Second-Order Tensor Fields with Hypersstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.
- [4] Claudio Delrieux. A Fast and Accurate Implementation of the Line Integral Convolution. En *CISST 2000 Conference*, págs 467–474, ISBN 1-892512-57-2, 2000.
- [5] Claudio Delrieux, Julián Dominguez, y Andrés Repetto. Mejorando la Visualización de Campos Vectoriales. In *VI ICIE*, págs. 397–416, 2001.
- [6] Claudio Delrieux, Julián Dominguez, y Andrés Repetto. Towards a CLIC in Vector Field Visualization. En *CISST 2001 Conference*, págs. 695–702, ISBN 1-892512-73-4, 2001.
- [7] Claudio Delrieux y Andrés Repetto. Visualización de Sistemas no Lineales y Caóticos. En *XXV CLEI '99*, págs. 241–253, 1999.

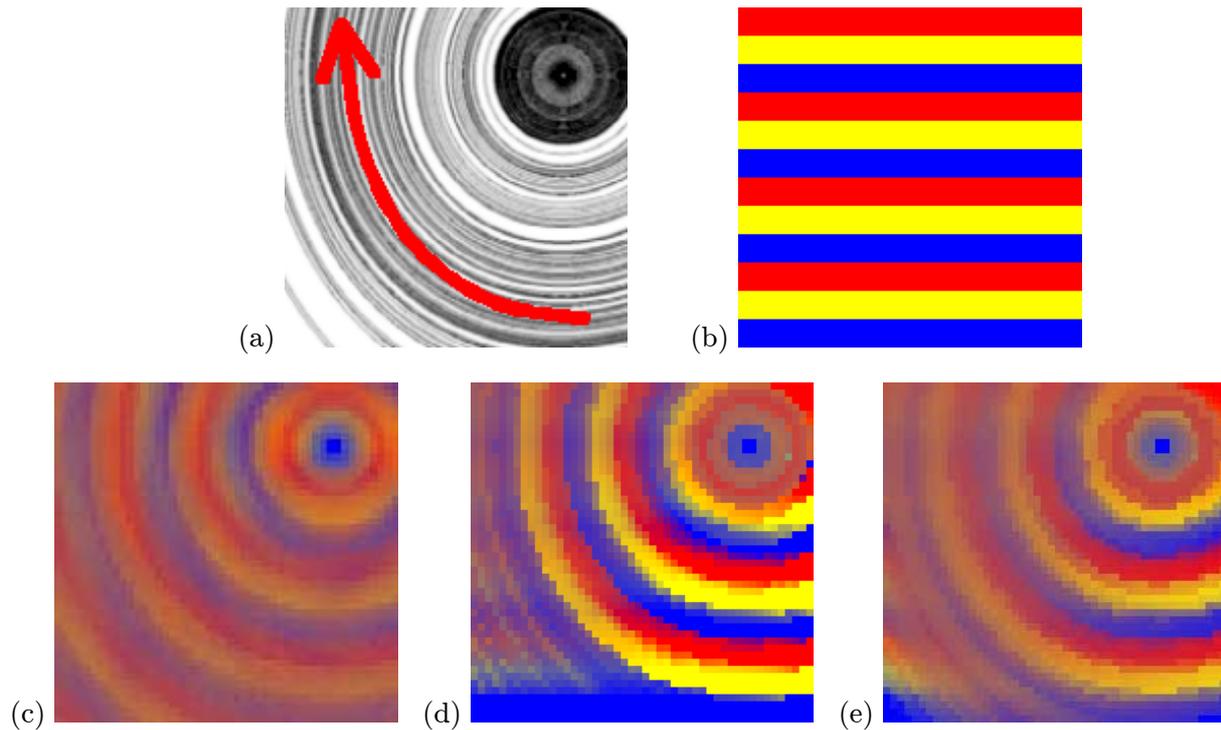


Figura 4: (a) Un campo vectorial, y (b) la textura utilizada. En (c) vemos la LIC respectiva, en (d) el CLIC y (e) el doble CLIC.

- [8] Claudio Delrieux y Andrés Repetto. Métodos para la Visualización de la Dinámica de Sistemas no Lineales. En *Argentine Symposium on Computing Technology*, págs. 153–162, 29 JAIIO, 2000.
- [9] Robert Devaney. *A First Course in Chaotic Dynamical Systems*. Addison Wesley, Reading, MA, 1992.
- [10] L.K. Forsell y S.D. Cohen. Using Line Integral Convolution for Flow Visualization: Curvilinear Grids, Variable-speed Animation, and Unsteady Flows. *IEEE Transactions on Visualization and Computer Graphics*, 1(2):133–141, 1995.
- [11] Hans-Christian Hege y Detlev Stalling. Fast LIC with Piecewise Polynomial Filter Kernels. En *Mathematical Visualization*, págs. 295–314. Springer-Verlag, Amsterdam, 1998.
- [12] James Helman y Lambertus Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, 1991.
- [13] D. Stalling y H.-C. Hege. Fast and Resolution Independent Line Integral Convolution. En *SIGGRAPH 95 Conference Proceedings*, págs. 249–256.
- [14] Stefen H. Strogatz. *Nonlinear Dynamics and Chaos*. Addison-Wesley, Reading, Massachusetts, 1994.
- [15] J. J. van Wijk. Spot Noise: Texture Synthesis for Data Visualization. *ACM Computer Graphics*, 25(4):309–318, 1991.
- [16] J. J. van Wijk. Flow Visualization with Surface Particles. *IEEE Computer Graphics & Applications*, 13(7):18–24, 1993.
- [17] R. Wegenkittl, E. Gröller, y W. Purgathofer. Animating Flow-fields: Rendering of Oriented Line Integral Convolution. En *Computer Animation '97 Proceedings*, págs. 15–21.

# Interfaces en la Visualización de Información

Lic. Sandra Di Luca   Lic. Sergio Martig   Mg. Silvia Castro  
Guillermo Trutner   Lic. Mercedes Vitturini

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur  
Laboratorio de Investigación en Visualización y Computación Gráfica  
Universidad Nacional del Sur – Bahía Blanca  
Instituto de Investigación en Ciencia y Tecnología Informática (IICyTI)  
{sdiluca,gtrutner}@criba.edu.ar, {smartig,smc,mmv}@cs.uns.edu.ar

## Introducción

La creciente cantidad de aplicaciones en Visualización de Información hace que la misma sea actualmente un área de activo desarrollo. Así como los usuarios creativos empujan los límites de las herramientas actuales, los diseñadores serán presionados para proveer aún mayor funcionalidad.

La visualización y la exploración de colecciones de información se vuelve cada vez más dificultosa a medida que el volumen y la diversidad aumenta. El proceso de extracción de información es un proceso impreciso y, cuando los usuarios pretenden un acercamiento al sistema, tienen a menudo un entendimiento vago de cómo lograr sus objetivos.

Existen diversas técnicas de Visualización de Información que ayudan en la tarea de exploración de la información. Estas son atractivas ya que permiten tanto la presentación de la información bajo distintos aspectos como una exploración controlada de la misma por parte del usuario. Si bien estas tareas son propias de la Visualización de Información, deben integrarse adecuadamente a la Interface. En este contexto la interface podrá asistir a los usuarios a formular sus consultas, navegar y seleccionar entre las fuentes de información disponibles y entender los resultados de la visualización.

Las mejoras logradas en las interfaces de usuario para consultas en Bases de Datos y búsquedas en documentos de texto están dando como resultado nuevos productos. En la actualidad son posibles acercamientos gráficos originales y de manipulación directa tanto en la formulación de consultas como en la visualización de información y en la interacción.

Mientras los prototipos han tratado solamente con un tipo de dato (uni-, bi-, y tri-dimensional, datos temporales y multidimensionales, árboles y redes), los futuros productos comerciales deberán combinar varios. Estos productos necesitarán proveer una buena integración con el software existente y soportar tareas como zoom, filtros, detalles sobre demanda e historia. Estos métodos son atractivos porque presentan la información rápidamente y permiten una exploración controlada por parte del usuario. Si todos ellos deben ser completamente efectivos, requeriremos estructuras de datos avanzadas, displays de alta resolución y nuevas formas de entrenamiento de los usuarios. En la actualidad, por ejemplo, están siendo construidas varias interfaces de usuario para especificar filtros avanzados que están siendo evaluadas para incluirlas en distintos proyectos comerciales.

La búsqueda de información con estructura compleja (incluyendo, entre ellos, gráficos, imágenes, sonido o video) y la presentación de la información misma brindan una gran oportunidad para el diseño de interfaces de usuario avanzadas y motores de búsqueda poderosos.

Se necesita de una mejor integración con la psicología perceptual y con la toma de decisiones, ya que son fundamentos teóricos y medidas prácticas para la elección a través de las diversas técnicas de visualización. Los estudios empíricos ayudarán a ordenar la situación específica en la cual la visualización sea mas útil. Finalmente, las herramientas de software para la construcción de visualizaciones innovadoras, integradas adecuadamente a la interface, facilitarán el proceso de exploración.

Así es que el objetivo de nuestro trabajo consiste en delinear criterios para el diseño de Interfaces que asistan al usuario particularmente en todas las etapas del proceso de Visualización de Información. Además cabe señalar que, como una aplicación significativa de la Visualización de Información lo constituye la Visualización de Bases de Datos, se comenzará teniendo en cuenta los principios para el diseño de Interfaces instanciados adecuadamente a Bases de Datos. El enfoque sobre una aplicación ayudará a cubrir la distancia entre las propuestas teóricas y las aplicaciones de las mismas y es un modo de establecer el valor de las técnicas de Interfaces.

## **Diseño de la Interfaces a lo largo del proceso de Visualización de Información**

Debido a la gran variedad de dominios de la Visualización de Información, el desafío es diseñar un ambiente que permita a los usuarios llevar a cabo, de manera intuitiva, una variedad de tareas de visualización. Aunque diferentes dominios de aplicación requieren representaciones visuales distintas, muchos de estos comparten operaciones de transformación de los datos y manipulaciones similares a lo largo del pipeline de visualización.

Las interfaces deben ayudar en las distintas etapas del proceso de visualización: deben soportar el proceso de búsqueda mediante la especificación de las consultas, debe ayudar en el proceso mismo de la exploración de la información presentada y deben, sin duda, ser un apoyo en el proceso de interacción a lo largo de todo el proceso, de modo tal que el usuario pueda satisfacer su necesidad de acceder y explorar la información.

En todas las etapas de la Visualización de Información, debemos contar con interfaces apropiadas que apoyen al usuario en sus tareas específicas. Desde el punto de vista del diseño de las interfaces, los usuarios tienen habilidades, diferencias y predilecciones muy variadas; son muchos los factores que pueden conducir tanto al rechazo como a la aceptación de las distintas técnicas de Interfaces; además, esta diversidad plantea la necesidad de un diseño muy flexible y de la incorporación de características, en cada etapa, que las haga efectivas. De acuerdo a esto podemos detallar lo particular en cada uno de los procesos involucrados en la Visualización de Información:

- En la etapa de la presentación inicial de la visualización se le debe proveer al usuario una buena manera de comenzar el proceso exploratorio. Una pantalla en blanco no brinda una forma adecuada de comenzar este proceso. Al comienzo, un punto importante es que la interface ayude a los usuarios a seleccionar las fuentes de datos a las que quiere interrogar y que éstas se presenten de manera útil. A partir de este momento, el usuario tratará de satisfacer la o las metas que tenga y dará por comenzado el pipeline de visualización.
- Luego comenzará la interacción con la visualización introduciendo consultas, observando los resultados y modificando las consultas en un ciclo interactivo hasta ver satisfechas sus

necesidades de información. Esta segunda etapa lo constituye la especificación de las consultas. La formulación de una consulta debe permitir al usuario *seleccionar* colecciones, describir metadatos o conjuntos de información de acuerdo a lo pedido en la misma y también debe permitir *especificar* información que pueda coincidir con la información en la base de datos. A partir de la consulta, el sistema creará la información derivada que será mostrada de alguna manera adecuada. En este punto deben verse las posibles interfaces y sus características de acuerdo al estilo de interacción elegido para la especificación de las consultas.

- Se pondrá además especial énfasis en las etapas específicas que correspondan a la presentación de la Información. En esta etapa se deben tener en cuenta los distintos métodos que hay para la presentación de datos multidimensionales y los desafíos que plantean las Interfaces en lo referente a mostrar los datos y sus interrelaciones.
- La interacción del usuario en cada una de las etapas del pipeline de visualización provocará seguramente una realimentación. Es por ello que el sistema deberá brindar un ambiente en el que la interface sea clara y amigable para lograr aprovechar al máximo la flexibilidad, creatividad y conocimiento general que el usuario posee del problema real. De esta manera se establecerá un flujo interactivo coherente entre el usuario y el sistema de visualización.

Actualmente se está trabajando en el diseño de un prototipo integrado de visualización, en el que la interacción del usuario esté presente en las distintas etapas del proceso de Visualización. En esta primera etapa se está realizando un primer abordamiento en la visualización de Base de Datos seleccionando dentro del conjunto de técnicas generales de visualización a las Coordenadas Paralelas por considerar la multidimensionalidad de los datos. Nuestro objetivo es el diseño de interfaces efectivas en cada una de las etapas del proceso de visualización siguiendo los principios para el diseño de la misma e instanciándolos adecuadamente de acuerdo a la aplicación señalada.

## Conclusiones

La importancia de la Interacción Humano-Computadora está siendo reconocida como un campo importante de Ciencias de la Computación, y de acuerdo a lo visto surge sin duda el rol relevante de la interface en todo el proceso de acceso e interacción con la información.

Por ello es necesario contar con interfaces efectivas que apoyen las distintas tareas a realizar en todo el proceso de visualización. Las interfaces deben seguir los principios de diseño de interfaces de usuario y cada uno de estos principios debe ser instanciado adecuadamente dependiendo de la etapa particular en la que el usuario interactúa con la visualización.

## Bibliografía

1. Apple Computer, *Human Interface Guidelines: The Apple Desktop Interface*, Addison-Wesley, 1987.
2. Baecker, R. M. and Buxton, W. A. S. , *Readings in Human-Computer Interaction*. San Mateo CA.: Morgan Kaufmann Publishers.
3. Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval*, Addison Wesley, 1999.

4. Card, S., Shneiderman, B., Mackinlay, J., *Readings en Information Visualization – Using Vision to Think*, Morgan Kaufmann, 1999.
5. Chaomei, Ch. *Information Visualization and Virtual Environments*, Springer Verlag, 1999.
6. Dix, A., Finlay, J., Abowd, G., Beale, R. , *Human-Computer Interaction*, Prentice Hall Europe, Second Edition, 1998.
7. Foley, J., Van Dam, A., *Fundamentals of Interactive Computers Graphics*, Addison-Wesley, Reading, Massachussetts, segunda edición, 1992.
8. Newman,W., Sproull,R., *Principles of Interactive Computer Graphics*, McGraw-Hill, New York, 1973.
9. Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T., *Human-Computer Interaction*, Addison Wesley, 1997.
10. Raskin, J., *The Human Interface*, Addison Wesley, ACM Press, 2000.
11. Shneiderman, B., *Designing the User Interface*, Addison-Wesley Publishing Company, 1998.
12. Strothotte, T., *Computational Visualization*, Springer Verlag, 1998.
13. Tufte, E., *The Visual Display of Quantitative Information*, Graphics Press, 1983.

# Visualización de Reglas de Asociación

## Un acercamiento comparativo

G. Filocamo W. Grandinetti S. Castro<sup>1</sup> S. Martig<sup>1</sup> C. Chesñevar<sup>2</sup>

Departamento de Ciencias e Ingeniería de la Computación  
UNIVERSIDAD NACIONAL DEL SUR  
Av. Alem 1253 – B8000CPB Bahía Blanca – REPÚBLICA ARGENTINA  
TEL/FAX: (+54) (291) 459 5135/5136 – EMAIL: {smc, smartig, cic}@cs.uns.edu.ar  
PALABRAS CLAVE: Reglas de Asociación, Datamining, Visualización

### Resumen

La minería de datos en búsqueda de *reglas de asociación* intenta encontrar relaciones de asociación o correlación entre grandes colecciones de datos. Si bien las reglas de asociación brindan una técnica que sintetiza una gran colección de datos, las mismas por sí solas pueden resultar poco informativas si no son visualizadas correctamente.

Este trabajo presenta distintos aspectos vinculados a la integración de reglas de asociación con técnicas de visualización. Se discuten las características de un software (**Miner.ar**) para obtención de reglas de asociación, y distintas alternativas para su integración con técnicas de visualización.

## 1 Introducción y motivaciones

El *descubrimiento de conocimiento a partir de bases de datos*, denominado genéricamente *datamining*, es un proceso de extracción no trivial de información implícita, previamente desconocida y potencialmente útil a partir de bases de datos. Dentro de las diversas técnicas existentes, las *reglas de asociación* constituyen una de las técnicas de datamining que intenta encontrar relaciones de asociación o correlación entre grandes colecciones de datos.

La obtención de reglas de asociación a partir de grandes bases de datos transaccionales resulta atractiva en diversas áreas. Algunos ejemplos de aplicación son la determinación de nuevas estrategias de marketing para organizaciones comerciales, detección de fraudes financieros, determinación de la causa de enfermedades y utilización de patrones de información climática para predicción meteorológica [Toi96]. En tal sentido, cabe señalar que, si bien las reglas de asociación brindan una técnica que sintetiza una gran colección de datos, las mismas por sí solas pueden resultar poco informativas si no son presentadas al usuario final de manera correcta. Es aquí donde resulta importante contar con *técnicas de visualización* [CMS99, Chi00] apropiadas para resolver este problema.

Este trabajo describe los principales aspectos de una línea de investigación que ha comenzado a ser desarrollada en el ámbito del Departamento de Cs. e Ingeniería de Computación de la Universidad Nacional del Sur, que involucra la integración de reglas de asociación con técnicas de visualización. El trabajo se estructura como sigue: primeramente se describen los principales conceptos teóricos para obtención de las reglas de asociación. Luego se mencionan brevemente las características del software **Miner.ar** [FG02] para obtención de reglas de asociación, y finalmente se discuten los elementos más relevantes al momento de integrar dichas reglas con técnicas de visualización.

---

<sup>1</sup>Laboratorio de Visualización y Gráfica (VyGLab), UNS.

<sup>2</sup>Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA), UNS.

## 2 Reglas de Asociación en Bases de Datos

Seguidamente introduciremos las definiciones básicas necesarias para la caracterización de reglas de asociación. Como punto de partida se toma un conjunto de ítems  $\mathcal{I} = \{i_1, i_2, \dots, i_m\}$  y un conjunto  $\mathcal{D}$  de transacciones, donde cada transacción  $d \in \mathcal{D}$  tiene un identificador único y contiene un conjunto de ítems, también llamado *itemset*. Un itemset con  $k$  ítems es también llamado un *k-itemset*. El *soporte* de un itemset  $X$ , denotado como  $\sigma(X)$ , es el número de transacciones en las cuales  $X$  aparece como subconjunto. Un itemset es *maximal* si no es subconjunto de ningún otro itemset. Un itemset es *frecuente* si su soporte es superior al *mínimo soporte* (*min-sup*) especificado por el usuario.

Una **regla de asociación** es una implicación de la forma  $X \Rightarrow Y$ ,  $X \subset \mathcal{I}$ ,  $Y \subset \mathcal{I}$  y  $X \cap Y = \emptyset$ . El *soporte* de una regla está dado por  $\sigma(X \cup Y)$  y la *confianza* como  $\sigma(X \cup Y) / \sigma(X)$  (i.e., la probabilidad condicional que una transacción contenga  $Y$ , dado que contiene  $X$ ). Una regla es *fuerte* si su confianza es mayor a la mínima confianza especificada por el usuario (*min-conf*). NOTA: nótese que las reglas de asociación no son transitivas, en virtud de que  $S \Rightarrow T$  y  $T \Rightarrow U$  no implica necesariamente  $S \Rightarrow U$ .

El problema de extraer reglas de asociación de un conjunto de transacciones  $\mathcal{D}$  consiste en generar todas las reglas de asociación que tengan un soporte y confianza mayor que el mínimo especificado por el usuario (*min-sup* y *min-conf* respectivamente). Esta tarea puede ser descompuesta en dos partes:

- *Buscar los itemset frecuentes.* Dados  $m$  ítems existen potencialmente  $2^m$  itemset frecuentes, por lo que su computación por fuerza bruta es inviable. Se requieren algoritmos especializados para esta tarea.
- *Generar todas las reglas de asociación fuertes.* Estas serán reglas de la forma  $X \setminus Y \Rightarrow Y$  donde  $Y \subset X$ , para todos los itemsets  $X$  frecuentes que superen el mínimo de confianza.

La obtención de los itemset frecuentes es en consecuencia un aspecto clave en la determinación de reglas de asociación. Existen diversas técnicas, destacándose las siguientes:

**Apriori** [AS94] Ampliamente utilizado por su simplicidad. Utilizando el principio anterior se generan los itemset de tamaño  $k$  candidatos a partir de los itemset frecuentes previos.

**FP-Growth** [HPY00] Es un método más reciente y supera al anterior. Aplica la técnica de *dividir y conquistar* y con ello reduce el espacio de búsqueda, transformando el problema de buscar largos fragmentos de patrones en buscar fragmentos más cortos y luego realizar la concatenación de sufijos construyendo un árbol y trabajando sobre el mismo.

### 2.1 El software Miner.ar. Características

El sistema **Miner.ar** [FG02] es una herramienta desarrollada en Java para extracción de reglas de asociación. Provee una interface gráfica que permite tomar una base de datos arbitraria, seleccionar atributos específicos y extraer reglas de asociación a partir de parámetros definidos por el usuario. El método de extracción usado es *FP-Growth*. Se realizaron experimentos de su performance a partir de una base de datos con información administrativa de la Universidad Nacional del Sur (más de 12.000 transacciones).<sup>3</sup>

<sup>3</sup>Para mayores detalles consultar [FG02].

### 3 Visualización de Reglas de Asociación

Las *reglas de asociación* constituyen uno de los patrones más importantes que pueden ser descubiertos utilizando datamining. Su exploración es una de las tareas centrales en el proceso de la minería de datos. Se ha invertido mucho tiempo y esfuerzo de investigación en el estudio de las mismas, principalmente en lo que hace a su generación o descubrimiento de una manera eficiente. La inclusión de algoritmos que permitan su descubrimiento de manera automática en la mayoría, sino en todas, las herramientas de datamining es un indicativo de su importancia. Centramos nuestra atención en dos aspectos de la manipulación de las reglas de asociación que aún no han sido totalmente resueltos:

- La gran cantidad de reglas generadas.
- La dificultad en el entendimiento de las reglas de asociación.

Realizaremos nuestro abordaje a los problemas planteados desde la Visualización de Información.

#### 3.1 Visualización: Conceptos preliminares

La *Visualización de Información* favorece el entendimiento y el análisis de datos abstractos mediante el uso de Computación Gráfica Interactiva y de Técnicas de Visualización. Los datos abstractos no son inherentemente geométricos y presentan desafíos a los investigadores en Visualización porque evidentemente no es obvio poner los datos abstractos en formas visuales efectivas. La Visualización de Información es la representación gráfica adecuada tanto de los datos con parámetros múltiples como de las tendencias y las relaciones subyacentes que existen entre ellos. Su propósito no es la creación de las imágenes en sí mismas sino la asimilación rápida de información o monitoreo de grandes cantidades de datos. Este medio es promisorio fundamentalmente por varias razones:

- Acrecienta los recursos del humano en la forma de procesamiento perceptual expandiendo su memoria de trabajo.
- Favorece el reconocimiento de patrones.
- Permite el uso de inferencia y monitoreo perceptual.
- El medio mismo es manipulable e interactivo.

Un aspecto fundamental en las técnicas de Visualización de Información es el de la interacción, pues ésta es central para el proceso exploratorio del espacio de información, siendo la generadora de un proceso de retroalimentación que potencia la tarea. En un sentido amplio, podemos atacar el problema de visualizar grandes volúmenes de información desde distintas perspectivas:

- En uno de los extremos tenemos la visualización de los datos propiamente dichos, tratando de mostrar visualmente todos los items de datos, o su mayoría, y proveer técnicas que favorezcan su exploración. La principal limitación de estas técnicas es que podemos mostrar simultáneamente un número limitado de ítems de datos.

- Una alternativa es mostrar visualmente algún tipo de metainformación (por ej. algún tipo de resultado producido por datamining como son las reglas de asociación) y favorecer la exploración visual de ésta en un segundo nivel de abstracción o en una etapa de sintonía fina. Esta estrategia, si bien manipula conjuntos de datos de cardinalidad inferior a la del conjunto original, debe permitir visualizar los conjuntos de datos generados por los algoritmos de datamining que suelen ser de gran tamaño y difíciles de entender para los usuarios.

### 3.2 Aspectos relevantes a visualizar en Reglas de Asociación

Como se vio anteriormente, las reglas de asociación de la forma  $X \Rightarrow Y$ , son interpretadas como: “Si en una transacción ocurre  $X$ , entonces es frecuente que en la misma transacción también ocurra  $Y$ ”. En este contexto, la noción de “frecuente” es cuantificada por dos medidas (el soporte y la confianza), las cuales reflejan la utilidad y la certeza de la regla respectivamente. Es indudable que la manera en que se visualicen las reglas deberá proveer información sobre:

- Los valores que componen  $X$
- Los valores que componen  $Y$
- La confianza de cada regla
- El soporte que tiene cada regla

Previamente se estableció que el problema de las reglas de asociación ofrecía por lo menos dos frentes: **cantidad de las reglas** y el **entendimiento de las mismas**. En cuanto a la cantidad de las reglas generadas por los algoritmos, se ha realizado mucho trabajo para filtrar y generar conjuntos de reglas de asociación fuertes. El problema surge del hecho de que las reglas desechadas pueden aportar al usuario información que le sería de utilidad para entender las reglas fuertes que quedaron en el conjunto. Lo anterior no significa que los algoritmos no deban realizar un filtrado de las reglas, todo lo contrario, lo que sí se debe considerar es la manera de ofrecer información adicional y herramientas que le permitan al usuario comprender la estructura subyacente de las reglas de asociación.

## 4 Línea de investigación a desarrollar

En una primer instancia se realizará un relevamiento de las distintas técnicas utilizadas en distintas herramientas disponibles para los fines propuestos. Se evaluarán las mismas con respecto a diferentes perfiles de usuario desde el punto de vista de la efectividad y la facilidad de uso de las mismas.

Se realizará en particular un análisis de los acercamientos utilizados en los principales paquetes de software para datamining ya existentes (ej. DBMINER). Se contrastarán las técnicas usadas en estos paquetes con aquellas técnicas que han sido exploradas e implementadas en el ámbito del VyGLab.

En una etapa posterior se tratará de combinar el software **Miner.ar** con las herramientas visuales para potenciar el software existente, a partir de la incorporación de diferentes técnicas de visualización.

## 5 Trabajos relacionados. Conclusiones

La visualización de reglas de asociación ha sido explorada en distintas variantes. Entre algunas de las muchas alternativas propuestas pueden mencionarse el uso de un *ball graph* (grafo de bolas) que interrelaciona distintas reglas de asociación, la utilización de una ‘grilla’ de reglas<sup>4</sup> y propuestas basadas en ‘Mosaic plots’ [HSW00]. También se ha estudiado el análisis visual de reglas de asociación a lo largo de un período de tiempo [ZL00].

La extracción de reglas de asociación a partir de grandes bases de datos transaccionales resulta atractiva en diversas áreas, y su aplicabilidad está condicionada en gran medida a la posibilidad de visualizarlas adecuadamente.

Si bien se han desarrollado diferentes abordajes en la visualización de reglas de asociación, entendemos que los mismos carecen muchas veces de un enfoque unificador y obedecen a intuiciones de los diseñadores de software. Consideramos que dicho enfoque puede ser provisto y sistematizado a partir de las técnicas de visualización existentes, muchas de ellas desarrolladas a nivel prototípico en el VyGLab. En esta dirección se desarrollarán las etapas de investigación futuras.

## Referencias

- [AS94] AGRAWAL, R., AND SRIKANT, R. Fast algorithms for mining association rules. In *Proceedings of the 20th VLDB Conference. Santiago, Chile (1994)*.
- [Chi00] CHI, E. A taxonomy of visualization techniques using the data state reference model. In *IEEE Visualization 2000, Actas en CD-Rom (2000)*.
- [CMS99] CARD, S., MACKINLAY, J., AND SHNEIDERMAN, B. *Readings in Information Visualization - Using Vision to Think*. 1999.
- [FG02] FILOCAMO, G. R., AND GRANDINETTI, W. Reglas de asociación para datamining – teoría y aplicaciones. *Tesis de Licenciatura – Dep. de Cs. e Ing. de la Computación – Universidad Nacional del Sur (2002)*.
- [HPY00] HAN, J., PEI, J., AND YIN, Y. Mining frequent patterns without candidate generation. In *Proc. 2000 ACM-SIGMOD Int. Conf. on Management of Data, Dallas, TX (May 2000)*.
- [HSW00] HOFMANN, H., SIEBES, A., AND WILHELM, A. Visualizing association rules with interactive mosaic plots. In *KDD 2000 (2000)*, ACM.
- [Toi96] TOIVONEN, H. *Discovery of Frequent Patterns in Large Data Collections*. PhD thesis, University of Helsinki, Finland, 1996.
- [Tuf97] TUFTE, E. *Visual Explanations: Images and Quantities, Evidence and Narrative*. 1997.
- [WB95] WONG, P. C., AND BERGERON, R. D. 30 years of multidimensional multivariate visualization. In *Proc. Workshop on Scientific Visualization (1995)*, IEEE Computer Society Press.
- [ZL00] ZHAO, K., AND LIU, B. Visual analysis of the behavior of discovered rules. In *KDD 2000 (2000)*, ACM.

---

<sup>4</sup>Estas técnicas en particular han sido utilizadas en el software comercial DBMINER.

# Generación de Objetos Flexibles 3D

Adela González

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial<sup>1</sup>  
Departamento de Informática  
Universidad Nacional de San Luis (UNSL)  
Ejército de los Andes 950  
5700, San Luis, Argentina  
e-mail: adelag@unsl.edu.ar

## Resumen

El objetivo de esta investigación es desarrollar una herramienta para la generación de objetos 3D modelados con materiales flexibles. Se desea priorizar la simplicidad de la interfaz con el usuario en el tratamiento interactivo de las deformaciones del objeto. Computar las deformaciones del objeto sujeto a fuerzas externas, como son los elementos con los que se simulará el moldeado lo que implica el tratamiento entre el objeto deformable y objetos rígidos.

## 1. Introducción

La investigación sobre modelado de objetos deformables se ha incrementado aceleradamente en los últimos años [1, 6, 7, 11], la simulación de este tipo de objetos es necesaria en varias áreas en computación gráfica:

- En animación gráfica: entre otros para la generación de movimientos de ropas [5], expresiones faciales [3, 12, 13], movimiento del cuerpo humano, video juegos interactivos, animación de dibujos animados, en sistemas de entrenamiento en cirugías para proveer un rendering gráfico realístico del tejido humano [1, 7], etc.
- En modelado interactivo: [15] donde diseñar objetos poliédricos de forma libre 3D es en general una tarea pesada, aquí la aplicación de técnicas de deformación a algún objeto creado puede resultar beneficiosa.
- En la simulación del comportamiento de ciertos materiales con propiedades que permiten la deformación (por ejemplo los plásticos o elásticos) para resaltar interacción dinámica con el entorno como ser en la simulación de una pelota que rebota o de la colisión de un auto en carrera.

El tratamiento computacional de objetos deformables involucra una combinación compleja de resultados que se extienden desde estimar parámetros mecánicos a solucionar grandes sistemas de ecuaciones diferenciales, detectar colisiones, modelar respuestas a las colisiones, entre otros. Estos procesos generan problemas cuando las simulaciones deben ser ejecutadas en tiempo real [1, 3, 8, 9, 10, 14].

Por otro lado el diseño de objetos poliédricos de forma libre 3D es en general una tarea ardua. Las técnicas constructivas de sólidos geométricos son muy útiles en el contexto del diseño industrial, pero parecen ser menos flexibles para aplicaciones con animación de caracteres o simulación de

---

<sup>1</sup> Director: PhD. Raúl Gallard. El grupo de investigación está soportado por la Univ. Nacional de San Luis, y la ANPCyT.  
<http://www.lidic.unsl.edu.ar/>

objetos biológicos. Frecuentemente los objetos son modelados por medio del muestreo de objetos existentes 3D y modificaciones subsecuentes. Un método alternativo es el modelado a través de superficies implícitas [17, 18].

Algunos sistemas permiten el diseño de modelos 3D [5] posibilitando un prototipo rápido en las primeras etapas de diseño, esto favorece a aplicaciones de interés educativo o recreativas.

## 2. Modelo y Técnicas de deformaciones

La deformación de los objetos ocurre por ejemplo cuando se mueven los vértices de un objeto poligonal o cuando se alteran los puntos de control de una curva paramétrica, la deformación exitosa de un objeto poligonal requiere que existan suficientes números de polígonos en el mismo ya que si la resolución es baja las deformaciones podrían resultar en una degradación de los bordes de la silueta o en aliasing de bordes.

La técnica de deformación de forma libre (FDD) desarrollada por Watt [2], sumerge al objeto en un espacio que puede ser deformado, una analogía comúnmente usada es considerar al objeto embebido en un paralelepípedo imaginario de plástico flexible, transparente. Cuando esta estructura es deformada el objeto en su interior también lo será ya que los objetos se consideran que son flexibles por lo que pueden ser deformados con el plástico que los rodea.

En el desarrollo de interfaces simples para el modelado encontramos el trabajo de Igarashi [5] que desarrolló un sistema de modelado de forma libre basado en un bosquejo 3D, donde el usuario simplemente realiza un bosquejo 2D con movimientos del cursor del mouse y el sistema automáticamente genera una geometría 3D razonable de forma libre con un método novedoso para inflación de polígonos, en este trabajo el objetivo fue explorar el concepto de *UI de forma libre*, donde utilizó un rendering no-fotorealístico interactivo con el propósito de evitar que el usuario se preocupe de los detalles y se dedique a una tarea explorativa.

Soporta funciones de transformación: doblar, alargar y torcer al objeto pero a partir de indicaciones de formas objetivos finales (target) preservando la topología poligonal del acoplamiento, no de la realización de una operación directa sobre el objeto a deformar.

La principal motivación de la técnica de inflado de polígonos es tener una alternativa computacionalmente menos costosa de modelado [4]. La técnica de inflación del polígono, puede ser útil en la etapa del diseño de un modelo poliedral 3-D. Puesto que en muchos aspectos es similar en alcance y propósito al modelado de superficies implícitas, puede considerarse como una alternativa a esta técnica de modelado; ofrece una definición de forma intuitiva vía un esqueleto ubicado en espacio, y soporta una gran variedad de formas realizables. Es una generalización de la operación de extrusión para aumentar polígonos a objetos prismáticos poligonales. La configuración inicial del polígono y el mesh poligonal que resulta son estructuras topológicas 2D embebidas en el espacio 3-D.

## 3. Desarrollo:

Se estudia el hecho que las deformaciones sean realizadas con alguna herramienta (por ejemplo al moldear plastilina se emplean los dedos) que deberá simularse, luego será necesario un estudio sobre su interacción con el objeto a deformar.

Aquí, nos centramos en dos problemas principales:

- Computar la deformación de un objeto conforme a fuerzas externas,
- Manejar la interacción entre objetos que se deforman y objetos rígidos, como son el objeto con la herramienta que lo moldeará.

Nuestra meta es el desarrollo de estructuras y algoritmos eficientes que pueden procesar modelos en tiempos compatibles para una función interactiva. Para ello, se explora el hecho de realizar deformaciones *locales*, limitando los cálculos a las porciones de los objetos que experimentan deformaciones significativas.

En este sentido se estudian diversas técnicas con el objetivo de generar objetos blandos (o con capacidad de deformarse) forma libre 3D interactivamente, favoreciendo la simplicidad de modelado e incorporando la posibilidad de deformar los objetos que se definan flexibles simulando distintos tipos de materiales.

Otra dirección importante de la investigación es desarrollar operaciones de modelado para crear pliegues, torcer el modelo, etc. Es necesario definir entonces técnicas para el desarrollo de una interfaz simple que trate la complejidad inevitable del soporte de estas operaciones.

#### 4. Bibliografía

1. Joel Brown, Stephen Sorkin Cynthia Bruyns Jean-Claude Latombe Kevin Montgomery Michael Stephanides *Real-Time Simulation of Deformable Objects: Tools and Application*, Stanford University
2. Watt Alan, Watt Mark y Parry, Scott. *Free Form Deformation of solid Geometric Models*. SIGGRAPH, volume 20, Number 4, 1986. 151-159
3. P.Kalra, A. Mangili, N. M.Thalmann,D. Thalmann *3D Interactive Free Form Deformations for Facial Expressions*, University of Geneva, Switzerland
4. C.W.A.M. van Overveld, B.Wyvill *Polygon Inflation for Animated Models: a method for the extrusion of arbitrary polygon meshes*. University of Calgary October 18, 1996
5. Takeo Igarashi†, Satoshi Matsuoka†, Hidehiko Tanaka *Teddy: A Sketching Interface for 3D Freeform Design* ACM SIGGRAPH 99. University of Tokyo, Tokyo Institute of Technology.
6. D. Baraff and A. Witkin. *Large steps in cloth simulation*. In *ACM SIGGRAPH 98 Conference Proceedings*, pages 43–52,1998.
7. M. Bro-Nielsen and S. Cotin. *Real-time volumetric deformable models for surgery simulation using finite elements and condensation*. *Computer Graphics Forum (Eurographics '96)*, 15(3):57–66, 1996.
8. J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Ponamgi. *ICOLLIDE: An interactive and exact collision detection system for large-scale environments*. In *Proceedings of ACM Interactive 3D Graphics Conference*, pages 189–196, 1995.
9. S. Gottschalk, M. C. Lin, and D. Manocha. *OBB-tree: A hierarchical structure for rapid interference detection*. In *ACM SIGGRAPH 96 Conference Proceedings*, pages 171– 180, 1996.
10. S. Quinlan. *Efficient distance computation between non convex objects*. In *Proceedings of the IEEE International Conference On Robotics and Automation*, pages 3324–3329, 1994.

11. G. van den Bergen. Efficient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.
12. Terzopoulos D and Waters K (1990) Physically Based Facial Modeling, Analysis, and Animation, Visualization and Computer Animation, Vol. 1, No. 2, pp. 73-80.
13. Kalra P, Mangili A, Magnenat-Thalmann N and Thalmann D (1991) SMILE : A Multilayered Facial Animation System, Proc IFIP WG 5.10, Tokyo, Japan (Ed Kunii Toshiyasu L) pp. 189-198.
14. Sederberg TW and Parry SR (1986), Free Form Deformation of Solid Geometric Models, Proc. SIGGRAPH '86, pp. 151-160.
15. LeBlanc A, Kalra P, Magnenat-Thalmann N and Thalmann D (1991), Sculpting with the "Ball & Mouse" Metaphor, Proc. Graphics Interface '91, Calgary, Canada,
16. James D. Foley, Andries van Dam, Steven Feiner, and John Hughes. *Computer Graphics Principles and Practice*. Addison-Wesley, 1990.
17. J.Vince. *3-D computer animation*. AddisonWesley, 1992. [Loop 94] Charles Loop. Smooth Spline Surfaces over Irregular Meshes. *Computer Graphics (Proc. SIGGRAPH 94)*, 28:303–309, July 1994.
18. M.Desbrun, N. Tsingos, and M.P.Gascuel. *Adaptive sampling of implicit surfaces for interactive modeling and animation*. pages 171–186, 1995.

## Reconocimiento de Formas en Imágenes Médicas

Adela González      Roberto Guerrero      Alfredo Zavala.

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional<sup>1</sup>

Departamento de informática

Universidad Nacional de San Luis

Av. Ejército de los Andes 950 Local 106

5700- San Luis

Argentina

e-mail: {adelag, rag, azavala}@unsl.edu.ar

### Resumen

Las imágenes son potenciales herramientas para la toma de decisiones dado que permiten extraer información a partir de ellas. Las áreas del procesamiento de imágenes y visión tratan el análisis y manipulación de imágenes en 2D para la extracción de información que permita facilitar la toma de decisiones en áreas específicas ampliando la información originalmente proporcionada. La propuesta del presente grupo pretende a través del procesamiento de un conjunto de imágenes la identificación automática de objetos particulares y su posterior presentación en 2D o 3D.

### Introducción

El procesamiento de imágenes en conjunción con la visión por computadora [1, 2, 3] provee varias técnicas que permiten obtener una representación del mundo a partir del análisis de imágenes obtenidas desde cámaras de video.

El amplio espectro de aplicaciones cubierto por la visión por computadora, se debe a que permite extraer y analizar información espectral, espacial y temporal de distintos objetos. La información espectral incluye frecuencia (color) e intensidad (tonos de gris). La información espacial se refiere a aspectos como forma y posición (una, dos y tres dimensiones). La información temporal comprende aspectos estacionarios (presencia y/o ausencia) y dependientes del tiempo (eventos, movimientos, procesos).

Según el tipo de aplicación, serán el tipo de imagen que será necesario adquirir (imágenes de rayos X, infrarrojo, etc.) y el análisis que se aplicará.

Entre las aplicaciones en las que la Visión por Computadora ofrece importantes aportes se destacan las siguientes:

- En el ramo industrial se aplica para la inspección automatizada; permite identificar que las piezas no tengan defectos.
- En la geología permite estudiar las modificaciones de los causes de los ríos; también sirve para detectar bancos de pesca o variaciones en las corrientes marinas por cambios de clima.

---

<sup>1</sup> El laboratorio es dirigido por el Dr. Raúl Gallard y subvencionado por la Universidad Nacional de San Luis y la ANPCyT (Agencia Nacional para la Promoción de la Ciencia y la Tecnología).

- En la medicina facilita el diagnóstico automático de las enfermedades del corazón, a partir de fotogramas del movimiento del mismo y de un análisis de las deformaciones que se producen, etc..
- En el ámbito militar se desarrollan complejas armas como misiles que se guían por sí solos por zonas previamente grabadas; en la identificación de aeropuertos, barcos, tanques o cualquier imagen tomada desde aviones o satélites.

Entre las operaciones básicas con las que se construyen diversas aplicaciones específicas de Visión por Computadora, podemos destacar las siguientes:

- Identificación de objetos (por ejemplo identificar personas a partir de su fotografía).
- Determinación de la posición en que se halla un objeto.
- Establecimiento de relaciones espaciales entre varios objetos. (Para guiar a un robot en el ensamblaje de piezas).
- Determinación de las coordenadas de puntos importantes de un objeto.
- Realización de mediciones angulares.
- Mediciones tridimensionales (por ejemplo para ayudar a un robot a librar obstáculos)

Existe numerosa bibliografía relacionada con el tratamiento de problemas específicos en diferentes áreas de aplicación [4, 8]. Todas ellas comienzan con la identificación del o los objetos a ser tratados (en su mayoría a través de técnicas de segmentación); las posteriores operaciones básicas (y su procesamiento asociado) dependerán de la característica del problema bajo análisis. Si bien dichas operaciones son básicas (de ahí su nombre) en algunos casos será necesaria la presentación de la información mediante técnicas especializadas de visualización tales como coloreado 2D, reconstrucción 3D, etc. [11, 15].

### **Campo de Aplicación Específico**

El estudio de la representación tridimensional (3D) en medicina se ha expandido considerablemente en los últimos años[10, 12, 13, 14]. En el campo de la clínica la visualización en 3D puede ser de mucha utilidad en numerosos áreas (cirugía, ortopedia, etc.). Partiendo de estudios de imagenología (ultrasonido, tomografía axial computarizada, etc.) se extrae información anatómica o funcional de algún órgano de interés.

En la actualidad el poder de la tecnología permite obtener muy buenos resultados en el dominio de la representación 3D. El área de la medicina, por ejemplo, necesita de dicha representación para la evaluación de las imágenes y la posterior emisión de un diagnóstico.

Se pueden definir los diversos pasos que intervienen en la reconstrucción tridimensional de las imágenes médicas, desde su obtención en el equipo correspondiente hasta su despliegue en una computadora.

1. Adquisición y Registro: Normalmente, se obtienen las imágenes en dos dimensiones directamente del equipo. En caso de imágenes multimodales, éstas deberán alinearse.

3. Segmentación: La segmentación consiste en extraer objetos de interés a partir de las imágenes en tonos de gris. La segmentación proporciona información superficial al generar una imagen binaria de la estructura de interés o proporciona una imagen con información hacia adentro del volumen.

Este tema ha sido ampliamente investigado, pero no se ha llegado a obtener un método universal que clasifique automáticamente (aproximadamente) cualquier estructura anatómica de interés, a partir de cualquier modalidad de imagen, de ahí que se continúe esta investigación según la aplicación [5, 6, 7, 16].

4. Interpolación: Los contornos obtenidos en la etapa precedente deben ser alineados en pila y en caso de que la resolución sea insuficiente, se requiere aplicar un algoritmo de interpolación entre datos para todas las dimensiones deseadas. Dentro de los métodos de interpolación existen aquellos que resaltan características de calidad o de tiempo de ejecución. Es necesario realizar un estudio para determinar el mejor método de interpolación que se adapte a la aplicación que en particular se desee desarrollar.

5. Presentación de Resultados en 3D: La etapa final de la reconstrucción es quizás la más importante. Consiste en desplegar el volumen del órgano en estudio o de una de sus estructuras en la pantalla de la computadora.

Toda representación en 3D involucra transformaciones de movimiento, eliminación de áreas ocultas, asignación de sombreado o transparencias, etc.. Actualmente existen paquetes comerciales que realizan funciones de despliegue de estructuras en 3D de una manera automática, principalmente para aplicaciones de imágenes sintéticas. Sin embargo, por ejemplo, no cualquier método de sombreado entrega un resultado acorde con lo que se vería en la realidad. Nuevamente, es necesaria una evaluación de los métodos empleados en esta etapa, de acuerdo a criterios cualitativos y cuantitativos del especialista.

## **Propuesta en Desarrollo**

En la presente propuesta presentamos el análisis y evaluación de imágenes provenientes de estudios médicos mediante diversas técnicas, en ellas se pretenderá en una primera etapa lograr el reconocimiento de formas específicas (calcificaciones, tumores, órganos, etc.) a determinar dependiendo del caso en particular. Ésta inicialmente se realizará mediante la implementación de técnicas ya existentes que serán confrontadas mediante una evaluación subjetiva de sus resultados; las cuales servirán a posteriori como información para la elaboración de una técnica de aprendizaje que permita el reconocimiento en forma automática.

Una segunda etapa consistirá de la inferencia de información estructural de la forma detectada, mediante la interpolación de diferentes vistas de la misma para, a posteriori, la representación de los datos estructurales mediante alguna técnica de visualización.

## **Bibliografía**

- [1] Parker J. R., [1997], *Algorithms for Image Processing and Computer Vision*, John Wiley and Sons.
- [2] Gonzalez R. and Woods R., *Digital Image Processing*, Addison - Wesley, Reading Mass, 1997.
- [3] Galbiati, *Machine Vision & Digital Image Processing*, Phipe, 1997
- [4] Levine M., *Vision in Man and Machine*, Mc Graw-Hill, New York, 1985
- [5] Forsyth D., *Shape from Texture and Integrability*. International Conference on Computer Vision, Vancouver, Canada, 2001.
- [6] Felzenszwalb P. and Huttenlocher D., *Image Segmentation using Local Variation*. Proc. IEEE Conference Computer Vision and Pattern Recognition, 98-104, 1998.

- [7] Jain A. and Dubes R., [1998], *Algorithms for Clustering Data*, Prentice Hall.
- [8] Shi J. and Malik J., *Normalized Cuts and Image Segmentation*. Proc. IEEE Conference Computer Vision and Pattern Recognition, 731-737, 1997.
- [9] Wu Z. and Leahi R., *An Optimal graph-theoretic-approach to Data Clustering: Theory and its applications to image segmentation*. PAMI, vol 11, 1101-1113, nov, 1993.
- [10] Mori J. and Malik J., *Estimating human body configurations using shape context matching*. Workshop on Models versus Exemplars in Computer Vision (CVPR 2001), 2001.
- [11] Foley J. D., Van Dam A., Feiner S. K., And Hughes J. H. [1990], *Computer Graphics Principles and Practice, 2nd Edition*, Addison-Wesley.
- [12] Bullitt E, Aylward S, "*Fast, high-quality, volume rendering of vascular anatomy via segmentation*", IEEE-TMI Abril, 2001, pp. 218-228.
- [13] Aylward S, Coggins J, Cizadlo T, Andreasen N, "*Spatially Invariant Classification of Tissues in MR Images*", Proceedings of Visualization in Biomedical Computing Conference, Rochester, MN, Octubre 1994. pp. 352-361.
- [14] Aylward S, Hemminger, B., Pisano, D., "*Mixture Modeling for Digital Mammogram Display and Analysis*", International Workshop on Digital Mammography, Junio 1998. pp. 305-312.
- [15] Bullitt E, Aylward S, "*Volume rendering of segmented tubular objects*", Medical Image Computing and Computer-Assisted Intervention, vol. 2208 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 2001. pp 161-168.
- [16] Felzenszwalb P. and Huttenlochel D, *Efficiently Computing a Good Segmentation*, DARPA Image Understanding Workshop, 1998.

# Un Ambiente Gráfico para Testear Estrategias Evolutivas en el Problema Inverso de las IFS

María Laura Ivanissevich<sup>1</sup>, Paula Millado<sup>1</sup>, Luis Sierpe<sup>1</sup> y Claudio Delrieux<sup>2</sup>

<sup>1</sup>*Univ. Nacional de la Patagonia Austral, L. de la Torre 1070, (9300) Río Gallegos,*

<sup>2</sup>*Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.*

*Voice: (54)(291)4595101 ext. 3381 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org*

**Palabras Clave:** SISTEMAS DE FUNCIONES ITERADAS (IFS), PROGRAMACIÓN EVOLUTIVA, PROGRAMACIÓN GENÉTICA.

## 1 Introducción

Los sistemas de funciones iteradas (IFS) constituyen una manera de representar imágenes por medio de conjuntos fractales. La factibilidad de representar imágenes con IFS se fundamenta en el Teorema del Collage, según el cual una imagen cualquiera puede ser arbitrariamente aproximada por un atractor fractal adecuadamente representado por un IFS. Esta propiedad hace tentadora la idea de buscar un método de compresión de imágenes que encuentre el collage adecuado para cualquier imagen de entrada. Esto constituye el problema inverso del IFS. Sin embargo, la búsqueda exhaustiva de un IFS para aproximar una imagen dada está fuera de las posibilidades de los sistemas de cómputo actuales. Un IFS como el del helecho consta de 24 parámetros, por lo que encontrar sus respectivos valores por “fuerza bruta” requeriría un tiempo de cómputo imposible de cuantificar.

Las estrategias hasta ahora propuestas para solucionar esta dificultad son dos. Por un lado, es posible restringir el espacio de búsqueda a un subconjunto de las transformaciones afines, como se propone en la compresión fractal en bloques (CFB) [4, 6]. En la CFB se utiliza un conjunto relativamente grande de mapas entre segmentos de la imagen, utilizando escalas fijas, y rotaciones cuantizadas a cuartos de circunferencia. Por lo tanto, el espacio de búsqueda es relativamente pequeño para cada transformación. Los resultados obtenidos con la CFB poseen una buena relación de compromiso entre tiempo de cómputo, compresión resultante, y calidad final, aunque están evidentemente lejos de las posibilidades teóricas [5].

Por otro lado, es posible utilizar un mecanismo de búsqueda adaptativo, que tenga cierta inteligencia como para guiarse dentro del un subconjunto del espacio de búsqueda en el cual los resultados se acercan a la solución buscada. Este es el tipo de estrategia buscada con la programación evolutiva (PE) [8] y con la programación genética (PG) [7]. El uso de PE y PG para la solución del problema inverso ha sido mucho menos estudiado, pese a que permitiría obtener mejores compresiones, tal vez con mayores tiempos de cómputo, que la CFB. Al mismo tiempo, el problema inverso representa un caso interesante de estudio

para comparar las diversas estrategias evolutivas y genéticas, combinarlas, y eventualmente proponer algunas nuevas. Este es el objetivo de esta línea de investigación. Para obtener estos resultados se implementó un ambiente gráfico que permite testear y comparar los resultados del uso de dichas estrategias, y evaluar numérica y visualmente los resultados que se van produciendo a lo largo de las generaciones.

## 2 Sistemas de funciones iteradas

Un sistema de funciones iteradas (IFS) consiste en una colección de transformaciones afines contractivas que mapea al plano  $\mathbb{R}^2$  sobre sí mismo. Esta colección de transformaciones define un mapa

$$W(\cdot) = \bigcup_{i=1}^n w_i(\cdot).$$

Las transformaciones afines contractivas tienen la importante característica que cuando son aplicadas repetidamente, convergen a un punto fijo. El mapa  $W$  no es aplicado a todo el plano sino a un subconjunto, es decir, a una colección de puntos del plano. Dado un conjunto inicial  $S$ , podemos calcular  $w_i(S)$  para cada  $i$ , tomar la unión de estos conjuntos y obtener un nuevo conjunto  $W(S)$ .  $W$  es un mapa en el espacio de los subconjuntos del plano. Según el teorema de Hutchinson [2], cuando las transformaciones  $w_i$  son contractivas en el plano, entonces  $W$  es contractiva en el espacio (cerrado y acotado) de subconjuntos del plano. Por lo tanto, la iteración de un IFS converge a un único subconjunto del plano, el cual es un punto fijo del mapa  $W$ , denominado *atractor*  $A$  del IFS. Para la obtención del código IFS de una figura  $I$  arbitraria, se parte de una representación inicial aproximada de la silueta  $S$  del objeto que se desea modelar. La idea del método es buscar un grupo de mosaicos —copias transformadas de  $S$ — que en conjunto cubran la silueta original (Teorema del Collage [3]). Si por  $d(I, J)$  denotamos la distancia de Hausdorff entre subconjuntos  $I$  y  $J$  del plano en una métrica adecuada, y  $r$  es la contractividad global de  $W$ , entonces el teorema del collage permite afirmar la desigualdad

$$d(A, I) \leq \frac{1}{1-r} d(I, \bigcup w_i(I)),$$

lo cual significa que la distancia entre la figura  $I$  y el atractor  $A$  del IFS está acotada por la distancia entre  $I$  y el cubrimiento determinado por  $W$ . A mejores cubrimientos, mejor aproximación a la imagen final. Como ya mencionáramos, este cubrimiento hasta ahora no ha podido implementarse en forma automática, debiéndose trabajar con asistencia humana. A partir de la silueta  $S$ , son introducidas una a una copias reducidas de  $S$ , de modo de poder ir cubriéndola con sus copias transformadas. Cada copia podrá ser torcida, escalada, rotada o trasladada según convenga para lograr el propósito buscado. Es importante que el tamaño de cada  $w_n(S)$  sea menor que el de  $S$  para asegurar que cada transformación sea contractiva.

En otras palabras, para lograr la codificación se deben determinar un conjunto de transformaciones afines contractivas  $\{w_1, w_2, \dots, w_n\}$  con la siguiente propiedad: La silueta original  $S$  y el conjunto

$$\tilde{S} = \bigcup_{n=1}^N w_n(S)$$

deben ser visualmente cercanos, siendo el número de transformaciones el menor posible.  $w_n(S)$  es llamado el  $n$ -ésimo mosaico del collage. En la Figura 1 se muestra un atrac-

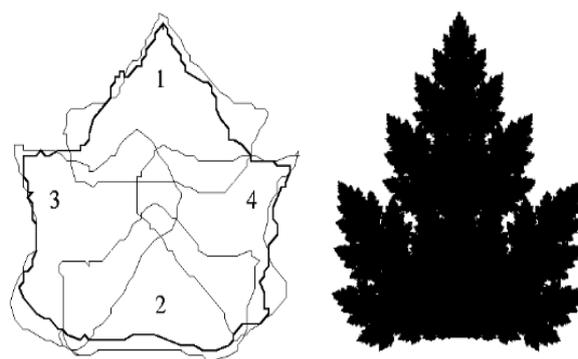


Figura 1: Cubrimiento de una hoja para la obtención de los códigos IFS y atractor que se logra con ellos.

tor obtenido utilizando el procedimiento descrito. Un algoritmo utilizado para generar el atractor o imagen a partir de las transformaciones, comienza con el código IFS  $\{w_n, p_n : n = 1, 2, \dots, N\}$  junto con la ventana de graficación  $V$  de resolución  $L \times M$ . El algoritmo, denominado “juego del caos” [3], realiza una recorrida aleatoria del atractor del siguiente modo: parte de un punto inicial, escoge al azar un mapa  $w_n$ , y transforma el punto mediante este mapa, obteniendo las coordenadas de un pixel. Este pixel es graficado, y su punto correspondiente es luego sometido al mismo proceso, y así una cantidad predefinida de iteraciones.

### 3 Estrategias evolutivas y genéticas

La programación evolutiva (PE) y la programación genética (PG) son técnicas bien conocidas que surgen de imitar lo que conocemos de la evolución natural. La PE y la PG se utilizan, entre otros contextos, para resolver problemas que con los métodos de Inteligencia Artificial de alto nivel tradicionales (búsqueda heurística, lógica, etc.) son o bien intratables o bien imprácticos.

Las ideas principales provienen de una metáfora de la evolución natural, en la cual existen individuos (fenotipos) que expresan una información genética (genotipo), y además están sujetos a la presión evolutiva del medio (fitness). Los individuos más exitosos (con mejor fitness) ven aumentadas sus posibilidades reproductivas, generando nuevos individuos con su mismo genotipo. Sin embargo, en el proceso reproductivo ocurre el fenómeno distintivo de la evolución natural: la aparición de mutaciones al azar.

Este mismo esquema dinámico, con un sinnúmero de variaciones, ha sido desarrollado para diversas aplicaciones. En este trabajo veremos cómo el problema inverso de la IFS puede resolverse por medio de estrategias evolutivas y genéticas. Supongamos la existencia de una población con  $\mu$  ancestros, cuyo genotipo  $\bar{g}_i$  se encuentra codificado con números. Entonces se crea un conjunto  $\lambda$  de descendientes siguiendo el esquema

$$\bar{p}_i = (\bar{g}_i, \bar{\sigma}_i) \xrightarrow{\text{mut}} (\bar{g}_i + n_0(\bar{\sigma}_i), \alpha(\bar{\sigma}_i)),$$

donde  $n_0(\bar{\sigma}_i)$  denota la elección de un número de una aleatoria normal con media cero y variancia  $\sigma_i$ .  $\alpha$  define una función de adaptación, la cual puede ser la distancia de Hamming entre el atractor  $A$  del IFS resultante y la imagen  $I$  de entrada.

En nuestro caso,  $\bar{g}_i$  representa los parámetros de la IFS  $W$  que estamos tratando de encontrar. Cada mapa  $w_k$  requiere seis reales para su objetivo:

$$w_k(x, y) = \begin{pmatrix} a_k & b_k \\ c_k & d_k \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_k \\ f_k \end{pmatrix} \quad (1)$$

$$= \begin{pmatrix} r_{1k} \cos \theta_{1k} & -r_{2k} \sin \theta_{1k} \\ r_{1k} \sin \theta_{2k} & r_{2k} \cos \theta_{2k} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_k \\ f_k \end{pmatrix} \quad (2)$$

En general, la expresión 2 es preferible a la 1 por tener un significado geométrico más intuitivo. Por lo tanto,  $\bar{g}_i$  representará las  $n$  sextuplas  $(r_{1k}, r_{2k}, \theta_{1k}, \theta_{2k}, e_k, f_k)$ ,  $k \in [1..n]$  en un determinado individuo de una generación. Como estrategia híbrida se puede introducir, además de la mutación, el *crossover* o cruzamiento entre el genotipo de dos ancestros para obtener el genotipo del sucesor.

Para evitar los mínimos locales, y quedar atrapado en un “nicho ecológico”, la estrategia mantiene una diversidad genotípica seleccionando un grupo numeroso de individuos con la mejor adaptación. Entre estas técnicas de *niching* podemos contar el *fitness sharing* (escalar el fitness en un entorno de cada individuo), el *crowding* (reemplazar al padre más cercano del individuo recientemente creado), y el *clustering* o agrupamiento, la cual resulta ser la más eficiente en la práctica.

## 4 Implementación de un ambiente para testear estrategias

Una vez definido el propósito y la metodología de trabajo, se implementó un prototipo en el lenguaje Matlab para testear el funcionamiento de los algoritmos y la eficacia de las estrategias. Luego, para contar con una implementación más adecuada en velocidad y capacidad de cómputo, se reescribió el programa en el lenguaje visual Delphi, construyéndose un ambiente gráfico para testear y evaluar las diferentes estrategias. En la Fig. 2 es posible observar la interfase gráfica del sistema. Se puede observar abajo a la izquierda el atractor resultante, y abajo a la derecha, la curva que muestra la evolución del *fitness* del atractor a lo largo de las generaciones. Los controles de la parte superior izquierda permiten elegir un conjunto de imágenes de entrada, para testear el correcto funcionamiento en varios casos, mientras que los controles a la derecha permiten controlar la ejecución del sistema, modificar los parámetros, las diversas estrategias, etc. La implementación en una herramienta como Delphi permitió migrar con cierto trabajo el prototipo original, pero con el beneficio de la velocidad de cómputo, la flexibilidad del uso de las componentes visuales, y otras posibilidades importantes. En la Fig. 2 se pueden ver los parámetros de la simulación realizada. Se realizaron 2000 generaciones, a 3000 órbitas por generación, con 3 transformaciones por genotipo, 10 antecesores y 5 sucesores. El tiempo total en un equipo PC standard fue de menos de dos minutos.

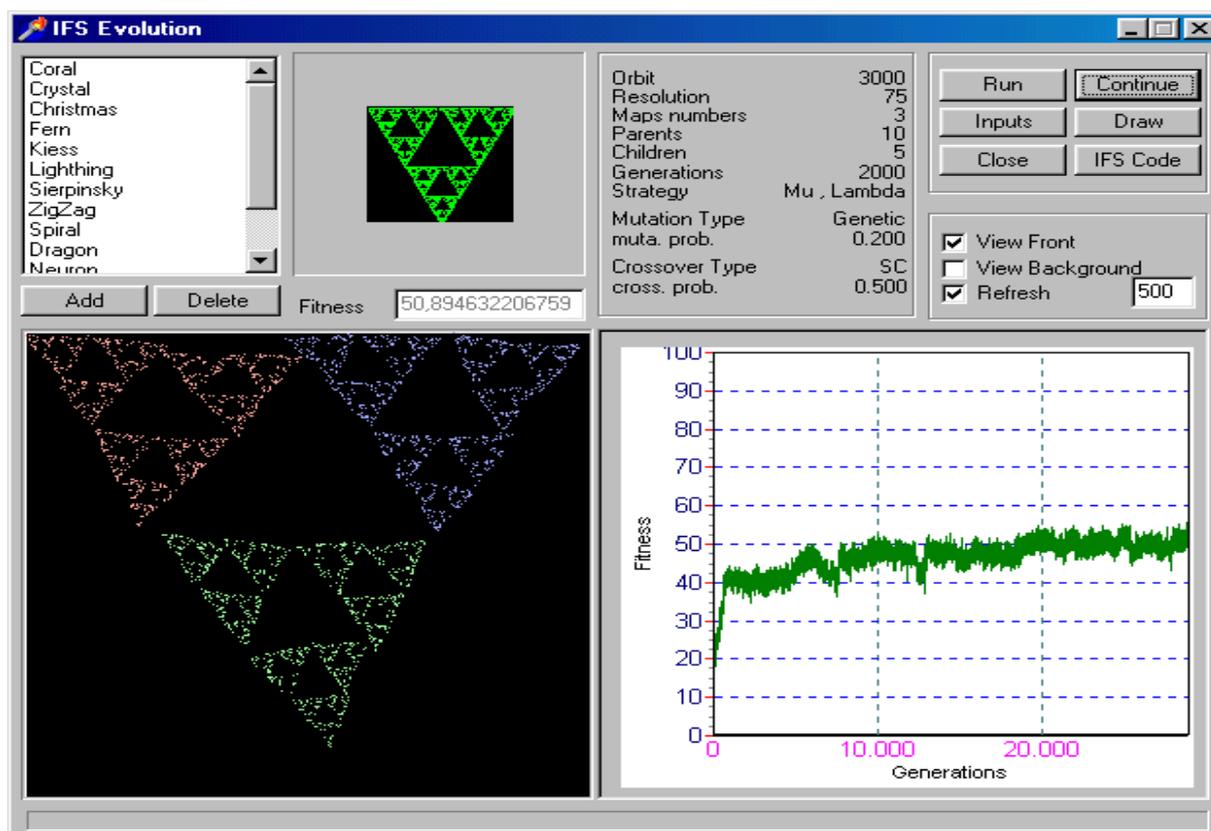


Figura 2: La interfase del sistema, mostrando la convergencia evolutiva de un conjunto de transformaciones al atractor de la imagen inicial.

## Referencias

- [1] M. Barnsley y S. Demko. Iterated function systems and the global construction of fractals. *Proc. Roy. Soc. London*, A399:243–275, 1985.
- [2] M. Barnsley, A. Jacquin, y F. Malassenet. Harnessing Chaos for Image Synthesis. *ACM Computer Graphics*, 22(3):131–14, 1988.
- [3] M. F. Barnsley. *Fractals Everywhere*. Academic Press, Boston, 1988.
- [4] Yuval Fisher, editor. *Fractal Image Compression: Theory and Application*. Springer-Verlag, New York, NY, USA, 1995.
- [5] Yuval Fisher, editor. *Fractal Image Encoding and Analysis*, Springer-Verlag, New York, NY, USA, 1997.
- [6] Arnaud E. Jacquin. Image Coding Based on a Fractal Theory of Iterated Contractive Image Transformations. *SPIE Proceedings*, Vol. 1360, págs. 227–239, Lausanne, 1990.
- [7] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, segunda edición, 1994.
- [8] I. Rechenberg. *Evolution strategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, 1973.

# Sistemas Híbridos para el Reconocimiento de Patrones

Román Katz y Claudio Delrieux

*Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.  
Voice: (54)(291)5710098 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org*

**Palabras Clave:** Procesamiento de Imágenes, Reconocimiento de Patrones, Inteligencia Artificial, Programación Lógica, Visión Robótica.

## 1 Introducción

El Procesamiento Digital de Imágenes (PDI) es un área de permanente importancia tecnológica. Sus objetivos van desde procesar datos adquiridos satelitalmente para mejorar la percepción, detección o interpretación de algún patrón específico [5, 6], o aplicar filtros a imágenes fotográficas para reconstruir o retocar sus características visuales [3, 4], o comprimir información gráfica para facilitar su transporte por las redes de comunicaciones [7, 2]. Entre las áreas de mayor importancia tecnológica del PDI podemos mencionar el reconocimiento de patrones y la visión robótica, en las cuales se requiere el uso combinado de técnicas de todo tipo, desde las puramente numéricas hasta las más cercanas a la inteligencia artificial. Actualmente la mayor parte de las técnicas avanzadas utilizadas en el reconocimiento de patrones se basa en un procesamiento numérico, donde mediante un tratamiento heurístico se busca conjugar los resultados de ciertos estimadores con valores específicos de una distribución [8, 10].

Las técnicas asociadas están adaptadas para cada problema en particular con lo que se logran excelentes rendimientos en tareas específicas, pero su posibilidad de aplicación satisfactoria en otras áreas es muy reducida. En tal sentido el procesamiento simbólico, en donde se busca reconocer la presencia de determinados patrones para disparar luego un proceso de interpretación, se presenta como una alternativa posible. Es indudable que ambos tipos de procesamiento son indispensables en aplicaciones tecnológicas como las mencionadas. En un sistema de visión robótica, por ejemplo, la mayor parte del procesamiento debe realizarse en forma numérica para acomodar las imágenes de entrada de manera de poder facilitar la ejecución de los algoritmos de alto nivel que puedan requerirse (reconocimiento de patrones, identificación, etc.). Sin embargo, no tiene sentido práctico implementar estos últimos desde el bajo nivel, así como tampoco parece adecuado implementar los algoritmos fuertemente numéricos de bajo nivel en un lenguaje de alto nivel.

Por lo tanto, la solución tecnológica parece ser la de construir una arquitectura de software que integre simultáneamente en una misma aplicación el procesamiento numérico de bajo nivel en una plataforma adecuada (C++, Delphi), con la implementación de algoritmos de inteligencia artificial implementados en los lenguajes específicamente desarrollados (Lisp,

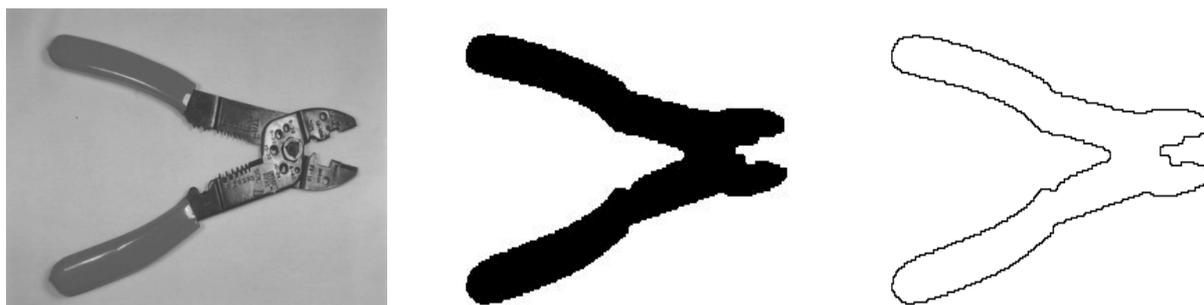


Figura 1: (a) Imagen en escala de grises, (b) umbralizada, y (c) segmentada.

PROLOG). Es este último el enfoque metodológico adoptado para este trabajo, en el cual presentamos una aplicación implementada en Delphi, donde se conjugan las técnicas de procesamiento numérico para el procesamiento de bajo nivel de las imágenes, con las de alto nivel implementadas en el lenguaje PROLOG. El objetivo en esta etapa de nuestro desarrollo es implementar un sistema de reconocimiento de objetos en imágenes digitales.

## 2 Procesamiento numérico

Los primeros pasos del tratamiento que recibe una imagen luego de su adquisición se compendian bajo la denominación de preprocesado y son tendientes a producir una mejora en alguna de sus características, de forma de incrementar el rendimiento de las etapas posteriores [5]. En general son operaciones que requieren algoritmos simples pero de gran insumo de operaciones numéricas en su ejecución, entre los que se destacan conversión a escala de grises, filtrado, umbralización, escalamiento, extracción de regiones de interés, etc. [10]. Un paso posterior a la mejora de la imagen es el de la segmentación. Aquí se busca simplificar la imagen reduciéndola a un subconjunto predeterminado de objetos básicos (puntos, segmentos u otras primitivas geométricas sencillas), con el objeto de facilitar luego un procesamiento de un nivel superior.

En la etapa siguiente, la descripción de la imagen, se genera una representación útil para el problema de interpretación de interés a partir de la información proveniente de las operaciones de filtrado y segmentación previas. Existe una gran variedad de descriptores [5] entre los que se pueden encontrar los relacionados con características de forma, de Fourier, momentos estadísticos, códigos de cadena, segmentos del contorno, textura, etc. Bajo estas condiciones una alternativa razonable para nuestro problema en particular es utilizar los segmentos del contorno como descriptor, considerando los determinados por puntos regularmente espaciados y obteniendo de esta forma una representación del perímetro del objeto con una cantidad de información simplificada. Como se observa en la Fig. 2, aún reduciendo la cantidad de vértices para describir el contorno a porcentajes muy bajos (hasta un 25% o menos de los vértices originalmente segmentados) el objeto se presenta altamente distinguible.

## 3 Reconocimiento

El paso final del proceso de reconocimiento de imágenes es el de la interpretación. En el mismo se busca descubrir, identificar y comprender la información contenida en los descriptores.

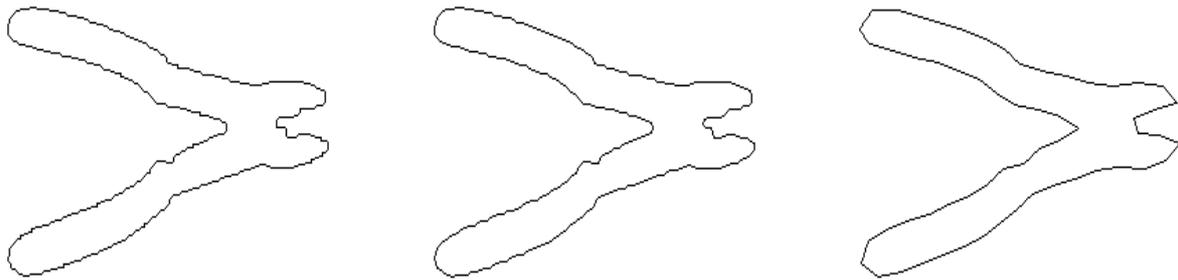


Figura 2: Contorno descrito con diferentes proporciones de los vértices originalmente segmentados: (a) 50%, (b) 25% y (c) 5%.

Se desea además que este grado de inteligencia, representado por una base de conocimientos apropiada, tenga un alto nivel de independencia con respecto a la aplicación, de manera tal que esta experiencia pueda servir en un futuro para otro tipo de reconocimientos. Expresar nuestro conocimiento mediante la lógica de predicados, específicamente utilizando el lenguaje PROLOG [9], nos permitirá verificar la condición de independencia buscada. Su naturaleza declarativa facilitará notablemente la tarea de expresar tanto formas geométricas complejas [1] como relaciones estructurales y posicionales entre las mismas. La representación de conocimiento y de los patrones de inferencia utilizando lógica es lo suficientemente modular como para poder testear y modificar las diferentes partes del sistema independientemente unas de otras, lo cual también garantiza su portabilidad y reuso en futuras aplicaciones. El uso de PROLOG facilita además el desarrollo del sistema de razonamiento o “motor” de inferencia, tanto para implementar los patrones deductivos como para realizar otro tipo de inferencias ampliativas en general (abducción, razonamiento por *default*, razonamiento por intervalos de confianza, etc.).

La formulación lógica de los patrones a reconocer se expresa mediante un *modelo de detección* genérico. Dicho modelo es una estructuración de reglas que reduce el costo de explicitación en predicado de nuevos patrones. De esta forma posibilita insertar el predicado que define a una estructura geométrica dada en una “plantilla” conocida, impidiendo que se repitan resultados y que se ejecuten *backtracking* innecesarios. Dicha plantilla tiene una estructura esquemática como se observa en la Tabla 1. Manteniendo esta estructura, sólo se deben elaborar las “relaciones geométricas” del patrón, es decir la formulación en forma de predicado del elemento particular a detectar.

---

```

patron geometrico (patron):-
    lista segmentos que verifican patron
    cut[!] (para que la lista sea única)
    patrón está en lista

lista segmentos que verifican patron:-
    relaciones geométricas
    verifica que no sea repetido
    agrega a lista

```

---

Tabla 1: Modelo de Detección.

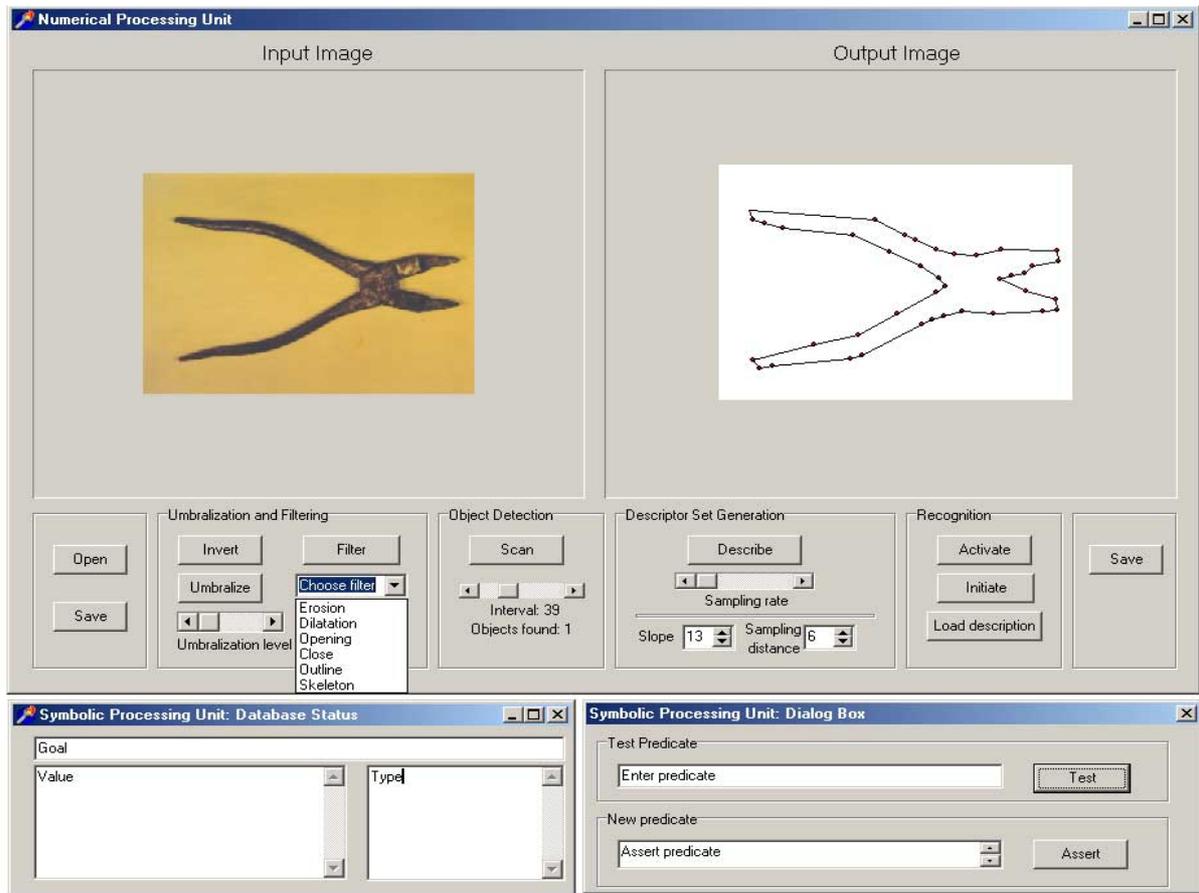


Figura 3: Interfase gráfica del sistema de reconocimiento.

La implementación de estos conceptos se realizó mediante el desarrollo de un sistema para la identificación de patrones mediante su contorno o su esqueleto. En el sistema se conjuga el procesamiento de imágenes de forma numérica con la utilización de un mecanismo de inferencia adecuado para el reconocimiento. El procesamiento numérico genera una representación adecuada de la imagen mientras que el “motor” de inferencia busca cotejar la información contenida en dicha representación con una base de conocimientos estipulada. Se observa en la Fig. 3 la interfase gráfica del sistema, con los comandos asociados a las etapas antes descritas contenidos en la ventana de **Procesamiento Numérico** y con una imagen de entrada arbitraria, junto con la de su descripción final asociada. En el proceso de reconocimiento, el mecanismo de inferencia se obtiene mediante el manejo de una componente de software que contiene una librería DLL del lenguaje PROLOG. La librería de PROLOG requiere una inicialización apropiada, luego de la cual el sistema queda preparado para recibir consultas. En nuestro caso se realiza, junto con la inicialización, la carga de una base de conocimientos estipulada a través de archivos. La misma contiene la formulación lógica de las relaciones geométricas que conforman el patrón que se desea detectar y los predicados apropiados para generar una base de datos a partir de los segmentos que provienen de la descripción del objeto. Las consultas se realizan a través de la ventana **Consola de consultas**, en la que también se pueden generar nuevos predicados para ser testeados y posiblemente agregados a la base de conocimiento. La ventana **Consola base de conocimientos** muestra los resultados obtenidos a partir de las consultas, incluyendo los

tipos y valores finales de las variables involucradas en la misma. Ambas consolas se pueden ver en la Fig. 3.

## 4 Conclusiones y trabajo futuro

Se presentó un sistema integrado de reconocimiento de patrones que utiliza simultáneamente algoritmos básicos de tratamiento numérico de imágenes y técnicas de reconocimiento simbólico en lenguaje PROLOG. La complementación de procesamiento de bajo y alto nivel en una plataforma de programación común rescata las capacidades salientes de ambos enfoques. Nos permite conjugar la excelente respuesta en velocidad de cálculo, la disponibilidad de librerías gráficas y la versatilidad de implementación de los sistemas en entornos *RAD (Rapid Application Development)* con las facilidades de interpretación del lenguaje PROLOG. La posibilidad de inferir resultados a partir de información incompleta, de “explicitar” el conocimiento de una forma simple, natural y altamente portable y reciclable y la robustez frente a variaciones posicionales y de orientación hacen que dicho mecanismo de identificación resulte ideal para el problema de detección patrones en multiles aplicaciones. El sistema mostrado permite la adaptación de diversas técnicas numéricas de procesamiento adicionales y la ampliación y reemplazo de la base de conocimientos para problemas de detección particulares.

Entre los trabajos futuros se considera la posibilidad de realizar una extensión del sistema para contemplar esquemas de descripción alternativos para los objetos. Se contempla también la incorporación de *meta-interpretés* en el proceso de reconocimiento, realizando un seguimiento en el mecanismo de inferencia de forma de obtener información particular del objeto y “niveles” lógicos que cuantifiquen el éxito obtenido en la tarea. Esto permitiría, entre otras cosas, darle un manejo operacional a las situaciones de reconocimiento parcial, de manera de poder elaborar razonamientos tentativos, explorarlos modificando el procesamiento numérico, y proveyendo respuestas basadas en intervalos de confianza.

## Referencias

- [1] B. Batchelor. *Intelligent Image Processing in Prolog*. Springer-Verlag, London, 1991.
- [2] K. Castleman. *Digital Image Processing*. Prentice-Hall, New York, 1989.
- [3] A. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufman, 1995.
- [4] J. Gomes y L. Velho. *Image Processing for Computer Graphics*. Springer, 1997.
- [5] R. González y R. Woods. *Digital Image Processing*. Addison-Wesley, 1996.
- [6] Anil Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Cambridge, 1996.
- [7] Jae Lin. *2D Signal and Image Processing*. Prentice-Hall, Cambridge, 1991.
- [8] Arie Roszenfeld. *Digital Picture Processing*. MIT Press, Cambridge, MA, 1995.
- [9] L. Sterling y E. Shapiro. *The Art of Prolog*. MIT Press, 1994.
- [10] S. Umbaugh. *Computer Vision and Image Processing*. Prentice-Hall, 1998.

# Visualización de Datos Multivariados

Lidia Marina López  
Dpto Informática y Estadística  
Universidad Nacional del Comahue  
llopez@uncoma.edu.ar  
Tel 0299-4490312 Int 435 - Fax 0299-4490313

La visualización de datos multivariados implica la representación de los datos en un entorno de dimensiones mucho menores tal como una pantalla de computadora u otro medio de impresión que permita al experto inferir conclusiones. Cuando la base de datos es de un tamaño considerable, poder contar con una representación visual habilita para descubrir relaciones implícitas.

Los datos que se pretenderán visualizar corresponden al proyecto de investigación de la Facultad de Humanidades de la Universidad Nacional del Comahue: *Situación socio-profesional de género en la planta docente y órganos de gobierno de la Universidad Nacional del Comahue* y están formados por los datos personales y profesionales de dichos miembros de la Universidad.

Las técnicas de minería de datos permiten predecir situaciones de acuerdo a la distribución de la información, pero si se pudiera obtener una estructura visual, los miembros del grupo de investigación mencionado, contarían con una nueva representación que, según sus opiniones, darían un valor agregado importante.

En este marco, una línea del proyecto se está dedicando a probar las distintas técnicas de transformación de datos crudos a estructuras visuales generando la visualización propiamente dicha.

## 1 Presentación

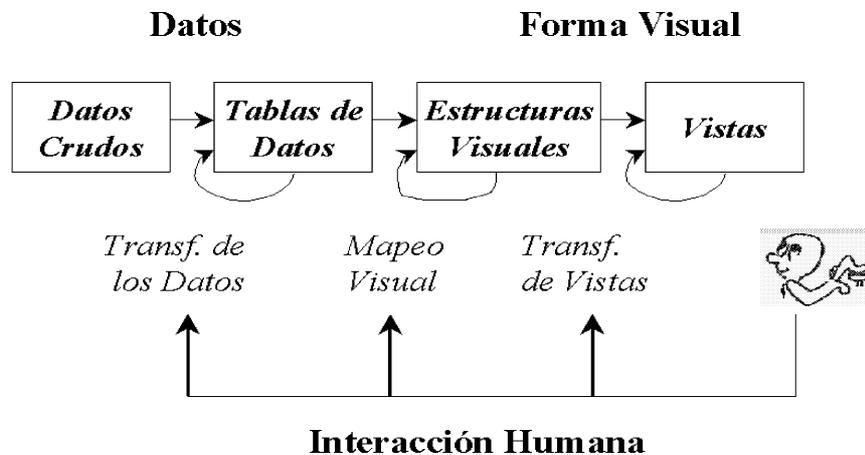
Las bases de datos personales manejan principalmente información descriptiva que, parte de ella, debe responder a un nivel de seguridad que no permita descubrir su individuo. Se debe buscar clasificarla, calcular estadísticos, cuantificar información descriptiva.

Estamos trabajando en encontrar una manera de preprocesar estos datos para luego utilizar técnicas de visualización de forma efectiva[S.C99].

Los datos crudos son transformados en relaciones o conjuntos de relaciones estructurados y fáciles de manipular para poder mapearse a formas visuales. Como este tratamiento matemático omite la información descriptiva importante para visualizar, las visualizaciones pueden pensarse como mapas ajustables que van desde datos a formas visuales, y hacia el perceptor humano. Se obtiene así, la estructura visual que representa la situación para visualizarla adecuadamente.

Mostraremos en la primera parte de este trabajo, el método de transformación de datos en *Tablas de Datos* para obtener un prototipo que mapee alguna Formas Visuales, y en la segunda parte, un breve recorrido de las técnicas de visualización multivariada. Actualmente se está trabajando sobre ambas partes y pretendemos que luego resulte en encontrar un prototipo de visualización adecuado.

## 2 Mapeando Datos



Este esquema muestra las transformaciones que sufren los datos crudos, ajustándose con la interacción del hombre que restringe la vista de ciertos rangos de datos o cambios en la naturaleza de la transformación. El modelo de referencia utilizado [S.C99] es el mapeado de Tabla de Datos a Estructura Visual. Las Tablas de Datos están basadas en relaciones matemáticas; las Estructuras Visuales están basadas en propiedades gráficas efectivamente procesadas por la visión humana. Si bien el dato crudo puede ser visualizado directamente, las Tablas de Datos son un importante paso intermedio para producir un componente espacial directo.

**Tablas de Datos:** Matemáticamente, una relación es un conjunto de *tuplas*:  $\langle Value_{ix}, Value_{iy}, \dots \rangle, \langle Value_{jx}, Value_{jy}, \dots \rangle, \dots$ . Como este tratamiento matemático omite la información descriptiva que es importante para visualizar, se crea la noción de *Tabla de Datos*:

	$Case_i$	$Case_j$	$Case_k$	...
$Variable_x$	$Value_{ix}$	$Value_{jx}$	$Value_{kx}$	...
$Variable_y$	$Value_{iy}$	$Value_{jy}$	$Value_{ky}$	...
...	...	...	...	...

Una Tabla de Datos combina relaciones con *metadatos* que describen aquellas relaciones. Las etiquetas para las filas y las columnas son un ejemplo de metadatos. Las filas representan *variables*, para los rangos de los valores en las tuplas. Las columnas representan los casos, que son conjuntos de valores para cada una de las variables. La doble línea vertical distingue una Tabla de Datos de otros tipos de tablas. El orden de las filas y las columnas en una Tabla de Datos puede o no ser significativo. Este orden es otro ejemplo de metadatos que es importante para visualización.

Una de las ventajas de las Tablas de Datos es que claramente describen el número de variables asociado con un conjunto de datos, una consideración importante cuando se seleccionan visualizaciones. La dimensionalidad se usa para referirse al número de variables de entrada, al número de variables de salida, a ambos, o más aún, al número de dimensiones espaciales en los datos.

**Transformaciones de Datos:** A menudo los datos crudos contienen errores o valores perdidos que deben ser direccionados antes de que los datos puedan ser visualizados. Los cálculos estadísticos pueden también agregar información adicional. Por estas razones, las Tablas de Datos contienen valores o estructuras derivadas. Existen cuatro tipos de estas transformaciones de datos:

*Valor* → *Valor Derivado*

*Estructura* → *Estructura Derivada*

*Valor* → *Estructura Derivada*

*Estructura* → *Valor Derivado*

Las transformaciones de datos pueden ser concatenadas para formar cadenas de agregación y clasificación como parte del proceso de cristalización del conocimiento. Se pueden descubrir patrones y agregarlos como un nuevo dato del esquema codificándolos en las variables de la Tabla de Datos. Las visualizaciones de Tabla de Datos pueden ser usadas para detectar más patrones. Los controles operados por el usuario sobre transformaciones estructurales de la Tabla de Datos pueden ser usados como controles sobre la visualización.

### 3 Visualizando Datos

Muchas técnicas para representar Estructuras Visuales han sido desarrolladas a partir de 1987 [Nie97]. Estas técnicas combinan las métricas espaciales, las marcas y las propiedades gráficas de manera realmente exitosa consiguiendo resultados excelentes. De acuerdo a estas combinaciones, las hemos clasificado en: *Representaciones Matriciales* (composición de ejes ortogonales)

*Iconografía* (métricas espaciales y propiedades gráficas)

*Representaciones Jerárquicas* (conexión y contenido)

*Otras Representaciones* (Brushing y representaciones no cartesianas)

**Representaciones Matriciales** Se refieren a visualizaciones que codifican información posicionando marcas sobre ejes ortogonales. El desarrollo más importante aquí es el análisis de datos exploratorio de Tukey [Tuk77], pilar importante en la visualización de datos. Raramente se utiliza color. Muchas de estas herramientas muestran correlaciones entre dos variables. Aportes importantes se realizaron durante los 80 con Cleveland [S.C93].

**Iconografía** Esta técnica utiliza pequeños objetos gráficos, llamados íconos o glyfos, Este término se utiliza para describir cualquier entidad gráfica que mapea puntos de datos. El glyfo tiene atributos varios tales como tamaño, forma, y color controlados por el valor del punto de información en dimensiones diferentes.[War94][R.P98] Los glyfos proveen una manera efectiva de visualizar arriba de 9 atributos concurrentemente: 3D posición, 3D tamaño, color, forma, opacidad.

**Representaciones Jerárquicas** El *apilado dimensional* es una técnica donde las dimensiones están dentro de otras dimensiones. El orden más alto de dimensión divide la representación en regiones rectangulares. Dentro de cada una de estas regiones, la siguiente dimensión más alta divide la representación en más regiones rectangulares, y así sucesivamente.[JN90] Otra técnica importante es *Mundos dentro de Mundos*, método que explora e intenta entender la visualización de funciones de muchas variables. [SC90]

Para grafos existen algoritmos que producen buenas representaciones tanto top-down como left-right de grafos en forma arbórea [EJ81][II90]. Una técnica popular es representar grafos en 3D en lugar de 2D con la esperanza de que la dimensión extra brinde, literalmente, más espacio y puedan representarse grandes estructuras. La representación hiperbólica (principalmente árboles) es una de las formas nuevas de representar grafos que ha sido desarrollada teniendo en cuenta visualización e interacción. Los primeros trabajos se desarrollaron sobre visualizadores del contenido de la Web.[JP95][JR96]

**Otras Representaciones** -*Brushing*, definición estándar aceptada para *brush*: entidad gráfica que contiene un subconjunto de datos que está siendo visualizado y controlado por el usuario en una manera rápida, intuitiva e interactiva.[BC87][MW95]

-*Coordenadas Paralelas*, en este método cada dimensión corresponde a uno de los ejes verticales espaciados uniformemente. Un punto de dato mapea a un conjunto de puntos a lo largo de cada eje. El punto se representa dibujando una línea de todos los ejes conectando los puntos. [AB90][War94]

## 4 Estado actual y trabajo futuro

El proyecto mencionado anteriormente, donde pretendemos aplicar la visualización, tiene como objetivos específicos mostrar la distribución por género de cargos y funciones de la planta docente y órganos de gobierno, establecer modelos de relaciones existentes entre las diferentes jerarquías según las normas estatutarias y las prácticas establecidas para poder formar el entramado de relaciones socio-educativas al interior de la Universidad.

El estado actual de la rama de visualización se encuentra en la etapa de construcción de la Tabla de Datos y análisis y evaluación de las distintas técnicas de representación. En estos momentos hemos conseguido lograr representaciones de hasta seis dimensiones con algunas de las técnicas mencionadas. El objetivo final es lograr una visualización de por lo menos nueve dimensiones.

## Referencias

- [AB90] A.Inselberg and B.Dimsdale. *Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry*, *IEEE Visualization 90*. IEEE Computer Society Press, Los Alamitos, California, 1990.
- [BC87] Richard Becker and Williams Cleveland. Brushing scatterplots. *Technometrics*, 29:127–142, 1987.
- [EJ81] E.M.Reingold and J.S.Tilford. Tidier drawing of trees. *IEEE Trans. Software Engineering*, 7(2):223–228, 1981.
- [II90] Q.Walker II. A node-positioning algorithm for general trees. *Software -Practice and Experience*, 20(7):685–705, 1990.
- [JN90] M.Ward J.LeBlanc and N.Wittels. Exploring n-dimensional databases, 1990.

- [JP95] R.Rao J.Lamping and P.Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Human Factors in Computing Systems*. CHI 95 Conf.Proc., 1995.
- [JR96] J.Lamping and R.Rao. The hyperbolic browser: A focus+context technique for visualizing large hierarchies. pages 33–55. *J. Visual Languages and Computing* - vol7, no 1, 1996.
- [MW95] Allen R. Martin and Matthew O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Proceeding of IEEE Visualization 95*, pages 271–278, Los Alamitos, California. Oct 1995, 1995. Gregory M. Nielson and Deborah Silver, IEEE Computer Society Press.
- [Nie97] Muller Nielson, Hagen. *Scientific Visualization. Overviews-Methodologies-Techniques*. IEEE Computer Society, California, USA, 1997.
- [R.P98] G.Grinstein R.Pickett. Iconographics displays for visualizing multidimensional data, 1998.
- [SC90] S.Feiner and C.Beshers. Visualizing n-dimensional virtual worlds with n-vision, 1990.
- [S.C93] William S.Cleveland. *Visualization Data*. Hobart Press, Summit, NJ, 1993.
- [S.C99] B.Shneiderman S.Card, J.Mackinlay. *Readings in Information Visualization - Using Vision to Think*. Morgan Kaufmann Publisher, Inc, San Francisco, California, 1999.
- [Tuk77] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
- [War94] Matthew O. Ward. Xmdvtool: Integrating multiple methods for visualizing multivariate data. In *Proceeding of IEEE Visualization 94*, pages 326–336, Los Alamitos, California. Oct 1994, 1994. Daniel Bergeron and Arie E. Kaufman, IEEE Computer Society Press.

## Interacción en la Visualización de Información

Sergio Martig – Silvia Castro – Sandra Di Luca

Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica (VyGLab)  
Instituto de Investigación en Ciencias y Tecnología Informática (IICTI)  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Bahía Blanca  
[smartig/smc]@cs.uns.edu.ar

### Introducción

En un sistema de Visualización, un conjunto de interacciones bien diseñadas puede utilizarse para responder a una gran variedad de preguntas. A pesar de esto, aún no hay una sistematización que permita identificar qué técnicas de interacción aplicar en los distintos puntos del proceso de Visualización de Información. Esta es la razón de por qué se debe caracterizar la interacción en las distintas etapas del proceso de Visualización.

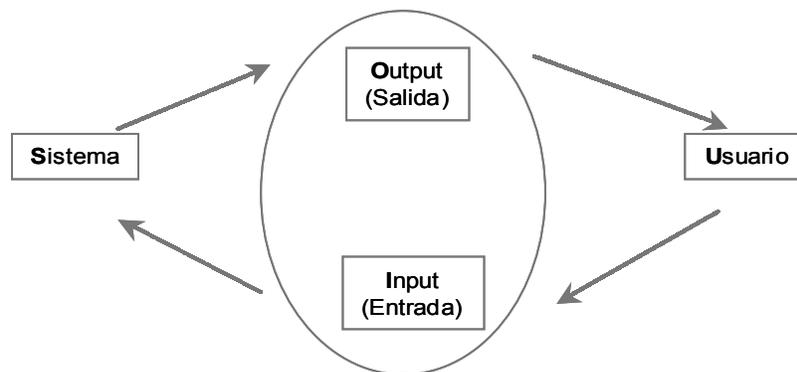
### 1 Interacción

El término interacción implica la existencia de dos partes que se influyen mutuamente en su accionar, para que tenga lugar deben existir como mínimo dos actores. Desde una óptica computacional, interacción es el fenómeno que ocurre cada vez que un usuario (el humano) se involucra en la operación de un producto computacional (el sistema).

Para que la interacción sea efectiva, debe ser completa. Debe contemplar los dos componentes, al humano y a la computadora. Ambos componentes son complejos y difieren mucho en la manera en que ellos se comunican, requieren un protocolo de comunicación que permita una interacción tan rápida y natural como sea posible.

La interface debe en consecuencia ser el medio efectivo de traducción entre el humano y la computadora, para que la interacción sea exitosa. El proceso de traducción puede fracasar en distintos puntos y debido a diferentes razones.

El uso de modelos de interacción pueden ayudarnos a entender lo que está ocurriendo exactamente y detectar los problemas que puedan surgir en ese proceso. Ellos también nos proveen de un marco válido para la comparación de los distintos estilos de interacción y el análisis de los posibles problemas.



Modelo de Interacción (Abowd y Beale)

## 2 Visualización de Información

El campo de Visualización de Información surgió en respuesta a la búsqueda de los investigadores de herramientas que favorecieran el análisis y el entendimiento de datos abstractos mediante el uso de de computación gráfica interactiva y el uso de técnicas de visualización. Los datos abstractos no son inherentemente geométricos y presentan desafíos a los investigadores en Visualización porque evidentemente no es obvio poner los datos abstractos en formas visuales efectivas.

Concretamente podemos decir que la Visualización de Información es el uso interactivo de representaciones visuales de datos abstractos soportadas en computadoras con el objetivo de amplificar el conocimiento. El principal objetivo de esta área de la Visualización es la representación gráfica adecuada tanto de los datos con parámetros múltiples como de las tendencias y las relaciones subyacentes que existen entre ellos. Su propósito no es la creación de las imágenes en sí mismas sino el *insighth*, es decir, la asimilación rápida de información o monitoreo de grandes cantidades de datos.

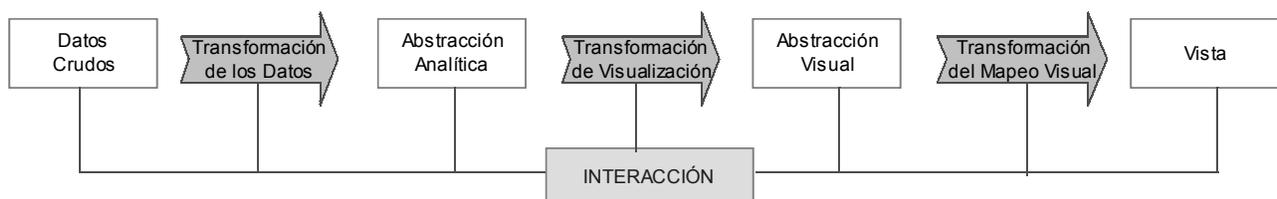
La Visualización de Información puede ser vista como una herramienta de exploración, tiene que asistir e incentivar al usuario en el proceso de análisis del espacio de información. Como dice Jacques Bertin en su *Semiology of Graphics*: “Un gráfico no es dibujado una vez y para siempre; es construido y reconstruido hasta que revela todas las relaciones entre los datos...Un gráfico nunca es el fin en sí mismo; es un momento en el proceso de la toma de decisiones”.

### 2.1 Modelo de referencia:

La Visualización puede ser vista como un proceso de transformación que convierte valores de datos en vistas gráficas. En Visualización Científica, varios investigadores han examinado el uso de un modelo de red de flujo de datos para construir visualizaciones.

El modelo tradicional de red de flujo de datos usado en Visualización Científica es insuficiente para describir a la Visualización de Información. Esto en parte se debe a que los requerimientos de la Visualización de Información difieren de los de la Visualización Científica. En Visualización de Información nos enfrentamos con el hecho de que los datos abstractos no tienen un mapeo espacial inherente lo que nos lleva a tener que resolver cuestiones acerca de cómo representar los datos visualmente, qué tipo de interacción exploratoria incluir y cómo estructurarla. Necesitamos un modelo que no solo describa los procesos de transformación de los datos, sino también los distintos estados de los datos. En particular lo que necesitamos es un modelo que permita unificar el modelo de interacción con el del proceso de visualización.

El modelo (Pipeline de Visualización de Información) propuesto por Ed Huan-shi Chi [2], que surge como modificación del modelo de Stuart Card [1] consta de cuatro estados de datos y de procesos de transformación entre ellos. En los extremos se encuentran los datos y las vistas respectivamente y sus transformaciones van derivando en los estado intermedios. El modelo contempla la interacción del usuario tanto en los diferentes estadios de los datos, como en las transformaciones.



**Pipeline de Visualización de Información**

### 3 Interacción en la Visualización de Información

Avances en las interfaces para visualización de información han mostrado que el análisis visual no solo se beneficia de buenos métodos de representación, sino también de buenas interacciones con esas representaciones. Estas interfaces permiten que los usuarios realicen las operaciones de análisis directamente sobre la representación visual, teniendo realimentación directamente a nivel visual.

En un sistema de visualización un conjunto de interacciones bien definidas pueden ser usadas para responder a una amplia gama de necesidades. En una primera instancia el diseño de un conjunto de interacciones bien definidas requiere de conocimiento en el dominio específico de la aplicación, esto es natural ya que cada disciplina y las cuestiones particulares dentro de las mismas determinan los análisis a los que someterán los datos. Afortunadamente aunque sea frecuente que distintos dominios de aplicación requieran representaciones visuales diferentes, varios de ellos pueden compartir estados intermedios de los datos, o requerir manipulaciones a nivel de vistas similares o incluso necesitar de las mismas transformaciones de datos.

En el marco de un modelo conceptual para la Visualización de Información, podemos analizar y categorizar las similitudes entre los distintos dominios de aplicación y caracterizar las interacciones que puedan tener lugar en las distintas etapas del proceso. El objetivo es lograr caracterizar los distintos tipos de interacción que puedan requerirse en procesos típicos de visualización de información. Se quiere lograr una caracterización del espacio de diseño de las interacciones considerando:

- la etapa del pipeline sobre la que se quiere interactuar
- el estado de los datos

determinando y evaluando los estilos de interacción apropiados para cada caso y su impacto en el resto del pipeline.

### 4 Conclusiones

La persona que, en posesión de un dominio del conocimiento, está frente a una visualización, puede decidir reacomodar los datos o posiblemente adicionar datos relevantes; ésta es la esencia de la visualización interactiva: un examen y una interpretación iterativos de los datos presentados gráficamente. De esto se desprende que un mero dibujo estático no es suficiente para extraer información de los datos. El gráfico debe ser construido y reconstruido (manipulado) hasta que todas las relaciones que subyacen en los datos hayan sido percibidas.

Esto pone en evidencia la imperiosa necesidad de explorar los datos interactivamente. Esta capacidad es tan valiosa que se ha invertido mucho esfuerzo en la creación y la implementación de herramientas de visualización interactiva que exploten este potencial. El análisis visual se beneficia no sólo con buenos métodos de representación visual sino también recibe beneficios de buenas interacciones con estas representaciones.

Si bien es indudable la importancia de la interacción en el proceso de Visualización de Información, aún no hay una sistematización que permita identificar qué técnicas de interacción aplicar en los distintos puntos del proceso de Visualización. De este hecho surge la necesidad de caracterizar la interacción en las distintas etapas del proceso de Visualización.

### Bibliografía

- 1 Card, S., Mackinlay, J., Shneiderman, B., *Readings in Information Visualization – Using Vision to Think*, Morgan Kaufmann, 1999.
- 2 Chi, Ed H. *A Framework for Information Visualization Spreadsheets*. Ph.D. Thesis. University of Minnesota, Computer Science Department. March, 1999

- 3 Dix, A., Finlay, J., Abowd, G., Beale, R. , *Human-Computer Interaction*, Prentice Hall Europe, Second Edition, 1998.
- 4 Feiner, S., Beshers, C., *Worlds within Worlds: Metaphors for Exploring n-Dimensional Virtual Worlds*. Proceedings UIST'90, pp. 76-83.
- 5 Foley, J., Van Dam, A., *Fundamentals of Interactive Computers Graphics*, Addison-Wesley, Reading, Massachussetts, segunda edición, 1992.
- 6 Gallagher, R., *Computer Visualization: Graphics Techniques for Sc. And Eng. Analysis* , 1996.
- 7 Johnson, B., Shneiderman, B., *Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures*, Proceedings of IEEE Information Visualization '91, pp. 275-282.
- 8 Keim, D., Kriegel, H., *VisDB: Database Exploration using Multidimensional Visualization*, IEEE Computer Graphics & Applications, Sept. 1994, pp. 40-49. .
- 9 Levkowitz, H., Herman,G., *Color Scales for Image Data*, IEEE Computer Graphics and Applications, 12, pp. 78-80.
- 10 Lin, X., *Visualization for the Document Space*, Proceedings of IEEE Visualization '92, pp. 957-968.
- 11 Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., Carey, T., *Human-Computer Interaction*, Addison Wesley, 1997
- 12 Rao, R., Card, S., *The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information*, Proceedings CHI'94, 318-322.
- 13 Robertson, G., Card, S., Mackinlay, J., *Information Visualization Using 3D Interactive Animation*, Communications of the ACM, 36(4), pp. 56-71, 1993.
- 14 Shneiderman, B., *Designing the User Interface*, Addison-Wesley Publishing Company, 1998
- 15 Tufte, E.R., *Envisioning Information*, Cheshire, CT Graphics Press, 1990.

# Implementación de un Ambiente para el Estudio Experimental de la Dinámica de Poblaciones

Mirta Padín<sup>1</sup>, Claudio Delrieux<sup>2</sup>, Julián Dominguez<sup>2</sup>, y Héctor Soto<sup>1</sup>

<sup>1</sup>Univ. Nacional de la Patagonia Austral, L. de la Torre 1070, (9300) Río Gallegos,

<sup>2</sup>Universidad Nacional del Sur, Alem 1253, (8000) Bahía Blanca, ARGENTINA.

Voice: (54)(291)4595101 ext. 3381 — Fax: (54)(291)4595154 — e-mail: claudio@acm.org

**Palabras Clave:** Visualización Científica, Sistemas Dinámicos, Dinámica de Poblaciones.

## 1 Motivaciones

En este trabajo mostraremos las ventajas de explorar computacionalmente las diferentes características de los modelos de la dinámica de población y de las interacciones entre el predador y la presa, basados en las ecuaciones de Lotka-Volterra [5]. Este trabajo pretende constituirse a futuro en una herramienta dinámica y versátil para la enseñanza y aplicación en sistemas biológicos, incluyendo la posibilidad explorar alternativas que atiendan a realidades particulares de las poblaciones en análisis.

## 2 Modelo de Lotka-Volterra

En 1920 A. Lotka y V. Volterra proponen un modelo para la dinámica de poblaciones, el cual supone una región cerrada en la cual una función  $F(t)$  representa la cantidad de individuos *presa* contenidos en la región en el instante  $t$ . Se asume una *tasa de mortalidad "natural"*  $\mu_0$ , y que la población produce un número  $\beta$  de nacimientos por unidad de población y de tiempo. En ausencia de predación, la población de presas crece exponencialmente con una *tasa neta de crecimiento* característica  $r = \beta - \mu_0$ . Para estimar la *tasa de mortalidad* adicional per cápita impuesta en la presa por la predación, asumimos que las presas están aleatoriamente distribuidas dentro de un área geográfica  $A$ , y que cada predador  $s$  busca su presa en un área  $A_s$  por unidad de tiempo [2]. Si una fracción  $\sigma$  de los encuentros predador-presa resulta en que la presa es muerta, entonces cada predador consume un promedio de  $\sigma(\frac{A_s}{A})F(t)$  presas por unidad de tiempo. Definimos entonces la *tasa de ataque*  $\alpha = \sigma(\frac{A_s}{A})$  y reescribimos la *tasa de actualización* per cápita de un predador como  $\alpha F(t)$ . Si la región de interés contiene  $C(t)$  predadores en el instante  $t$ , entonces podemos escribir la *tasa de mortalidad* por predación per cápita experimentada por una presa como  $\alpha C(t)$ . Así, la tasa neta de crecimiento per cápita de una población de presas explotada por  $C$  predadores es  $r - \alpha C$ , y podemos escribir

$$\dot{F} = (r - \alpha C)F. \quad (1)$$

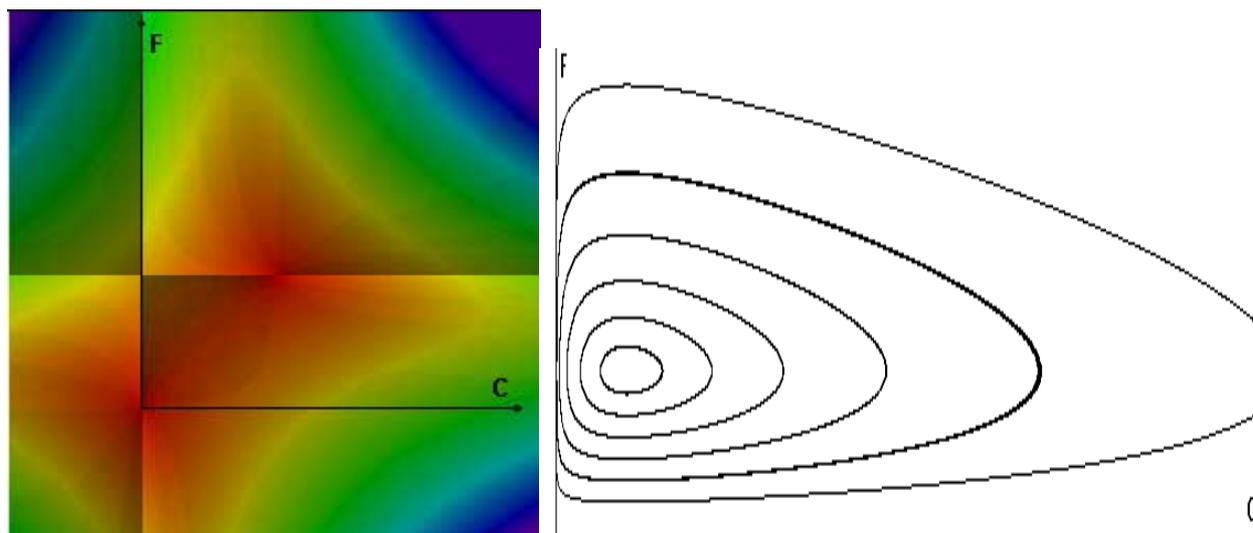


Figura 1: (a) El módulo del Jacobiano del sistema de Lotka-Volterra, representado con una escala de colores, y (b) el diagrama de fases representado por medio de trayectorias.

En la obtención de la ecuación anterior, hemos asumido que cada predador consume  $\alpha F(t)$  presas por unidad de tiempo. Si asumimos que cada presa consumida resulta en la producción de  $\epsilon$  nacimientos, entonces la *fecundidad per cápita* de los predadores en el instante  $t$  es  $\epsilon\alpha F(t)$ . Si los predadores están sujetos a una *tasa de mortalidad per cápita* constante  $\delta$ , entonces su dinámica poblacional está descrita por

$$\dot{C} = (\epsilon\alpha F - \delta)C. \quad (2)$$

Un punto de equilibrio requiere que las presas experimenten una tasa de mortalidad por predación exactamente balanceada con su tasa de crecimiento neto, y que cada predador consuma exactamente suficientes presas para que sus tasas de reproducción y mortalidad se balanceen. En muchas circunstancias, la solución matemática no es posible, por lo que resulta indispensable considerar como alternativa el estudio del sistema por medio de su representación gráfica. En la Fig. 1(a) podemos observar el espacio de fases del sistema, en el cual hemos representado el módulo del Jacobiano del sistema, empleando una escala de colores, y la fase como intensidad. Se observan los dos puntos donde existen estados estacionarios biológicamente posibles, dada una elección de valores para los parámetros del sistema dinámico, en función del valor de  $F$  y  $C$ . Una forma conveniente de pensar sobre las consecuencias de lo mencionado hasta aquí es imaginar que en el tiempo  $t$  contamos el número de presas y predadores en el sistema (o proponemos un valor hipotético, llamado *semilla*) y graficamos el resultado como un punto en el espacio de fases [4]. Pasado cierto tiempo, recensamos el sistema y graficamos el resultado en el mismo gráfico. Uniendo los puntos en orden temporal definimos la *trayectoria* seguida por el cambio de estado del sistema. El conjunto total de trayectorias encontradas para todas las semillas posibles constituye el *diagrama de fases* del sistema. Algunas de estas trayectorias están representadas en la Fig. 1(b), en las cuales hemos comenzado con seis valores iniciales diferentes de presas  $F$  y predadores  $C$ . En cada caso es posible observar que las trayectorias producen curvas cerradas. En realidad, este gráfico nos dice que es esperable que el estado del sistema se mueva en un lazo en torno al estacionario, involucrando una sucesión de episodios en los cuales una baja población de predadores nos conduce a una sobreproducción de presas. El tamaño de los ciclos está determinado por la desviación de la condición inicial respecto

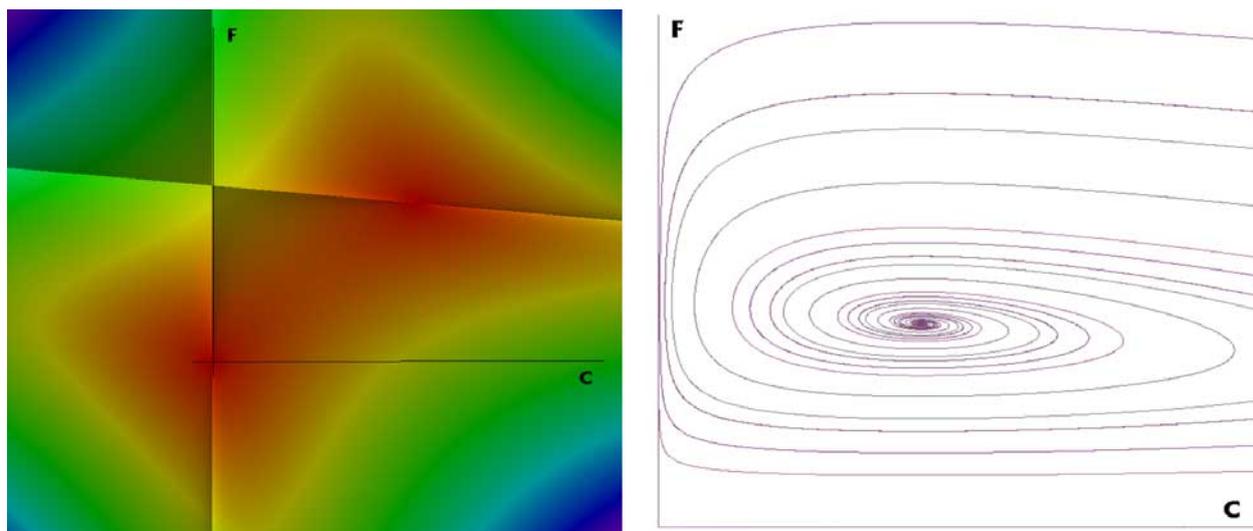


Figura 2: (a) Estados estacionarios del sistema Lotka-Volterra con autolimitación de las presas, y (b) Diagrama de fases de este sistema.

del estado estacionario. El modelo, entonces, implica que la amplitud de cualquier ciclo ecológico, aún en un futuro arbitrario, depende de la condición inicial del sistema en un pasado lejano.

### 3 Modificaciones al modelo

La ventaja más singular del modelo de Lotka-Volterra es su simplicidad. No obstante, esta simplicidad extrema se obtuvo omitiendo muchas características de los procesos de poblaciones reales, y es necesario saber hasta qué grado tales omisiones han determinado su comportamiento. Por dicha razón agregaremos algunas modificaciones a las ecuaciones, una por vez [5]. La primera es que en ausencia de predación, el crecimiento exponencial de la presa no puede continuar indefinidamente. Para modelar la inevitable limitación del recurso y su efecto en el crecimiento de la población de presas, asumimos que, en ausencia de predación, la presa crecería de acuerdo a la ecuación logística, con *tasa de crecimiento intrínseca*  $r$ , hasta alcanzar una población máxima determinada por una *capacidad de carga*  $K$ . Reteniendo todas las otras características de la formulación de Lotka-Volterra, nuestro nuevo modelo es ahora

$$\begin{aligned}\dot{F} &= \left[ r \left( 1 - \frac{F}{K} \right) - \alpha C \right] F, \\ \dot{C} &= (\epsilon \alpha F - \delta) C.\end{aligned}\tag{3}$$

Las simulaciones numéricas de la Ec. 3 muestran que el nuevo modelo se comporta en forma bastante diferente del modelo básico de Lotka-Volterra. En la Fig. 2(b) podemos observar cómo se configuran las trayectorias biológicamente significativas dentro del diagrama de fases de este sistema.

El comportamiento del modelo básico de Lotka-Volterra queda determinado por las condiciones iniciales (conclusión inadecuada desde el punto de vista de la biología), pero la introducción de aún la más mínima autolimitación de la presa hace que el comportamiento a largo término sea independiente de las condiciones iniciales. Para explorar esta restricción

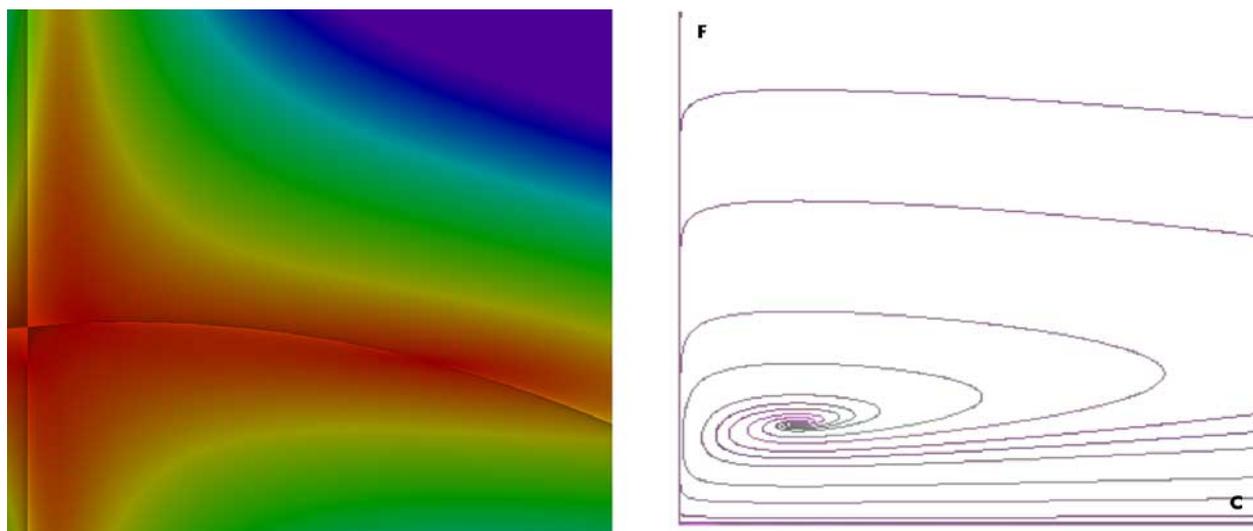


Figura 3: (a) Puntos estables con presas autolimitantes y predadores con saturación, y (b) Diagrama de fases del sistema

al modelo, asumimos que el predador tiene una respuesta funcional de Holling tipo II [6], esto es, la tasa de actualización per cápita  $U(F)$  está dada por

$$U(F) = U_m \left[ \frac{F}{F + F_h} \right]. \quad (4)$$

Sin cambiar las restantes hipótesis planteadas, pero escribiendo ahora la tasa neta de crecimiento de la presa como  $L(F) = rF(1 - \frac{F}{K})$ , tenemos que

$$\begin{aligned} \dot{F} &= L(F) - U(F)C, \\ \dot{C} &= (\epsilon U(F) - \delta)C. \end{aligned} \quad (5)$$

El modelo básico siempre predice oscilaciones continuadas predador-presa, pero el modelo con presa autolimitante, aunque frecuentemente muestra un transitorio oscilatorio, es estable en el largo plazo.

## 4 Un sistema integrado de visualización

Todas las figuras vistas hasta ahora (y una copiosa cantidad no mostrada) fueron generadas con el L-V Workbench, una aplicación computacional especialmente desarrollada por los autores para facilitar la exploración y visualización de las diferentes características del comportamiento dinámico de poblaciones basado en el modelo de Lotka-Volterra y sus variantes. El L-V Workbench permite elegir el sistema dinámico básico o cualquiera de sus variantes, modificar los parámetros de evaluación, y generar representaciones por medio de integración numérica utilizando el algoritmo de Runge-Kutta. La representación visual se realiza por medio de los métodos clásicos de Streamlines y LIC [7, 1], así como otras técnicas intermedias pero más ventajosas desarrolladas por los autores [3]. En el caso de elegir un método basado en texturas, el sistema permite elegir entre un amplio conjunto de texturas base, las cuales se generan dinámicamente por medio de parámetros elegidos por el usuario según sus necesidades de visualización (ver Fig. 4). También es posible representar la evolución temporal de las poblaciones.

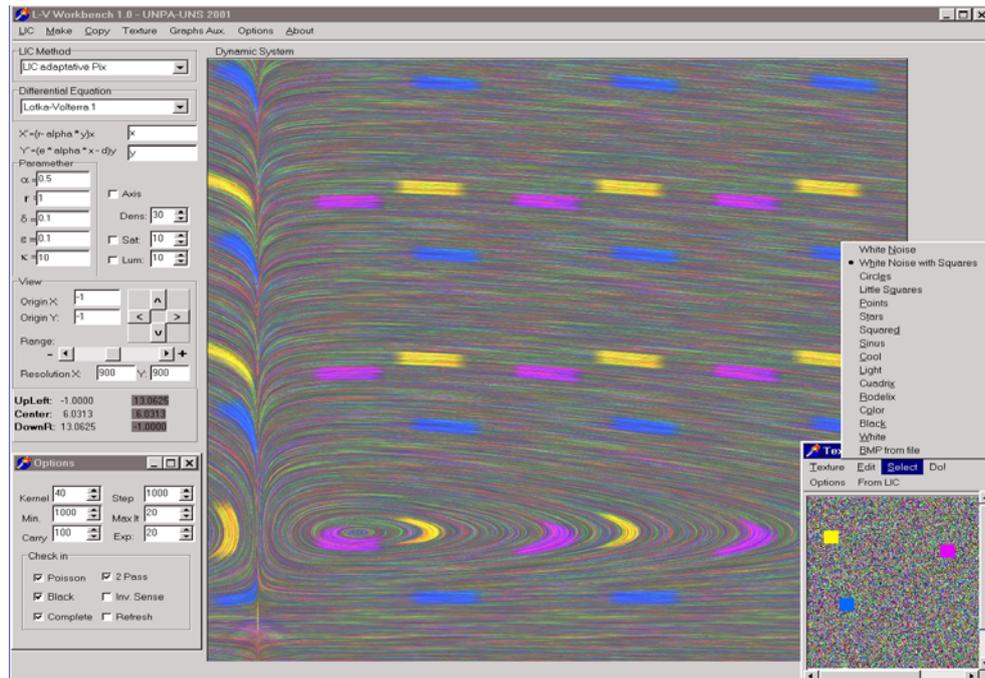


Figura 4: El L-V Workbench en acción.

## Referencias

- [1] B. Cabral y L. Leedom. Imaging Vector Fields Using Line Integral Convolution. *ACM Computer Graphics (SIGGRAPH Proceedings)*, 25(3):263–270, 1993.
- [2] T. J. Case. *An Illustrated Guide to Theoretical Ecology*. Oxford University Press, Oxford, 2000.
- [3] Claudio Delrieux, Julián Domínguez, y Andrés Repetto. Towards a CLIC in Vector Field Visualization. In *CISST 2001 Conference*, págs. 695–702, ISBN 1-892512-73-4, 2001.
- [4] Robert Devaney. *A First Course in Chaotic Dynamical Systems*. Addison Wesley, Reading, MA, 1992.
- [5] W. S. Gurney y R.M. Nisbet. *Ecological Dynamics*. Oxford University Press, Oxford, 1998.
- [6] A. P. Gutiérrez. *Applied Population Ecology*. John Wiley & Sons, New York, 1996.
- [7] J. M. Ottino. *The Kinematics of Mixing*. Cambridge University Press, Cambridge, Massachusetts, 1989.

## "Estimación y Compensación de Movimientos en Video Codificado"

*Magister Claudia Cecilia Russo (UNLP)<sup>1</sup>*

*Magister Hugo Ramón (UNLP)<sup>2</sup>*

### Resumen

En los últimos años han ido apareciendo nuevas aplicaciones en el mundo de las telecomunicaciones, que requieren una continua mejora de los equipos terminales y de los canales de comunicación. El desarrollo y la explotación de nuevos sistemas de comunicación como la videoconferencia o la videotelefonía, deben tratar de compaginarse con la necesidad de aprovechar canales de transmisión de baja capacidad, que en un principio fueron diseñados para transmitir voz o texto. La necesidad anterior ha impulsado el diseño de técnicas de codificación que permitan trabajar con unas tasas de transmisión restringidas, manteniendo una *calidad aceptable*[1], para ello se hace hincapié en el estudio de técnicas de segmentación como también en la implementación de métodos de estimación y compensación de movimiento.

**Palabras Claves:** Compensación de movimiento, estimación de movimiento, segmentación, video, MPEG.

### Introducción

La compresión de video surge de la necesidad de transmitir imágenes a través de un canal que contenga un ancho de banda aceptable. Los métodos de compresión, recurren a los procedimientos generales de compresión de datos, aprovechando además la redundancia espacial de una imagen (áreas uniformes), la correlación entre puntos cercanos y la menor sensibilidad del ojo a los detalles finos de las imágenes fijas (JPEG) y, para imágenes animadas (MPEG), se saca provecho también de la redundancia temporal entre imágenes sucesivas[1] [2] [3].

Cuando las imágenes individuales son comprimidas sin referencia a las demás, el eje del tiempo no entra en el proceso de compresión, esto por lo tanto se denomina codificación intra (intra=dentro) o codificación espacial. A medida que la codificación espacial trata cada imagen independientemente, esta puede emplear ciertas técnicas de compresión desarrolladas para las imágenes fijas. El estándar de compresión ISO (International Standards Organization) JPEG (Joint Photographic Experts Group), está en esta categoría. Donde una sucesión de imágenes codificadas en JPEG también se usan para la televisión, esto es llamado "JPEG en movimiento"[4].

Se pueden obtener grandes factores de compresión teniendo en cuenta la redundancia entre imágenes sucesivas. Esto involucra al eje del tiempo. Este proceso se denomina codificación inter (inter=entre) o codificación temporal.

La codificación temporal permite altos factores de compresión, pero con la desventaja de que una imagen individual existe en términos de la diferencia entre imágenes previas. Si una imagen previa es quitada en la edición, entonces los datos de diferencia pueden ser

---

<sup>1</sup> Magister en Informática - Profesor Adjunto Dedicación Exclusiva, Facultad de Informática, U.N.L.P. E-mail: crusso@lidi.info.unlp.edu.ar

<sup>2</sup> Magister en Informática - Profesor Adjunto Dedicación Exclusiva, Facultad de Informática, U.N.L.P. E-mail: hramon@lidi.info.unlp.edu.ar

insuficientes para recrear la siguiente imagen. El estándar ISO MPEG (Motion Pictures Experts Group) utiliza esta técnica.

### **Compresión de Video en el estándar MPEG [4]**

En el año de 1990, la ISO, preocupada por la necesidad de almacenar y reproducir imágenes de video digitales y su sonido estereofónico correspondiente, creó un grupo de expertos que llamó MPEG (Moving Pictures Expert Group) procedentes de aquellas áreas implicadas en el problema (telecomunicaciones, informática, electrónica, radio difusión, etc).

El primer trabajo de este grupo se conoció como la norma ISO/IEC 11172, mucho más conocida como MPEG-1, en el año 1992. La idea inicial era la de permitir el almacenamiento y reproducción en soporte CD-ROM con un flujo de transmisión de datos del orden de 1,5 Mbits/s, transportando tanto imagen como sonido.

El estándar MPEG además de aprovechar la redundancia espacial intrínseca de una imagen fija utilizada en la codificación JPEG, aprovecha la redundancia temporal que aparece en la codificación de imágenes animadas, permitiendo encontrar similitudes entre las imágenes sucesivas de video. [5]

Debido a que la calidad en la compresión de video en el estándar MPEG-1 era de baja calidad y no servía para otras aplicaciones, se creó la norma ISO/IEC 13818, mucho más conocida con el nombre de MPEG-2. Esta norma permite un flujo de transmisión hasta el orden de los 20 Mbits/s, transportando tanto imagen como sonido. Norma que se utilizaría en la televisión de alta definición.

En la actualidad, se está trabajando en una norma llamada MPEG-4 y está encaminada a la transmisión de datos del orden de los 8 a 32 Kbits/s, norma que será utilizada en las aplicaciones de video conferencia o video teléfono.

La compresión de video utiliza los mismos principios que JPEG con pérdidas, a la que se le añaden nuevas técnicas que, juntas, forman el MPEG-1, que permiten reducir considerablemente la cantidad de información necesaria para la transmisión de imágenes sucesivas muy correlacionadas temporalmente. [7]

Estas técnicas, llamadas de "predicción con compensación de movimiento", consisten en reducir, con un mínimo de información adicional, la mayoría de las imágenes precedentes (incluso las que le siguen).

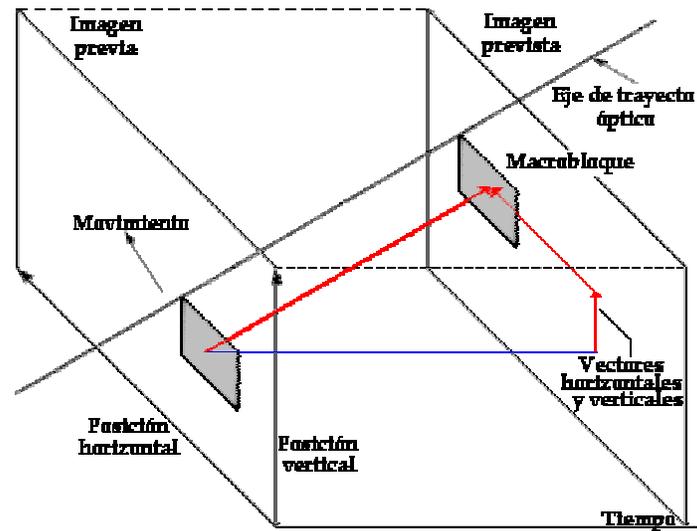
Esto requiere un dispositivo de estimación de movimiento en el decodificador, que es la parte más compleja.

Tratándose de imágenes en movimiento o animadas, la descompresión deberá poder hacerse en "tiempo real" durante la reproducción. Por otro lado, la necesidad de un tiempo de sincronización y de una respuesta de acceso aleatorio a una secuencia no demasiado largos (0.5 segundos máximo) limita el número de imágenes que pueden depender de la misma primera imagen a diez o doce para un sistema de 25 imágenes por segundo. [6]

Es de nuestro interés poner énfasis en el estudio de la optimización de algunos de los procesos involucrados en la compresión de video como la segmentación que será la base para aplicar los métodos de compensación y estimación de movimiento.

### Compensación de movimiento [7]

Es un proceso mediante el cual se mide eficazmente el movimiento de los objetos de una imagen a otra. De este modo se consigue medir qué tipos de movimientos redundan entre imágenes. La siguiente muestra que la imagen en movimiento puede ser expresada en un espacio de tres dimensiones que resulta de mover a lo largo del eje del tiempo dos imágenes consecutivas.



**Objetos viajando en un espacio de tres dimensiones**

En el caso de un objeto que permanezca estático, el movimiento de este solo se ve en el eje del tiempo. Sin embargo, cuando un objeto está en movimiento, este se mueve en el eje de trayecto óptico (eje horizontal y vertical en el tiempo) que no es paralelo al eje del tiempo.

El eje de trayecto óptico une los puntos de movimiento de un objeto a medida que este se mueve a través de varias imágenes. Este trayecto puede ser nulo en valores que representen un objeto en movimiento que solo cambia con respecto al eje del tiempo. De igual manera, al mirar un objeto en movimiento que cambia su apariencia; uno de estos movimientos es el de rotación.

Para un simple movimiento de traslación los datos que representan un objeto son altamente redundantes con respecto al eje de trayecto óptico. Así, que de este modo, el eje de trayecto óptico puede ser hallado, generando un código de ganancia cada vez que se observa la presencia de un mismo objeto en movimiento.

Un codificador de compensación de movimiento trabaja de la siguiente forma. Una imagen I es enviada, pero esta es almacenada de tal modo que pueda ser comparada con la siguiente imagen de entrada para encontrar así varios vectores de movimiento, los cuales pueden ser utilizados en diferentes áreas de la imagen. Luego la imagen I es combinada de acuerdo a

estos vectores o cancelada a una codificación espacial debido a su no conveniencia. La imagen prevista resultante es comparada con la imagen actual para producir una predicción de error también llamada residual. La predicción de error es transmitida con los vectores de movimiento. En el receptor la imagen I original es también retenida en la memoria, esta es cambiada de acuerdo con los vectores de movimiento transmitidos para crear la imagen prevista y luego la predicción de error es adicionada recreando la imagen original. Cuando una imagen es codificada de esta manera, es llamada imagen P en MPEG.

### **Métodos de estimación [8] [9]**

Existen varios criterios para clasificar los diferentes métodos de estimación.

Podemos distinguir entre métodos que emplean la imagen en proceso y los que no la emplean si estamos trabajando con información la cual se emplea para el cálculo de vectores.

Entre los métodos que no emplean la imagen en proceso se encuentran las técnicas recursivas, basadas de forma directa en la ecuación de flujo óptico y que poseen la ventaja de que no necesitan transmitir ninguna información de movimiento; el precio que pagan dichas técnicas, es una predicción con un error mayor que el que se obtiene con el otro grupo de técnicas. Por otro lado, las técnicas que emplean para realizar la estimación información sobre el propio cuadro en proceso, necesitan transmitir información de movimiento al decodificador, pero obtienen una predicción mucho mejor que las anteriores, por lo que el análisis se ajustará a éstas.

Otra posible clasificación de los métodos de estimación, se realiza de acuerdo a si llevan a cabo una estimación del movimiento del plano de la imagen (movimiento 2D) o bien una estimación del movimiento de objetos (movimiento 3D, donde el tiempo es la tercera dimensión).

### **Sentido de la estimación de movimiento. [9]**

La estimación de movimiento trata de obtener los vectores que describen el movimiento aparente entre dos cuadros de una secuencia. Al igual que en la compensación, existen dos posibilidades para su aplicación:

La estimación hacia adelante consiste en la selección de puntos en la imagen de referencia. Dicha imagen esta disponible tanto en el codificador como en el decodificador, por lo que si se emplea en ambos extremos el mismo mecanismo de obtención de esos puntos, sólo resulta necesaria la transmisión de los vectores de movimiento.

La estimación hacia atrás toma los puntos en la imagen actual, obligando a transmitir al decodificador tanto los vectores como las coordenadas de dichos puntos, porque dicha imagen no existe aún en el decodificador.

### **Temas actuales de estudio e investigación**

Nuestro estudio se centra tanto en las técnicas de estimación y compensación como en el análisis de la segmentación de las imágenes para lograr un mayor nivel de estimación y compensación, para esto se estudia la segmentación por bloques fijos, por bloques variables quadtree y la segmentación basada en regiones tomando los objetos dentro de la imagen.

### **Bibliografía**

- [1] Introduction to Data Compression. Khalid Sayood, Morgan Kaufmann 1996.
- [2] Digital Image Processing, Bernd Jahne, Springer 1997.
- [3] Digital Image Processing Algorithms, Ioannis Pitas, Prentice Hall 1995.
- [4] Digital Video Processing, A. Murat Tekalp, Prentice Hall Signal Processing Series 1995.
- [5] Digital Compression of Still Images and Video, R. J. Clake, Academic Press 1995.
- [6] Image and Video Compression, Antonio Ortega and Kannan Ramchandran, IEEE Signal Processing Nov. 1998.
- [7] Video Compression, Gary J. Sullivan and Thomas Wiegand, IEEE Signal Processing Nov. 1998.
- [8] Estimating Motion in Image Sequences, Christoph Stiller and Januz Konrad, IEEE Signal Processing, July. 1999.
- [9] Correspondence Estimation in Image Pairs, Andre Redert, Emile Hendrinks and Jan Biemond, IEEE Signal Processing, May 1999.

## *Clasificación de imágenes digitales utilizando patrones N-dimensionales*

Sanz, Cecilia<sup>1</sup> – De Giusti, Armando<sup>2</sup>  
{csanz, degiusti}@lidi.info.unlp.edu.ar

*LIDI. Laboratorio de Investigación y Desarrollo en Informática<sup>3</sup>.  
Facultad de Informática. UNLP.  
50 y 115. 1er Piso. La Plata*

**Palabras Clave:** *Procesamiento de Imágenes, Técnicas de Clasificación, Razonamiento Evidencial, Imágenes Hiperespectrales, Sensado Remoto, Algoritmos paralelos*

### **Resumen**

En esta línea de investigación se estudian y analizan diferentes técnicas de clasificación de imágenes digitales, en particular se centra la atención en aquellas que permiten integrar un alto número de variables de entrada, aún cuando las mismas sean de diferente naturaleza. Se analiza el método de Razonamiento Evidencial y se propone una modificación para el mismo (Razonamiento Evidencial Dinámico). También se realiza una comparación con otros métodos tradicionales de clasificación sobre aplicaciones reales. Se presentan aquí algunos resultados obtenidos hasta el momento.

En una segunda etapa se estudia la paralelización de los algoritmos de tratamiento de imágenes, buscando optimizar los tiempos de procesamiento. Como primer paso en esta etapa, se ha desarrollado una versión paralela para el algoritmo de Razonamiento Evidencial Dinámico y se ha comparado la misma con su versión secuencial.

### **Introducción**

En esta línea de investigación se estudian distintas técnicas de clasificación para imágenes digitales, en particular la de Razonamiento Evidencial [1], a la cual se le han introducido modificaciones (Razonamiento Evidencial Dinámico).

Con el avance de la tecnología se obtienen actualmente imágenes con sensores hiperespectrales que se caracterizan por tener de 10 a cientos de bandas continuas de un intervalo ancho del espectro, y proveen mayor información respecto a la provista por los sensores multiespectrales. Muchas de las aplicaciones que involucran el análisis de este tipo de imágenes también utilizan información auxiliar almacenada en base de datos GIS e integran datos espaciales, espectrales y temporales. Es por ello que es necesario explorar nuevos métodos de clasificación que permitan una adecuada utilización de los datos de entrada disponibles para lograr una mejor precisión en los resultados.

Se analizan las ventajas y desventajas de los diferentes métodos cuando se trata de clasificar imágenes hiperespectrales. Se trabaja en una aplicación utilizando dicho tipo de imágenes, en la que se busca el reconocimiento de diferentes tipos de cultivos (maíz, sorgo, soja) y de las rutas que aparecen en las mismas.

---

<sup>1</sup> Profesor Dedicación exclusiva - Facultad de Informática. UNLP.

<sup>2</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva.

<sup>3</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.  
TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

Clasificar e identificar posibles problemas que se presentan en alguna área geográfica en particular, es un objetivo no sólo de interés científico sino también gubernamental que puede permitir evitar catástrofes como inundaciones, incendios, plagas, condiciones climatológicas adversas, etc.

### **Temas de Investigación y desarrollo**

Dentro de esta línea, se han investigado y estudiado métodos del reconocimiento de patrones que van desde métodos basados en regionalización a métodos estadísticos como Bayes o el de Máxima Verosimilitud. Se ha realizado un estudio detallado de técnicas de clasificación no supervisada como ISODATA, K-medias, y el algoritmo de Batchelor y Wilkinson [2]. Pero principalmente se ha centrado la atención sobre los métodos de clasificación relacionados con el área de sensado remoto como el de Razonamiento Evidencial, que se ha utilizado para muchas aplicaciones como clasificación de bosques, detección de patrones marinos, control de recursos hídricos, etc. Este método permite integrar datos de diferentes fuentes de información que pueden ser de distinta naturaleza.

Se ha propuesto una modificación del método, al que se lo llama Razonamiento Evidencial Dinámico, de manera tal de mejorar su precisión. Esta modificación consiste en permitir tener una etapa de aprendizaje supervisado, y de este modo incorporar nueva evidencia para el sistema. Además se modifica su regla de decisión, y se la compara con las distintas alternativas propuestas por diversos autores y otras surgidas de la presente investigación [3]. Se evalúa su comportamiento para la clasificación de los cultivos en imágenes hiperespectrales. Esto se realiza sobre una etapa específica de la evolución de los cultivos y para un área geográfica particular (Nebraska –USA). Las imágenes fueron provistas por un grupo de trabajo de USA [4], que las ha utilizado para hacer investigación sobre el estrés de los cultivos.

Para esta aplicación se busca estudiar la integración de información espectral (se toman medidas derivadas de las bandas espectrales de la imagen que para este caso son 10) y cierta información espacial como son los rasgos de textura. Para este caso se utilizaron estadísticas de primer orden como la media, el “skewness”(simetría), la varianza y el rango.

Como extensión de esta línea de investigación se propone la paralelización del algoritmo [5], esto es importante dado la carga computacional que implica el trabajo con las imágenes hiperespectrales las cuales tienen de 10 a cientos de bandas. Además el algoritmo trabaja integrando la evidencia provista por diferentes fuentes de información lo que supone un tiempo de respuesta mayor a medida que se incrementa la cantidad de dichas fuentes. Se desarrollaron ya algunas versiones paralelas obteniéndose los resultados esperados como se detallará en la próxima sección.

### **Algunos resultados obtenidos. Líneas de Investigación en curso.**

Dentro de los resultados obtenidos se pueden mencionar los siguientes:

- Se han estudiado diferentes métodos de clasificación y se los ha analizado, destacando sus puntos fuertes y débiles (algoritmos de clasificación supervisados: Máxima Verosimilitud, Bayes, basados en distancia y en regionalización y no supervisados: ISODATA, k-medias). También se analizó el comportamiento de las redes neuronales y modelos biofísicos para el tipo de problemas planteados en el área de sensado remoto.
- Se ha realizado un análisis comparativo teórico del comportamiento del método de Razonamiento Evidencial Dinámico respecto de otras técnicas de clasificación

En particular de los dos puntos anteriores, se destaca que los métodos de clasificación tradicionales como el de Bayes o Máxima Verosimilitud no fueron diseñados para procesar conjuntos de datos modernos, los cuales suelen tener:

- Alta dimensión (alto número de bandas espectrales)
- Propiedades inapropiadas para el análisis paramétrico
- Información surgida de múltiples fuentes con disparidades inherentes, inconsistencias, y errores.
- Datos con diferentes escalas de medición o con distintos niveles (por ejemplo datos nominales, de radio, direccionales, etc).

Del método de Razonamiento evidencial y el de Razonamiento Evidencial Dinámico, se destacan los siguientes puntos:

- No es necesario presuponer una distribución para los datos.
- La teoría provee una base general y heurística para integrar cuerpos distintos de información a partir de fuentes independientes. De esta forma, pueden incorporarse un alto número de datos y de diferente naturaleza. Incluso es posible incorporar los resultados de una clasificación previa, lo cual puede resultar muy útil para el análisis de la evolución de los cultivos de un área geográfica particular.
- Se permite manejar la evidencia no asignada o incertidumbre.

El método de Razonamiento Evidencial Dinámico (RED) propuesto mantiene las ventajas presentadas en el método de Razonamiento Evidencial, incorporando una serie de variaciones que modifican en forma positiva el comportamiento del clasificador. La etapa de aprendizaje propuesta por el mismo permite la incorporación de nueva evidencia y esto tiende a un notable mejoramiento en los resultados obtenidos. Claro está, que es necesario contar con información confiable sobre la pertenencia de las muestras que se agregan por medio del proceso de aprendizaje a una cierta categoría o clase.

En cuanto a la regla de decisión propuesta por RED, la misma permite establecer un umbral de seguridad para no arriesgar por una clase a menos que se disponga de una cierta evidencia. Además, la regla trabaja analizando cuál es la clase con mayor soporte, pero no sólo utiliza esta información para la decisión, sino que también estudia la medida de plausibilidad y la distancia entre los soportes de la clase de mayor soporte y el resto de las clases con el fin de saber si han recibido una evidencia similar. En tal caso, se considera que la clase a asignar es aquella que posee el mayor valor resultante de la combinación de un índice de cantidad de fuentes (**ICF**), los pesos asignados a cada relación fuente-clase, y el soporte [6].

Se probó el clasificador RED con diferentes conjuntos de tests para cada clase, tomados de diferentes imágenes. Se obtuvieron los siguientes promedios :

Maíz: 13 regiones(500 pixeles) -> 90.07 % precisión

Ruta: 14 regiones (678 pixeles) -> 99.85 % precisión

Soja: 10 regiones (678 pixeles) ->96.4 % precisión

Sorgo: 5 regiones (779 pixeles) -> 92.2 % precisión

También se realizó un análisis comparativo con el Razonamiento Evidencial, utilizando la regla del máximo soporte:

**Clase Maíz**

RED: 90.07 % RE: 71.76 %

**Clase Ruta**

RED: 99.85 % RE: 99.57 %

**Clase Soja**

RED: 96.4 % RE: 92.9 %

**Clase Sorgo**

RED: 91.8 % RE: 91.8 %

- Se estudiaron y analizaron diferentes fuentes de información para utilizar como entrada al clasificador.
- Se desarrolló e implementó una versión paralela de RED, en donde se dividió la imagen a procesar en diferentes procesadores, cada uno realizando la clasificación de la parte correspondiente. Un proceso maestro es el encargado de reunir los resultados parciales de los procesos clasificadores. Los resultados obtenidos fueron los esperados [7].

Como líneas de investigación futuras se propone:

- Estudiar otras posibilidades de paralelización además de las ya estudiadas
- Aplicar y analizar la técnica de Razonamiento Evidencial Dinámico paralelo para la detección de cultivos específicos en imágenes multiespectrales y/o hiperespectrales para un período específico de su evolución.
- Estudiar las diferentes etapas de evolución de cultivos y las características que los mismos presentan. Luego, extender la aplicación para el estudio de los cultivos en sus diferentes etapas evolución en el tiempo.

**Bibliografía Básica****[1]** Peddle

“Mercury<sup>⊕</sup>: An evidential reasoning image classifier”

Computers & Geosciences.

Vol. 21 – Nro. 10- 1995

**[2]** R. Duda, P. Hart

"Pattern Classification and Scene Analysis"

John Wiley. 1973.

**[3]** C. Sanz

“Razonamiento Evidencial Dinámico. Un método de clasificación aplicado al análisis de imágenes hiperespectrales”.

Tesis Doctoral. Facultad de Ciencias Exactas. 4/01/2002

**[4]** <http://www.earthscan.com>

EarthScan. Realizan estudios de áreas cultivadas.

[5] C. Sanz – A. De Giusti

“A distributed solution for dynamic evidential reasoning applied to the classification of hyperspectral images”.

5th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) y 7th International Conference on Information

Systems Analysis and Synthesis (ISAS 2001), Orlando, USA

Julio de 2001.

[6] C. Sanz

“DER (Dynamic Evidential Reasoning), applied to the classification of hyperspectral images”,

International Geoscience and Remote Sensing Symposium (IGARSS 2001 – IEEE), Sydney, Australia

Julio, 2001.

ISBN: 0-7803-7033-3

[7] Sanz, De Giusti.

“Paralelización del Algoritmo de Clasificación DER usando imágenes hiperespectrales”.

VII Congreso Argentino de Ciencias de la Computación, Calafate, Argentina

Octubre, 2001.

## Simulación de Medios Artísticos Tradicionales

Alfredo Zavala

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional<sup>1</sup>  
Departamento de Informática  
Universidad Nacional de San Luis  
Av. Ejercito de los Andes 950 Local 106  
5700 – San Luis  
Argentina  
e-mail: azavala@unsl.edu.ar

### Resumen

Entre los estilos de rendering podemos mencionar al fotorealístico<sup>2</sup> y al nofotorealístico<sup>3</sup>. Esta investigación, que forma parte del área de computación gráfica, se basa en éste último estilo de generación de imágenes. En particular, en la simulación de un medio tradicional de artista, en el cual la escena es generada usando algún medio tradicional de pintura. La generación NPR de escenas basadas en la simulación de un medio tradicional de artista ha sido de gran interés en los últimos años y se sigue desarrollando dado que provee una herramienta a usuarios poco interiorizados con los distintos estilos de pintura, de generación de imágenes basadas en técnicas de artista.

El objetivo de esta investigación es desarrollar una herramienta capaz de simular los lápices de colores como medio de artista y las distintas técnicas usadas en el mundo real para la generación de pinturas usando lápices de colores.

### Introducción y Trabajos Previos

No solamente la generación de imágenes fotorealísticas es un tópico importante de investigación en la síntesis de imágenes sino también el rendering nofotorealístico y en particular la síntesis de imágenes artísticas también lo es. Dentro de este campo muy pocos desarrollos se han hecho en la generación de pinturas usando lápices de colores, tal vez por el prejuicio de considerarlo como un medio para principiantes y no para profesionales de la pintura, pero nada más alejado de la realidad.

Dentro del campo de simulación de medios artísticos tradicionales podemos mencionar:

- Strassman[1] quien simula el arte tradicional japonés del *sumi-e* modelando las cerdas, el movimiento del pincel y la absorción de tinta en el papel.
- Salisbury[2] presentó un sistema interactivo para crear ilustraciones en pen-and-ink usando trazos texturados, esto es un trazo que lleva tono y textura. Winkenbach[3]

---

<sup>1</sup> El laboratorio es dirigido por el Dr. Raúl Gallard y subvencionado por la Universidad Nacional de San Luis y la ANPCyT (Agencia Nacional para la Promoción de la Ciencia y la Tecnología).

<sup>2</sup> Inicialmente la computación gráfica tuvo como objetivo la generación de escenas con un alto grado de realismo.

<sup>3</sup> Posterior al estilo fotorealístico. Surge como una forma alternativa de generación de una escena basada en la idea de que a veces un bosquejo con algunas pocas líneas es más demostrativo que una foto; por ejemplo las ilustraciones de un manual de mecánica son mucho más demostrativas que una foto de la pieza.

continuó el desarrollo en pen-and-ink y mostró como los principios de las ilustraciones en pen-and-ink pueden ser implementadas como parte de un sistema de rendering automatizado.

- Curtis[4] describe varios efectos artísticos de la acuarela y como ellos pueden ser simulados automáticamente, usa el modelo de Kubelka-Munk para la composición óptica de capas coloreadas y demuestra como la generación por computadora de acuarelas puede ser usado en un sistema de pintura como un método para, automáticamente, transformar una imagen en una pintura de acuarela y como un mecanismo para generar escenas 3D no fotorealísticas. Este trabajo se basa en el desarrollo de Small[5], que basado en una arquitectura paralela, modela acuarelas simulando la difusión, los pigmentos y la fibra del papel.
- Sousa[6, 7, 8] hace todo el desarrollo para la simulación de pinturas usando lápices de grafito. Comienza modelando los distintos tipos de lápices de grafito (desde 9H al 8B), el papel y su interacción, basado en la observación al microscopio de los mismos, posteriormente le incorpora borrador y blender y por último realiza la generación por computadora de modelos 3D usando lápices de grafito.
- Takagi[9, 10] realiza un modelado volumétrico para dibujos de lápices de colores. El modelo consiste de tres submodelos que describen: la microestructura del papel, la distribución de pigmentos sobre el papel y la redistribución de pigmentos. Para el modelado usa como visualizador volumétrico a VolVis.

## Desarrollo

Es objetivo de esta investigación el desarrollo de una herramienta capaz de simular lápices de colores como así también las distintas técnicas de artista asociadas a dicho medio.

Si bien existen trabajos de simulación en lápices de colores, los mismos han sido encarados modelando geoméricamente cada uno de una de sus componentes, esto es: se usan cilindros y discos para modelar el papel (fibra y talco). Como resultado se observa que haciendo un corte transversal del mismo la estructura del papel describe una poligonal *Manhattan*. La punta del lápiz y el pincel (usado para la técnica del agregado de agua) son modelados como esferas y el borrador es modelado como una elipse.

Distinto es el enfoque en la simulación de lápices de grafito. Allí se hace un modelado de papel usando el concepto de grano y volumen del mismo y se usa para ello información de los distintos pesos de los vecinos para obtener el volumen del grano que, posteriormente, durante el proceso de interacción lápiz/papel se irá llenado. Un corte transversal sobre dicho modelo de papel describe una poligonal. Además se modela la mina del lápiz como un polígono convexo y esto da la posibilidad de poder, usando el mismo lápiz (esto es sin variar la punta de la mina), cambiar el ángulo de incidencia sobre el papel y lograr distintos efectos. El modelado del borrador se hace, al igual que con la punta del lápiz, usando un polígono convexo.

Del estudio realizado sobre la pintura en lápices de colores[13, 14] surge que para su implementación necesitaremos modelar los siguientes elementos: *papel, lápices de colores, fijadores, sacapuntas, borradores, solventes y mezcladores* de los cuales sólo nos limitaremos, en esta primera etapa, al papel y los lápices dejando para futuras discusiones los restantes elementos, como así también todo lo que tiene que ver con las técnicas para coloreo entre las que se encuentran: *sombreado amplio, circular, círculo, crosshatch, líneas diagonales, horizontales o verticales, aplicación pesada*, entre otras.

- Papel:** El papel usado para pintura de lápices de colores puede estar hecho de una mezcla de fibras de madera dura y madera blanda o fibras de algodón, de lino, de yute. La fabricación del papel se puede ubicar en 3 categorías: **hecho a mano**, **hecho en molde** y **hecho a máquina**. Teniendo en cuenta su composición y el método de fabricación, se tienen las siguientes características: a) **Permanencia**, de los pigmentos en la superficie del papel y b) **Superficie o textura**, pudiendo ser *Papel áspero* (irregular) es la superficie natural del papel hecho a mano; *Levemente granular* (mate, medio, regular, satén) o *Liso* (esmalado, de alto brillo). Para modelar la superficie usamos ruido Perlin el cual permite, cambiando los valores de frecuencia y amplitud, generar los tres tipos de texturas (desde el más al menos rugoso). Además, aquí, hay que tener en cuenta el color del papel, el cual puede ser elegido por el usuario.

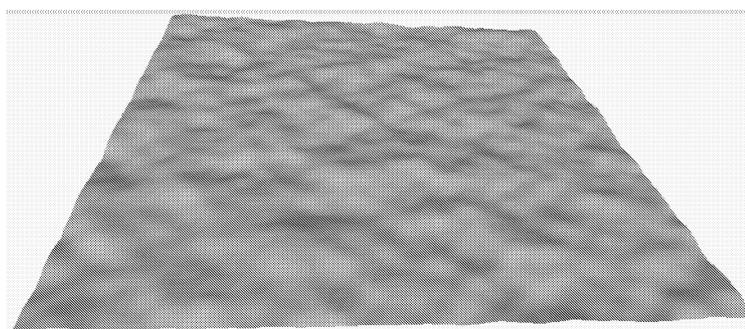


Fig. 1: Papel generado usando ruido Perlin

- Lápices de colores:** compuesto de los siguientes materiales: *masilla*, *colorante* (el pigmento o el tinte), *material de ligadura* y *cera*. Como característica saliente tenemos el color. Para la implementación, y teniendo en cuenta su composición, los lápices de colores agregan un componente respecto a los tres de los lápices de grafito (grafito, cera y material de ligadura) pero al igual que éstos, solamente se tomará en cuenta el pigmento al momento de reflejar el color. La diferencia con los lápices de grafito es que vienen en una gran variedad basados en la composición de grafito y cera, el cual define el grado de dureza de la mina y la tonalidad del trazo. Para los lápices de colores lo importante es el color (pigmento) y de allí la necesidad de elegir un conjunto adecuado de colores. Para este último punto se prevé realizar dos aproximaciones:
  - ✓ **Por un lado dejar que el usuario decida el color de lápiz a usar:** en este caso la aproximación al tono deseado se hará en función de cómo se ve reflejado contrastándolo con fondo negro y blanco.
  - ✓ **Sólo disponer de un conjunto limitado de colores**[11, 12]: en este caso los colores de los lápices estarán preestablecidos y se compondrán entre ellos para lograr otro tono.

## Conclusiones

Esta investigación tiene como objetivo el desarrollo de una herramienta capaz de simular el medio artístico de los lápices de colores como así también las distintas técnicas usadas por los artistas para colorear. La finalidad es proveer una herramienta a usuarios con y sin entrenamiento en lápices de colores.

Se presenta aquí una primera aproximación a los lápices de colores junto con una breve discusión acerca de los trabajos realizados en este y otros medios. Así mismo, se muestra que parte de la simulación comparte con los trabajos existentes y cuales son nuevo. Cabe mencionar que la interacción lápiz/papel seguirá la línea propuesta por Sousa con algunas alteraciones, mientras que otras (ej. fijador/papel) serán simulaciones exclusivas para este medio.

### Bibliografía

- [1] Steve Strassmann, *Hairy brushes*. En SIGGRAPH '86 Proceedings, pp. 225–232. August 1986.
- [2] Michael P. Salisbury and Sean E. Anderson and Ronen Barzel and David H. Salesin, *Interactive Pen--And--Ink Illustration*, Proceedings of SIGGRAPH '94, 1994, july, pp. 101-108.
- [3] George Winkenbach and David H. Salesin, *Computer-Generated pen-and-ink Illustration*. Proceedings of SIGGRAPH '94, 1994, july, pp. 91-100.
- [4] Cassidy. J. Curtis, Sean E. Anderson, Joshua. E. Seims, Kurt W. Fleischer, and David. H. Salesin, *Computer-Generated Watercolor*. Proceeding of SIGGRAPH '97, 1997, august, pp. 421-430.
- [5] David Small, *Simulating watercolor by modeling diffusion, pigment, and paper fibers*. Proceeding of SPIE, Image Handling and Reproduction Systems Integration, vol. 1460, 1991, august, pp. 140-146.
- [6] Mario C. Sousa and John W. Buchanan, *Observational model of blenders and erasers in computer generated pencil rendering*. Proceeding of Graphics Interface '99 1999, june, pp. 157-166.
- [7] Mario C. Sousa and John W. Buchanan, *Computer-generated graphite pencil rendering of 3d polygonal models*. Proceeding of Eurographics, 1999, september, pp. 195-207.
- [8] Mario C. Sousa and John W. Buchanan, *Observational models of graphite pencil drawing materials for non-photorealistic rendering*. Computer Graphics Forum 19(1), pp. 27-49, March 2000.
- [9] S. Takagi and I. Fujishiro, *Microscopic structural modeling of colored pencil drawings*. En ACM SIGGRAPH 97 The Art and Interdisciplinary Programs (Technical Sketches), pag. 187, August 1997.
- [10] S. Takagi, M. Nakajima and I. Fujishiro, *Volumetric Modeling of Colored Pencil Drawing*. En Pacific Graphics '99, October 1999.
- [11] Joanna L. Power, Brad S. West, Eric J. Stollnitz, and David H. Salesin. *Reproducing color images as duotones*. Proceedings of SIGGRAPH 96, in Computer Graphics Proceedings, Annual Conference Series, pp. 237-248, August 1996.
- [12] Eric J. Stollnitz, Victor Ostromoukhov, and David Salesin. *Reproducing Color Images Using Custom Inks*. Proceedings of SIGGRAPH 98, in Computer Graphics Proceedings, Annual Conference Series, 1998.
- [13] Tiffanie L. Gray, *Colored Pencil Techniques*. <http://elfwood.lysator.liu.se/farp/cptech/>
- [14] Colored Pencil Society of America - CPSA. <http://www.cpsa.org/>

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

# WICC 2002

*16 y 17 de mayo de 2002*

## Tecnología Informática Aplicada en la Educación



## Una propuesta para Ambientes Colaborativos Inteligentes

Proyecto de Investigación 430301

G. Aguirre, M. Chiarani, B. García, G. Leguizamón, M. Lucero, L. Manzur, e I. Pianucci

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950 - (5700) San Luis

Tel/Fax: +54 2652 420823/430224

e-mail: {gaguirre, mcchi, bgarcia, legui, margos, lman, pianucci}@unsl.edu.ar

### Resumen

El presente trabajo pretende comunicar los avances del proyecto de investigación "Ambientes Colaborativos Inteligentes" (ACI) aplicado en ambientes de trabajo y/o aprendizaje colaborativos. En el diseño de dicho ambiente se pretende integrar aspectos relacionados a entornos de trabajo/aprendizaje colaborativo respaldado por computadoras, CSCW y CSCL (siglas en inglés para *Computer Supported Collaborative Work* y *Computer Supported Collaborative Learning* respectivamente) y en una etapa posterior, la incorporación de agentes inteligentes con diferentes propiedades y/o funcionalidades.

El desarrollo de la plataforma se realizará de acuerdo a un modelo cliente/servidor basada en la Web. Sin duda éste es el espacio más propicio, no sólo para compartir la información sino para ofrecer un entorno de trabajo/aprendizaje colaborativo para usuarios que eventualmente se encuentren dispersos geográficamente.

### Fundamentación

El creciente avance de las nuevas tecnologías de información ha permitido visualizar diversos campos para su utilización, particularmente aquellos en donde la comunicación, velocidad de acceso a los datos y trabajo colaborativos son factores importantes. Especialmente en el contexto de ambientes colaborativos, existen diferentes enfoques: el trabajo colaborativo respaldado por computadoras CSCW (sus siglas en inglés para *Computer-Supported Collaborative Work*) el cual se define como una red de computadoras (por ejemplo una LAN o WAN) que respalda un grupo de trabajo con objetivos o tareas en común para lo cual existe una interfaz adecuada para comunicar a los participantes. Por otro lado, como un caso particular de CSCW, se introduce el concepto de "aprendizaje colaborativo" CSCL (sus siglas en inglés para *Computer Supported Collaborative Learning*) cuya definición es similar a la de CSCW, con la diferencia que CSCW tiende a enfocarse más a las técnicas de comunicación en si mismas, antes que centrarse en lo que está siendo comunicado, lo que caracteriza a CSCL. Ambos enfoques se sostienen en la premisa de la disponibilidad de sistemas de computación que puedan respaldar y facilitar los procesos y dinámicas propias de grupos en situaciones donde éstas no puedan ser alcanzadas a través del contacto directo pero que tampoco pretendan reemplazar la comunicación directa o *cara-a-cara*. De esta manera, sistemas del tipo CSCW y CSCL pueden ser diseñados para ser usados por múltiples usuarios trabajando ya sea, en la misma computadora (en diferentes tiempos) o bien a través de computadoras dispuestas en red con el objetivo de generar un flujo de datos entre los participantes como por ejemplo, ideas, documentos, información de acceso y principalmente, proveer de retro-alimentación en las actividades involucradas en la resolución de problemas comunes.

### Propuesta

En los últimos años la Web se ha convertido en el servicio más importante de Internet y proporciona acceso universal a una amplia gama de información a millones de usuarios. Utilizando un simple programa para navegar en Internet, los usuarios pueden acceder a su área de trabajo, transferir

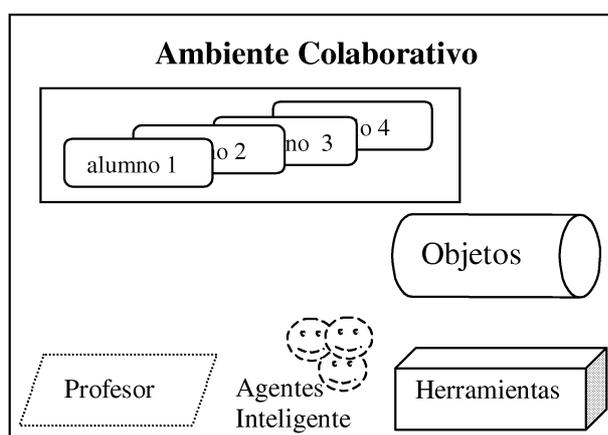
documentos de un área compartida a su máquina y viceversa. Pero, a diferencia de los sistemas comerciales que hablan de trabajo "colaborativo" que son sólo meros repositorios de documentos, se busca desarrollar una plataforma que vaya más allá de esta situación y permita que personas que conforman un grupo de trabajo puedan brindarse cooperación sin importar el hardware y software disponible. Si pensamos que el aprendizaje debe ser activo y cooperativo, los medios tradicionales de la educación a distancia son pasivos y proporcionan una interacción mínima entre estudiantes y profesores y entre los propios estudiantes, por lo tanto, disminuir el aislamiento y ofrecer un ambiente estimulante y cooperativo de aprendizaje son objetivos a los que deben contribuir las nuevas tecnologías.

La Web proporciona una infraestructura fácil y de gran alcance para el desarrollo de aplicaciones, porque fue creada con la intención de utilizar la colaboración entre los investigadores dispersos geográficamente. Siempre que nos comunicamos o cooperamos, compartimos el entorno y el contenido. En la red, sin embargo, la infraestructura de la misma puede darnos más que un lugar para crear y compartir la información. Puede también decirnos lo que están haciendo otros en ese espacio; informar sobre cambios relevantes; la pista de cómo se encuentra y se utiliza la información; y ofrecer un entorno para las discusiones sobre contenido compartido.

La propuesta de nuestro grupo de investigación es desarrollar una Plataforma Colaborativa Inteligente basada en agentes que facilite la interacción entre miembros de un grupo de trabajo. Nuestro caso de estudio está dirigido a potenciales grupos de docente/alumnos involucrados en un proceso de enseñanza/aprendizaje, en los cuales profesores y alumnos acceden sincrónica o asincrónicamente al ambiente para realizar distintas actividades. En especial los alumnos podrán trabajar en el ambiente colaborativo con su grupo de compañeros, pudiendo además acceder a distintas herramientas (chat, correo electrónico, foros, pizarra, etc.) y compartir diferentes objetos (bibliografía, material en la web, Unidades de aprendizaje).

## Componentes del ambiente colaborativo

### 1. Usuarios:



- Profesor:** Responsable del contenido curricular y dictado del curso. Se encarga de supervisar e intervenir en los momentos que cree conveniente en la actividad colaborativa.
- Alumnos:** Pueden trabajar en forma individual o en grupos colaborativos. Los grupos estarán formados hasta 4 integrantes, donde cada alumno tendrá un rol diferenciado.

2. **Objetos a compartir:** bibliografía, material en la web, unidades de aprendizaje.

4. **Herramientas:** email, chat, foros, pizarra, etc.

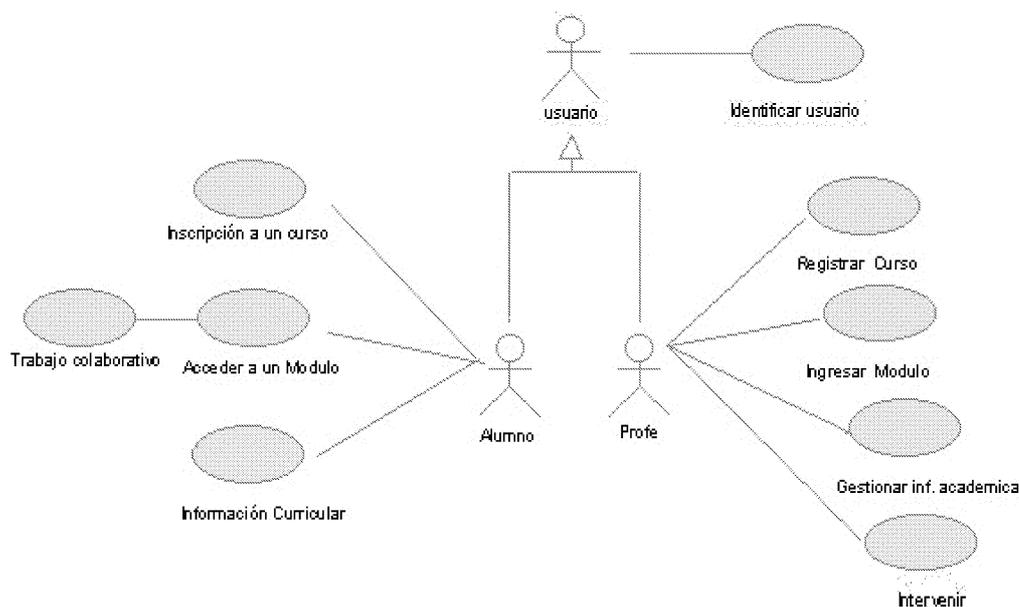
5. **Agentes Inteligentes:** Existirán algunos que acompañan al alumno en su estadía en el ambiente; otros que monitorean su actividad, con el objetivo de recolectar información para la tarea seguimiento.

## Diseño del Ambiente Colaborativo Inteligente

Para el diseño del ambiente utilizamos el UML (Unified Modeling Language). El cual es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos. Un artefacto es una pieza de información que es utilizada o producida mediante un proceso de desarrollo de software. En esta primera instancia sólo presentaremos el modelo del Negocio (según la nomenclatura UML), que permite capturar los tipos de objetos más importantes en el contexto del sistema y describir los procesos existentes. Los procesos del negocio son descriptos en términos de caso de uso sus actores.

### Modelo del Negocio

En el desarrollo del ACI identificamos dos tipos de *actores*, el Profesor y los Alumnos. Las actividades que puede realizar el profesor son las siguientes: Identificarse para ingresar al sistema, registrar curso determinado, ingresar los módulos del curso, gestionar información académica e intervenir en la clase. Mientras que las actividades que puede realizar el alumno incluyen: Identificarse para ingresar al sistema, inscribirse en un curso, acceder a un módulo de un curso (Trabajo Colaborativo), enviar e-mails,



participar de foros y/o chat con sus compañeros de grupo, ingresar actividades resueltas, consultar información curricular

### Casos de uso del modelo del negocio

A continuación se describen las tareas que se realizan en los casos de usos identificados en nuestro modelo.

#### Identificar Usuario

El profesor y/o alumno para ingresar al ambiente debe ingresar su clave y password.

**Registrar Curso**

Profesor ingresa para dar de alta a un curso. Modificar fechas. Dar de baja a un Curso. Solicitar listado de alumnos de un curso. Programa del Curso y correlativas

**Ingresar Módulo**

El profesor ingresa el material de estudios, prácticos, etc.; del módulo de un curso determinado. Devolución de prácticos. Listado de alumnos y e-mails de ese curso. Seguimiento del avance de sus alumnos. Formar equipos colaborativos de alumnos. Publicación y/o modificación de fechas límite para presentación de trabajos prácticos o exámenes parciales. Acceso a la pizarra.

**Gestionar Información académica**

El profesor busca toda la información para un determinado curso. El docente puede realizar las siguientes tareas: poner calificaciones, modificar programas, publicar fechas de exámenes. Consultar el desempeño académico de sus alumnos. Obtención de listado de alumnos por curso.

**Intervenir (Colaborar)**

El profesor interviene en los grupos colaborativos

**Inscripción a un curso**

El alumno debe seleccionar el curso a inscribirse e ingresar los datos que reflejen el perfil del alumno.

**Acceder a un Módulo**

El alumno accede al listado de clases disponibles de un curso determinado. Consultar programas del curso. Puede enviar y recibir e-mails de su profesor y compañeros. Consultar fechas de exámenes. Acceder al pizarrón. Requerir información académica.

**Trabajo Colaborativo**

El equipo colaborativo estará formado por un grupo reducido de alumnos, los mismos podrán ingresar para trabajar en dicha modalidad; leer y/o dejar mensajes en la pizarra; enviar y recibir e-mails; conectarse con sus compañeros y resolver trabajos en forma conjunta y/o simultánea.

**Conclusiones**

Si bien se ha avanzado en el diseño del ambiente, aún se está trabajando junto a especialistas en contenido y pedagogos para la implementación de un caso particular a los efectos de observar su comportamiento y realizar sucesivas correcciones al modelo.

En relación a la incorporación de agentes inteligentes, el grupo se encuentra en la etapa de estudio, la que incluye una revisión del estado del arte en cuanto su uso en este tipo de entornos. Sin embargo, uno de los primeros problemas que se pretenden resolver es la inferencia del perfil (o intereses) de usuario basados en indicadores explícitos e implícitos de manera tal de poder conformar grupos de trabajo nivelados en función de las capacidades y/o intereses de cada uno de los alumnos participantes. Asimismo, se pretende que de la interacción del alumno con el entorno se pueda obtener información suficiente para modelar adecuadamente al usuario con el objetivo de automatizar la tarea de aprendizaje. Dicho modelo del usuario podría permitir al sistema realizar alguna de las siguientes recomendaciones: con quien integrarse, qué documentos debería leer y qué ejercicios debería realizar a fin de entender determinados tópicos, entre otras.

## Bibliografía

- Bigus. J. Y Bigus J. - Constructing Intelligents Agents Using Java. Second Edition. John Wiley & Son Inc. 2001.
- Boticario, J.; Gaudioso, E. y Catalina, C. - *Towards personalised learning communities on the Web. The 5TH International Work-Conference on Artificial and Natural Neural Networks (IWANN'99). Alicante, Spain, June 2-4. 1999. In: IWANN'99 Proceedings (Volume II). Eds. J. Mira and J.V. Sánchez-Andrés. Lecture Notes in Computer Science 1607 (pp. 740-749).*
- Boticario, J.G., Gaudioso E. "Personalización de las Prácticas de Aprendizaje a Través de un Sistema Interactivo en Internet". En: Actas de las Jornadas Nacionales de la Enseñanza de la Informática (Jenui'99). EUPLA (Zaragoza), España,  
-Octubre 25-26. 1999.
- Cabera Almenara, J. y Martínez Sanchez F., *Nuevos canales de comunicación en la enseñanza.* Colección de Enseñanza y Medios. Ed. Centro de estudios Ramón Areces.S.A. Madrid.1995.-
- Claypool, M.; Phong Le, D.; Waseda, M. -*Inferring User Interest.* IEEE Internet Computing. Noviembre, 2001.
- Collins J.; Ndumu, D.; Nwana, H.; Lee, L. - A Tool-Kit for Building Distributed Multi-Agent Systems, In Applied Artificial Intelligence Journal, Vol 13 (1), 1999, p129-186, Link: <http://www.labs.bt.com/projects/agents/aaij-zeus.zip>
- Giraffa, L. (1998) - *The Use of Agents Techniques on Intelligent tutoring System.* RIBIE -98. Brazil.
- Booch G., Rumbaugh J., Jacobson I. - *El Lenguaje Unificado de Modelado* . Addison Wesley. Madrid y otros. 1999
- Mena, Marta: *Nuevos enfoques pedagógicos para mejorar la producción de materiales en la Educación a Distancia.* Journal of Distance Education . Vol. VII No. Pp 121-130.
- Mena Marchan, B. y Marcos Porrás, M.: *Nuevas Tecnologías para la Educación. Didáctica y Metodología.* Ediciones La Torre. 1994
- Johnson, D. y Otros. *El aprendizaje Cooperativo en el Aula.* Ed. Paidós Educador. Bs.As. 1999.
- Silverman, B. (1995): "*Computer Supported Collaborative Learning (CSCL)*", en *Computers and Education*, 25(3), pp. 81-91.
- Learner-centered design.** Communications of the ACM, abril 1996,vol.39, n°4. *Advanced Educational Technology. Researh Issues and Future Potential*, T. Liao (ed.) NATO ASI Series F145, Springer-verlag.1996.

## EXPERIENCIA INTERDISCIPLINAR EN LA PRACTICA DOCENTE DEL PROFESORADO EN COMPUTACION

**P. Camiletti, R.A. Pizarro**

*Departamento de Matemática - Facultad de Ciencias Exactas y Naturales- Universidad Nacional de La Pampa.*

*Uruguay 151 - (6300) - Santa Rosa (LP) - Argentina.*

Tel. 02954-425166 Fax 02954-432679 e-mail: [pablocamiletti@exactas.unlpam.edu.ar](mailto:pablocamiletti@exactas.unlpam.edu.ar) , [ruben@exactas.unlpam.edu.ar](mailto:ruben@exactas.unlpam.edu.ar)

*WICC 2002*

### RESUMEN

En la cátedra Práctica Educativa III del Profesorado en Computación los alumnos de esta carrera desarrollan su práctica docente. En el año 2001, en el que se implementó por primera vez esta cátedra, nos propusimos concretar su desarrollo con algunas modificaciones substanciales.

Hasta el momento en la "Residencia Docente en Computación" del anterior plan, los alumnos se limitaban a observar clases de computación para poder seleccionar el curso en el cual con posterioridad se desempeñarían como docente practicante, desarrollando una unidad temática.

Esta situación se modificó ya que los alumnos de Práctica Educativa III, del nuevo Profesorado en Computación, desarrollaron un proyecto interdisciplinar de carácter integrador que les permitió concretar sus prácticas. En dichas prácticas pudieron contactarse con la realidad de la institución educativa en la cual se desempeñaron como así también interactuar con las autoridades y docentes de los colegios, desarrollando desde su práctica contenidos de distintas disciplinas. Los resultados obtenidos fueron ampliamente satisfactorios ya que tanto los alumnos practicantes, los docentes y autoridades del colegio como los alumnos del mismo se mostraron gratificados por las nuevas alternativas que nos presentó este tipo de trabajo.

### INTRODUCCIÓN

Desde las cátedras Práctica Educativa II y III pertenecientes al cuarto año del profesorado en computación, nos propusimos modificar las características de la "residencia docente" del antiguo plan, para lograr entre otros objetivos que nuestros alumnos: Se vinculen efectivamente con su futuro ámbito laboral; Conozcan la situación de las NTIC y la Educación; Escriban Proyectos y todo esto usando las nuevas tecnologías. De esta forma harán propia una experiencia de formación que querrán ampliar con sus alumnos y los iniciara en el camino de la formación permanente.

Partiendo de experiencias realizadas en la Universidad de Río Cuarto, nos propusimos que los alumnos diseñaran e implementaran en su práctica docente un proyecto interdisciplinar que tendiera a facilitar la incorporación de las tecnologías de la información y la comunicación en el proceso de enseñanza aprendizaje.

Para el diseño del mencionado proyecto, en la cátedra Práctica Educativa II, los alumnos debieron indagar entre los distintos profesores de la institución elegida, para contactarse con los interesados en llevar a cabo un trabajo interdisciplinar entre su asignatura específica y la asignatura relacionada con computación (Tecnología de la Información y la Comunicación, Computación, Taller de Computación, etc.). De esta forma se establecen importantes vínculos del alumno practicante con Institución seleccionada y se comienza a elaborar el Proyecto por medio del cual el practicante podrá desarrollar su práctica docente.

Desde Práctica Educativa III se desarrollaron las observaciones de clases y el análisis Institucional, herramientas que posibilitan formular el proyecto en forma adecuada. Al mismo tiempo se fue escribiendo el plan y cada una de las clases elaborando el material a utilizar. Se mantuvo además un vínculo constante con los docentes de la institución seleccionada, coordinando y consultando con ellos las actividades a implementar. Finalmente se desarrollaron en forma exitosa todos los contenidos planificados, logrando una gran interacción entre las asignaturas del colegio

implicadas en el proyecto e incorporando en el desarrollo del proceso enseñanza aprendizaje diferentes tecnologías que lo favorecieron.

## **DESARROLLO**

### **La práctica docente hasta la implementación del Proyecto Interdisciplinar**

Hasta este año los alumnos del Profesorado en Matemática y Computación realizaban su practica docente como parte del desarrollo de la asignatura anual Residencia Docente en Computación. Luego de realizar las observaciones correspondientes los alumnos seleccionaban un curso y desarrollaban la unidad temática o parte de esta que correspondía a la planificación de la materia en ese momento. En la mayoría de los casos la asignatura relacionada con computación no se desarrollaba en forma interdisciplinar o teniendo en cuenta el contenido de otras asignaturas.

En muchas oportunidades los practicantes y los docentes de computación en general enfrentamos la dificultad de los diferentes niveles de conocimiento entre los alumnos o el desinterés en muchos casos ya que algunos de ellos poseen en sus hogares equipos superiores a los de los colegios. Creemos que el desarrollo de las asignaturas relacionadas con computación desde un punto de vista interdisciplinar puede contribuir a la solución de estos inconvenientes.

### **La práctica docente actual**

En base a la bibliografía consultada y teniendo en cuenta varias guías de observaciones los alumnos de Práctica Educativa III, del profesorado en Computación, comienzan sus actividades elaborando su propia guía de observaciones para esta asignatura. Luego asistieron a las diferentes instituciones educativas en las cuales observaron las clases de computación, además de recabar información necesaria para un primer análisis institucional.

Una vez realizadas las observaciones cada alumno, eligió un colegio e informó a los profesores de la institución la idea de implementar un proyecto en forma interdisciplinar. Si bien la mayoría de los profesores se manifestaban con intensiones de participar en muchos casos también manifestaban su desconocimiento o temor a incluir la computadora en el desarrollo de actividades, por las complicaciones que ello pudiera implicar.

Una vez que el docente del colegio y el alumno practicante coincidieron en los tiempos y el tema a desarrollar, comenzó la elaboración del Proyecto desde Práctica Educativa II, teniendo en cuenta los diferentes aspectos institucionales, las competencias que se quería promover en los alumnos y los contenidos.

Uno de los proyectos, denominados *Representación de datos en Geografía utilizando las NTIC (Nuevas tecnologías de la Información y la Comunicación)*, relacionó las asignaturas "Tecnología de la Información y la Comunicación" y "Geografía" en un curso de Primer año del Polimodal.

La profesora de "Geografía" que debía desarrollar la unidad temática en la cual se estudiaba el MERCOSUR proporcionó la información necesaria, "Tecnología de la Información y la Comunicación" orientó y facilitó el manejo de las herramientas adecuadas para procesar dicha información. Es así que desde esta asignatura se desarrollaron los siguientes contenidos

- Obtención de información de Internet.
- Gráficos en planilla de Calculo ( Excel)
- Edición de Imágenes ( Paint)
- Presentación de la información (Power Point)

Si bien en la asignatura Práctica Educativa II, del primer cuatrimestre, los practicante desarrollaron el proyecto a implementar; desde Práctica Educativa III se comenzó a planificar las diferentes clases. Sus contenidos y actividades se fueron elaborando teniendo en cuenta el material proporcionado por la Profesora de Geografía. Los alumnos del profesorado planificaron cada clase teniendo en cuenta los diferentes factores observados y como estos podían influenciar en el desarrollo de cada una.

El curso en el que se implementó el mencionado proyecto estaba formado por 32 alumnos que podían desarrollar sus actividades en una sala de cómputos con 10 equipos.

En un lapso de tiempo estimado en el proyecto, el alumno practicante presentó actividades a un curso de Polimodal desde el espacio curricular "Tecnología de la Información y la Comunicación". Estas actividades consistieron en la representación de datos poblacionales, económicos y demográficos de los países integrantes del MERCOSUR, utilizando una "Planilla de cálculos". Los datos fueron obtenidos de diversas páginas de Internet, luego por medio de un editor de imágenes ampliaron y crearon referencias en los diferentes mapas de los países integrantes. En forma simultánea durante las clases de geografía, los alumnos trabajaron en torno y a partir de esta misma información. Finalmente se realizó en un diseñador de diapositivas (Power Point) una presentación con animaciones y sonidos en la cual se relacionaron los datos y mapas preparados en "Tecnología de la Información y la Comunicación" con las conclusiones trabajadas en geografía.

La concreción en el aula de este trabajo demandó 13 clases de ochenta minutos cada una, desarrolladas durante los meses de Agosto y Setiembre.

Los alumnos del Polimodal fueron evaluados por su desempeño semanal y finalmente con la presentación realizada en Power Point que expusieron a sus compañeros.

En todas las experiencias de los practicantes se pudo observar claramente, que los alumnos de los cursos en los cuales se realizaron las prácticas presentaban una mayor predisposición a trabajar con información y material que luego utilizaban en otras asignaturas.

## CONCLUSIONES

Luego del desarrollo de las Prácticas Educativas del 4to año de la carrera "profesorado en computación" ha quedado claramente reflejado que:

- los alumnos del profesorado han realizado una experiencia de formación utilizando las tecnologías de la información y la comunicación (TIC).
- los alumnos del profesorado han conocido la situación de las TIC y la Educación además de establecer importantes vínculos en sus futuros ámbitos laborales.
- los alumnos del profesorado generan materiales y experiencias para la posterior planificación y desarrollo de contenidos en forma interdisciplinar en su futuro como docentes, estos materiales se van organizando en un banco de datos.
- los docentes de otras asignaturas se sienten respaldados por el alumno practicante y están dispuestos en su mayoría a participar en los nuevos proyectos
- el desarrollo de temas específicos de computación desde un punto de vista interdisciplinar favorece el proceso "enseñanza y aprendizaje"
- los alumnos de los colegios se muestran más implicados en el desarrollo de las actividades, y se ven claramente atraídos por la posibilidad de manejar imágenes y sonidos

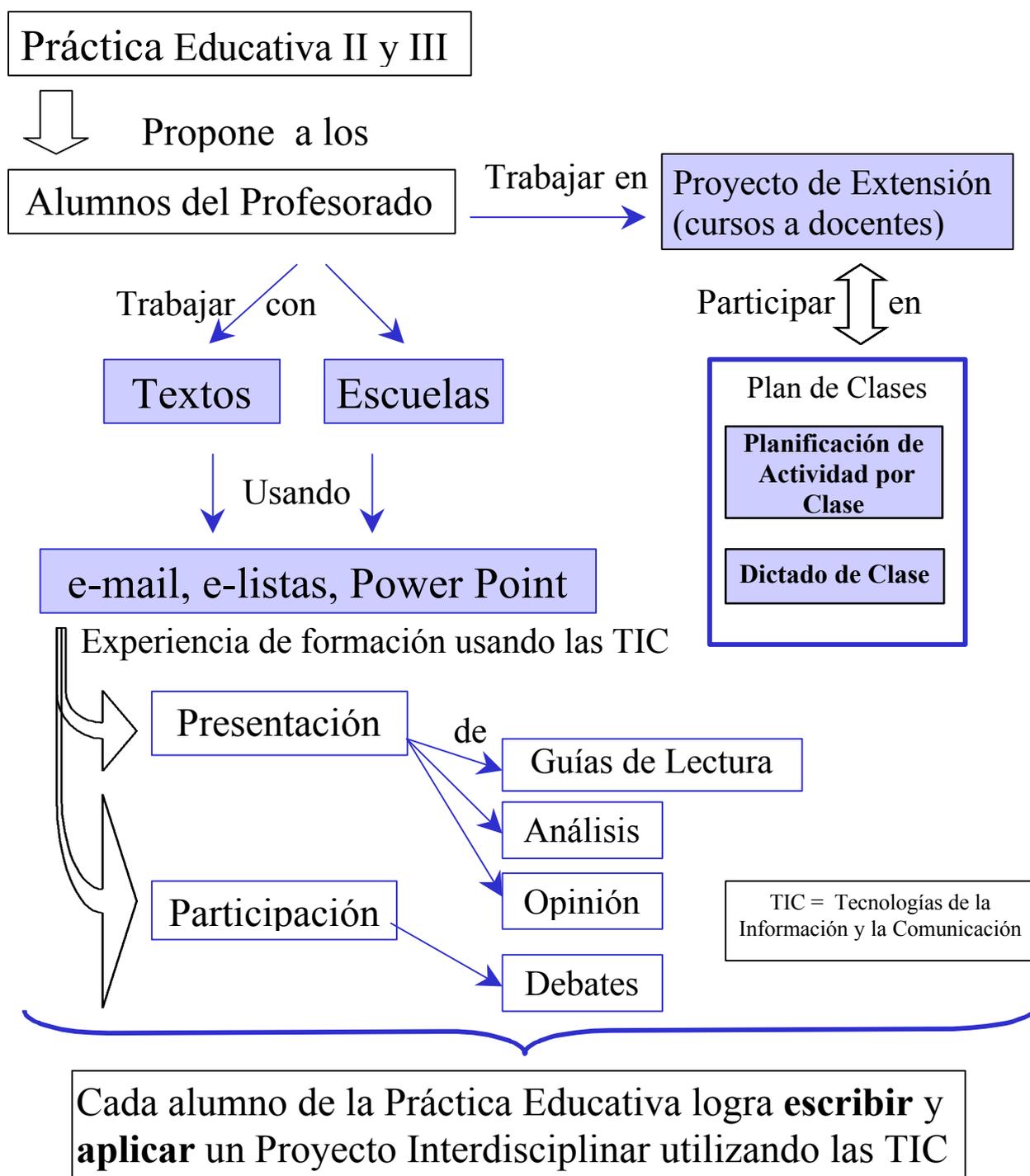
Si bien las instituciones educativas siempre cooperaban con los alumnos practicantes del profesorado, estaba claro que estos últimos aparecían siempre solicitando tiempos y espacio a los colegios pero pocas veces beneficiaban a estas instituciones con la devolución de actividades o algún beneficio concreto. En esta oportunidad los docentes de las instituciones que colaboraron con el trabajo de los practicantes, se han beneficiado participando en un trabajo interdisciplinar apoyado por las tecnologías de la información y la comunicación. En este sentido luego del desarrollo de esta experiencia se ha propuesto para el próximo año la realización de un curso en los colegios seleccionados por los practicantes.

Los nuevos cursos organizados y certificados por la Facultad, servirán para que los futuros practicantes realicen una primera experiencia como capacitadores, en la utilización de las tecnologías en los procesos de enseñanza aprendizaje.

Este curso dirigido a los mismos profesores de la institución en la que harán su práctica docente, favorecerá la anticipación de posibles usos de las tecnologías en actividades específicas de

cada disciplina. Además de incrementar la comunicación entre la Universidad y los demás niveles educativos.

*En el siguiente esquema representamos las actividades propuestas por las cátedras, las desarrolladas por los alumnos y los resultados a los que se arribaron.*



**BIBLIOGRAFÍA**

- 1 Ander-Egg, E. "La Planificación Educativa". Magisterio del Río de La Plata.(1996).
- 2 Medina Rivilla, A. - Sevillano García, M. "Diseño y Desarrollo Curricular". Docencia. (1997).
- 3 Valdéz Velos, H. "Como evaluar", Ponencia presentada por Cuba, Encuentro Iberoamericano sobre Evaluación del Desempeño Docente. ([www.oei.es](http://www.oei.es)) (2000)
- 4 Poole, Bernard J.; "Tecnologías Educativas", Mc Graw Hill, (1999)

## Specifying Quality Characteristics and Attributes for E-Learning Sites

Guillermo Covella; Luis Olsina

GIDIS, Departamento Informática, Facultad de Ingeniería, UNLPam,  
Calle 9 y 110, 6360, General Pico, La Pampa, Argentina

TE./Fax 02302 430497; E-mail [covellag, olsinal]@ing.unlpam.edu.ar

**Abstract.** Education involves a continuous process of exploration, acquisition, organization, integration and delivery of information and knowledge. In recent years, e-learning sites and applications have exponentially started to grow, being a central phenomenon for the industry and education fields. Key factors for the present and future success of these applications and organizations are, on one hand, the fulfillment of technological standards for e-learning environments, and, on the other hand, the utilization of well-defined development processes and the production of required quality. One of our current concerns is the quality assessment of e-learning sites and applications. In this paper, a taxonomy of e-learning sites and applications, and ultimately the quality requirements for the *Functionality* characteristic are thoroughly described, regarding an intentional audience. An evaluation case study for this domain, using the WebQEM (Quality Evaluation Methodology) strategy is being runned.

### 1. Introduction.

By e-learning is meant any form of learning or training (basically both CBT -Computer-based Training, and WBT -Web-based Training), which is partially or totally delivered through an electronic way, either via a Web browser (in an Intranet or Internet), or through multimedia platforms like CD-ROM, DVD or video. Although this definition is all-embracing, e-learning has very often been identified with Web. For the intended audience, the course material and communication services are potentially 24x7, 365 days per year available, and can be synchronously or asynchronously accessed from anyplace around the world.

For the actors involved in the development of face-to-face courses, Web-based teaching and learning resources and processes have implied a substantial change in relation to traditional approaches. This does not only imply the change of formats and means to access materials used in face-to-face courses, but also implies the integral use of resources and services that boost the interaction and access to contents organized in a variety of different media (i.e., text, illustrations, animations, audio and video). The variety of these multimedia and intercommunication resources, together with the permanent availability on the Web, tends to motivate students where no physical interaction among them exists. Also, some other desirable characteristics of Web-based education and learning sites and applications are: accessibility, navigability, on-line searching, on-line evaluation support, collaborative and controlled learning services, etc. [1, 4], in a context of a high degree of independence of time-distance-platform dimensions.

Several international IT consulting reports (IDC, Gartner Group, etc.) have given formidable numbers of investment and world-wide growth projections for the new phenomenon of e-learning. However, this explosive growth has often been poorly systematized from several points of view. On the side of *Learning Technology Systems* (LTSs) there are currently several initiatives of recognized standardization bodies with the purpose of reaching interoperability among systems or components of them (for example, the *IEEE Learning Technology Standards Committee* [5], the *Global IMS Learning Consortium* [6], the *ISO/IEC JTC1 SC36* [8] initiatives, among others). Groups of these bodies have started works on specification to learning objects metadata, course material packaging, required structure for on-line contents, tracking learner progress, student profiles, among other issues. Together with these standardization processes, tools and environments for e-learning development have also been strengthened, as WebCT, ([www.webct.com](http://www.webct.com)); BlackBoard, ([www.blackboard.com](http://www.blackboard.com)), to mention just a few. However, we can raise open research questions such as, are in the standards analyzed learning theories and pedagogical aspects in order to incorporate them in specifications? Or, are there factors such as quality considered in these standards? On the other hand, from the point of view of the use of systematic evaluation processes and methods for quality assurance of e-learning sites and applications, the strategies and practices informed so far have mostly been ad hoc in spite of the rapid growth and increasing complexity of such sites and

applications. It is worthwhile to emphasize the initial efforts made in classifying and reviewing many characteristics of LTS environments and supporting tools, such as those documented in [1, 9], or the comparative analyses reported in [www.c2t2.ca/landonline/](http://www.c2t2.ca/landonline/).

One of our current concerns is the assessment of the quality of e-learning sites and applications [12]. We are engaged in assessing the current e-learning quality (i.e., product quality) in order to give recommendations for improvements. In this ongoing work, we discuss and specify aspects of the product quality requirements to be used in a process of e-learning evaluation, from an intentional visitor standpoint. Requirements for the evaluation of quality in use are not treated here for space reasons. We are performing an evaluation case study in the e-learning domain. For this purpose, the WebQEM quantitative methodology is used [12]. Since in previous performed cases studies for academic and e-commerce domains [10, 11] attributes to *Usability*, *Reliability* and *Efficiency* characteristics were specified, in this article we focus on the *Functionality* characteristic and derived sub-characteristics (such as *Course Features*, *Student Features*, *Virtual Learning Environment Features*, etc.), which are e-learning specific.

The rest of this paper is organized as follows. In Section 2, a detailed taxonomy of sites and applications for the e-learning domain is described. In Section 3, we focus on the analysis of quality requirements for the *E-learning Functionality and Contents* sub-characteristic, being the definition of non-functional requirements, one of the initial and core activities of any evaluation process. Finally, in Section 4, the conclusions and in-progress works are drawn.

## 2. A Taxonomy of Sites and Applications for E-learning.

From a general point of view, *Learning Technology Systems* (LTSs) [2, 3, 4, 5, 6] are managing, training and learning educational systems supported by certain ITs. Classes of these systems are *Computer Based Training Systems* (CBTs), and *Web-based Instructional Systems* (WbIS). Among the main features supported by WbISs are, namely: *Course Management*, *Class Management*, *Communication Tools and Mechanisms*, *Content Management*, *Student Services*, and *Institutional Management*. More specifically, WbISs are LTSs, which use the last Web and Internet technologies in order to offering education and training following the open distance learning paradigm. The WbIS is basically integrated by three components, namely: *Human Agents*, *Web-based Learning Resources*, and the *Technological Infrastructure*. In turn, the main component of the Technological Infrastructure of a WbIS is the *Virtual Learning Environment* (VLE). “A VLE is a middleware that acts and interfaces between the low-level infrastructure of the Internet and the WWW from the one side and the customized domain-specific learning education and training systems on the other side” [2].

From the point of view of sites and applications in the e-learning domain, table 1 details the proposed taxonomy. This classification shows four main categories of WbIS sites and applications, namely: *Portal*, *E-learning Catalogue*, *Tools and Support*, and *E-Learning Functionality* sites. It is important to notice that a specific Web site could implement one or more categories. Particularly, for the last described category (i.e., the *E-Learning Functionality* one), a sub-categorization in *Complementary* and *Indispensable* sites can be established, regarding mainly the amplitude of implemented characteristics in the VLE, in the *Indispensable* subcategory, the courses (independent ones or as part of a program or career), are mainly fulfilled by using the Web. Thus, all student-oriented administrative tasks, from registration to graduation, should be available on line through the Web.

**Table 1.** *A taxonomy of sites and applications in the domain of WbIS.*

Site or Application Category	Description and Examples of URLs (checked by 01-03-02).
Portal	Educative Portal is mainly centered in delivering information rather than functionality. It can contain general information on educational institutions, additional courses and careers; resources like news, e-articles both of research and general interest, e-books, evaluation guidelines for educational tools, an other contents. It can also provide some synchronous and asynchronous communication mechanisms. <a href="http://e-learning.start4all.com/">http://e-learning.start4all.com/</a> ; <a href="http://www.educ.ar/">http://www.educ.ar/</a>
Catalogue	An E-learning Catalogue Site can contain listings of institutions, careers, courses, modules and support material offered by a great variety of instructional organizations. In fact, a catalogue acts as an index of products and services offered on the Web. It could have neither a strict control over the on-line offering nor a responsibility in the editing of contents of linked sites. Catalogues give added value to offers by grouping and categorizing them, so visitors are able to browse and search referenced contents, prices, in addition to

	registration functionality, shopping cart and e-payment capabilities, among other characteristics. <a href="http://www.ccae.org/">http://www.ccae.org/</a>
Tools and Support	E-learning Tools and Support site is a proprietary site that offers administration, authoring and support tools for e-learning environments. It can generally contain e-learning software packages for sale, templates and facilities to develop functionality for virtual classroom, manage contents, etc., in addition to provide online advise and support for their products and services. <a href="http://www.webct.com/">http://www.webct.com/</a>
E-learning Functionality	It is a WbIS that implements partially or totally the e-learning functionality and contents. It provides necessary mechanisms and resources so that members of a learning community interact in a virtual classroom. Specifically, it supports and exploits mechanisms and services that a VLE offers, such as [2]: <i>Managing Courses</i> : it groups characteristics for the creation, customization, and monitoring of courses; <i>Managing Classes</i> : it groups characteristics for user management, workgroups organization, projects assignments, etc.; <i>Managing Contents</i> : it groups characteristics that allow the creation, authoring, management and delivery of contents and files; <i>Providing Communication Mechanisms</i> : it groups characteristics allowing the synchronous and asynchronous communication mechanisms such as chat rooms, e-mail, discussion forums, audio and video conferences, announcement boards, etc.; <i>Providing Student Services</i> : it groups characteristics that give support to students in handling the resources and contents of courses, such as private and public annotations, bookmarks, online searching and retrieving of information, and so on; <i>Providing Assessment Mechanisms</i> : it groups characteristics for the implementation of evaluations, quizzes and tests, self-evaluated assignment, etc.; <i>Providing School-Management</i> : it groups characteristics for managing student registrations, grades, fee issues, etc. <a href="http://www.cvq.edu.ar/">http://www.cvq.edu.ar/</a> ; <a href="http://www.uoc.es">http://www.uoc.es</a>

### 3. Quality Requirements for E-learning Sites and Applications.

One of the aims of this work is to systematically evaluate attributes and characteristics that allow determining the degree of fulfillment of elementary and global quality requirements in the e-learning domain. Particularly, the quality requirements to be used in a process of evaluation of e-learning sites and applications (focused mainly on the *E-learning Functionality* category as shown in table 1) are in detail specified. The analyzed requirement tree serves as input for the implementation of an evaluation case study, using for this end the WebQEM [12] quantitative methodology. It is based on a hierarchical quality requirement model formally composed by *Characteristics*, *sub-characteristics* and *attributes*. At the highest level of the tree, the prescribed ISO 9126-1 [7] characteristics for software product are used. For a given evaluation project, starting from a selection of characteristics, sub-characteristics can be derived; in turn, following a hierarchical decomposition process, attributes can be specified. The methodology can be summarized by a set of five technical steps, namely:

**Step 1. Definition of the evaluation goal and selection of the user profile.** The evaluators should define the goal and scope of the evaluation process. On the other hand, the relative importance of characteristics and attributes depend on the evaluation goal, the selected user profile, and the application domain. For Web evaluation purposes, three user profiles at a high level of abstraction have been considered, namely: visitors, developers, and managers. For the design of the e-learning case study, the intentional visitor profile, an specific visitor class, was considered.

**Step 2. Quality requirements definition.** In this step, the evaluators should define, categorize, and specify the quality characteristics and attributes that are going to be present in the process, grouping them in a requirement tree. From the ISO-based characteristics, sub-characteristics are derived, and, in turn, measurable attributes are specified. For each quantifiable attribute of the empirical domain, a variable in the numeric domain can be associated. This variable can take a real value that can objectively or subjectively measured.

**Step 3. Definition of elementary preference criteria and measurement procedures.** (see [11, 12] for details).

**Step 4. Definition of aggregation structures and implementation of the global evaluation** (ibidem).

**Step 5. Analysis of Results and Recommendations.** Once the evaluation project was designed and implemented, the process finishes with the documentation of the conclusions and recommendations. The evaluators analyze the results considering the established goals, user profile, and criteria. Thus, the strengths and weaknesses of the assessed product can be justified, and recommendations for improvements can be made.

Given that in previous case studies [10, 11], attributes and sub-characteristics for the *Usability*, *Reliability* and *Efficiency* characteristics, in addition to the *Searching*, *Retrieving*, and *Navigation* sub-characteristics (for the *Functionality*) were specified, here we analyze the *Specific Functionality and Contents for E-learning* sub-characteristic, 2.3 coded, as shown in Fig. 1. Next, five categories for the 2.3 sub-characteristic are described:

**Course Features:** this category allows to offer visitors general information about the course. Thus, it is composed by sub-characteristics like *General Information* (2.3.1.1 coded) -the student should have information about the course contents, demos, hands-on availability, modality (whether it is totally online or not), class scheduling, instructors, and assessment information, amongst other attributes; *Requirements Information* (2.3.1.2) -if there exists information on academic requirements for admission as well as technical requirements in order to participate in proposed activities; and *Application Information* (2.3.1.3).

***Student Features:*** this sub-characteristic should reflect the availability of on-line mechanisms for registration and enrolment management. Specifically, it should be provided *On-line Student Account/File* (2.3.2.1) feature in order to add or update personal data, preferably under secure conditions of navigation, in addition to *On-line Paperwork* (2.3.2.2) attributes for the career and/or courses.

***Virtual Learning Environment Features:*** the availability of mechanisms and resources to participate in classes by the Web in order to access and use instructional materials, to interact and collaboratively learn in the virtual classroom, and to fulfill the assessment process, are here considered. The *Virtual Learning Environment Features* includes the following sub-characteristics: *Synchronous Communication and Collaboration Resources* (2.3.3.1), *Asynchronous Communication and Collaboration Resources* (2.3.3.2), and *Assessment Mechanisms* (2.3.3.3). In the *Synchronous Communication and Collaboration Resources*, we find attributes that represent services and resources as for example the real-time student participation in the classroom, the visualization of actions that instructors are carrying out so that students and instructors can interact each other, and resources such as chat sessions, audio/video conferences, etc. In the *Asynchronous Communication and Collaboration Resources* sub-characteristic, mechanisms and resources to complete the learning process with no need of staying neither synchronized nor permanently online are grouped. These include participation mechanisms such as messaging (e-mail, discussion lists, etc.), and self-learning resources such as posted material and software, e.g., previously recorded classes, class related links, simulators, amongst others. Finally, in the *Assessment Mechanisms* sub-characteristic, attributes for evaluation of self-assessment, controlled assessment, scoring, and grading mechanisms are grouped.

***Student Services:*** it is a sub-characteristic that groups services, resources and mechanisms that promote and contribute the permanence of student visitors in the virtual campus. It includes three sub-characteristics: *Virtual Community* (2.3.4.1), which is a service intended to recreate a communitarian life within virtual campus, mainly through bulletin board, forums, etc.; *Help and Support* (2.3.4.2), which allows to evaluate if there are mechanisms for helping visitors, either by means of information on how to interpret icons or specific functionalities, or by means of the well-known FAQ attribute. Finally, in *Other Services* (2.3.4.3), the availability of a *Virtual Library*, *Web Mail*, among others, are taken into consideration.

***Privacy, Certification and Guarantee Policies:*** in this category the existence of pages with specific information is evaluated, particularly information about *Privacy Policy* (2.3.5.1.), *Certification Policy* (2.3.5.2.), i.e., the certificate scope and validity to be granted, and *Guarantee Policy* (2.3.5.3), including information of *Terms of Use*, *Return Policy*, etc. Generally, these attributes convey and warn in advance about the scope and limitations of the contractual relationship between the provider and the user.

### 2.3 Specific Functionality and Contents for E-learning

#### 2.3.1. Course Features

- 2.3.1.1 General Information
- 2.3.1.2 Requirements Information
- 2.3.1.3 Application Information

#### 2.3.2. Student Features

- 2.3.2.1 On-line Student Account/File
- 2.3.2.2 Online Paperwork

#### 2.3.3. Virtual Learning Environment Features (for a VbIS)

- 2.3.3.1 Synchronous Communication and Collaboration Resources
  - 2.3.3.1.1 Class Session Palette
  - 2.3.3.1.2 Participation/Collaboration Mechanisms
    - 2.3.3.1.2.1 Hand raising
    - 2.3.3.1.2.2 Class Member List
    - 2.3.3.1.2.3 Synchronized Web Navigation
    - 2.3.3.1.2.4 Assistant Instructor Participation
    - 2.3.3.1.2.5 Workgroup Facility
  - 2.3.3.1.3 Synchronous Resources.
    - 2.3.3.1.3.1 Streaming Slides
    - 2.3.3.1.3.2 Streaming Video
    - 2.3.3.1.3.3 Shared Applications (e.g., Whiteboard)
    - 2.3.3.1.3.4 Audio/Video Conference
    - 2.3.3.1.3.5 Chat
  - 2.3.3.1.4 Virtual Lab
- 2.3.3.2 Asynchronous Communication and Collaboration Resources

#### 2.3.3.2.1 Participation Mechanisms.

- 2.3.3.2.1.1 Messaging (e-mail, discussion lists, Message Board)
- 2.3.3.2.1.2 Agenda Facility

#### 2.3.3.2.2 Asynchronous Resources

- 2.3.3.2.2.1 Class Playing (already recorded)
- 2.3.3.2.2.2 Case Studies
- 2.3.3.2.2.3 Class Related Links
- 2.3.3.2.2.4 Simulators
- 2.3.3.2.2.5 Personalized Resources

#### 2.3.3.2.2.5.1 Documents Repository

#### 2.3.3.2.2.5.2 Recommendations.(e.g., people, related courses)

#### 2.3.3.3 Assessment Mechanisms

- 2.3.3.3.1 Self-assessment
- 2.3.3.3.2 Proctored/Supervised Assessment
- 2.3.3.3.3 Delivery/Return Mechanisms
- 2.3.3.3.4 Scoring Mechanisms
- 2.3.3.3.5 Grading Mechanisms

#### 2.3.4. Student Services

- 2.3.4.1 Virtual Community
- 2.3.4.2 Help and Support
- 2.3.4.3 Other Services.

#### 2.3.5 Privacy, Certification and Guarantee Policies

- 2.3.5.1 Information on Privacy Policy
- 2.3.5.2 Information on Certification Policy
- 2.3.5.3 Guarantee Policy

**Figure 1.** The Specific Functionality and Contents sub-characteristic for Web sites and applications in the E-learning Functionality category, with the Virtual Learning Environment subcategory expanded.

Finally, in order to give an attribute example, for the *Synchronous Communication and Collaboration Resources* sub-characteristic, the *Class Session Palette* (2.3.3.1.1) attribute and assessment criteria are summarized. This attribute

is an on-line class session panel whereupon students (and instructors) can activate it in order to participate in the virtual classroom. From the student point of view, the main synchronous collaboration and interaction tools as well status fields are grouped in a panel, such as handraising, class list, and feedback progress, amongst others. This attribute can be assessed by a multi-level discrete criterion defined as a subset, where 0 implies no such mechanism available; 1 implies a basic mechanism (where only some tools are just available in the panel, accomplishing 60% of the requirement); and 2 implies a fully functional mechanism (where all synchronous tools are available in the panel, accomplishing 100% of the requirement).

#### 4. Conclusions and Future Works.

In recent years, the building and employment of e-learning sites and applications have been an increasing and relevant phenomenon for training, teaching and learning activities in different fields. A key for the present and future success of these applications and organizations is the use of quality approaches for the generation and assessment of these sites and applications. In this direction, one of our current line of work is the quality evaluation of e-learning sites and applications from the product quality and quality in use perspectives.

In the preliminary requirements elicitation, techniques such as undergraduate student interviews, enrolled in virtual Argentinean universities, and a non-intrusive task observation protocol were used.

In the other side, it is important to remark that in the requirements elicitation for sites and applications with e-learning functionality, neither content quality of the instructional material nor pedagogic aspects have been considered in the evaluation goals of the case study in order to diminish the assessment complexity. With regard to the discussed sub-characteristic (2.3 coded, in Fig. 1), each sub-characteristic is composed of a set of attributes, directly or indirectly quantifiable according to a set of absolute evaluation criteria of discrete or continuous variable [14]. Therefore, by means of the systematic use of the WebQEM methodology and its supporting tool, the degree of fulfillment of elementary, partial and global quality requirements can be determined.

Currently, we are running an evaluation and comparison case study to determine the quality of four typical sites and applications with e-learning functionality, so we are using the requirements described in this paper, and the ones given in previous works. This study will allow us to corroborate or refuse the following hypothesis: *“that generally, in typical sites and applications with e-learning functionality, the global quality satisfies the quality requirements for a given user’s profile. Particularly, that each site satisfies, at least, the critical point of acceptability of a 60% of the global preference, according to the quality requirements specified for an intentional visitor”*

#### References.

- [1]. Avgeriou, P. et al.2001, "*Learning Technology Systems: issues, trends, challenges*"; Proceedings of 1<sup>st</sup> IOSTE Symposium in Southern Europe, Science and Technology Education, Cyprus.
- [2]. Avgeriou, P.; Retalis S.; Skordalakis, M.; Psaromiligos Y., 2001, "*A Layered Component-Based Architecture of a Virtual Learning Environment*"; Paper to be published.
- [3]. Beer, V.; 2000, *The Web Learning Fieldbook, Using the WWW to Build WLEs*, Jossey-Bass Pfeiffer.
- [4]. Berge, Z.; Collins, M. y Dougherty, K.; 2000, "*Design Guidelines for Web Based Courses*" Instructional and Cognitive Impacts of Web-Based Education; Idea Group Publishing.
- [5]. IEEE; *Draft Standard for Learning Technology Systems Architecture (LTSA)*. Draft-8-Apr-2001.
- [6]. IMS Global Learning Consortium, Inc, 2001, Specifications, [www.imsproject.com/specifications.html](http://www.imsproject.com/specifications.html)
- [7]. ISO/IEC; 9126-1: Software Engineering - Product Quality - Part 1: Quality Model; 2001
- [8]. ISO/IEC JTC1 SC36; Standards for: Information Technology for Learning, Education, and Training; 2001.
- [9]. Mamoukaris, K.; Economides, A., 1999, "*Assessment and Comparison of Web-Based Educational Environments*", Proceedings of IV Int'l WebNet Conference, Hawaii, USA, Vol.1, 677-682.
- [10]. Olsina, L.; Lafuente, G.; Rossi, G.; 2000, "*E-commerce Site Evaluation: a Case Study*"; LNCS 1875 of Springer, 1st International Conference on Electronic Commerce and Web Technologies, EC-Web 2000, London, UK, pp. 239-252.
- [11]. Olsina L., Rossi, G.; 2001, "*A Quantitative Method for Quality Evaluation of Web Sites and Applications*", To appear in IEEE Multimedia Magazine.
- [12]. Covella G.J., Olsina L.; 2002, "*Evaluación de Calidad de Sitios Web con Funcionalidad E-Learning*", Proceedings of 5th Iberoamerican Workshop Requirements Engineering and Software Environments IDEAS'2002, La Habana, Cuba.

**Acknowledgments.** This research is supported by the UNLPam-09/F022 research project.

## ***Tecnología informática en Software Educativo: Herramientas para el diseño, administración y evaluación de cursos no presenciales***

**De Giusti A<sup>1</sup>., Sanz C<sup>1</sup>., Madoz C<sup>1</sup>., Gorga G<sup>1</sup>., Champredonde R<sup>1</sup>., Pesado P.<sup>1</sup>, Pasini A.<sup>1</sup>, Barbieri A.<sup>1</sup>, Rodriguez I.<sup>1</sup>, Feierherd G.<sup>2</sup>, Depetris B.<sup>2</sup>**

*LIDI. – Facultad de Informática – Universidad Nacional de La Plata  
Dpto Computación – Facultad de Ingeniería- Sede Ushuaia -Univ. Nacional San Juan Bosco*

### **Palabras Claves**

Tecnología Informática aplicada en Educación. Software educativo. Aprendizaje centrado en la WEB. Educación a distancia. Herramientas de diseño y administración de cursos. Evaluación.

### **Resumen**

Investigar metodologías para la transformación de cursos presenciales en cursos semi presenciales y/o no presenciales, en diferentes modalidades (según el recurso tecnológico utilizado). Esta investigación debe contemplar los aspectos pedagógicos y la factibilidad técnica y económica de las diferentes soluciones.

Investigar la problemática de la evaluación en cursos no presenciales, analizando diferentes alternativas para la evaluación continua del alumno no presencial.

Desarrollar herramientas específicas para la educación no presencial, en particular para Educación a Distancia basada en WEB.

Investigar los aspectos a considerar en la evaluación de software educativo, y en líneas generales de la tecnología informática que se incorpora en el aula. Aplicar estos criterios como ejes para la evaluación de las herramientas desarrolladas en el proyecto, de modo de perfeccionarlo.

***Esta línea de investigación se realiza en conjunto entre el LIDI (Informática-UNLP) y la Sede Ushuaia de la Facultad de Ingeniería de la UNPSJB.***

<sup>1</sup> **Docente-Investigador del LIDI. Facultad de Informática. Universidad Nacional de La Plata. E-Mail [csanz@lidi.info.unlp.edu.ar](mailto:csanz@lidi.info.unlp.edu.ar)**

<sup>2</sup> **Docente-Investigador de la Sede Ushuaia de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco. E-Mail [feierherdge@ciudad.com.ar](mailto:feierherdge@ciudad.com.ar)**

## Breve Introducción

La Informática en general, y los recursos tecnológicos como los sistemas distribuidos, las redes y la multimedia, ofrecen alternativas y herramientas para favorecer los procesos de aprendizaje, autoaprendizaje, aprendizaje a distancia con guía docente y aprendizaje cooperativo a distancia.

Uno de los factores claves en esta concepción es la identificación explícita del "profesor como el agente del cambio crítico" para posibilitar una transferencia de tecnología desde los tradicionales materiales basados en la impresión a otros medios tecnológicos.

Ser un usuario de tecnologías no significa necesariamente ser un conocedor de las mismas. Ese desconocimiento permite hablar de un analfabetismo tecnológico. La educación en tecnología debería diseñarse para que una buena parte de la población comprenda el impacto de la tecnología en la sociedad y en el ambiente, sus relaciones con la ciencia y las actividades productivas.

Numerosas experiencias destacan el potencial educativo, tanto en cuanto a la posibilidad de facilitar los procesos de aprendizaje, adecuándolos a las necesidades y a las habilidades individuales, como para servir, en particular, a la evaluación y autoevaluación de aptitudes e intereses por la disciplina Informática.

A su vez la aplicación de estas herramientas ha producido un aumento en la utilización de software educativo no presencial que requiere una cuidadosa evaluación metodológica y de implementación concreta sobre diferentes poblaciones (alumnos de escuela primaria o secundaria, alumnos de grado universitario, alumnos de postgrado, usuarios calificados, etc).

El diseño de metodologías y el desarrollo de herramientas que permitan incorporar la Informática en el proceso de enseñanza - aprendizaje se han constituido entonces en temas de investigación y aplicación permanente dentro de las Ciencias de la Educación y de la Computación.

Este proyecto trata de dar respuesta al desarrollo concreto de experiencias educativas utilizando tecnología informática, para lo cual se propone trabajar en forma multidisciplinaria en la definición y diseño de un ambiente para el desarrollo y administración de cursos semi-presenciales y no presenciales. Posteriormente, se utilizarán los resultados de la línea de I/D en métricas para software educativo con el fin de corregir y optimizar la herramienta.

## Temas de Investigación y desarrollo

- Analizar y comparar metodologías en Educación a Distancia. Estudiar las técnicas de transformación de cursos presenciales en semi presenciales y/o no presenciales. Analizar los casos específicos según el tipo de población a la que pertenece el alumno y la temática y metodología del curso. Plantear pautas metodológicas para la transformación de cursos presenciales en semi presenciales y/o no presenciales sobre diferentes soportes tecnológicos.
- Analizar y comparar técnicas de evaluación en Educación no presencial. Investigar la incorporación de tecnología automática y semiautomática en evaluación no presencial basada en WEB. Plantear pautas de evaluación para alumnos de cursos no presenciales, analizando el seguimiento y evaluación continua de los mismos.
- Estudiar las métricas orientadas a productos de software educativos aplicados en Educación a Distancia.
- Estudiar diferentes modelos de interfaz hombre-máquina y su impacto en la percepción y el aprendizaje, según la edad y conocimientos de los aprendientes. Analizar el impacto de la interactividad y los modos de generar acciones interactivas concretas a través de software educativo.

- Diseñar experiencias con alumnos de diferentes niveles, en el ámbito de la UNLP y la UNPSJB y evaluar la utilización de las tecnologías estudiadas.

### Productos concretos a desarrollar y resultados esperados

1. Extender el lenguaje Visual Da Vinci, de modo de permitir su instalación y ejecución como una aplicación dentro de un sitio WEB o desde una herramienta de educación a distancia permitiendo su ejecución interactiva desde sitios remotos.
2. Desarrollar ambientes/lenguajes orientados a no informáticos para el diseño de cursos centrados en la WEB. El modelo puede ser un lenguaje tipo Toolbook.
3. Desarrollar un sistema (WEB-LIDI) para el diseño, administración y evaluación de cursos a distancia centrados en WEB, cuyas funcionalidades principales son:
  - Administración de Usuarios (alumnos, profesores, administradores, etc.).
  - Administración de Cursos.
  - Administración de Evaluaciones.
  - Administración de Archivos.
  - Herramientas de comunicación.
  - Herramientas de Estudio

A estas funcionalidades básicas se agregarían las siguientes funcionalidades:

- Administración de Carreras y Correlatividades.
- Coordinación y / o comunicación entre profesores y / o alumnos.
- Biblioteca de recursos, temas de interés, etc.
- Segmentación de usuarios y autorizaciones automáticas
- Evaluaciones automáticas y semi-automáticas.

### Bibliografía Básica

[Ble 86] Blease, Evaluating Educational Software. Londres.

[Cat 99] Zulma Cataldi, Una metodología para el diseño, desarrollo y evaluación de Software Educativo. Trabajo de Tesis del Magister en Automatización de Oficinas, UNLP 2000.

[Dor 98] <http://www.xtec.es/~pmarques/eva2.htm>

[Mad 01] Madoz C., Gorga G., Hacia una propuesta de métricas para la evaluación de Software Educativo, Workshop de Investigadores de Ciencias de la Computación (WICC 2001). San Luis. Mayo, 2001

[Mal 01] Malbrán, M. y Villar, C. , "Las nuevas Tecnologías en el Nivel de Posgrado. Un estudio de caso." Ponencia presentada en el International Council on Education for Teaching. 46ta. Asamblea Mundial. Santiago de Chile. 2001

[Mar 95] Marques, P. Metodología para la elaboración del software educativo en Software Educativo. Guía de uso y Metodología de diseño, Barcelona 1995.

[Mar 98] Marques, P., La evaluación de programas didácticos. Comunicación y Pedagogía, Barcelona, 1998.

[Mar 98] Marques, P., Programas didácticos: diseño y evaluación, Universidad Autónoma de Barcelona, 1998.

[Tar 99] Tarouco Liane, Educación a Distancia, Magister de Tecnología en Educación, UNLP, 1999.

IEEE Transactions on Education- 2001-2002

<http://www.webct.com>

**www.lea.com.ar: “Un portal de accesibilidad para disminuidos visuales y ciegos”**Prof. Javier Díaz ([jdiaz@info.unlp.edu.ar](mailto:jdiaz@info.unlp.edu.ar))Lic. Ivana Harari ([iharari@info.unlp.edu.ar](mailto:iharari@info.unlp.edu.ar))Sr. José Ferreyra ([jferreyra@info.unlp.edu.ar](mailto:jferreyra@info.unlp.edu.ar))

L.I.N.T.I. “Laboratorio de Investigación de Nuevas Tecnologías Informáticas” - U.N.L.P.

Calle 50 y 115 – 1° Piso – La Plata – Bs. As. – Argentina – Tel.: (0221) 422-3528

**Introducción**

El desarrollo y avance tecnológico exigen cada vez mayor preparación por parte de los hombres. Si las personas con algún impedimento tienen dificultades para acceder a recursos que satisfagan sus necesidades básicas, con el advenimiento y expansión de nuevas tecnologías, se les abre una laguna inconmensurable entre ellas y el mundo que les rodea.

Las discapacidades físicas siempre han sido un impedimento en un mundo diseñado por, y para personas “normales”. Las arquitecturas [Ref.1], no sólo las urbanas, sino que también las de distintos elementos cotidianos a los que se ve enfrentado una persona con capacidades especiales, no son la excepción. Los artefactos y utensilios cotidianos no están pensados para un uso universal sino que para una parte de la sociedad que lo compone.

Internet, como nuevo medio de comunicación y reunión virtual, no está al margen de estas críticas. Por medio de la red mundial no sólo podemos comunicarnos, sino también hacer infinidad de actividades que mejoran nuestra calidad de vida. Pero estos logros que parecen acercar y brindar nuevas posibilidades al mundo, han originado una nueva barrera para los discapacitados, en particular a los minusválidos visuales. Los individuos pertenecientes a este sector de la comunidad, se ven imposibilitados de aprovechar esta nueva herramienta informática, excluyéndolos cada vez más.

Si bien existen iniciativas por parte de diferentes consorcios y de organizaciones internacionales [Ref.2], en el sentido de establecer documentos que agrupan un conjunto de reglas y normas de accesibilidad, sus esfuerzos suelen permanecer ocultos en los propios entornos que los generan, suelen tener dificultades de difusión, sufren cambios con bastante frecuencia y resultan difíciles de conseguir. Esto conlleva a que la gran parte de la comunidad informática dedicada al desarrollo y programación de sitios Web, no tome conciencia de la necesidad de aplicar dichas normas.

Por otro lado, muchos de los productos y herramientas desarrolladas por distintas empresas dedicadas a la informática, [Ref.3] las cuáles intentan integrar al discapacitado, presentan problemas de integración entre ellas, ya sea porque son de distintas empresas o porque las distintas versiones de un mismo producto son incompatibles, o porque su metáfora de trabajo varía de una versión a otra. Además, dichas aplicaciones muchas veces no tienen en cuenta las distintas reglas de accesibilidad.

Como solución a esta problemática, se ha diseñado e implementado un portal de accesibilidad a disminuidos visuales y ciegos, el cuál tiene como finalidad transformar y adecuar una página Web solicitada, basándose en un conjunto de normas publicadas por las organizaciones y consorcios más destacados en el tema. Además, es una herramienta integradora de distintas aplicaciones de soporte al discapacitado visual, y brinda una solución conjunta al problema de la accesibilidad.

Por medio de este artículo, se presentarán las características más sobresalientes del portal, qué normativas se implementaron y cómo ayuda al minusválido visual en el funcionamiento y utilización de Internet. Por último, se expondrán como conclusión una serie de comentarios sobre el producto, haciendo hincapié en las actividades a seguir en el futuro.

## La problemática que presenta Internet

En la sociedad de la información en que nos hallamos inmersos, todas las personas deberían tener la posibilidad de acceder a los recursos que nos ofrece Internet, sin que los usuarios con discapacidad sean una excepción. Para ellos, este medio resulta muy interesante ya que permite el acceso a diferentes recursos como catálogos o bases de datos, tanto desde su propio domicilio como desde bibliotecas o centros de documentación. Esta red mundial de información sin fronteras será el medio de comunicación más importante del siglo XXI, revolucionando el área de las comunicaciones, y permitiendo un acercamiento cada vez más estrecho de diferentes culturas y razas.

Estos logros alcanzados por Internet que parecen acercar y brindar nuevas posibilidades a todo el mundo, ha originado una nueva barrera para los discapacitados, en particular a los minusválidos visuales [Ref.4]. Esto se debe principalmente a su naturaleza multimedial, y a su representación visual de la información, con páginas que incluyen animaciones, videos, sonidos, elementos de control, frames, tablas, texto parpadeante, imágenes de fondo, y otros componentes, los cuáles hacen menos accesibles la página.

Este y otros motivos llevaron a profesionales y expertos de todo el mundo a la creación de consorcios y organizaciones internacionales, la mayoría en EEUU y Canadá, las cuáles se ocupan de establecer y publicar normas de accesibilidad. El más importante es W3C (World Wide Web Consortium) en el que se encuentra la WAI (Web Accessibility Initiative) que es una organización internacional que orienta y estructura el desarrollo global de la WWW centrándose en el desarrollo tecnológico. La WAI ha elaborado la *Guía para la accesibilidad y la autoría de páginas* que contiene recomendaciones y pautas para la creación de páginas. Además de estos, también trabajan en este campo: EASI (Equal Access to Software and Information), WeABLE, DO-IT, NCAM (National Center for Accesible Media), ATRC (Adaptive Technology Resource Centre de la Universidad de Toronto), Include (Finlandia dependiente de la Unión Europea), Microsoft Enable, Starling Access Services (Canadá), entre otros.

Estos organismos, con el consorcio WAI a la cabeza, trabajan principalmente orientando a los creadores de páginas Web en la utilización de herramientas de diseño y de validación de sitios Web, como así también brindando pautas de diseño accesibles y promoviendo el uso de elementos estándares.

Pero la existencia de estas normas no es ninguna garantía de que los fabricantes de hardware y software la vayan a seguir. La toma de conciencia y adiestramiento de los desarrolladores de sitios y páginas Web de todo el mundo resulta imposible, es por eso que se debe plantear una solución al desfase entre las normas de accesibilidad y los diseñadores.

Nuestra inquietud está centrada en brindar una solución a la gran demanda de personas discapacitadas, por acceder e incorporarse al mundo virtual que ofrece Internet.

## El objetivo del portal [www.lea.com.ar](http://www.lea.com.ar)

“[www.lea.com.ar](http://www.lea.com.ar)” es un portal de accesibilidad para discapacitados, enfocando su funcionalidad en las problemáticas que presentan los minusválidos visuales y ciegos al momento de interactuar con el contenido de las páginas Web. Es preciso destacar que también, debido a la naturaleza de los distintos tipos de discapacidades, mejora la funcionalidad con respecto a usuarios con otros tipos de minusvalías, como por ejemplo los discapacitados motrices, psíquicos, entre otros.

El objetivo primordial de este sitio es el de poder adaptar cualquier tipo de página de Internet, y traducirla en una página accesible para un usuario no vidente o disminuido visual, de acuerdo a su complejidad. Y por medio de dicha transformación construirles, “una puerta de entrada

a Internet” y a su mundo de información. Este mecanismo de adaptación, que se ejecuta en el servidor, involucra dos pasos de transformación:

- **El Proceso de Normalización:** que consiste en la conversión de la página que el usuario desea acceder a una página bien diseñada, que sea simple de utilizar, y que respete las normativas de accesibilidad estándares establecidas por los organismos y consorcios antes mencionados.
- **El Proceso de Personalización:** que consiste en la conversión de la página ya normalizada a una página adaptada al tipo de afeción visual detectada, como así también al tipo de hardware y software que dispone.

Para cumplir con este objetivo, el portal debe ser capaz de captar la problemática visual del usuario y resolver a partir de allí, el esquema de adaptación adecuado para él. Esto lo beneficiará y asistirá en su proceso de lectura, navegación, búsqueda en Internet y demás actividades.

Para esto, y teniendo como referencia a las normas de accesibilidad, se tuvo en cuenta primeramente aquellas reglas que involucran de manera directa el buen desempeño de un usuario discapacitado visual. Cabe aclarar que continuamente se están haciendo un relevamiento y actualización de estos estándares, con el fin de lograr un resultado más óptimo de la herramienta, incorporando a la funcionalidad no sólo elementos de ayuda a los minusválidos visuales, sino también abriendo el horizonte a todos los usuarios discapacitados.

### **Las normas empleadas para la normalización de páginas**

Al momento de implementar el proceso de adaptación del portal, se realizó una recolección, análisis y selección de reglas de accesibilidad [Ref.5]. Se hizo énfasis en aquellas normas cuya aplicación automática resultara posible, por lo que el proceso normalizador está basado en las siguientes reglas:

#### 1.- Referentes al diseño de página:

- Evitar fondos (background) en mosaicos llamativos, ya que el texto puede oscurecerse, imposibilitando al usuario de una lectura clara. De igual manera se deben evitar colores oscuros o brillantes de fondo, modificando el atributo de color de fondo, si es que se comprueba que es inapropiado.
- Ofrecer un alto grado de contraste entre el texto y el fondo, esto es importante en la elección de colores. De esta manera se ofrece una buena legibilidad a personas con limitaciones en la percepción del color o en caso de usuarios que trabajen con monitores en blanco y negro. El usuario puede elegir el color del texto (ya sea títulos o texto normal), de los hipervínculos y del fondo de la página, o en su defecto elegir configuraciones predefinidas establecidas por el portal, las cuales son optimas en cuando a contraste de colores se refiere.
- Eliminar emoticones (caritas sonrientes) u otros dibujos que provee el texto ASCII, ya que son de difícil identificación por parte de los disminuidos visuales, y además no pueden ser interpretados de manera correcta por los sintetizadores de voz o lectores de pantalla. De la misma forma se deben evitar abreviaturas y/o palabras claves, o formatos de fecha xx/yy/zz, llevándolo la forma extendida de la misma (28 de Abril de 2002).
- Usar listas verticales de hipervínculos y links y no que todos estén en una misma línea.

#### 2.- Referentes a HTML:

- Evitar texto parpadeante o con algún otro efecto visual molesto, ya que estos dificultan leer texto. Por el mismo motivo es conveniente traducir el texto con efecto marquesina en texto plano.

### 3.- Referentes a HTML avanzado:

- Hacer que los hipervínculos sea accesibles para personas con discapacidad visual mediante el desplazamiento con la tecla Tab. De igual forma se colocan espacios separando cada hipervínculo.
- Traducir, siempre que sea posible, las tablas colocando la información en un formato más lineal y jerárquico.
- Referente a las listas, se rotulan numéricamente los elementos de la lista para facilitar su ubicación, también en el caso de las listas anidadas, siendo la forma jerárquica (1, 1.1, 1.2, 1.2.1.) la mas aconsejada. También se Incluye un salto de línea entre los elementos de la lista.

Si bien la aplicación de estas reglas se puede implementar mediante un proceso automático, existe una gran cantidad de normas que no pueden ser aplicadas inmediatamente, sino que deben ser debidamente estudiadas antes de ser empleadas, ya que su mala utilización, puede crear un efecto contrario al deseado.

Si la página original presenta defectos en su etapa de diseño, o en la parte de codificación avanzada de HTML, estos son más difíciles de normalizar. Téngase en cuenta, por ejemplo, la norma que dice: <<No utilizar expresiones como “Clickear acá” o “versión completa” o “entrar”>>, en este caso si al momento de la traducción, el portal toma una decisión inapropiada, lo único que se logrará será oscurecer el documento original, llevando sin lugar a dudas al usuario al desconcierto.

Por esto el proceso de traducción es la parte más crítica del portal de accesibilidad, y por lo tanto el que continuamente debe ser mejorado y analizado para un buen funcionamiento de la herramienta.

### Los servicios del portal

“*www.lea.com.ar*” incorpora una serie de servicios especiales, los cuáles están orientados al manejo simplificado del portal por parte de los minusválidos visuales. Si bien existen en el mercado diferentes herramientas como sintetizadores de voz, magnificadores de pantalla, controladores de teclados y mouse, navegadores especiales, etc., que sirven como apoyo al momento de interactuar con páginas de Internet u otras aplicaciones, dichas herramientas generalmente llevan un gran costo de instalación previa. En la mayoría de las veces son incompatibles entre sí, o su funcionalidad difiere entre ellas, exigiendo un gran entrenamiento por parte del usuario, para poder aplicarlas con eficacia.

Por otra parte, muchas de estas aplicaciones no cuentan con tutoriales, su ayuda es escasa o están en un idioma desconocido por el usuario. Es por ello que como proveedores de un portal de accesibilidad coherente, se incorporó de manera automática un sintetizador de voz, y un magnificador de pantalla (para le texto lineal), los cuáles no necesitan de instalaciones costosas, ni tampoco del apoyo de programas externos, sino que todo el manejo de los mismos se realiza directamente sobre el portal y utilizando el teclado como entrada estándar. Además, provee una ayuda verbal de las mismas en idioma castellano y de manera natural. Esta versatilidad lograda hace que la utilización del portal sea más natural que los dispositivos de soporte para disminuidos visuales antes comentados.

Otros servicios que ofrece el portal de accesibilidad son los siguientes:

- **Registración del usuario:** Tanto de sus datos personales, como así también de su afección visual. Se realiza también pruebas referente al tamaño de letra, colores y contrastes que el usuario pueda percibir mejor, como así también pruebas sobre del sonido a utilizar, velocidad

del sintetizador de voz, etc. De igual forma se registran las características de su equipo, tipo de monitor, impresora, y otros periféricos con el fin de aprovechar los recursos que cuente el usuario.

- **Asistencia en la navegación:** Estableciendo Información de los hipervínculos, avisando sobre el tamaño, tipo de contenido, tiempo de lectura estimada al acceder a la página de dicho vínculo. Clasificación, calificación de los hipervínculos, sugiriéndole al usuario sobre el grado de utilidad de acceder a dicha página, basado en estudios estadísticos y en una proyección sobre el grado de eficacia del proceso de la conversión.
- **Agenda de direcciones:** Denominada “*Páginas Azules*”, consiste en un conjunto de direcciones de sitios Web, analizados de acuerdo a las normas de accesibilidad, y que de permiten ser transformadas sin una gran pérdida de información. Dichas direcciones de Internet se encuentran divididas en distintas categorías de acuerdo a su naturaleza, permitiendo que un discapacitado visual pueda llegar a su contenido de manera casi inmediata, sin tener que realizar una búsqueda profunda y costosa sobre un tema en particular.

## Conclusiones

La sociedad en su conjunto esta involucrándose cada vez más en las problemáticas que se les presentan a los discapacitados. En lo referente a la informática también han comenzado a hacerse esfuerzos en esta área, comenzando por los entornos operativos y aplicaciones y comprometiéndose en estos últimos años con las dificultades que trae aparejada la inserción de Internet en la sociedad.

Distintas organizaciones y consorcios continuamente están definiendo y publicando normas estándares con el fin de establecer pautas en el diseño de páginas y sitios Web, pero esto sólo es la mitad de la solución. Lograr que la comunidad informática dedicada a la construcción de sitios Web tenga el conocimiento total y esclarecido de normas, y que éstas sean aplicadas de forma ordenada, es la otra mitad del camino por recorrer. Resulta casi imposible tratar de enseñar y capacitar en esta área a todas las personas del mundo que se dedican a diseñar contenidos.

“*www.lea.com.ar*” es un portal creado a partir de la necesidad de establecer una política de normalización diferente, basado en un conjunto de reglas y normas homogeneizadas. Tiene como objetivo principal hacer accesibles las páginas ya publicadas en la Web, permitiendo al usuario minusválido visual determinar la forma en que desea que se le presente la página solicitada, de acuerdo a la naturaleza de su discapacidad.

Ofrece también una serie de servicios ya integrados como “*Sintetizador de voz*” y “*Magnificador de pantalla*”, entre otros, evitando de esta forma problemas de incompatibilidades entre distintos productos del mercado.

Actualmente, se sigue realizando mejoras del portal antes de publicarlo, evaluándolo frente a usuarios discapacitados, y de esta manera establecer las potencialidades del producto.

## Referencias

[Ref.1]- Revista “VIVA”, suplemento dominical del diario Clarín del domingo 24 de febrero de 2002.

[Ref.2]- “Web Site Accessibility Guidelines”, Gregg C. Vanderheiden Ph.D., Wendy A. Chisholm, Neal Ewers, Shannon M. Dunphy, Trace R. University of Wisconsin, Marzo de 1997.

[Ref.3]- “El proyecto exhib los discapacitados visuales y el acceso a la información”. Consorcio exhib. O.N.C.E., Organización Nacional de Ciegos Españoles, Junio de 1994.

[Ref.4]- “People with Disabilities Can't Access the Web”, M. Paciello - <http://www.yuri.org/webable/mp-pwLCA.html>

[Ref.5]- “W3C Recommendation”, <http://www.w3.org/pub/WWW/TR/REC-html32.html>

## *Hacia una experiencia de aprendizaje inicial de Ciencias de la Computación basada en la web*

**Guillermo Feierherd, Beatriz Depetris (1)**

*Departamento Informática – Sede Ushuaia*

*Facultad de Ingeniería – UNPSJB*

*Darwin y Canga – (9410) - Ushuaia – Tierra del Fuego – Argentina*

*TE/FAX: +54 (2901) 44 3513*

*feierherdge@ciudad.com.ar / depetrisb@ciudad.com.ar*

### **Resumen**

La necesidad de satisfacer las crecientes demandas sociales de educación requieren optimizar el uso de los recursos disponibles. Ello implica –particularmente en nuestra región patagónica y en la educación de nivel superior–, emplear todas las combinaciones posibles de técnicas de educación presencial y a distancia.

En este caso particular se tratará de extender los resultados auspiciosos alcanzados mediante experiencias presenciales realizadas con alumnos del último año del nivel Polimodal e ingresantes a la carrera de Informática –originadas inicialmente en una preocupación por los elevados niveles de deserción observados en la misma–, al resto de la población de la Provincia de Tierra del Fuego, incluyendo los residentes en las bases de la Antártida Argentina.

La transformación del curso presencial en un curso a distancia basado en la web, deberá contemplar tanto aspectos técnicos como educacionales y económicos.

EL proyecto está coordinado con la Facultad de Informática de la UNLP<sup>2</sup>.

### **Introducción**

Las demandas educacionales de la sociedad son cada vez mayores y proveer a su satisfacción se ha convertido en un imperativo ético. Imprescindiblemente, este objetivo debe lograrse mediante una optimización de los recursos físicos, económicos y humanos. En todo el mundo, pero particularmente en regiones como nuestra Patagonia, caracterizada por su extensión y su baja densidad de población, dicha optimización implica renunciar a utilizar exclusivamente las modalidades tradicionales de enseñanza. En consecuencia, cómo dice Greville, **“si existe insuficiencia de maestros y de escuelas disponibles deberán encontrarse alternativas distintas de la enseñanza presencial dentro de los linderos de la escuela.”**

Las distintas variantes de educación a distancia –eventualmente combinadas con instancias presenciales–, conforman hoy gran parte de las alternativas requeridas por Greville, y en ellas los gobiernos, las universidades, las empresas y los organismos no gubernamentales depositan sus mayores esperanzas para satisfacer las demandas de educación.

Así, según Prieto Castillo, para millones de estudiantes la oportunidad de aprender a distancia se presenta como una posibilidad –en algunos casos la única–, real, cualquiera sea el nivel de enseñanza. Las posibilidades que ofrece, entre las que se pueden citar la masividad espacial, el menor costo por estudiante, la diversificación de la población escolar, la individualización del aprendizaje, la provisión de cantidad sin desmedro de calidad y la generación de una autodisciplina de estudio indispensable para una educación continua, la convierten en una herramienta poderosa.

En el caso particular de las carreras de informática en la educación superior, un aspecto no menor en la optimización de recursos lo constituye la búsqueda de metodologías y herramientas que permitan disminuir los elevados niveles de deserción. Con ese objetivo, en coordinación con la

---

<sup>1</sup> Profesores de las Facultades de Ingeniería y Ciencias Económicas en la Sede Ushuaia de la Universidad Nacional de la Patagonia San Juan Bosco

<sup>2</sup> Laboratorio de Investigación y Desarrollo en Informática – Facultad de Informática – Universidad Nacional de La Plata

Facultad de Informática de la Universidad Nacional de La Plata, implementamos desde el año 2000 un proyecto para actuar sobre aquellas variables que entendíamos como las principales responsables de esos índices de deserción: el desconocimiento de las características del campo (en el sentido de Bourdieu), y por ende de las competencias y aptitudes para intervenir en el mismo, y distintos problemas metodológico-didácticos en el proceso de enseñanza en las etapas iniciales de las carreras universitarias en informática.

El proyecto implicó la realización de experiencias presenciales en la ciudad de Ushuaia, donde se halla asentada una Sede de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco en la que se dicta la carrera de Informática. Los resultados preliminares permiten afirmar que dichos ensayos –tanto los llevados a cabo con los aspirantes a ingresar como con los alumnos de los últimos años del Nivel Polimodal–, pueden considerarse como positivos.

Por otra parte, visitas realizadas a colegios de Tolhuin y Río Grande (las otras dos localidades de la Provincia de Tierra del Fuego) para difundir las características de las carreras que se ofrecen en la Sede Ushuaia y comentar las actividades que se desarrollan en ella, nos permitieron detectar el interés de sus alumnos por participar de experiencias como las mencionadas. Finalmente, existen solicitudes para arbitrar mecanismos que hagan accesibles algunas de las actividades de la Sede local de la Universidad al personal de las bases permanentes en la Antártida Argentina que, en el caso de la base Esperanza, incluye jóvenes en los niveles de EGB y Polimodal. Tanto por razones geográficas como económicas, estos nuevos requerimientos sólo podrán ser satisfechos utilizando estrategias de educación a distancia.

Se plantea entonces la necesidad de extender las experiencias en dos dimensiones: la primera, temporal, permitiendo su realización desde etapas más tempranas del nivel medio; la segunda, espacial, poniéndola al alcance de un mayor número de alumnos del nivel medio y aspirantes a ingresar a estudios superiores de Ciencias de la Computación.

Esto constituye todo un desafío. Para repetir los resultados logrados no debemos olvidar que la educación a distancia no consiste simplemente en reproducir los contenidos de un curso presencial para que puedan ser accedidos a distancia. Implica mucho más, fundamentalmente en este caso en que pretendemos enseñar principios básicos de programación, tarea que ya presenta dificultades en forma presencial.

### **Temas de investigación, desarrollo y experimentación**

- **Analizar y comparar metodologías para la implementación de cursos de computación a distancia**

Se trata de realizar un análisis comparativo de las variantes existentes en educación a distancia, incluyendo sistemas asincrónicos y sincrónicos (en línea) y, fundamentalmente aquellos basados en la web o soportados por ella, eventualmente combinados con instancias presenciales. Este primer análisis considerará exclusivamente los aspectos pedagógicos que puedan favorecer la enseñanza de la programación. El resultado de esta etapa debe arrojar un orden relativo de alternativas posibles.

Cabe un comentario adicional: disponerse a seguir un curso a distancia a través de la web implica adquirir las habilidades necesarias para manejar el sistema informático que permite conectarse a ella. En muchos casos, cuando los contenidos del curso que se pretende seguir están totalmente alejados de estas cuestiones tecnológicas, el tiempo requerido para lograr dichas habilidades (si no se dispone ya de las mismas), puede verse como un tiempo perdido. Esto no es así en el caso que nos ocupa, ya que las destrezas que se adquieran son útiles para desenvolverse en el campo al que se pretende ingresar.

- **Aplicación de limitaciones técnicas y económicas a las alternativas diseñadas en el paso previo.**

Se filtrarán las alternativas diseñadas aplicando criterios de factibilidad técnica y económica.

Se deberán considerar tanto las propias de la institución como la de los alumnos potenciales. Esta última condición implica la necesidad de un relevamiento, no necesariamente exhaustivo, que las determine. Si bien no es posible actuar sobre las posibilidades de hardware y de comunicaciones de que dispone cada alumno, se privilegiará en materia de software el uso de software libre, lo que permitirá su distribución sin costo.

- **Diseño y desarrollo del curso para su implementación a distancia**

El diseño y desarrollo del curso sobre la alternativa elegida deberá contemplar no una simple transcripción de los materiales sino un rediseño que lo adapte al nuevo contexto y que permita aprovechar las ventajas potenciales que éste ofrece.

- **Implementación del curso**

Se llevarán a cabo las distintas instancias del curso.

- **Evaluación de los resultados y retroalimentación**

Se evaluarán los resultados obtenidos y, en función de los mismos se determinará la conveniencia de modificaciones y mejoras.

### **Algunos resultados obtenidos y tareas en curso**

- Durante los años 2001 y 2002 se desarrollaron Talleres de Introducción a la Programación (a nivel del ingreso a la carrera de Informática) y Talleres de Análisis de Aptitudes y Competencias para alumnos de los dos últimos años del Nivel Polimodal residentes en la ciudad de Ushuaia. En todos los casos se utilizó el entorno de programación Visual Da Vinci. Las experiencias realizadas permitieron sugerir modificaciones y extensiones fundadas al ambiente, las que se están desarrollando en coordinación con la UNLP.
- Los Talleres de Introducción a la Programación se prolongaron en actividades de autoaprendizaje y desarrollo de algoritmos (con asistencia de tutores), durante el primer cuatrimestre de cada año, lo que permitió a los alumnos iniciar el Curso de Algorítmica y Programación en mejores condiciones. Se acompañó la investigación experimental con evaluaciones, encuestas a los alumnos y estudios comparativos con alumnos de ciclos anteriores.
- La formación de recursos humanos en el proyecto ha concretado tres Especializaciones en Docencia Universitaria y están en curso cuatro estudios de Magister en temas relacionados.

## Bibliografía Básica

Prieto Castillo D., Gutiérrez Pérez F. *La mediación Pedagógica* Editorial Ciccus – La Crujía, Sexta Edición, Argentina, 1999.

Greville R. *La UNED: Una Evaluación* Editorial UNED, Costa Rica, 1987.

Fainholc A. *La Interactividad en la Educación a Distancia* Editorial Paidós, Argentina, 1999.

Pérez Gómez E. *Volver a pensar la Educación* Editorial Morata, Madrid, 1996

Prieto Castillo D. *La comunicación en la Educación* Editorial Ciccus – La Crujía, Argentina, 1999

Colección de Transactions on Education del IEEE

Colección de Boletines del SIGCSE de ACM

Champredonde R., Palacios A., Ainchil V. *Programming Teaching Based on Thinking Skills* Publicado en First International Congress on Tools for Teaching Logic, Universidad de Salamanca, Junio 2000.

Lucarelli E. *La construcción de la articulación teoría-práctica en las cátedras universitarias: búsqueda y avance* Ponencia en el Instituto de Investigaciones en Ciencias de la Educación, Facultad de Filosofía y Letras, UBA, 1996

Feierherd G., Depetris B., Jerez M. *Tecnología Informática aplicada al aprendizaje inicial de Ciencias de la Computación* Publicado en el III Workshop de Investigadores en Ciencias de la Computación (WICC 2001), San Luis, 2001

Feierherd G., Depetris B., Jerez M. *Una evaluación sobre la incorporación temprana de algorítmica y programación en el ingreso a Informática* Publicado en el VII Congreso Argentino de Ciencias de la Computación, El Calafate (Santa Cruz), 2001

Feierherd G., Depetris B. Proyecto *Evaluación y desarrollo de herramientas multimediales para análisis de competencias y aplicación de una metodología didáctica para mejorar el aprendizaje inicial en Informática* Universidad Nacional de Patagonia San Juan Bosco, 2000

## APRENDIZAJE BASADO EN PROBLEMAS COMO APOYO A LAS CLASES PRESENCIALES A TRAVÉS DE SOFTWARE PARA GROUPWARE

Nancy Figueroa, Fernando Salgueiro, Sabrina Cánepa,  
Fernando Lage & Zulma Cataldi

Laboratorio de Informática Educativa. Facultad de Ingeniería. UBA  
Paseo Colón 850. 1063 Ciudad de Buenos Aires. Informat@mara.fi.uba.ar

### Resumen

*En este trabajo se resumen algunos de los resultados de la línea de investigación del Laboratorio de Informática Educativa (LIE) centrada en la problemática de los alumnos en el área de programación cuando ingresan a la universidad.*

*A tal efecto, se llevaron a cabo varias experiencias de aprendizaje basado en problemas (ABP) utilizando un software para groupware, para alumnos de cursos de Algoritmos y Programación de primer cuatrimestre de la carrera de Ingeniería Informática, como apoyo a las clases presenciales.*

*Se describe el marco teórico que da lugar a las experiencias, para las que se utilizó el protocolo de trabajo descrito en trabajos previos (Cataldi, Lage, 2001a) que se ha ajustado a este caso particular.*

*Luego, se presenta una perspectiva acerca de los análisis cualitativo y cuantitativo realizados a partir de los datos obtenidos a través de los mensajes intercambiados por los estudiantes en forma asincrónica, dejando para comunicaciones posteriores el análisis de las comunicaciones sincrónicas tal como el chat.*

*Finalmente se presentan algunas de las líneas de investigación que se están llevando a cabo mediante experiencias de aplicación del modelo en cursos de grado universitarios incorporando las pautas del aprendizaje basado en problemas. Por otra parte, se señala que se está ensayando un software de iguales características con herramientas más flexibles, confiables y amigables acorde con las necesidades detectadas mediante encuestas realizadas a los usuarios estudiantes y docentes.*

### Introducción

Las mediaciones debidamente contextualizadas deben permitir que los alumnos construyan sus propios aprendizajes, que sean capaces de crear y desarrollar proyectos, que sean capaces de investigar, evaluar y resolver problemas. Por ello, es pertinente tratar con metodologías como la de resolución de problemas que permitan aprendizajes trabajando cooperativa y colaborativamente.

El *aprendizaje colaborativo* se refiere a la formación de grupos o equipos de trabajo orientados hacia ciertos objetivos de aprendizaje, donde *cada participante del grupo interviene en todas y en cada una de las partes del proyecto o problema*. En cambio en la *cooperación*, cada uno de los integrantes del grupo, tiene destinada *una tarea específica dentro del proyecto o problema*, realizando en este caso un trabajo más individual como parte del trabajo total. (Johnson y Johnson, 1995, 1999)

En trabajos anteriores se señalaron una serie de problemas relativos al bajo rendimiento de los alumnos en un curso inicial de algoritmia. (Lage, Cataldi, Denazis, 2000). A partir de la información provista por los alumnos a través de los guiones en la preparación de sus exámenes se listó una serie de evidencias tales como: la falta de consultas a bibliografía, el poco de tiempo dedicado al estudio, la falta de metodologías de estudio, la ausencia de estrategias de estudio y de resolución de problemas, ya que muchos de los estudiantes no leían correctamente el enunciado o lo interpretaban mal. Se evidenciaron también dificultades para diferenciar la información relevante de la que no lo es. Esto revela problemas tales como la falta de atención y el conocimiento frágil de los alumnos (Perkins, 1995). Centrados en estos cinco indicadores se pensó en buscar y diseñar estrategias de estudio que permitan: mejorar el rendimiento de los alumnos e inculcarles hábitos de estudio, fomentar la investigación y lograr la motivación. (Cataldi, Lage et al., 1998)

Se consideró en cambiar el escenario de trabajo desde el aula, hacia las nuevas tecnologías a fin de ir más allá del efecto novedad (Cabero, 2001), y de mostrarles a los estudiantes las ventajas del trabajo en grupos cooperativos y colaborativos, además de inculcarles algunos hábitos y valores tales como: el respeto al otro, la tolerancia y la solidaridad.

### **Las experiencias**

Se han realizado varias experiencias durante los últimos años tendientes a una mejora en los aprendizajes. A continuación se describe una de ellas efectuada a un grupo de 60 alumnos de primer cuatrimestre de la carrera de Ingeniería Informática, utilizando el software de groupware que se describe en Cataldi. Lage et al. (2001b,f). Durante la experiencia de resolución de problemas se usó el modelo de trabajo 4C (Cataldi. Lage et al., 2001b,f). simplificado en tres etapas: dirigida, cooperativa y colaborativa puras. (Lage, 2002)

Se registraron las interacciones obtenidas mediante el uso del correo electrónico y del chat, si bien éstas últimas serán analizadas posteriormente, en una primera etapa se efectuará el análisis cualitativo y cuantitativo de los correos electrónicos intercambiados por los alumnos. La sistematización de la información obtenida, proporcionará datos acerca de los diferentes tipos de interacciones, de las cuales se han elegido: alumno-alumno (AA), alumno-docente (AD), alumno-contenido (AC) y alumno-medio (AM) por ser las que podrían brindar más información acerca del mejoramiento de los aprendizajes quedando el resto para un análisis posterior.

Para realizar el análisis correspondiente se definieron algunas categorías o unidades de análisis de acuerdo a Gairín (1998), a fin de sistematizar las relaciones entre el docente, los alumnos, los contenidos y el medio

### **Metodología de trabajo.**

Como se ha planteado en trabajos previos, (Cataldi, Lage, 2001c,d,e) la metodología a seguir consta de los tres momentos que se indican:

1. Diseño de los problemas
2. Período de entrega del trabajo y desarrollo
3. Evaluación

En cada uno de dichos momentos, se deben tomar diferentes decisiones que determinan el protocolo de trabajo a seguir, como se ha analizado oportunamente en trabajos previos (Figuroa, Lage, Cataldi et al. 2001 a,b)

### **Desarrollo de las experiencias**

El ambiente de aprendizaje queda definido de esta forma como un *ecosistema de aprendizaje* (Murillo, 2000) en el cual todos participan con determinadas tareas específicas y realizan una serie de acciones (plantean preguntas, emiten respuestas, solicitan ayuda, información, etc.). Se puede pensar por lo tanto que se establecen entre los alumnos y el docente un tipo de relaciones que producen retroacciones. Tomando en consideración estos aspectos se puede definir la *interacción electrónica* como el intercambio de mensajes electrónicos que tienen lugar entre dos o más personas que se influyen mutuamente intercambiando información y producen resultados que probablemente ninguno de los actores hubiera producido por separado. La información y los datos que se van a recoger, construir y analizar están centrados en las acciones entre pares de componentes del proceso educativo (alumno, contenido, medio y docente)

El proceso interactivo muestra: respuestas, réplicas y contrarréplicas que se encadenan, lo que evidenciará las diferentes categorías, según la interacción analizada sea AA o AD u otras.

Para la evaluación de las experiencias se tuvieron en cuenta aspectos generales a través del análisis del número de correos, la clasificación por categoría de los correos, la longitud de encadenamientos producidos y aspectos particulares tales como el seguimiento de los alumnos participantes, que muestra si las interacciones fueron positivas, (como se ve en la tabla 1) en cuyo caso se concluye que hubo aprendizaje.

Desde el aspecto general se puede decir que en la primera experiencia que se describe se inscribieron 60 alumnos para interactuar a través del foro de discusión y se emitieron 323 correos. Hubo algunos alumnos del

curso que por no poseer computadora con acceso a Internet prefirieron el sistema presencial tradicional de clases de consulta individual o grupal. Los alumnos participantes emitieron el 81% de los mensajes y los docentes el 19%. Del total de los correos de los alumnos el 57% han sido de la categoría Alumno-Alumno y el 43% restante han sido Alumno-Docente.

Respecto del total de los correos de los alumnos, el 16% se refiere a la categoría Alumno-Medio y el 56% se refiere a Alumno-Contenido. El movimiento de los alumnos presenta una media de 5,3 correos y una desviación media de 1,75.

Se detectaron 18 *encadenamientos* de mensajes establecidos con longitudes de la cadena promedio de 12 que representa un tráfico de 216 mensajes. Estos encadenamientos fueron estudiados obteniéndose un modelo básico de secuenciamiento de mensajes, de acuerdo con las observaciones de Rodríguez Marcos (1995) e integrando su visión como se puede observar en Figueroa, Lage, Cataldi (2001). Si bien puede haber encadenamientos mucho más complejos, existe una secuencia básica que subyace a todos ellos, la que se corroborará mediante experiencias posteriores con grupos nuevos.

Dentro de cada tipo de interacción, se observan categorías como las que se muestran en la Tabla 1 a modo de ejemplo. En este caso se han ilustrado cuatro tipos de interacciones que involucran al alumno participante con todos los elementos presentes en el acto didáctico mediado y son: AA, AC, AM y AD.

ALUMNO-ALUMNO		ALUMNO-MEDIO	
Solicitan aclaraciones	41,1%	Preguntan	36,4%
Clarifican	23,7%	Informan	54,5%
Valoran	3,7%	Sugieren	9,1%
Ayudan a la Gestión	2,5%	ALUMNO-CONTENIDO	
Aportan Bibliografía	1,2%	Solicita aclaraciones	60%
Piden Información	15%	Clarifican-Explican	33%
Dan Información	12,5%	Definen	3,5%
ALUMNO-DOCENTE		Responden	3,5%
Gestión	82%		
Solicitan aclaraciones del contenido	18%		

Tabla 1: Tipos de interacciones y categorías para la interacción Alumnos-x

Por otra partes, del total de mensajes emitidos por el Docente a la categoría Docente-Alumno le corresponde el 54,5%, a Docente-Medio 12%, a Docente-Contenido 21,5% y hubo un 12% de mensajes entre docentes. El análisis de la experiencia completo se puede ver en Figueroa, Cataldi, Lage (2002)

### Evaluación a través de la evolución

La evolución de las interacciones permite determinar la evolución del aprendizaje de cada alumno. A fin de poder establecer si el estudiante ha *aprendido* se tendrán en cuenta las reglas siguientes:

- Se considera que una interacción es **positiva** si manifiesta que hubo aprendizaje.
- Será **neutra** si permanece en un estado igual al inicial
- Será **negativa** en el caso en que el alumno no hubiere evolucionado en su aprendizaje.

Estas reglas permiten determinar la **evolución** del alumno, por este motivo, para cada alumno del grupo se establece la secuencia de mensajes que él ha generado acerca de un tema en particular. Luego, se analizará dicha secuencia, es decir desde el mensaje de partida hasta el mensaje final. Como lo que se observa es **la evolución** en el aprendizaje del estudiante, a través de cada interacción, la variable que muestra el aprendizaje se relevará desde indicadores considerados como pares ordenados tal como se observa al la Tabla 2:

PAR ORDENADO	INDICADOR
(no entiende un concepto, no entiende el concepto)	<b>Interacción neutra</b>
(no entiende un concepto, entiende parcialmente)	<b>Interacción positiva</b>
(no entiende un concepto, entiende el concepto)	<b>Interacción positiva</b>

Tabla 2: interacciones referidas a preguntas

Estos pares se toman para casos en que las interacciones no se refieran a una respuesta, pero, en el caso de que las interacciones sean una respuesta se tendrá otra tabla con indicadores similares a los presentados en la Tabla 2

Si bien en las experiencias no se consideraron interacciones negativas, ya que lo que se quería analizar era si hubo o no aprendizaje, es posible combinar una condición donde la salida sea peor que la llegada, y es el caso mostrado en la Tabla 3.

PAR ORDENADO	INDICADOR
(respuesta aceptable, respuesta incorrecta)	<b>Interacción negativa</b>

Tabla 3: Condición de interacción negativa

Este tipo de interacción hubiera mostrado la debilidad del conocimiento adquirido, en otras palabras el *conocimiento frágil* del que habla Perkins (1995) como se señaló antes.

## Conclusiones

Los resultados obtenidos confirman que los aprendizajes han mejorado, pero como se trabaja con grupos humanos y con una metodología, además del uso de las nuevas tecnologías, se considera conveniente replicar las experiencias a fin de validar los resultados obtenidos, durante períodos sucesivos, para ver como evolucionan los indicadores.

El nivel de desempeño de los alumnos en las diferentes etapas de la evaluación mostró una mejoría respecto de los grupos que no han trabajado con tecnologías de esta índole. En los exámenes finales la mejora resultó estar en el orden de un 10% respecto del período previo.

Esto se explica por:

1. La posibilidad de comunicarse en cualquier momento según las necesidades de las personas genera un sentimiento de "*comunidad*" y de confianza que hace que el alumno exponga sus inquietudes sin problemas. En resumen la tecnología ha permitido incrementar la interacción en la que se basa el aprendizaje.
2. El incremento de la interacción ha favorecido el trabajo colaborativo para la resolución de problemas.
3. El rol docente como facilitador o mentor, sostenido por los modelos de aprendizaje colaborativo y por resolución de problemas, ha sido favorecido por el sistema al permitir una forma de trabajo en la que el alumno sólo obtiene la ayuda que requiere.
4. Las interacciones Alumno-Alumno permiten: confrontar perspectivas, es decir experiencia-juzgar-juzgar-decidir. En colaboración "*significar*" es hacer un juicio y tomar decisiones junto con otros y la búsqueda del significado es el resultado de los procesos de negociación los pares a través la pluralidad de juicios. El uso del correo electrónico en un foro de discusión implica que los estudiantes deban utilizar cuidadosamente el lenguaje, es decir con momentos de presentación, exposición de la inquietud y cierre del mensaje, lo cual involucra un estilo, que sin ser totalmente protocolar, requiere de un uso adecuado desde el punto de vista semántico y sintáctico de manera que resulte inteligible para los interlocutores.

En las experiencias también se han notado algunos inconvenientes tales como:

1. Falta de ubicuidad de la tecnología, ya que no todos los alumnos tienen disponible el equipamiento, o bien la conexión a la red necesaria en sus domicilios para que las comunicaciones sean realmente fluidas.
2. Sobre otra característica del sistema que es la interactividad es necesario decir que sus ventajas pueden minimizarse si los docentes no son prestos en las respuestas a los requerimientos de los alumnos
3. Si bien no fue un inconveniente, pocos alumnos utilizaron un "alias", prefirieron identificarse en todo momento.

## Trabajos a futuro

Como ya se dijo la interacción juega un papel muy importante en el éxito del modelo de resolución de problemas a través de trabajo cooperativo y colaborativo. Por ello resulta de sumo interés precisar y optimizar los mecanismos que la potencien. Esto significa realizar investigaciones en las siguientes líneas de trabajo:

1. Estudiar la interacción en las comunicaciones sincrónica y asincrónica en diferentes contextos pedagógicos para establecer cuando cada tipo de interacción es más apropiada para soportar la construcción del conocimiento. Se espera que la diferencia de contextos arroje categorizaciones dentro de los diferentes tipos de interacciones propuestos.
2. Estudiar la mejor forma de estructurar la interface de comunicación, ya que esta afecta los aprendizajes. Esto conduce a identificar y tipificar los modos y canales de comunicación de los alumnos.
3. Analizar cómo evolucionan las habilidades adquiridas por los estudiantes a lo largo del tiempo.

## Referencias Bibliográficas

- Cabero 2001. *Tecnología Educativa*. Editorial Síntesis
- Cataldi, Z., Lage, F. y Perichinsky, G. 1998. *Enseñanza de Computación: una disciplina en vertiginoso cambio dentro de una educación en cambio*. Proceedings del IV Congreso Internacional de Ingeniería Informática. Páginas 286-295. Editado por Departamento de Publicaciones de la Facultad de Ingeniería.
- Figueroa N.; Lage, F. y Cataldi Z. et al. 2001b. *Una experiencia para la construcción del conocimiento en cooperación y colaboración para alumnos que ingresan a la universidad*. Taller Internacional de Software Educativo. TISE'01. 3-5 de diciembre. Universidad de Chile. Sgo de Chile.
- Figueroa N., Cataldi Z. y Lage F. (2002). Aprendizaje basado el problemas. Informe interno LIE-FI-UBA-02
- Figueroa, N.; Lage, F., Cataldi, Z., et al.. 2001a. *La resolución de problemas en cooperación y colaboración: una experiencia para la construcción social del conocimiento en alumnos de Algoritmos y Programación I*. Paper IE-00098, págs. 161-171. ISBN 9879628855-6-1. VII CACIC: Congreso Argentino de Ciencias de la Computación, 16-20 de octubre. El Calafate.
- Gairín, J. M. (1998): *Sistemas de representación de números racionales positivos. Un estudio con maestros en formación*. Tesis Doctoral. Departamento de Matemáticas. Universidad de Zaragoza.
- Johnson D. y Johnson R. 1995. *Teaching Students to be Peace Makers*. Interaction Book Co.
- Johnson D. y Johnson R. 1999. *Aprender Juntos y Solos*. Aique
- Lage F. J., Cataldi Z., Denazis J. 2000. *The Scripts of University Students and Experts in the Preparation of the Examinations: A study in Process*. Proceedings of 30<sup>th</sup>SEE/IEEE Frontiers in Education Conference. FIE'2000. Kansas, 18-20 de octubre. Sesión FIG
- Lage F. y Cataldi Z., (2001e). *Una experiencia de resolución de problemas a través de modelos cooperativos-colaborativos aplicada a algoritmia usando nuevas tecnologías de comunicación*. IN-MAT 2001. I Congreso Internacional de Matemática Aplicada a la Ingeniería y Enseñanza de la Matemática en Ingeniería. 7-9 de noviembre. Facultad de Ingeniería. UBA.
- Lage, F. & Cataldi (2001d) *Modelo 4C Cooperativo-Colaborativo Para Capacitación De Recursos Humanos A Través De Intranet/Extranet*. WICC 2001. Workshop de Investigadores en Ciencias de la Computación. San Luis. 22-24 de mayo.
- Lage, F. y Cataldi Z., 2001f. *Un modelo cooperativo-colaborativo de resolución de problemas para favorecer los aprendizajes significativos*. en I Encuentro Internacional de Didáctica de la Educación Superior. Universidad de Pinar del Río. Cuba. 19-22 de junio.
- Lage, F. y Cataldi, Z. et al. (2001c): *El modelo 4C para capacitación de recursos humanos ooperativo-colaborativo*. LIE; Informe interno: Aceptado en EDUTEC 2001. Universidad de Murcia.
- Lage, F.; Cataldi, Z. et al. (2001a): *Una experiencia cooperativa-colaborativa asincrónica aplicada a la resolución de problemas*. III Simposio de Educación Matemática. 1 al 4 de mayo. Memorias en CD-ROM. ISBN 987-98741-0-2. Universidad de Luján. Chivilcoy.
- Lage, F.; Cataldi, Z.; Villagarcía W., H. e Iglesias O., (2001b). *Aplicación del modelo 4C cooperativo-colaborativo para capacitación de recursos humanos mediante el uso de software para groupware*. Paper IE-00118, págs. 71-83. ISBN 9879628855-6-1. VII CACIC: Congreso Argentino de Ciencias de la Computación, 16-20 de octubre. El Calafate.
- Lage F. (2002). Tesis de Magister en Automatización de Oficinas. Facultad de Informática. UNLP
- Murillo (2000): *Un entorno interactivo de aprendizaje con Cabri-actividades aplicado a la enseñanza de la geometría en la ESO*. Tesis Doctoral. Universidad autónoma de Barcelona.
- Perkins David (1995) *La Escuela Inteligente*. Gedisa.
- Rodríguez Marcos (1995): *Un enfoque interdisciplinario en la formación de maestros*. Narcea.
- Sánchez Ilabaca (2001). *Aprendizaje visible, tecnología invisible*. Dolmen Ediciones.

### **Incorporación de las nuevas tecnologías de la Comunicación y de la Información al quehacer docente universitario**

*Maria V. Godoy Guglielmona, Sonia I. Mariño, Gladys N. Dapozo y Raquel H. Petris*

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.

Universidad Nacional del Nordeste. 9 de Julio N° 1449. CP: 3400. Corrientes. Argentina

TE (03783) 423126/423968/424606 - Fax (03783) 423968

[mvgodoy@exa.unne.edu.ar](mailto:mvgodoy@exa.unne.edu.ar), [msonia@exa.unne.edu.ar](mailto:msonia@exa.unne.edu.ar), [gndapozo@exa.unne.edu.ar](mailto:gndapozo@exa.unne.edu.ar),  
[rpetris@exa.unne.edu.ar](mailto:rpetris@exa.unne.edu.ar)

#### **Fundamentación**

La evolución de los sistemas educativos y de formación de recursos humanos a lo largo de la última década, se ha caracterizado por la búsqueda de medios de comunicación apropiados para brindar instrucción a más gente y con mayor grado de eficacia [5].

Las **Tecnologías de la Información y la Comunicación (TIC)** están introduciendo **importantes cambios en todos los sectores de la sociedad**, "... probablemente los cinco pilares estructurales e indispensables para su total comprensión son: el multimedia, el hipermedia, las grandes redes de ordenadores, las autopistas de información y especialmente Internet/Intranet/Extranet" [12]. Los profesionales futuros que formarán parte de la Sociedad de la Información necesitan imperiosamente, nuevos conocimientos, habilidades, destrezas y actitudes. Por ello, las nuevas tecnologías han de ser no sólo contenidos de formación, sino también medio formador.

La alfabetización en computación ocasiona una toma de conciencia del papel actual de las computadoras en nuestra sociedad y genera conocimiento acerca de sus diferentes funciones [16].

Ante estos retos, no solo se exige de los estudiantes de nuestra universidad nuevos saberes, sino que esto demanda a los docentes una preparación adecuada para facilitar y promover la formación de los jóvenes. Morales [13] expresa que los docentes deben dominar la tecnología para hacer un uso más racional y eficiente de las posibilidades que ésta ofrece.

Paralelamente, la **demanda de educación se intensifica**, la idea básica es la educación continua, cuyas características básicas sean la diversidad, la flexibilidad, y la accesibilidad tanto en espacio como en tiempo [9].

No es posible responder a estos requerimientos sino se asumen y desarrollan nuevas estrategias de enseñanza-aprendizaje. La incorporación de computadoras, conexiones a Internet, multimedios, etc. en las instituciones educativas y en organizaciones de distinta índole, así como en los hogares particulares, hace posible, que nuevas formas de educación lleguen hasta lugares muy dispersos geográficamente, brindando además cobertura a un gran número de personas y abreviando los costos demandados por la enseñanza presencial tradicional.

A pesar de esto, el avance que ha tenido la **tecnología no se ha compaginado con el desarrollo de los métodos didácticos en el aula de clase**, Fainholc [7] expresa que "al tratarse de espacios conservadores, las innovaciones actuales de omnipresencia y convergencia tecnológica se darían sobre todo en las propuestas no convencionales, entre ellas la de distancia, sin abolir sino integrando y enriqueciendo las tradicionales". Es por eso que la formación y el aprendizaje precisan de dinámicas acordes a este desarrollo tecnológico y en especial al uso apropiado de la metodología de la educación a distancia, en línea y virtual, trascendiendo el tiempo y el espacio para beneficio de aquellos usuarios de la educación que no pueden elegir físicamente un campus universitario o que no pueden permanecer allí por mucho tiempo.

En este proceso de innovación educativa, de construcciones "distintas de lo acostumbrado", es de innegable **importancia el rol** que le cabe cumplir a **los docentes** y los mismos deben estar capacitados adecuadamente para poder participar activamente como favorecedores de espacios de formación no tradicionales. Ya sea asumiendo las tareas de preparador o diseñador de materiales novedosos o bien como tutor, capaz de guiar y motivar al alumno hacia un aprendizaje más autónomo y responsable a través de las tecnologías [7].

La educación a distancia según Peters [14] “ .... es un método de impartir, conocimientos, habilidades y actitudes ....., haciendo uso extensivo de medios tecnológicos”, y otros como Tiffin [17] expresan “El sistema educativo para una sociedad de la información será independiente con respecto a la distancia..... Esta visión tiene como premisa la base del desarrollo de las tecnologías de la información que permiten la realidad virtual”. Desde este punto de vista, la combinación de diversas herramientas y actividades propias de **la educación a distancia y de la educación virtual**, tales como el diseño y elaboración de material multimedial, la construcción de tutoriales de autoestudio, la oferta de cursos a distancia, las tutorías mediante correo electrónico, la participación en foros de discusión, las teleconferencias, la posibilidad de que un alumno presencie clases dictadas en aulas virtuales, etc.; constituyen una batería de recursos **que entendemos deben ser utilizados como favorecedores y difusores de la educación**, buscando la conjunción más efectiva para cada una de las instancias educativas a construir.

### Objetivos del anteproyecto

El presente proyecto “**Incorporación de las nuevas tecnologías de la Comunicación y de la Información al quehacer docente universitario**” consiste en el “Dictado de Cursos de Capacitación” cuyo objetivo fundamental es dotar a los docentes de la Universidad Nacional del Nordeste de las habilidades necesarias para que sean capaces de desarrollar material didáctico multimedial, a través del cual podrán implementar nuevas estrategias de enseñanza-aprendizaje en entornos de educación a distancia y entornos virtuales. Los principales objetivos específicos, se formularon considerando que el cursante:

- maneje conceptos básicos de informática y de nuevas tecnologías de la comunicación.
- incorpore a sus prácticas cotidianas docentes, herramientas tales como: “libros electrónicos”, “documentos Web”, “material multimedial”, “tutoriales”, “acceso a redes e información en la Web”, “bibliotecas virtuales”, entre otros.
- construya material educativo para ser distribuido a través de CD ROM y de Internet, utilizando las TIC.
- elabore tutoriales e incorpore a los mismos los recursos que provee Internet.
- genere ofertas educativas novedosas para la región, tanto a nivel de Grado y de Posgrado, dada la demanda constante de educación continua que existe en el presente.

### Propuesta de capacitación tecnológica

Las herramientas de trabajo y de generación y obtención de información están cambiando, y con ellas nuestro alcance de comunicación y de aprendizaje [6]. La producción de material didáctico interactivo ajustado a éstos cambios, es una labor que requiere de una específica **estructuración y planificación** desde los inicios y requiere del docente un dominio de la tecnología para hacer un uso eficiente y racional de las posibilidades que ella ofrece [13]. La propuesta curricular consiste en cuatro módulos que a continuación se detallan.

La primera unidad temática, denominada “**Cursos básicos para el manejo de la herramienta computacional orientada a la producción de material educativo**”, comprende a) conceptos preliminares requeridos para el diseño de proyectos multimediales, hipertextuales e hipermediales. b) introducción a la producción de material educativo multimedial ([2], [3], [4] y [11]). c) introducción a Internet: acceso a redes, correo electrónico, foros de discusión, entre otros.

Los principales objetivos son que el cursante:

- identifique los conceptos informáticos que intervienen en la elaboración de material didáctico multimedial empleando las nuevas tecnologías de la comunicación y la información.
- adquiera las nociones básicas relacionadas con la construcción de proyectos educativos.
- aprenda a producir textos para multimedia.
- conozca el concepto de hipertexto y sus posibilidades educativas.

- adquiera los conocimientos, metodologías y destrezas que le permitan un manejo autónomo de los distintos servicios que ofrece Internet.

La multimedia ha reunido en las computadoras personales la capacidad de combinar sonido, vídeo, texto, gráficos y animación en sistemas de autores [software de creación de productos multimedia], que permiten el desarrollo de aplicaciones de capacitación, de entretenimiento, de información, de educación, etc. [6]. Se puede estructurar sistemas, que convergen en interfases multimediales, que facilitan a los usuarios navegar a través de la información de acuerdo a sus intereses, necesidades o simples preferencias, sintiéndose motivados y atraídos por esa forma de comunicación.

La segunda unidad temática, “**Producción de material educativo con vista a distribución off-line**”, consiste en una oferta de cursos definidos para distintos niveles de conocimiento orientado a instruir en el manejo de software, para generar aplicaciones educativas para ser ejecutadas en una PC. Entre las herramientas se mencionan: Neobooks para Windows, Power Point y Toolbook. Como objetivos generales se espera que el cursante:

- obtenga los conocimientos necesarios para la creación de libros multimediales.
- adquiera práctica en el empleo de la terminología específica.
- utilice y valore las herramientas que permiten un diseño de interfases multimedia atractivas.
- conozca y utilice herramientas orientadas a la creación de material didáctico interactivo.

Uno de los servicios, que por su espectacular repercusión muchas veces se lo utiliza como sinónimo de Internet, es el de WWW [12]. Éste permite la representación de información mediante sistemas hipermediales, a través de una red de documentos enlazados. Se pretende capacitar al docente para preparar material didáctico adaptado a estas nuevas formas de educación y de interacción con los alumnos [13].

La tercera unidad temática, “**Construcción de páginas Web utilizando diversas herramientas**”, propone el empleo de una variedad de herramientas informáticas destinadas a complementar este cometido, considerando distintos niveles de complejidad. a) Para el *Nivel inicial*, se propone el empleo de Netscape Composer y herramientas ofimáticas, como Word, Excel y Power Point. b) En el *Nivel medio*, se enfatiza el uso de FrontPage y la inclusión de otras herramientas que permiten la descarga de material en distintos formatos como archivos PDF, archivos .zip o archivos RTF, haciendo uso de la transferencia electrónica de ficheros (FTP). c) En el *Nivel avanzado*, se situó a productos tales como Dreamweaver, Flash, y lenguajes de programación como Java Script que permiten definir una mayor interactividad. Algunos de los objetivos principales planteados se definen en función de que el alumno:

- maneje conceptos de hipertexto e hipermedia y sus posibilidades educativas.
- aprenda a producir documentos para la Web.
- genere aplicaciones hipermediales empleando distintas herramientas informáticas.
- adquiera destreza en el manejo de editores visuales de sitios Web.
- aprenda y emplee las herramientas de construcción de tutoriales hipermediales, como elementos facilitadores de la transmisión de conocimientos.
- adquiera los conocimientos, metodologías y destrezas que le permitan un manejo autónomo de los distintos servicios que ofrece Internet.

Los documentos Web adoptan el modelo de "hipermedia" y se reconocen “... porque no se pueden leer imprimiéndolos en papel y requieren trabajarlos en la pantalla del ordenador” [15].

La última unidad temática “**Herramientas complementarias para el diseño en la Web**”, introduce a los educandos en el aprendizaje de herramientas de diseño gráfico (como Photoshop, CorelDraw, Corel Photo Paint) y producción de elementos multimediales como el sonido, las animaciones y los videos digitales. Se intenta que el cursante logre las siguientes capacidades:

- incorpore los conceptos de los objetos tales como gráficos, sonidos, animaciones y videos.

- adquiera las habilidades para el manejo de las herramientas de dibujo y de edición de imágenes
- ubique la importancia del diseño visual de productos multimedia.
- genere objetos multimediales en distintos formatos.
- conozca y maneje los tipos de imágenes, la resolución, y los conceptos de impresión.
- conozca y realice distintas variedades de animación por computadora.
- adquiera los conocimientos para realizar ediciones sencillas de video digital.

Para cada unidad temática: a) se determinaron los objetivos que se esperan alcanzar; b) se definió el perfil del alumno; c) se seleccionaron los contenidos a abordar; d) se definió la metodología del dictado de las clases; e) se confeccionó material didáctico, guías de trabajos prácticos y lista de bibliografía; f) se previó el modo de evaluación y requerimientos que los participantes deben cumplimentar para la acreditación del curso; g) se propuso que la certificación sea otorgada por el Rectorado de la Universidad, en el cual este proyecto ha sido presentado.

### **Consideraciones didácticas**

A pesar del enfoque tecnológico del presente proyecto, no se descarta ni minimiza la importancia de las consideraciones pedagógicas.

En relación a los cambios impuestos por las NTIC (Nuevas tecnologías de la información y de la comunicación) Elizondo [6] afirma, “...estos cambios no solo se refieren a las herramientas o técnicas que se emplean para diseminar el conocimiento, sino también en la manera en que se estructura el contenido que se pone a disposición de los alumnos, como en lo referente a los reactivos o evaluaciones a través de las cuales se mide la adquisición del conocimiento”.

Uno de los componentes de cualquier entorno de aprendizaje [8], es el material educativo que el mismo ofrece, pero por si mismo no es suficiente para lograr aprendizajes significativos [12]. Fundamentalmente, consideramos que todo material didáctico interactivo debe, al menos, contar con los siguientes componentes:

- Módulo de contenidos didácticos. Orientado a la presentación interactiva de los contenidos de la asignatura o curso seleccionado. Los nuevos contenidos deben basarse en criterios pedagógicos que permitan llegar a aprendizajes significativos.
- Realización de prácticas virtuales, mediante el empleo de simulaciones o de otras herramientas para realizar actividades prácticas requeridas para el aprendizaje.
- Autoevaluaciones y evaluaciones en línea. El ideal de un sistema de autoaprendizaje es la autoevaluación. Una evaluación alternativa a la tradicional se fundamenta en los siguientes aspectos: la identificación de los referentes básicos del proceso de evaluación y la identificación de los ejes básicos a evaluar [10].
- Funciones tutoriales. Deben ser vistas como guías que facilitan o exponen los puntos esenciales del contenido, proporciona esquemas de orientaciones para utilizar otros recursos o aclarar dudas cuando sea necesario [5]. Se puede implementarlas en forma sincrónica ( teleconferencia o chat), o asincrónica ( correo electrónico o foros de discusión ).

Es importante destacar que el éxito de un proyecto de educación virtual y/o a distancia depende de la conformación de grupos interdisciplinarios y de que los diseñadores sean capaces de seleccionar para cada desarrollo en particular, la tecnología educativa más apropiada para cada contexto.

### **Conclusiones**

La computadora como elemento de apoyo en el proceso de enseñanza-aprendizaje ha demostrado ampliamente sus utilidades. El empleo de Internet es hoy en día innegable, y las implicancias que esto trae en la divulgación de los saberes. Además, las Nuevas tecnologías de la Información y de la Comunicación son ampliamente difundidas en la sociedad global. En este trabajo se expone un proyecto de alfabetización dirigido a los docentes universitarios, que consiste en el uso de herramientas informáticas orientadas a la producción de material educativo interactivo, estas permitirán implementar nuevos ambientes de aprendizajes en los distintos dominios del

conocimiento. La propuesta incluye una variedad de productos que pueden ser incorporados considerando distintos niveles de complejidad y la formación previa de cada docente.

El proyecto de capacitación entrenará en el manejo de herramientas tecnológicas, pero además permitirá a los docentes visualizar los logros que podrán alcanzar con el empleo de las mismas en su quehacer docente. Las Nuevas tecnologías de la Comunicación y de la Información facilitarán a los docentes diseñar diversas ofertas educativas innovadoras, estructurarlas en módulos y unidades didácticas interactivas, incorporar programas orientados a prácticas, autoevaluaciones y evaluaciones en línea y disponer de tutorías virtuales, seleccionando las herramientas tecnológicas a su alcance y en función de los objetivos que el mismo se fije. De éste modo y contando con el asesoramiento pedagógico correspondiente, nuevas instancias educativas serán puestas a disposición de un mayor número de personas, intentando contribuir a lo que necesariamente entendemos como “democratización de los saberes”. Sin perder de vista, que esto realmente sucederá, cuando los resultados de instancias como el presente proyecto produzcan profesionales o individuos que resulten en agentes de cambio de nuestra realidad social y económica.

### Referencias

- [1] Benito Crosetti., B. de. Herramientas para la creación, distribución y gestión de cursos a través de Internet. Edutec. Revista Electrónica de Tecnología Educativa. Núm. 12. junio 2000.
- [2] Bustos Martin, I. de. Guía Práctica para usuarios de Multimedia. Ed. Anaya Multimedia. 1996
- [3] CCPM. Centro de Computación Profesional de México. Multimedia Aplicada. Mc. Graw Hill. 2001.
- [4] Codina, L. H de Hipertexto o la teoría de los hipertextos revisitada. En: <http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/codina.htm>
- [5] Chacón, F. y Arias Ferrero, M. L. Nuevas tecnologías para la educación a distancia. Universidad Nacional Abierta. 1996.
- [6] Elizondo, R. E. Nuevas Tecnologías para la Educación a Distancia. Módulo 9 – CREAD ITESM PP 1. 1999.
- [7] Fainholc, B. La interactividad en la educación a distancia. Editorial Paidós- Buenos Aires- Barcelona- Mexico-1era. Edición- 1999.
- [8] Ferrante, A. Materiales multimedia: uso, ventajas, desventajas. Extraído de Módulo 4: Multimedia y Virtualidad. Educaonline.com. 2000.
- [9] Garcia, V. El entorno cambiante de la Educación Superior: las nuevas tecnologías. JIS'98 Segundas Jornadas Informáticas y Sociedad. 305-312 pp. Ed. Universidad Pontificia de Salamanca (Campus Madrid). 1998.
- [10] Gutierrez Pérez, F y Prieto Castillo, D. La mediación pedagógica. Ed. Ciccus. 1999.
- [11] Isakowitz, T., Stohr E. A., & Balasubramanian P. Traducción y Interpretación realizada por Gabriel Muñoz. RMM: Metodología para el Diseño Estructurado de Hipermedios. En: <http://www.pmuc.udec.cl/~gmunoz/rmm/rmm.htm>
- [12] Joyanes, L. En nuevo orden en Informática: Globalización versus Internet. JIS'98 Segundas Jornadas Informáticas y Sociedad. 19-38 pp. Univ. P. de Salamanca (Campus Madrid). 1998
- [13] Morales Arcia, F. Educación a distancia y las nuevas tecnologías de la información. JIS'98 II Jornadas Informáticas y Sociedad. 313-316 pp. Univ. P.Salamanca (Campus Madrid). 1998.
- [14] Peters, O. Distance teaching and Industrial Production. 1983.
- [15] Pina, A. Multimedia en la enseñanza universitaria. En: [www.ub.edu/forum/pina.htm](http://www.ub.edu/forum/pina.htm)
- [16] Rubinstein, V. Enfoques para el abordaje de Contenidos Básicos Comunes desde la perspectiva de la informática. En: [www.currycap.me.gov.ar/publica/pdf/rubins.pdf](http://www.currycap.me.gov.ar/publica/pdf/rubins.pdf).
- [17] Tiffin, J. y Rajasingham, L. En busca de la clase virtual. 1997.

## Nuevos Entornos de Aprendizaje

### Los Ambientes Colaborativos Inteligentes

Grupo de Investigación “*Ambientes Colaborativos Inteligentes*”

Dpto. Informática-Fac. Cs. Fco. Mat. y Nat. - Univ. Nac. de San Luis

Integrantes: Lic. Guillermo Leguizamón, Lic. Margarita Lucero, Lic. Guillermo Aguirre,  
Mg. Marcela Chiarani, Lic. Irma Pianucci, Prof. Lilian Manzur, Prof. Berta García.

e-mail:

[legui@unsl.edu.ar](mailto:legui@unsl.edu.ar), [margos@unsl.edu.ar](mailto:margos@unsl.edu.ar), [gaguirre@unsl.edu.ar](mailto:gaguirre@unsl.edu.ar), [mcchi@unsl.edu.ar](mailto:mcchi@unsl.edu.ar), [pianucci@unsl.edu.ar](mailto:pianucci@unsl.edu.ar), [lman@unsl.edu.ar](mailto:lman@unsl.edu.ar), [bgarcia@unsl.edu.ar](mailto:bgarcia@unsl.edu.ar)

## RESUMEN

A través de este trabajo se comunica uno de los avances del Proyecto “Ambientes Colaborativos Inteligentes”, en cuanto a la investigación pedagógico-didáctica de las características esenciales de los diferentes ambientes de Aprendizaje que prevalecen aún en el Sistema Educativo y su relación con la propuesta de ambientes colaborativos inteligentes formulada por este grupo de investigación.

### I.- Introducción

Cada época ha tenido sus propias instituciones educativas, adaptando los procesos educativos a las circunstancias.

Ello dio lugar a paradigmas como los siguientes:

#### 1.- Proceso/Producto

Este modelo parte de un enfoque multidimensional de la *eficacia docente*, importa aquí la interacción en clase, caracterizando los criterios de eficacia – como señala Flanders (1977) – en aquellos aspectos de la **enseñanza** sobre los cuales el profesor tiene un control directo. Importan aquí las relaciones entre lo que el profesor hace y el efecto de estas acciones sobre los alumnos.

Consecuentemente: el énfasis está dado en la enseñanza, por lo que **el estilo de enseñanza** puede resultar peligroso ya que no existe una generalidad en la eficacia, esto es, puede resultar eficaz con un tipo de alumnos y no resultar con otros.

#### 2.- Mediacional

En este modelo el énfasis está dado en la actividad del alumno, es decir, en los **aprendizajes**, por ello cobran significación **los estilos de aprendizaje**.

El acento es colocado en los procesos cognitivos con que el alumno capta la enseñanza y elabora el aprendizaje, ya que se parte de la premisa de que el éxito de la labor del profesor depende de que haya aprendizaje, y éste sólo se produce cuando el alumno se implica personalmente en las tareas que se realizan, reelaborando sus contenidos hasta llegar a elaborar unos esquemas conceptuales propios. Por lo tanto, el elemento decisivo del éxito para la enseñanza estriba en *los procesos que median entre la enseñanza y el aprendizaje*, y éstos son procesos de carácter cognitivo que se desarrollan en el interior del alumno.

En este modelo se apuesta a considerar que el agente del aprendizaje es el alumno y el profesor un orientador o facilitador.

Los estudios más significativos sobre este modelo lo hallamos en Snow (1974), Olson y Bruner (1974), Rhotkopf (1976), Furth (1978) y Galagher (1979).

**Qué diferencias significativas se pueden observar?:**

<b>Ambientes de Aprendizaje Tradicionales</b>	<b>Nuevos Ambientes de Aprendizaje</b>
Instrucción centrada en el maestro	Aprendizaje centrado en el estudiante
Estímulo de un solo sentido	Estímulo multisensorial
Progresión por un solo camino	Progresión por muchos caminos
Medio de comunicación único	Medios Múltiples ("Multimedia")
Trabajo individual aislado	Trabajo cooperado
Transmisión de información	Intercambio de información
Aprendizaje pasivo	Aprendizaje activo/exploratorio basado en la indagación
Aprendizaje fáctico, basado en los saberes	Pensamiento crítico y toma de decisiones con conocimiento de causa
Respuesta reactiva	Acción proactiva/planeada
Contexto artificial, aislado	Contexto auténtico, del mundo real

El sistema educativo, se encuentra inmerso en un proceso de cambios, enmarcados en el conjunto de transformaciones sociales propiciadas por la innovación tecnológica y, sobre todo, por el desarrollo de las tecnologías de la información y de la comunicación, por los cambios en las relaciones sociales y por una nueva concepción de las relaciones tecnología-sociedad que determinan las relaciones tecnología-educación.

En la actualidad, los cambios en el contexto, exigen cambios en los modelos educativos, cambios en los usuarios de la formación y cambios en los escenarios donde ocurre el aprendizaje.

Ante ello, este grupo de investigación trabaja para el logro de una propuesta de un nuevo entorno de aprendizaje en ambientes colaborativos inteligentes.

**II.- Referencial Teórico**

Situándonos en el paradigma mediacional, nos preocupa y ocupa el desarrollo de ambientes colaborativos inteligentes.

Para ello es preciso definir en esta etapa del proyecto lo que se entiende por entorno virtual, por aprendizaje en "ambientes colaborativos".

**Qué es un Entorno Virtual?**

Una de las virtudes que caracterizan a los entornos virtuales, es que el usuario se siente formando parte de un mundo generado por un ordenador, estableciendo contacto con los diferentes objetos que componen estos mundos y estableciendo comunicación con otros usuarios como él o usuarios artificiales que habitan sólo dentro del mundo virtual.

**Podría definirse el aprendizaje colaborativo como:** El conjunto de métodos de instrucción y entrenamiento apoyados con tecnología así como de estrategias para propiciar el desarrollo de habilidades mixtas (aprendizaje y desarrollo personal y social), donde cada miembro del grupo es responsable tanto de su aprendizaje como del de los restantes miembros del grupo.

El aprendizaje es un proceso individual que puede ser enriquecido con actividades colaborativas tendientes a desarrollar en el individuo habilidades personales y de grupo.

### **Qué condiciones deben analizarse en el diseño de ambientes CSCL?**

David McConnell y un grupo de investigadores [ISBEN], en la conferencia internacional de *Computer Supported Collaborative Learning* (CSCL) del 95 compartieron lo que podrían ser condiciones claves para diseñar ambientes CSCL, sugieren tener en cuenta lo siguiente:

**a.- Apertura en el proceso educativo** : El estudiante debe estar en posición de tomar decisiones acerca de su aprendizaje y sentir que tiene la libertad para hacerlo. En CSCL un concepto que se aplica bien es el de comunidades de aprendizaje, las cuales se constituyen con miembros administrativos y estudiantes, quienes tienen igualdad de derechos para manejar los recursos de la comunidad y participar en los procesos de aprendizaje. En consecuencia deben tenerse los mecanismos para activar la apertura y la libertad.

**b.- Aprendizaje automanejado** : Cada persona toma la responsabilidad de identificar sus propias necesidades de aprendizaje, así como de ayudar a los demás a identificar las suyas, valorando la importancia de ofrecerse como un recurso flexible a la comunidad. En CSCL, un aspecto de automanejo es aprender cómo aprender. Este aprendizaje está inmerso en los procesos CSCL, asumiendo roles dentro del proceso y trabajando colaborativamente con otros miembros del grupo. La comunidad de aprendizaje comparte intereses, pero la escogencia del por qué y como se aprende es individual; si el grupo está trabajando (compartiendo, apoyándose, cuestionándolo), cada miembro del grupo constantemente estará profundizando sus niveles de aprendizaje y de conocimiento.

**c.- Un propósito real en el proceso cooperativo** : El aprendizaje grupal e individual requiere un propósito real en el proceso colaborativo; éste puede darse alrededor de la solución de un problema de interés grupal o individual, en cuyo caso, cada miembro del grupo define su propio problema y los otros integrantes del grupo ayudan a esa persona a resolverlo. El proceso de trabajar juntos tiene mucho en común con el ciclo natural de aprendizaje, acción e investigación: se inician una serie de acciones que al ser desarrolladas generan nuevas inquietudes y a su vez desencadenan nuevas acciones.

**d.- Un ambiente de aprendizaje soportado con computador** : Un aspecto importante de los ambientes CSCL es la necesidad de tener considerable interacción entre los miembros del grupo. Cada miembro del grupo debe sentir el apoyo del resto del grupo, para lo cual las redes virtuales apoyadas en tecnología de informática y comunicaciones permiten superar las barreras espacio temporales existentes entre los miembros de la red.

**e.- Evaluación del proceso de aprendizaje** : El proceso que se vive al interior del grupo debe estar sujeto a una evaluación constante personal y grupal, se debe tratar de desarrollar un sistema dinámico en el cual se hagan constantemente los ajustes necesarios para asegurar el buen desempeño del grupo, y de sus integrantes. En ambientes CSCL, se espera que la tecnología apoye: pensamiento creativo, autoaprendizaje, compromiso, responsabilidad, participación, organización, crecimiento individual y grupal. En ambientes heurísticos, abiertos y explorables.

### **Qué aportan estos ambientes en educación?**

El aprendizaje en ambientes colaborativos, busca propiciar espacios en los cuales se dé el desarrollo de habilidades individuales y grupales a partir de la discusión entre los estudiantes al momento de explorar nuevos conceptos, siendo cada quien responsable de su propio aprendizaje. Se busca que éstos ambientes sean ricos en posibilidades y más que organizadores de la información propicien el crecimiento del grupo. Diferentes teorías del aprendizaje encuentran aplicación en los ambientes

colaborativos; entre éstas, los enfoques de Piaget y de Vygotsky basados en la interacción social. Lo innovador en los ambientes colaborativos soportados en redes virtuales es la introducción de la informática a estos espacios, sirviendo las redes virtuales de soporte, lo que da origen a los ambientes CSCL (Computer-Support Collaborative Learning - Aprendizaje colaborativo asistido por computador).

Por otra parte, el Aprendizaje en estos ambientes propician espacios en los cuales se dé la discusión entre los estudiantes al momento de explorar conceptos que interesa dilucidar o situaciones problemáticas que se desea resolver; se busca que la combinación de situaciones e interacciones sociales pueda contribuir hacia un aprendizaje personal y grupal efectivo. La preocupación del aprendizaje colaborativo gira en torno a la experiencia en sí misma, más que a los resultados esperados. Se espera que el ambiente sea atrayente y significativo para cada uno de los miembros del grupo. Por lo tanto se maneja un tipo de motivación intrínseca antes que extrínseca.

Por otra parte, permiten la resolución de problemas en forma colaborativa, pudiendo ser aplicados en una diversidad de áreas temáticas.

Sin duda un aspecto a favor de las comunicaciones por medio de computadoras, es la posibilidad de tener acceso a numerosos recursos pedagógicos o de otro tipo disponibles en Internet. Muy en particular para el caso de estudiantes que trabajan en problemas similares y se hallan en diferentes ciudades con la posibilidad casi nula de que se encuentren, si no comparten un entorno virtual a través del cual comunicarse.

### III.- Nuestra Propuesta

En función de lo precedente, el prototipo que se halla en desarrollo y desde la perspectiva pedagógico didáctica se observaría del siguiente modo:

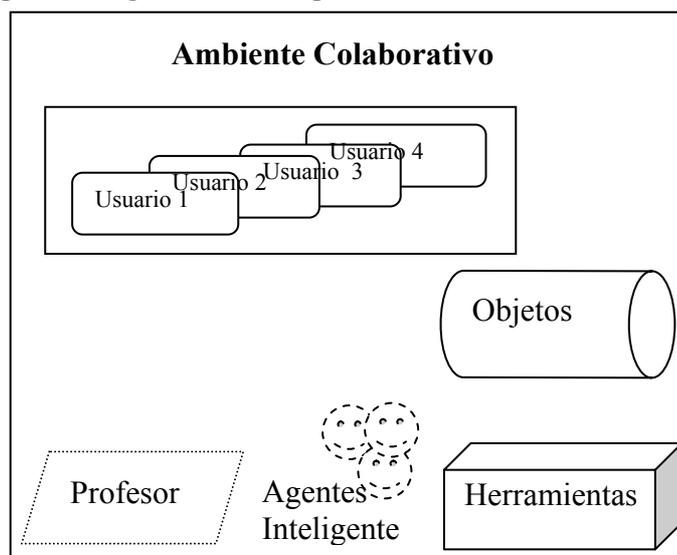
#### Componentes que intervienen en el ambiente

1.- Grupo Colaborativo formado por 4 integrantes, en el que cada uno tendrá un rol diferenciado.

2.- El Profesor responsable del dictado del curso se encarga de supervisar e interviene en los momentos que considera conveniente.

3.-Objetos a compartir (bibliografía, material en la web, Unidades de aprendizaje).

4.- Herramientas : email, chat, foros, pizarra, etc.



5.- Agentes Inteligentes: unos que acompañan al alumno en su estadía en el ambiente; otros que monitorean su actividad, siendo insumo para la tarea tutorial

Como se puede observar en la figura, se agregan a estos ambientes colaborativos, los agentes inteligentes que cumplen roles diferenciados y que tienen como objetivo complementar la tarea tutorial.

### **A modo de Conclusión:**

Si bien se halla expresado el Referencial Teórico, aún es prematuro ubicar el modelo en alguna de las Teorías Psicopedagógicas que se conocen, ya que dependerá del comportamiento de la plataforma en cuanto a los tipos de aprendizaje que posibilite.

Por otra parte, exige una redefinición del rol docente, de la participación de los alumnos, del material didáctico que deberá acompañar estos ambientes, quién y cómo deben ser elaborados, del sistema de evaluación permitirá esta plataforma, etc.

Finalmente, importa destacar respecto al ambiente aquí descrito, que en metas más avanzadas del proyecto, se analizarán los roles y funciones de los agentes inteligentes con los que contará el prototipo, los cuales han sido pensados para tener propiedades tales como: *autonomía, habilidad social, reactividad, pro-actividad*, etc.

### **Bibliografía**

- JENNY PREECE *et al* *Human-computer interaction* Addison-Wesley 1994
- Daniel D. Suthers *Computer Aided Education and Training Initiative* 1998.  
<http://advlearn.lrdc.pitt.edu/advlearn/papers/FINALREP.html>.
- Amy Bruckman *Programming for fun* ..1994.  
<http://www.cc.gatech.edu/~asb/papers/index.html#NECC>
- UCROS, Maria Eugenia. *Material del Curso Trabajo Cooperativo*. 1997
- ENLACES- RED INSTITUCIONAL. *Aprendizaje Cooperativo apoyado por computadores*. Centro Zonal Sur-Austral Proyecto Enlaces. Universidad de la Frontera. Mayo 1996.
- JOHNSON, David W (Et. All). *Circles Learning Cooperation in the Classroom*. 1988
- GALVIS, Alvaro H. *Ingeniería de Software Educativo*. Ediciones Uniandes. 1992.
- ISBEN, David J.S. *Collaborative Information Networks*. Conferencia internacional CSCL, 1995
- D.W. JOHNSON Y R. JOHNSON Y E. J. HOLUBEC – *El aprendizaje Cooperativo en el aula*. Ed. Paidós Educador. Bs.As. 1999.-
- COLEMAN, David. *Groupware - The Changing Environment*. Chapter 1
- DAVID MCARTHUR, MATTHEW LEWIS, AND MIRIAM BISHAY. - *The Roles of Artificial Intelligence in Education: Current Progress and Future Prospects*. -  
<http://www.rand.org/hot/mcarthur/Papers/role.html>.
- SALINAS, J. (1995): *Organización escolar y redes: Los nuevos escenarios del aprendizaje*, en CABERO, J. Y MARTINEZ, F. (Coord.): *Nuevos canales de comunicación en la enseñanza*. Centro de Estudios Ramón Areces, Madrid. 89-117
- STONIER, T. (1989): *Education: Society's number-one enterprise*. En PAINE, N. (ED.): *Open Learning in transition*. London: Kogan Page p.14-37
- TOPHAM, P. (1989): *The Concept of "Openness" in relation to Computer Based Learning Environments and Management Education*. *Interactive Learning International* Vol.5(1), 157-163.
- ESTEVE, José M. *La formación inicial de los Profesores de secundaria. Una reflexión sobre el curso de cualificación pedagógica*. Ed. Ariel S.A., Barcelona, 1997.-
- ALONSO, C., GALLEGO, D. y HONEY, P. *Los estilos de Aprendizaje. Procedimientos de diagnóstico y mejora*. Ed. Mensajero S.A. 1995.

## Entorno Web para la enseñanza del Método de Programación Lineal

*López, María Victoria; Golobisky, María Fernanda; Mariño, Sonia Itati*

*Departamento de Informática*

*Facultad de Ciencias Exactas y Naturales y Agrimensura*

*Universidad Nacional del Nordeste*

*9 de Julio N° 1449 - 3400 Corrientes - Argentina*

*TE (03783) 423126/423968/424606 - Fax (03783) 423968*

*mvlopez@exa.unne.edu.ar, mfgolo@exa.unne.edu.ar, msonia@exa.unne.edu.ar*

### Resumen

Este trabajo tiene como objetivo el desarrollo de una aplicación Web de tipo tutorial que permitirá transmitir a los estudiantes los contenidos temáticos referentes al Método de Programación Lineal, que conforman el Programa de la cátedra Investigación Operativa de la carrera de Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas y Naturales y Agrimensura (UNNE).

Se presentan los resultados consistentes en el diseño e implementación de esta herramienta.

Se logró un soporte educativo de uso masivo, que permite transmitir una cantidad importante de conceptos teóricos, que luego podrán ser aplicados para la obtención de resultados prácticos concretos.

Se pretende proponer el empleo de esta aplicación Web como recurso didáctico complementario en el dictado de la mencionada asignatura.

**Palabras Clave:** Educación. Programación Lineal. Hipertexto. WWW. Tutorial. Software Educativo. Aprendizaje asistido por computadora.

## Introducción

Jordi parafrasea a González, Gisbert et al. (1996), conceptualizando a las "nuevas tecnologías de la información y la comunicación" como el conjunto de procesos y productos derivados de las nuevas herramientas (hardware y software), soportes de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información.

Las tecnologías de la información tienen un gran potencial en todos los ámbitos de la educación, desde básica a universitaria, incluyendo educación continua y capacitación técnica, ya sea presencial o a distancia [Baeza].

La enseñanza superior se ha basado fundamentalmente en el método particular de cada uno, en aquello que el profesor hacía en el aula, y a menudo se confundía con la libertad de cátedra, con los conocimientos sobre estrategias metodológicas de enseñanza y aprendizaje [Duart]. Jordi respecto a esta cuestión menciona los cambios generados en "...la perspectiva tradicional en educación superior, por ejemplo, del profesor como única fuente de información y sabiduría y de los estudiantes como receptores pasivos debe dar paso a papeles bastante diferentes".

Las nuevas tecnologías de la información y la comunicación (TIC) aplicadas en la educación superior presentan nuevas oportunidades que deben ser valoradas. Principalmente, estas tecnologías permiten crear comunidades virtuales, es decir, espacios de interacción humana en los cuales el espacio y el tiempo, como coordenadas reales para cada uno de los miembros de la comunidad, pueden relacionarse de forma asincrónica con las relaciones entre cada uno de los miembros y constituir un entorno virtual. Un entorno virtual de aprendizaje permite aprender sin coincidir en el tiempo ni en el espacio y asume las funciones de contexto de aprendizaje que en los sistemas de formación presencial desarrolla el aula [Duart].

Usar la informática como apoyo a procesos de aprendizaje ha sido una inquietud que durante mucho tiempo ha sido investigada y probada por muchas personas. Su asimilación dentro de instituciones educativas, incluyendo el hogar, ha aumentado en los últimos años, con lo que la demanda por software educativo de alta calidad es cada vez mayor [Gom].

**Este trabajo tiene como objetivo el desarrollo de un software hipermedial de tipo tutorial que permitirá transmitir a los estudiantes los contenidos temáticos referentes al Método de Programación Lineal**, que conforman el Programa de la cátedra Investigación Operativa de la carrera de Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas y Naturales y Agrimensura (UNNE).

En lo referente al Método de Programación Lineal, en muchas situaciones reales se plantean limitaciones o restricciones, que varían de acuerdo al área analizada. Así, en el campo industrial se presenta el problema de contar con una existencia fija de materia prima almacenada, que entra en la fabricación de varios productos. Esto constituye una restricción, ya que limita la cantidad de cada producto a producir, sobre todo teniendo en cuenta que el incremento de uno ocasiona una reducción en los otros. Si se desea fabricar un alimento balanceado (problema que nos ocupa el presente trabajo), el contenido de proteínas y materias grasas, que oscila generalmente entre ciertos límites bien establecidos, constituye también una restricción del problema. Es decir entonces, que hay una cierta limitación de recursos, ya sean éstos materia prima o factores, tales como determinadas características del elemento a producir.

Por otro lado, el problema es de naturaleza tal que se presentan distintos productos que emplean el mismo recurso, y lo que se desea hallar es una combinación tal de productos que satisfaga las restricciones del problema. Las relaciones existentes entre los diferentes productos que emplean los mismos recursos, pueden ser lineales o no. En este estudio se parte de la base de que dichas relaciones son lineales; es decir, están expresadas por ecuaciones y desigualdades de primer grado.

Además, cada producto genera una cierta ganancia o involucra un determinado costo.

**El Problema de Programación Lineal surge cuando se cumplen las tres condiciones descriptas, a saber:**

- Restricciones, que hacen que los productos compitan entre si. Los productos (o variables del problema) deben cumplir indefectiblemente la condición de ser no negativos, ya que no tiene sentido determinar cuanto hay que fabricar de un producto, estando expresada esa cantidad por una cifra negativa.
- Relaciones lineales entre los productos o variables intervinientes.
- Optimización de la función económica del conjunto de productos, que establece que se deben maximizar los beneficios o minimizar los costos [Gass].

A pesar de los grandes adelantos en la optimización computacional ocurridos durante los últimos 20 años, **el método Simplex inventado por George B. Dantzig en 1947 es aún la herramienta principal en casi todas las aplicaciones de la programación lineal.** Este método provee un sistema rápido y efectivo para la resolución de los problemas de Programación Lineal. Es la metodología empleada en las aplicaciones prácticas, y permite resolver una gran cantidad de problemas de real importancia industrial. Su desarrollo parece complicado al principio, pero planteado el cuadro inicial correspondiente, el sistema se torna completamente mecánico, lo cual hace que su resolución sea perfectamente factible por medio de computadoras. El sistema llega a la solución óptima por medio de iteraciones o pasos sucesivos, ya que no existe ningún método o fórmula que alcance directamente esta solución [Piza].

## **Metodología empleada para el desarrollo de la aplicación Web tipo tutorial**

Se han considerado los siguientes pasos:

1. **Evaluación y estudio de las herramientas para la construcción de software educativo.** Generalmente, las tecnologías informáticas empleadas para el diseño y la elaboración de material educativo interactivo son los libros multimediales y los tutoriales hipermediales, implementados mediante la tecnología WWW (World Wide Web).
2. **Selección de los contenidos temáticos específicos.** Se incluyeron, entre otros, los siguientes temas: formulación general de un problema de programación lineal (condiciones de ligadura y no negatividad, restricciones, función objetivo, solución básica factible no degenerada, solución óptima), interpretación y solución gráfica, métodos algebraicos de soluciones, el método Simplex (metodología de transformación de la matriz, elemento pivote, variables auxiliares flotantes y artificiales), el problema dual, programación lineal paramétrica (análisis con variación de los coeficientes de la función objetivo, estudio de sensibilidad, cálculo del simplex con coeficientes parametrizados, análisis con variación de los coeficientes de las restricciones).
3. **Diseño e implementación de una aplicación Web de tipo tutorial en la (me parece que el “la” queda bien aquí) que se presentan los contenidos teóricos mencionados y ejercitación complementaria.** No basta con conocer la tecnología o los últimos avances en multimedia para desarrollar un buen material educativo. Hace falta diseñarlo de acuerdo a los objetivos de aprendizaje que se quieren alcanzar [Duart]. El Sistema de enseñanza-aprendizaje construido consiste en un tutorial hipermedial que actúa como eje integrador de los contenidos temáticos. Desde la página principal del sitio se dispone de un índice que se mantiene visible a lo largo de la navegación por el tutorial. El alumno puede acceder a los contenidos teóricos que desee, y cuanta además con una página de ayuda en línea en donde se recomienda el modo de navegación adecuado con el objeto de obtener el máximo rendimiento.

No obstante cabe destacar que el proceso de elaboración del software educativo no es un proceso lineal, sino iterativo. En determinados momentos de la realización se comprueba el funcionamiento, el resultado, se evalúa el producto... y frecuentemente se detecta la conveniencia de introducir cambios [Mar].

## Herramientas informáticas y lenguajes de programación utilizados

Se seleccionó el lenguaje de hipertextos **HTML**, para el desarrollo de la aplicación **Web**. Este es el lenguaje standard que “entiende” la World Wide Web (WWW) para crear y reconocer hipertextos. El lenguaje HTML está constituido por una serie de comandos que permiten:

- Definir la estructura lógica del documento.
- Incluir hiperenlaces.
- Dar formato al texto.
- Añadir otro tipos de elementos no textuales.

El desarrollo de la Informática en los últimos años, ha permitido la implementación del concepto de hipertexto, que se define como un documento multimedia con relaciones estructurales.

- **Documento multimedia:** indica que no se compone únicamente de texto, sino que contiene información en otros formatos (gráficos, sonido, video, etc.).
- **Relaciones estructurales:** posee enlaces (hiperenlaces o links) a otros documentos y en general a cualquier tipo de información residente localmente o en sistemas remotos. La información se estructura en forma no lineal.

Los hipertextos constituyen una verdadera revolución en la concepción del almacenamiento y presentación de la información, permitiendo:

- Almacenamiento de un gran volumen de información.
- Facilidad de acceso y consulta.
- Presentación de una forma mas agradable.
- Navegación a través de la información de una forma individualizada.
- Estructuración no lineal de la información.
- Adquisición de conocimientos de manera más participativa.
- Interacción entre la información y quien la consulta [Cobo].

## Resultados

El diseño educativo elaborado de acuerdo con criterios pedagógicos es la garantía para producir materiales didácticos multimedia de calidad y plenamente educativos [Duart].

Se desarrolló una aplicación Web tipo tutorial que permitirá a los estudiantes acceder a los contenidos temáticos referentes a la Programación Lineal, en una forma amena y ágil. Estos contenidos conforman el Programa de la cátedra Investigación Operativa de la carrera de Licenciatura en Sistemas de Información de la Facultad de Ciencias Exactas y Naturales y Agrimensura (UNNE).

Los programas tutoriales son programas que en mayor o menos medida dirigen, tutorizan, el trabajo de los alumnos. Pretenden que, a partir de información y mediante la realización de ciertas actividades previstas de antemano, los estudiantes pongan en juego determinadas capacidades y aprendan o refuercen conocimientos y/o habilidades [Mar].

Esta aplicación Web presenta una interfase amigable e interactiva. El software obtenido incorpora aspectos didácticos y pedagógicos que garantizan la satisfacción de las necesidades educativas, además de facilitar los procesos de enseñanza y de aprendizaje de manera productiva. Los contenidos son significativos para los estudiantes y están relacionados con situaciones y problemas de su interés. Asimismo, este software permite fomentar la práctica de contenidos aprendidos en el aula y la evaluación de los aprendizajes.

## Conclusiones

En este trabajo se describen las principales características de una herramienta Web orientada a la transmisión de los contenidos teóricos referentes a la Programación Lineal. La aplicación puede ser utilizada como material áulico, y presenta las siguientes ventajas:

- Proporciona a los alumnos una forma de estudio atractiva, en un entorno familiar y fácil de acceder, que los lleva a concentrarse en el contenido.
- El estudiante elige la forma de navegar por los contenidos ya que no existen secuencias predeterminadas para el acceso a la información.
- Posee una interfaz amigable de diseño simple apropiada para la comunicación, que proporciona fácil accesibilidad y rapidez de percepción.
- Permite incorporar las nuevas tecnologías y medios de comunicación a la formación y al proceso enseñanza/aprendizaje.
- Posibilita mejorar sustancialmente la calidad de la educación, evitando que los alumnos se distraigan (como ocurre a veces en el aula), y adaptándose a las particularidades de la enseñanza mediante la adecuación de los contenidos en cada momento.

Actualmente se está trabajando en:

- construcción de las animaciones que permitan realizar prácticas interactivas.
- programación de los sistemas de evaluación y seguimiento, con informes de las actividades realizadas por los estudiantes: temas, nivel de dificultad, tiempo invertido, errores, itinerarios seguidos para resolver los problemas.

En el futuro se propondrá el empleo de esta aplicación Web como recurso didáctico complementario en el dictado de la asignatura Investigación Operativa de la carrera de Licenciatura en Sistemas de Información de la Fac. de Cs. Exactas y Nat. y Agrimensura (UNNE).

## Referencias

- [Baeza] Baeza, R. Educación y Tecnología. <http://www.dcc.uchile.cl/~rbaeza/inf/educacion.html>
- [Cobo] Cobo Ortega, A. 1998. "Curso de Introducción a Internet". Universidad de Cantabria. España.
- [Duart] Duart, J. M y Sangrá, A. 2000. Aprender en la virtualidad. Ed. Gedisa.
- [Gass] Gass, Saúl I. 1977. "Programación Lineal. Métodos y aplicaciones". Compañía Editorial Continental S.A. México.
- [Gom] Gómez Castro, R. A.; Galvis Panqueva, A. H. y Mariño Drews, O. "Ingeniería de software educativo con modelaje orientado por objetos: un medio para desarrollar micromundos interactivos". <http://www.minerva.uevora.pt/simposio/comunicacoes/rigomezmarino.html>
- [Mar] Marquès, P. "El software educativo". [http://www.doe.d5.ub.es/te/any96/marques\\_software](http://www.doe.d5.ub.es/te/any96/marques_software)
- [Piza] Piza Dávila, Alejandro. 1994. Sitio web "George Dantzig: Fundador de la Programación Lineal".

**Aplicación del Método de Montecarlo para el cálculo de integrales definidas**

*López, María Victoria y Mariño, Sonia Itatí*

*Departamento de Informática*

*Facultad de Ciencias Exactas y Naturales y Agrimensura*

*Universidad Nacional del Nordeste*

*9 de Julio n° 1449. CP: 3400. Corrientes. Argentina*

*TE (03783) 423126/423968/424606 - Fax (03783) 423968*

*mvlopez@exa.unne.edu.ar; msonia@exa.unne.edu.ar*

**Resumen:**

En este trabajo se presenta un software educativo, desarrollado en *Mathematica*, para el cálculo de integrales definidas mediante el Método de Simulación o de Montecarlo.

El programa realiza llamadas a un paquete en *Mathematica* que lleva a cabo la generación de los números aleatorios.

Mediante el uso del software desarrollado los alumnos podrían adquirir habilidad y destreza en el manejo de modelos de simulación computarizados y afianzar los conceptos teóricos sobre simulación de sistemas adquiridos en el aula.

**Palabras clave:** Método de Simulación o de Montecarlo. Mathematica. Aprendizaje asistido por computadora. Software educativo.

### Introducción

Muchos problemas que se presentan en la práctica, carecen de solución dentro del campo analítico, es decir, de la matemática pura.

En este trabajo se considera el problema particular que consiste en el cálculo del valor de una integral definida. La imposibilidad o inconveniencia de la aplicación de los métodos tradicionales para la solución de este problema está dada, fundamentalmente, por las siguientes causas:

- Que no se conozca ninguna función primitiva de aquella que se desea integrar.
- Que aún conociéndose una función primitiva, resulte excesivamente compleja o extensa su aplicación.

Esto fundamenta el estudio de métodos más o menos precisos, no tradicionales, fundamentalmente numéricos, que permiten el cálculo de integrales definidas. Para resolver el problema, resulta necesaria una aproximación a una expresión del tipo

$$I = \int_a^b f(x) dx$$

que resulte comprensible por la computadora, mediante algún método numérico. Es posible realizarlo de muchas maneras diferentes (fórmula de los Trapecios, fórmula de Simpson, etc.) [Pace94].

En este trabajo se emplea el **método de simulación o de Montecarlo**. La simulación consiste en iterar un proceso lógico, expresado en términos de ecuaciones matemáticas y operaciones lógicas, constitutivas de algún modelo matemático cuyos coeficientes son parámetros que varían según un criterio determinado. La estructura iterativa o el carácter combinatorio de la simulación lo da la sistemática variación de sus parámetros en cada paso o iteración, cosa que se puede realizar haciendo intervenir al azar, o no, según el tipo de fenómeno que se trata de simular y el modelo utilizado para describirlo. Cuando al menos una de las variables que intervienen en el modelo es de tipo aleatorio, se estará en presencia de una simulación estocástica, en caso contrario, cuando todas las variables que intervienen en el modelo son deterministas, será una simulación determinista. En este trabajo se emplea la simulación determinista.

Para llevar a cabo este proceso de simulación, en la práctica generalmente se requieren sucesiones de números aleatorios. Reciben el nombre de números aleatorios o números al azar todas aquellas sucesiones de dígitos equiprobables, entre el cero y el nueve, ubicados aleatoriamente en toda su extensión [Pace95]. Uno de los requisitos principales que se imponen sobre las sucesiones obtenidas mediante procesos aleatorios de muestreo es la independencia estadística [Freund]. Se requiere disponer de un medio eficaz para generar los números aleatorios o al azar, a medida que son necesitados por el modelo matemático que representa el sistema [Pace95].

**Este trabajo tiene como objetivo la aplicación del Método de Simulación o de Montecarlo para el cálculo de integrales definidas. Para ello, se ha desarrollado en lenguaje *Mathematica* un paquete que permite calcular la integral definida de cualquier función dada. El programa realiza llamadas a otros subprogramas que llevan a cabo la generación de los números aleatorios.**

La simulación permite en algunos casos el experimentar situaciones, del tipo qué pasa si..., donde se ensayan las hipótesis y aparece el resultado [Rivera]. Finalmente, resulta interesante el desarrollo de un sistema que permita acceder a los alumnos a técnicas complejas desde el punto de vista numérico, como el método de Montecarlo.

### Metodología

El lenguaje de programación empleado es *Mathematica* [Castillo]. Se empleó un paquete propio de este programa para realizar las representaciones gráficas. Para la generación de los números aleatorios se accedió a un paquete [López] [Pace99] desarrollado en *Mathematica* que incluye el

Método de los Cuadrados Centrales de Vonm Neumann [Pardo], Método de Fibonacci [Pace95], Método Aditivo de Congruencias [Pardo], Método Multiplicativo de Congruencias [Coos], Método Mixto de Congruencias [Coos].

La metodología llevada a cabo para el cálculo de las integrales mediante el Método de Montecarlo consiste en:

1. Desarrollar un procedimiento para la generación de números aleatorios. En este caso se utiliza alguno de los procedimientos del paquete en *Mathematica* mencionado.
2. Inscribir la superficie encerrada entre la curva, el eje de las abscisas y las ordenadas extremas a y b, en un rectángulo determinado por la base (b-a) y la altura (d-c).
3. Ejecutar el procedimiento de generación de números aleatorios dos veces. Una de ellas para generar los números comprendidos en un intervalo para las "x" [a,b] y una segunda vez para generar los números comprendidos en un intervalo para las "y" [c,d]. Estos números son considerados de a pares ordenados y todos caen dentro del rectángulo considerado.
4. Determinar el valor de S, que indica la cantidad de puntos que caen dentro de la superficie determinada por la integral, contando los pares ordenados (x, y) que cumplen las condiciones establecidas.
5. Al finalizar las n iteraciones del proceso, calcular el valor de la integral de la siguiente manera:

$$\frac{Area}{(b-a)*(d-c)} = \frac{S}{N} \Rightarrow Area = \frac{S*(b-a)*(d-c)}{N}$$

donde:

N = Cantidad de puntos generados dentro del rectángulo.

S = Cantidad de puntos que caen dentro de la superficie correspondiente a la integral.

### Resultados

Se ha desarrollado un programa en *Mathematica* para el cálculo de integrales por el Método de Simulación. Este programa permite al usuario estimar el valor de la integral definida de una función dada como argumento, en un intervalo numérico también proporcionado como argumento, para los casos siguientes:

- la función toma valores positivos (Figura 1)
- la función toma valores negativos
- la función toma valores positivos y negativos (Figura 2)

El procedimiento en *Mathematica* requiere el ingreso de los siguientes parámetros para su ejecución:

n: número de las iteraciones de la simulación a realizar.

num: número semilla para inicializar el generador de números aleatorios.

n1: extremo inferior del intervalo de integración.

n2: extremo superior del intervalo de integración.

f: expresión de la función a integrar.

x: variable de integración.

Luego de sucesivas ejecuciones del procedimiento, se observó el valor de la superficie es más preciso, a medida que aumenta el número de iteraciones del proceso de simulación. En las Figuras 1 y 2 se muestran el comando que invoca al programa y los resultados obtenidos, para dos funciones ejemplo.

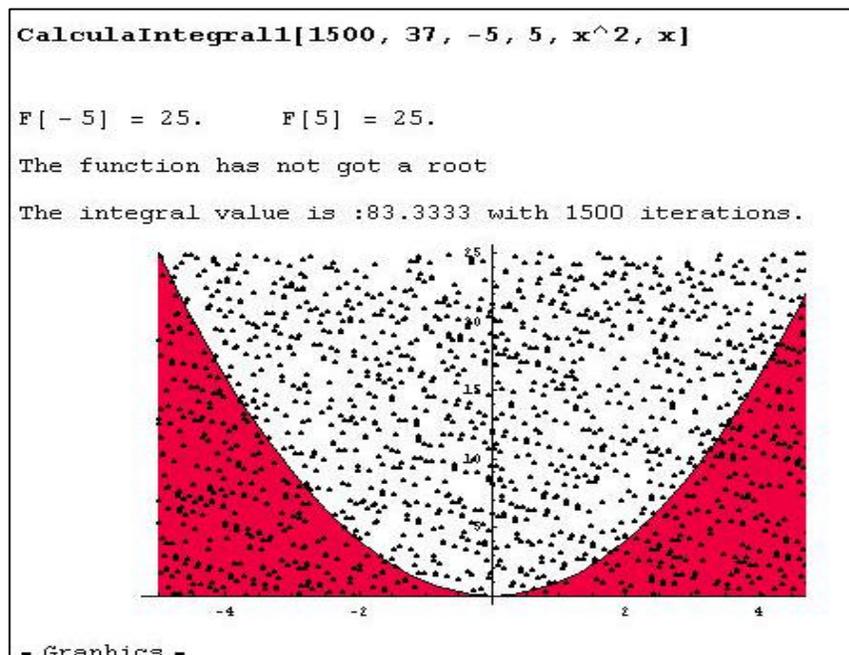


Figura 1.- Resultados obtenidos en la ejecución del programa CalculaIntegral, para la función  $y=x^2$ , con 1500 iteraciones

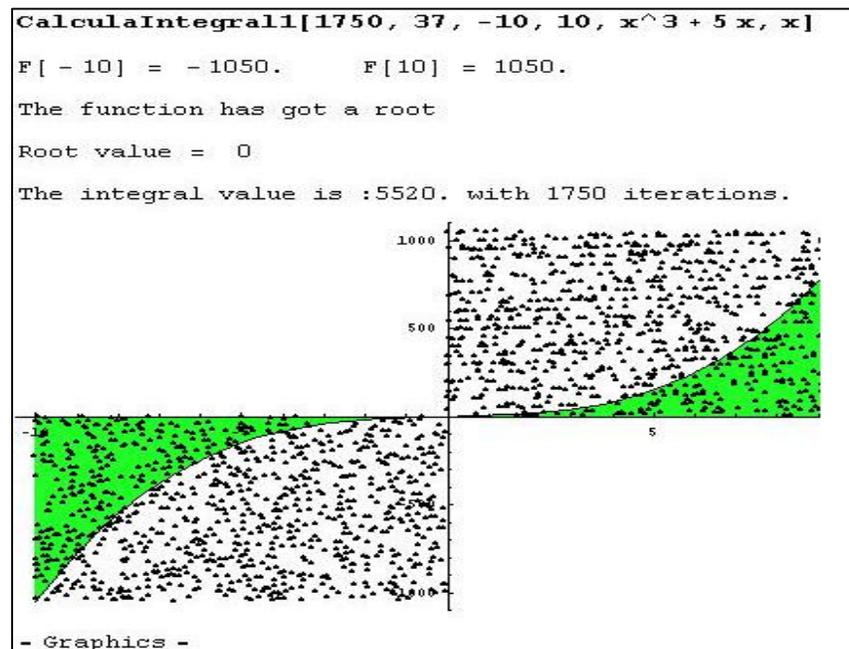


Figura 2.- Resultados obtenidos en la ejecución del programa CalculaIntegral, para la función  $y=x^3+5x$ , con 1750 iteraciones

### Conclusiones:

En este trabajo se presenta un software educativo, desarrollado en *Mathematica*, para el cálculo de integrales definidas mediante el Método de Simulación o de Montecarlo. El mismo permite comprobar la eficiencia del Método de Simulación en la resolución de problemas de este tipo. Se observa que con *Mathematica* es posible desarrollar los procedimientos de una manera sencilla,

disminuyendo las líneas de código, los tiempos de desarrollo, eliminando instrucciones repetitivas, empleando instrucciones específicas y obteniendo una buena calidad en las representaciones gráficas con muy pocos comandos. Se concluye en que las capacidades computacionales y gráficas de *Mathematica* lo convierten en una herramienta excelente para el desarrollo de simulaciones y su implementación con fines pedagógicos.

Se tiene pensado crear en el futuro el sitio Web de la asignatura “Modelos y Simulación” de la carrera de Licenciatura en Sistemas de Información de la Fac. de Cs. Exactas y Nat. y Agrimensura (UNNE). Desde esta página Web los alumnos podrán acceder a los contenidos teóricos de la materia, a las guías de trabajos prácticos, y a series de problemas complementarios. Asimismo, podrán establecer contacto con los docentes de la cátedra, realizar consultas, etc. El software desarrollado en este trabajo podría descargarse desde este sitio y mediante su utilización los alumnos podrían verse incentivados a:

- Adquirir habilidad y destreza en el manejo de modelos de simulación computarizados.
- Efectuar todos los pasos del procesamiento de modelos matemáticos mediante la simulación, no limitándose a la primera etapa de simple construcción de diagramas de flujo de los algoritmos correspondientes a los ejercicios prácticos. De esta manera se enfrentarán a dificultades tales como la selección de los parámetros iniciales, la ejecución de las corridas y el posterior análisis de los datos simulados.
- Afianzar los conceptos teóricos y técnicas utilizados en la simulación de sistemas, adquiridos en el aula.

### **Bibliografía:**

- [Castillo] Castillo, E., Iglesias, A., Gutiérrez, J. M., Álvarez, E. y Cobo, A. 1996. “Mathematica”. Ed. Paraninfo. Madrid, España. 3ra. Ed. 534pp.
- [Coos] Coos, R. 1992. “Simulación un enfoque práctico”. Limusa Grupo Noriega Editores. México. 157 pp.
- [Freund] Freund, J. E. 1962. “Mathematical Statistics”. Englewood Cliffs. Prentice-Hall.
- [López] López, María V., Mariño, Sonia I. y Petris, Raquel H. 1999. “Un análisis comparativo de generadores de números pseudo-aleatorios en Mathematica 3.0”. FACENA. Revista de la Facultad de Cs. Exactas y Nat. y Agrimensura. Univ. Nac. del Nordeste. Vol.15. Corrientes. Argentina.
- [Pace94] Pace, G. J. 1994. Métodos Numéricos. Editorial Universitaria de la Universidad Nacional del Nordeste. 310 pp.
- [Pace95] Pace, G. J. 1995. Material didáctico de la Cátedra “Modelos y Simulación”. FACENA. UNNE. Inédito. Corrientes. Argentina.
- [Pace99] Pace, G. J., López, María V., Mariño, Sonia I. y Petris, Raquel H. 1999. “Programación de un paquete de simulación con Mathematica”. Reunión de Comunicaciones Científicas y Tecnológicas. UNNE. Corrientes. Argentina. Tomo VIII. Cs. Exactas. 8:9-12.
- [Pardo] Pardo, L. y Valdés, T. 1987. “Simulación. Aplicación práctica en la empresa”. Ed. Díaz de Santos S. A. Madrid. España. 293 pp.
- [Rivera] Rivera Porto, E. “Aprendizaje asistido por computadora. Diseño y realización”. <http://msip.lce.org/erporto/libros/edu2>.

## **Un paquete de Mathematica para el aprendizaje de métodos de muestras artificiales de variables aleatorias no uniformes**

Sonia I. Mariño

Departamento de Informática. Facultad de Ciencias Exactas y Naturales y Agrimensura.  
Universidad Nacional del Nordeste. 9 de Julio N° 1449. CP: 3400. Corrientes. Argentina  
msonia@exa.unne.edu.ar

### **1. Introducción**

El uso de las nuevas tecnologías y el diseño interactivo de los materiales didácticos permiten la relación básica entre los diferentes elementos que forman cualquier proceso formativo: el estudiante, el profesor y los materiales complementarios (Duart & Sangrá 2000).

En la educación, el empleo de material interactivo es extendido (Miner and Topping 2001) donde es frecuentemente empleado como complemento virtual a la enseñanza tradicional.

En el desarrollo de material educativo implementado en la modalidad de software intervienen dos agentes: los docentes y los alumnos. El software educativo, proporciona al alumno una herramienta para afianzar las habilidades objeto de la enseñanza y a los docentes una herramienta de evaluación en un dominio del conocimiento.

El propósito de este trabajo es presentar un software educativo o entorno virtual de aprendizaje destinado a la enseñanza y/o profundización de métodos especiales para la construcción de muestras artificiales de variables aleatorias no uniformes.

En ocasiones el estudio de un modelo es tan complejo que un análisis matemático exacto no es posible y en otros casos no existen técnicas aproximadas de análisis razonables. En estas situaciones es necesario representar el modelo, mediante la simulación discreta en computadoras (Lavenberg 1983).

La simulación generalmente involucra alguna clase de modelo o una representación simplificada de la realidad (Roberts et al. 1983). Una vez formulado un modelo, el programa de simulación se escribe y se ejecuta con el objeto de obtener las estimaciones y evaluar el comportamiento de las variables de salida.

Una variable aleatoria es una función que asigna valores numéricos a los resultados de un experimento del azar (Lavenberg 1983).

Debido a que la mayoría de las simulaciones incorporan alguna forma de aleatoriedad en los valores de las variables de entrada, las entradas de un modelo constituyen las muestras independientes de distribuciones específicas y las salidas, producidas por los modelos, son valores simulados.

La mayoría de los lenguajes de programación incluyen funciones generadoras de variables aleatorias uniformes. Algunos, proporcionan funciones para generar muestras artificiales de las distribuciones más usadas (Lavenberg 1983; Bratley et al. 1983), sin embargo funcionan como cajas negras.

Este software educativo proporciona algoritmos específicos para la generación de muestras independientes de distintas distribuciones, así como una descripción de los métodos especiales empleados. Se pretende que la adopción del mismo facilite a los estudiantes el aprendizaje, la asimilación de los conceptos teóricos y los procedimientos específicos para generar muestras artificiales de variables pertenecientes a distribuciones discretas y continuas.

Este sistema educativo interactivo puede ser considerado como un modelo para la enseñanza de la matemática aplicada en carreras de sistemas de información, por lo que se considera relevante la incorporación de herramientas tecnológicas como son los lenguajes de programación utilizados para la implementación de las simulaciones.

## 2. Herramienta para el aprendizaje de muestras artificiales

El módulo de prácticas interactivas se desarrolló empleando Mathematica (Castillo et. al. 1996; Wolfram 1996). Mathematica, es un lenguaje de programación versátil que facilita el desarrollo de diversos tipos de cálculos de la manera más natural y lógica. Su elección, como herramienta de implementación, se basó en las siguientes ventajas:

- Programación mínima de código, el usuario dispone de una biblioteca de funciones predefinidas y aplicable a una gran variedad de problemas.
- Orientación a la programación de módulos, por medio de la codificación de procedimientos o funciones.
- Reusabilidad de módulos y funciones mediante la generación de paquetes.
- Legibilidad y facilidad en la modificación y depuración de los algoritmos.

## 3. Métodos implementados en el paquete

### a.- Paquete de Mathematica empleado

La generación de números pseudoaleatorios, uniformemente distribuidos en un intervalo específico es fundamental en la simulación (Bratley 1983). La mayoría de los programas presentan funciones para generar números aleatorios.

En la generación de los números pseudoaleatorios se empleó un paquete de Mathematica previamente desarrollado (López et al. 1999). Este paquete incluye el método de Von Neumann (Pardo 1987), Fibonacci (Pace 1995), el método aditivo, el método multiplicativo y el método mixto de las congruencias (Coos 1992; Pace 1995) y la prueba de hipótesis de Chi Cuadrado (Pace 1995) empleada para la validación de la serie de números pseudoaleatorios.

### b. Distribuciones no-uniformes

En la construcción de las muestras artificiales de las variables aleatorias no uniformes discretas y continuas, se consideraron procedimientos generales como: el método de la transformada inversa, el método del rechazo, el método de la composición, el método de la aproximación funcional a la transformada inversa y otros procedimientos especiales (Bratley et al. 1987).

El sistema educativo, incluye métodos para generar variables aleatorias pertenecientes a las siguientes distribuciones discretas: a) distribución de Bernoulli (Naylor et al. 1975), b) distribución binomial (Naylor et al. 1975; Bratley et al. 1987), c) distribución de Poisson (Bratley et al. 1987), d) distribución geométrica (Bratley et al. 1987), y la e) distribución hipergeométrica (Naylor et al. 1975; Bratley et al. 1987).

Se incluyen los siguientes métodos para generar muestras de distribuciones continuas como: a) variable aleatoria uniforme en el intervalo  $[a,b]$ , aplicando el método de la transformada inversa (Pace 1995); b) variables aleatorias con distribución normal no estandarizada (Pace 1995; Bratley et al. 1987), mediante la aplicación: del teorema central del límite, el método de Hasting usando aproximaciones funcionales de la transformada inversa (Bratley et al. 1987) y el método de Box-Muller (Bratley et al. 1987); c) variables aleatorias de Erlang generadas por el método de la transformada inversa (Bratley et al. 1987); d) variables aleatorias exponenciales basadas en el método de la transformada inversa (Bratley et al. 1987) y en el procedimiento de Von Neumann (Bratley et al. 1987).

Cada método especial, fue codificado en un módulo independiente, asegurando el mantenimiento, la reusabilidad y la escalabilidad del paquete educativo.

#### 4. Un paquete para el aprendizaje de variables aleatorias no-uniformes

La construcción de software educativo ofrece nuevas posibilidades de intercambio de información entre el alumno y la computadora, debido a que los datos, la información y las diversas representaciones del conocimiento se administran con una estructura dinámica y relacional que estimula la indagación, la integración y la transferencia del saber.

El desarrollo de entornos virtuales de aprendizaje (Duart & Sangrá 2000) empleando herramientas de programación para la web facilitan la disseminación de información. Internet proporciona diversos servicios aplicables a portales educativos. Mediante el desarrollo de portales verticales los estudiantes acceden al conocimiento y el docente crea enlaces a otros sitios relevantes e interesantes. Asimismo, en todo proceso de enseñanza-aprendizaje, brindan numerosas ventajas el correo electrónico, las listas de distribución, las charlas en la web, los foros entre otros servicios.

Se presenta un sistema de software educativo destinado a la enseñanza de procedimientos generadores de muestras artificiales de variables aleatorias no uniformes mediante la aplicación de métodos especiales. El entorno virtual de aprendizaje se compone de un sitio web, desde el cual se accede a contenidos teóricos que fundamentan las distribuciones teóricas estudiadas, y un paquete desarrollado en Mathematica.

En el paquete se incluyen los métodos para generar variables aleatorias de las distribuciones discretas: distribución de Bernoulli, distribución Binomial, distribución Hipergeométrica, la distribución de Poisson; y de distribuciones continuas como: distribución Uniforme, distribución Normal, distribución de Erlang, distribución Gamma; y la distribución Ji-Cuadrada. El paquete incluye un sistema de ayuda en línea, tendiente a orientar en el adecuado uso de cada uno de los métodos especiales descriptos.

El alumno podrá descargar el paquete desde un sitio web o entorno virtual de aprendizaje. Al acceder al paquete podrá seleccionar cualquiera de los módulos generadores disponibles.

En la ejecución de cada método, es posible seleccionar el generador de números pseudoaleatorios a emplear para la construcción de la muestra artificial. Los alumnos podrían evaluar mediante pruebas de hipótesis cual generador es estadísticamente más adecuado para la construcción de las distintas muestras artificiales.

El paquete de Mathematica, orientado a la enseñanza y el aprendizaje de muestras artificiales de variables aleatorias no uniformes, otorgará a los estudiantes un medio adicional para la comprensión del desarrollo de los procedimientos matemáticos generadores de las mismas, propiciando un entorno para la realización de las ejercitaciones interactivas.

#### 5. Conclusiones

Los sistemas educativos interactivos modifican la definición de las clases, ofreciendo a los estudiantes nuevas formas de aprender, propiciando el estudio independiente y la profundización de los contenidos sin restricciones de tiempo ni espacio. Pueden ser concebidos como sistemas dinámicos de aprendizaje, donde los lectores-alumnos adquieren paulatinamente los conocimientos. Mathematica como lenguaje de desarrollo de código abierto, actúa como una herramienta de asistencia a la comprensión de cada uno de los métodos especiales para la construcción de las muestras artificiales de las distribuciones más conocidas. Asimismo, permite crear software legible, fácil de modificar y depurar. El desarrollo de los procedimientos independientes facilita la actualización de esta herramienta educativa.

El uso de este paquete de Mathematica se orienta a reforzar y extender los conceptos específicos, identificando y corrigiendo los conceptos erróneos y proporcionando casos de estudio o ejemplos con el objeto de lograr aprendizajes significativos referentes a los métodos especiales para construir muestras aleatorias no uniformes.

## Referencias

- Bratley, P. Fox, B. and Schrage, L. 1987. A Guide to Simulation. Springer-Verlag. New York.
- Castillo E., Iglesias, A., Gutierrez, J. M., Alvarez, E. y Cobo, A. 1996. Mathematica. Ed. Paraninfo. Madrid, España. 3ra. Ed.
- Coos, R. 1992. Simulación un enfoque práctico. Limusa Grupo Noriega Editores. México.
- Duart, J. M y Sangrá, A. 2000. Aprender en la virtualidad. Ed. Gedisa.
- Lavenberg, S. S. 1983. Computer Performance Modeling Handbook. Academic Press. New York.
- López, María V., Mariño, Sonia I. y Petris, Raquel H. 1999. Un análisis comparativo de generadores de números pseudo-aleatorios en Mathematica 3.0. FACENA. Revista de la Facultad de Cs. Exactas y Nat. y Agrimensura. Univ. Nac. Del Nordeste. Vol. 15. 119-136.
- Miner, R. and Topping, P. 2001. Math on the Web: A Status Report
- Naylor, T. H., Balintfy, J. L., Burdick, D. S. & Chu, K., 1975. Técnicas de Simulación en computadoras. Ed. Limusa. Mexico.
- Pace, G. 1995. "Modelos y Simulación". Inédito. Corrientes. Argentina.
- Pardo, L. y Valdes, T. 1987. Simulación. Aplicación práctica en la empresa. Ed. Diaz de Santos S. A. Madrid. España.
- Roberts, N, Andersen, D., Deal, R., Garet, M. and Shaffer, W. 1983. Introduction to computer simulation: The system dynamics approach. Addison-Wesley Publishing Company.
- Wolfram, S. 1996. The Mathematica Book, 3rd ed. Wolfram Media/Cambridge University Press.  
En: <http://www.wolfram.com>

**Desarrollo de programas educativos para el modelado y la simulación de sistemas.  
Algunos estudios de casos.**

*Sonia Itatí Mariño y María Victoria López*

*Departamento de Informática.  
Facultad de Ciencias Exactas y Naturales y Agrimensura.  
Universidad Nacional del Nordeste. 9 de Julio 1449  
9 de Julio n° 1449. CP: 3400. Corrientes. Argentina  
TE (03783) 423126/423968/424606 - Fax (03783) 423968  
msonia@exa.unne.edu.ar, mvlopez@exa.unne.edu.ar*

**Palabras Claves:** Aprendizaje asistido por computadora. Generación de variables aleatorias. Método de Simulación. Sistema educativo interactivo.

### **Introducción**

Las nuevas tecnologías de la información y la comunicación (TIC) son ampliamente aplicadas en vastos dominios del conocimiento.

Uno de ellos se refiere a la educación en los distintos niveles. Algunos autores la denominan enseñanza asistida por computador (EAC), aprendizaje asistido por computadora (ASC) y otros los referencian como sistemas basados en las Tecnologías de la Educación (Learning Technology Systems o LTS) [Rod]. Para Rodríguez Artacho [Rod], estos sistemas educativos representan un enfoque más amplio que el que proporciona el desarrollo de sistemas tradicionales. Se trata de integrar unos sistemas educativos en un marco abierto donde se facilite la reusabilidad y la interoperatividad de los componentes de los mismos.

Desde la perspectiva docente, los trabajos se relacionan principalmente con el diseño y la elaboración de material educativo interactivo.

La simulación de un sistema es la operación de un modelo, el cual es una representación del sistema. Este modelo puede sujetarse a manipulaciones que en la realidad serían imposibles de realizar, demasiado costosas o imprácticas [Freund]. Una de las principales ventajas del **método de simulación o de Montecarlo** se refiere a que los modelos permiten inferir y estudiar las propiedades concernientes al comportamiento del sistema o subsistema real.

Desde la perspectiva de los alumnos, es aconsejable que los contenidos teóricos relacionados con la modelización de sistemas que son adquiridos en el aula, se afiancen mediante la implementación de modelos computacionales. La problemática del aprendizaje universitario, los procesos de apropiación de los conocimientos por parte de los alumnos constituyen una dimensión a considerar a la hora de diseñar una propuesta de enseñanza, dado que aunque no se puede establecer una relación causal entre la enseñanza y el aprendizaje, este último constituye la razón de ser de aquella [Mena].

En este trabajo, se describe un **software para la simulación de modelos**, empleando muestras artificiales pertenecientes a distintas distribuciones de probabilidad obtenidas mediante la generación de números pseudoaleatorios. El software o herramienta educativa fue desarrollada en el lenguaje orientado a eventos Visual Basic.

La aplicación desarrollada constituye un software educativo de tipo “simulador”. Los simuladores presentan un entorno dinámico que facilita su exploración y modificación por los alumnos, quienes pueden realizar aprendizajes inductivos y deductivos mediante la observación y la manipulación de la estructura subyacente; de esta manera pueden descubrir los elementos del modelo, sus interrelaciones, y pueden tomar decisiones y adquirir experiencia directa delante de unas situaciones

que frecuentemente resultarían difícilmente accesibles a la realidad. En cualquier caso, posibilitan un aprendizaje significativo por descubrimiento y la investigación de los estudiantes-experimentadores puede realizarse mediante preguntas del tipo: ¿Qué pasa con el modelo si modifico el valor de la variable X? ¿Y si modifico el parámetro Y? [Mar]

Lo importante de una simulación es el tratar de representar un fenómeno real y dinámico (o cambiante en el tiempo). La simulación permite adquirir habilidad o aprender las reglas para manipular un fenómeno, mecanismo o dispositivo dinámico y complejo. También permite el entender la dinámica compleja de una situación y ser entrenado a este tipo de medio ambiente (por ejemplo los simuladores de líneas de producción, de bolsa de valores o de negocios). La dinámica puede presentarse en gráficas comparativas o ser presentada como dibujos animados [Rivera].

El software no solo incluye prácticas interactivas de los principales estudios de casos presentados en el dictado de las clases teórico-prácticas, sino que ofrece los fundamentos teóricos en un formato hipermedial implementado mediante el sistema de ayuda adoptado por el entorno Windows.

El programa instalable del software educativo que se describe, podrá ser descargado del sitio web de la cátedra “Modelos y Simulación” de la FACENA, que incluirá los contenidos didácticos, el diagrama de flujo y el pseudocódigo correspondiente. Debido a que este software educativo podrá ser obtenido vía Internet, permitirá constituir el punto inicial para la elaboración de nuevos casos de estudios o su aplicación en la resolución de nuevos problemas.

### **Metodología para el desarrollo del sistema hipermedial de enseñanza – aprendizaje.**

La metodología adoptada para en el desarrollo de este software se basó en la técnica de los prototipos [Kendall], la metodología descrita para el diseño de aplicaciones multimediales [CCPM] y el diseño de interfaces de usuarios en aplicaciones educativas [Nog]. Las principales etapas consistieron en:

- Selección y transformación de los contenidos de los teóricos en formato digital.
- Elaboración de modelos de simulación representativos de estudios de casos determinísticos y estocásticos.
- Diseño del prototipo. Definición de las interfaces y el estilo de interacción representado mediante menús y botones.
- Desarrollo del prototipo informático para la enseñanza de modelos de simulación. Se diseñaron las interfaces, se codificaron las rutinas para modelar las situaciones planteadas en los distintos modelos especificando las llamadas correspondientes a los procedimientos generadores de muestras artificiales y se incluyeron los contenidos en formato hipertexto a las interfaces.
- Verificación y puesta a punto del correcto funcionamiento del sistema de enseñanza.

### **Métodos numéricos empleados**

Los métodos de simulación representan tanto fenómenos discretos como continuos, según el modelo que se intenta describir. La construcción de modelos de simulación requieren como entrada valores de variables que responden a distintas distribuciones teóricas. Si se desconoce la distribución que las origina, estas variables que componen las muestras artificiales pueden ser generadas aplicando el método de los números índice [Pace] [Mariño]. La construcción de muestras artificiales requiere de números pseudoaleatorios obtenidos mediante algún procedimiento. Los generadores de números pseudoaleatorios constituyen elementos imprescindibles para la implementación de los métodos de simulación. Se incorporaron los códigos desarrollados en un trabajo previo [Mariño] que incluye el Método de los Cuadrados Centrales de Vonn Neumann [Pardo], Método de Fibonacci [Pace], Método Aditivo de Congruencias [Pardo], Método

Multiplicativo de Congruencias [Coss], Método Mixto de Congruencias [Coos] y el método de los números índice [Pace].

### **Herramientas utilizadas**

El software educativo, integrador de los contenidos teóricos y las prácticas, se desarrolló usando el modelo de análisis y diseño orientado a eventos, programado con Visual Basic ([Balena], [Martins] y [Micro]) ejecutable en entornos Windows.

Se optó por este lenguaje de programación, con el objeto de incorporar módulos previamente desarrollados [Mariño] y relacionados a esta temática, orientando el desarrollo a una herramienta educativa escalable.

### **Resultados**

Marqués [Mar] menciona que las investigaciones en informática educativa se orientan a la elaboración de entornos de comunicación cada vez más intuitivos y capaces de proporcionar un diálogo abierto y próximo al lenguaje natural. Rodríguez Artacho [Rod] alude que fundamentalmente se abarca el diseño de los sistemas desde el punto de vista del intercambio de contenidos y en la integración con otros componentes.

Pautasso y colaboradores [Pau] mencionan que en la educación a distancia, la enseñanza no es guiada o controlada directamente por el profesor en el aula y la presencialidad es reemplazada por materiales autoinstruccionales. Aún cuando esta herramienta no se desarrolla con la finalidad de implementar un sistema educativo a distancia o virtual, se incorporan aquellos elementos facilitadores del autoaprendizaje de los alumnos o las prácticas aún cuando no exista un profesor en el aula.

El objetivo primordial de este desarrollo es que el estudiante comprenda y afiance las aplicaciones del método de Simulación o de MonteCarlo empleando una herramienta informática. El software educativo “Modelos de simulación”, se orienta a un modelo centrado en el estudiante [Duart] basándose en el autoaprendizaje y la autoformación. La herramienta se integra mediante interfaces amigables, consta principalmente de los siguientes módulos: i) Presentación de contenidos teóricos. Los fundamentos de la construcción de modelos de simulación se acceden mediante un sistema de hipertexto visualizados mediante el sistema de ayuda de Windows. ii) Acceso a una variada gama de prácticas parametrizables por el alumno. El estudiante tiene la posibilidad de ingresar en pantalla los valores de los parámetros requeridos por los modelos y por los métodos de generación de números aleatorios y muestras artificiales, pudiendo experimentar la ejecución de la simulación con diferentes valores iniciales, evaluando luego los resultados obtenidos en cada caso. También es posible guardar en un archivo o imprimir los resultados. iii) Guías de estudio orientativas, diseñadas para encaminar al estudiante en el acceso a los conocimientos temáticos. iv) Un completo sistema de ayuda en línea, que brinda información constante de las actividades propuestas y el uso del sistema.

La web es un medio o un sistema diferente a través del cual se relacionan los sujetos y los objetos que intervienen en el proceso educativo de aprendizaje, condiciona los recursos pedagógicos que se utilizan. Pero no se debe supeditar la tecnología a la educación, sino que la primera debe estar al servicio de la segunda [Duart]. Una vez implementado el espacio virtual de aprendizaje de la asignatura “Modelos y Simulación” que se dicta en nuestra facultad, esta herramienta educativa podrá ser obtenido mediante el servicio de transferencia electrónica de archivos, de modo que las prácticas puedan realizarse sin restricciones espacio-temporales.

## Conclusiones

Se exponen los fundamentos de un software educativo que implementa modelos matemáticos mediante la aplicación del Método de simulación o de Montecarlo, orientado a presentar casos de estudios de algunos modelos tomados como paradigmáticos de la simulación estocástica y la determinística.

El enfoque adoptado en la construcción de esta herramienta favorecerá, incentivando a los alumnos, el aprendizaje y la profundización de las unidades temáticas adquiridas en el aula.

Los módulos que componen el software se integran, con el propósito de apoyar al estudiante en el proceso de aprendizaje. Asimismo permite a los alumnos, que por diversos motivos no pueden asistir a las clases presenciales previstas, accedan a conocimientos específicos y estudien mediante herramientas propiciadoras de entornos virtuales.

## Referencias

- [Balena] Balena, F. 2000. "Programación Avanzada con Microsoft Visual Basic 6.0". Editorial McGraw-Hill.
- [CCPM] Centro de Computación Profesional de México. 2001. Multimedia Aplicada. Ed. Mc Graw Hill. 2da. Ed.
- [Coos] Coos, R. 1992. Simulación un enfoque práctico. Limusa Grupo Noriega Editores. México.
- [Duart] Duart, J. M y Sangrá, A. 2000. Aprender en la virtualidad. Ed. Gedisa.
- [Freund] Freund, J. E. 1962. "Mathematical Statistics". Englewood Cliffs. Prentice-Hall.
- [Kendall] Kendall, K. E. & Kendall, J. E. 1991. "Análisis y diseño de sistemas". Editorial Prentice - Hall Hispanoamericana, S.A.
- [Mar] Marquès, P. "El software educativo". [http://www.doe.d5.ub.es/te/any96/marques\\_software](http://www.doe.d5.ub.es/te/any96/marques_software)
- [Mariño] Mariño, S. I., y López, M. V. "Aprendizaje de muestras artificiales de variables aleatorias discretas asistido por computadora". CACIC 2001. El Calafate (Argentina).
- [Martins] Martins Carrizo, M.B. 2000. "Visual Basic 6.0 Edición Profesional en un solo libro". Editorial GYR.
- [Mena] Mena, M., Muriello, A., Rodriguez, L. Orecchia, V., Pascual, L., Ferrante, A., Langer, M. Misrahi, C. Suárez, M. A., Ursi, L. Enseñando en Ciencias Económicas Programa de Formación Docente Continua. Facultad de Ciencias Económicas. Universidad de Buenos Aires.
- [Micro] Sitio de Microsoft. Microsoft. <http://www.microsoft.com>
- [Nog] Noguera Oliver, M. López-Polín Hernanz, C. y Salinas Ibáñez, J. El interfaz de usuario. En: [http://www.filos.unam.mx/POSGRADO/seminarios/pag\\_robertp/paginas/interfaz.htm](http://www.filos.unam.mx/POSGRADO/seminarios/pag_robertp/paginas/interfaz.htm)
- [Pace] Pace, G. 1996. Material didáctico de la asignatura "Modelos y Simulación". FACENA. UNNE. Inédito.
- [Pardo] Pardo, L. y Valdes, T. 1987. Simulación. Aplicación práctica en la empresa. Ed. Diaz de Santos S. A. Madrid. España..
- [Pau] Pautasso, M. Gatica, M. J. y Diez, M. L. 1996. La educación a distancia en el sector público. INAP. Bs. As.
- [Rivera] Rivera Porto, E. "Aprendizaje asistido por computadora. Diseño y realización". <http://msip.lce.org/erporto/libros/edu2>.
- [Rod] Rodriguez Artacho, M. 2000. Las Tecnologías Educativas. En <http://sensei.ieec.uned.es/~miguel/tesis/node15.html> .

## Herramientas y estrategias visuales para sistemas de aprendizaje

Sergio Martig – Perla Señas

Laboratorio De Investigación y Desarrollo en Informática y Educación (LIDInE)  
Instituto de Investigación en Ciencias y Tecnología Informática (IICTI)  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur - Bahía Blanca  
[smartig/psenas]@cs.uns.edu.ar

### Introducción

En el marco educativo, cuando el desafío es el logro de aprendizajes significativos, cuando se trata de hacer asumir la propia elaboración de significados, en definitiva cuando se busca que “aprendan a aprender”, es importante contar con herramientas visuales capaces de representar el conocimiento.

En tal sentido se ha trabajado sobre los Mapas Conceptuales Hipermediales (MCH), enfatizando el diseño de herramientas que favorezcan la construcción y lectura de los mismos.

Un aspecto importante es potenciar la información representada en los mismos, ofreciendo acceso a la bibliografía, permitiendo acceder desde el esquema visual a documentos en los que se desarrollan y amplían los conceptos representados en las vistas de los MCH.

Desde lo cognitivo es proveer un mecanismo de adquisición de información dentro de un marco contextual, el brindado por el MCH.

### Ambientes MCH

Recordemos que los MCH están compuestos por dos planos de información:

#### ▪ Plano MCH propiamente dicho:

Esta capa de los MCH es la que corresponde al documento hipermedial; donde a cada nodo de la hipermedia se lo denomina vista del MCH. Cada vista puede ser visualizada en una ventana y es caracterizada por un color y por un nombre, por defecto el del concepto propio más abarcativo en dicha vista. El saldo positivo que se logra con la partición del mapa en vistas (desde lo pedagógico y desde lo operacional), puede tornarse negativo si el lector del mapa pierde las referencias conceptuales con respecto al todo mientras centra su atención en una vista particular. Con este objetivo se amplió ambiente MCH existente con un Grafo de Vistas ( $GV_{MCH}$ ) que brinde al usuario información contextual mientras está accediendo a una vista particular. [Mar01]

Por otro lado se consideró que era de suma importancia disponer de una herramienta que permitiera obtener una composición de todas las vistas que conforman un MCH, para lo cual se definió el Grafo Integrador de un MCH ( $GI_{MCH}$ ). [Mar00]

La idea del Grafo Integrador Anidado de un MCH surgió luego como síntesis de los trabajos anteriores, siendo una herramienta interactiva que permite obtener información contextual y detalle a demanda del usuario sobre un MCH dado.

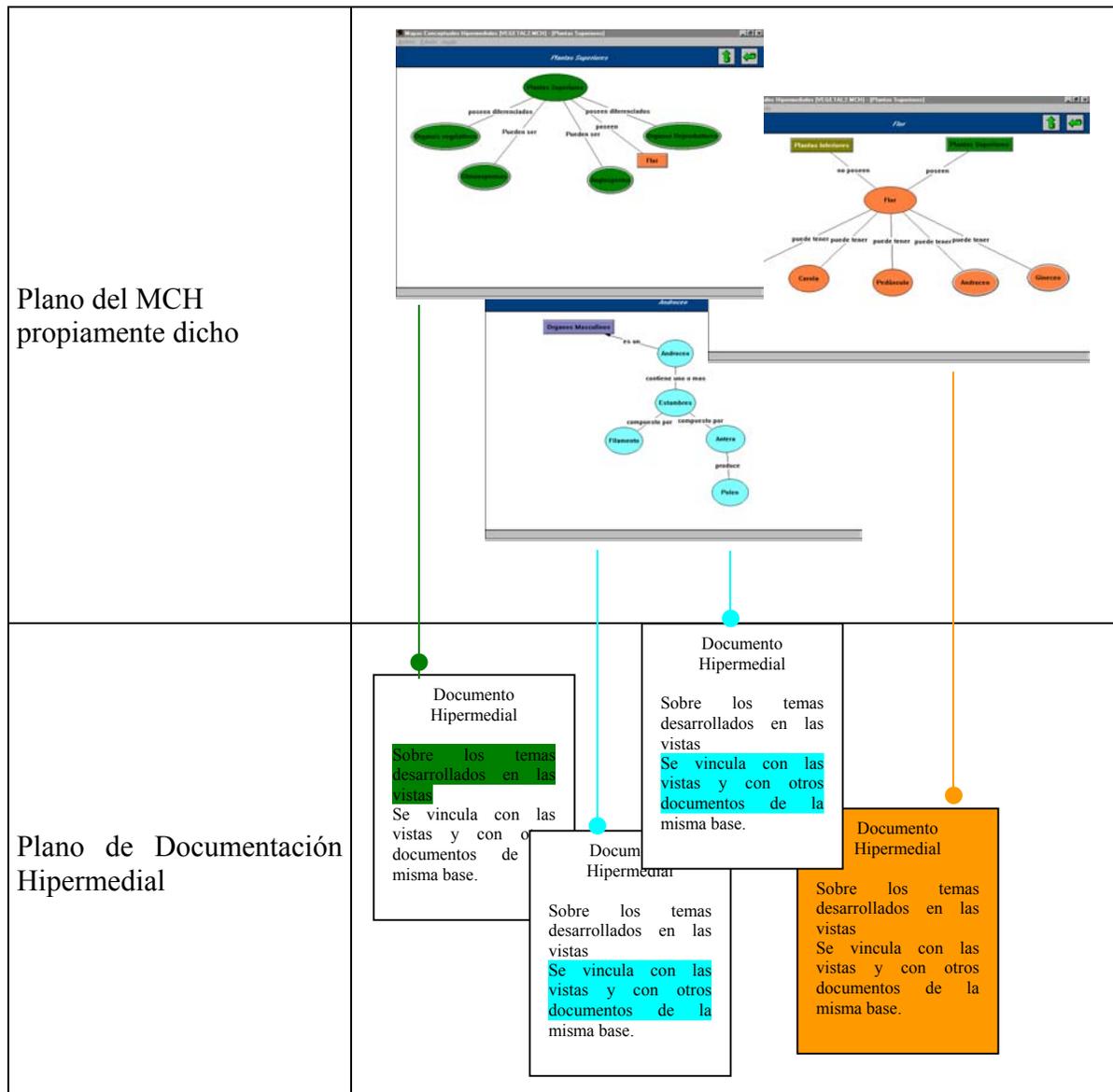
Este aspecto de los MCH es sobre el que más hemos trabajado, diseñando distintas herramientas que enriquecieron el ambiente original.

#### ▪ Plano de Documentación Hipermedial:

En el diseño de los MCH hay un segundo nivel de información compuesto por un conjunto de documentos hipermediales. En este nivel se almacena información relacionada con las distintas vistas.

En realidad la idea es que el lector del MCH pueda acceder a documentos vinculados con los conceptos contenidos en cada vista.

Estamos trabajando en la manera de establecer los vínculos entre las distintas vistas de un MCH y los documentos relacionados, puntualmente la que se desea es establecer los vínculos no sólo a los documentos, sino a secciones específicas de los mismos.



Establecer los vínculos entre las vistas que componen un MCH y la base hipermedial subyacente ofrece distintos frentes:

- El mantenimiento de los vínculos entre ambos planos. Su generación en la etapa de autoría.
- El acceso a los documentos desde el MCH en el modo lector.

Además lo que deseamos es reflejar qué parte del documento es de interés para una vista particular, esto determina un problema adicional que es el marcado dentro del documento, lo cual no es una tarea trivial, principalmente por la naturaleza hipermedial de los documentos.

## Aplicaciones

Dentro de un ambiente de aprendizaje los MCH pueden ser en sí mismos una herramienta válida para la caracterización de manera concisa y efectiva de documentos. Para lo cual se tiene planeado trabajar en la organización de repositorios de documentos. Disponer de los MCH asociados a los documentos y a través de ellos acceder al potencial de la plataforma constituye un poderoso recurso. Las facilidades de interconexión de los mapas brindan la posibilidad de tener una visión integrada de los contenidos de un conjunto de documentos.

## Bibliografía

- [Ber83] Bertin, J. "Semiology of Graphics: Diagrams, Networks, Maps". University of Wisconsin Press, Madison, Wisconsin, 1983.
- [Dix98] Dix, A., Finlay, J., Abowd, J., Beale, Russell, "Human-Computer Interaction", Prentice Hall Europe, 1998.
- [Fol92] Foley, J., Van Dam, A., "Fundamentals of Interactive Computers Graphics", Addison-Wesley, Reading, Massachusetts, segunda edición, 1992.
- [Gri95] Grinstein G - Levkowitz H. "Perceptual Issues in Visualization", Springer-Verlag, 1995.
- [Laj93] Lajoie, S. "Computer Environments as Cognitive Tools for Enhancing Learning". 1993. McGill University.
- [Leh93] Lehrer, R. "Authors of knowledge: Patterns of Hypermedia Design". 1993. University of Wisconsin-Madison.
- [Mar00] Martig, S. y Señas, P. "Herramientas para la construcción de conocimiento en ambientes de aprendizaje abiertos: Construcción y Visualización del Grafo Integrador de un MCH". VI CACIC. Argentina. 2000.
- [Mar01] Martig, S. y Señas, P. "GIA MCH". VII CACIC. Argentina. 2001.
- [Mar01] Martig, S. y Señas, P. "GV MCH". VII CACIC. Argentina. 2001.
- [Mor96] Moroni, N. - Vitturini, M. - Zanconi, M. - Señas, P. "Una plataforma para el desarrollo de mapas conceptuales hipermediales". Taller de Software Educativo - IV Jornadas Chilenas de Computación. Valdivia. 1996.
- [Mut97] Mutzel, P. "An Alternative Method to Crossing Minimization on Hierarchical Graphs".
- [Nov85] Novak, J. "Metalearning and metaknowledge strategies to help students learn how to learn. Cognitive Structure and Conceptual Change". New York. Academic Press. 1985.
- [Nor93] Norman, D. "Things that Make Us Smart". MA: Addison-Wesley. 1993.
- [Rao93] Rao, R., Card, S., "The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information", Proceedings CHI'94, 318-322.
- [Señ96] Señas, P., Moroni, N., Vitturini, M. y Zanconi, M.: "Hypermedial Conceptual Mapping: A Development Methodology". 13th International Conference on Technology and Education. University of Texas at Arlington, Department of Computer Science and Engineering. New Orleans 1996.
- [Señ98] Señas, P., Moroni. "Herramientas no convencionales para el aprendizaje de la programación". IV CACIC. Argentina. 1998.
- [Tuf90] Tufte, E.R., "Envisioning Information", Cheshire, CT Graphics Press, 1990.
- [Zan98] Zanconi, M., Moroni, N., Vitturini, M., Malet, A., Borel, C. y Señas, P. Tecnología computacional y meta-aprendizajes. RIBIE-98. 1998.

## La Visualización de Algoritmos como Recurso para la Enseñanza de la Programación

Norma Moroni - Perla Señas  
Grupo LIDInE  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur.  
Av. Alem 1253 - 8000 - Bahía Blanca - ARGENTINA  
moroni@criba.edu.ar - psenas@csuns.edu.ar

### Introducción.

En la búsqueda de nuevas estrategias metodológicas para la enseñanza de la programación, es interesante el paralelo que se puede realizar con los estudios sobre el aprendizaje de la lecto-escritura del lenguaje natural. Es muy común que al enseñar programación se centre la atención en la creación de algoritmos, descuidando el trabajo de lectura y comprensión de algoritmos existentes. Trabajar de esa manera puede pensarse en cierta forma equivalente a tratar de enseñarle a un alumno a escribir dejando de lado los contenidos de lectura mecánica y sobre todo de lectura comprensiva [27] [28].

En relación con la enseñanza de la programación se destaca la importancia de las actividades comprendidas en:

- construcción de algoritmos
- lectura comprensiva de algoritmos

Existe consenso en considerar un enfoque que parta de la resolución de problemas, para el primer ítem, y que se apoye en técnicas de visualización de algoritmos y programas para el segundo caso. Se aborda en este trabajo el estudio realizado sobre visualización de software, especialmente pensado para la lectura comprensiva de algoritmos y programas.

La visualización de software es el uso de gráfica de computadoras y animación interactiva para ilustrar y presentar programas, procesos y algoritmos. Se basa en el uso de diseño gráfico, de animación, de sonido, de video y tiene como característica sobresaliente la interacción entre el usuario y la computadora.

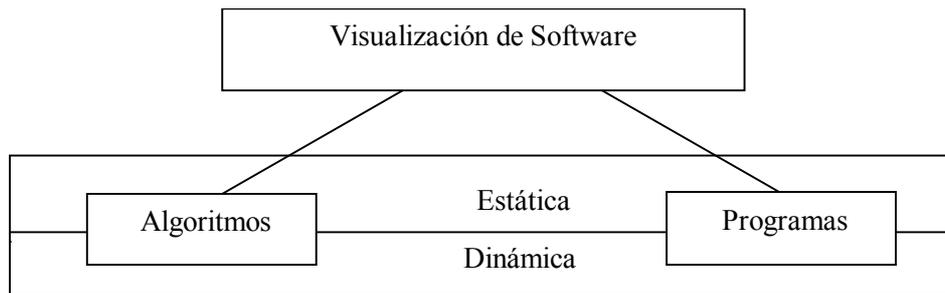
La visualización de la conducta dinámica de los programas y de sus modelos abstractos, los algoritmos, y consecuentemente, el beneficio psicopedagógico de su aplicación en la enseñanza, constituyen una gran área de investigación que aún no ha sido completamente explorada.

La motivación especial que condujo a realizar este trabajo es la necesidad de mejorar las herramientas existentes para facilitar la comprensión de los aspectos de la conducta dinámica de los algoritmos en las distintas fases del ciclo del software, en el diseño, en el desarrollo, en la búsqueda de errores, en la depuración y en el mantenimiento.

Generalmente, para lograr una comprensión acabada de un programa se realiza, por un lado, el estudio del código fuente, lo que es incómodo y de difícil aplicación y, por otro lado, la construcción de casos de prueba para explicar la conducta de un programa, lo que es una tarea penosa y especulativa. Estas dificultades motivan el desarrollo de programas especiales que son usados para ayudar a explicar la conducta de otros programas [1].

### Visualización de software

La Visualización de Software comprende la visualización interactiva de dos áreas fundamentales: la de algoritmos y la de programas, que en ambos casos, se pueden considerar transversalmente como presentaciones estáticas o dinámicas.



La visualización de algoritmos consiste en la exposición de abstracciones de alto nivel que describen semánticamente el software. La representación estática de los mismos está basada en la esquematización de la estructura de su especificación. La representación dinámica, que habitualmente se llama animación de algoritmos, muestra su comportamiento en tiempo de corrida. La visualización estática de programa realiza un análisis del texto del mismo y provee la información que es válida para todas las ejecuciones independientemente de su entrada [15] [11]. Una representación dinámica del programa provee información acerca de su ejecución sobre un conjunto de datos de entrada [15]. La información de la ejecución se puede ofrecer en forma instantánea, o en un tiempo posterior (análisis post mortem). El análisis post mortem puede realizar computaciones grandes para condensar la información de la ejecución y presentarla en una forma útil. Los métodos no son excluyentes.

Las herramientas en tiempo de corrida pueden ser pasivas o interactivas. En un sistema pasivo, la herramienta presenta información al usuario, pero el usuario tiene poco control sobre la actividad del mismo; en un sistema interactivo, el usuario puede tener control externo sobre la información que se está exhibiendo, puede estar disponible para modificar la computación que está siendo observada o bien el dato que está siendo procesado [11].

Los sistemas para comprender programas son usados en una variedad de aplicaciones. El debugger es uno de los programas más comunes que ayudan a la comprensión del software. Otras de las aplicaciones de sistemas de comprensión de programas es la encargada de poner a punto la efectividad o performance. Una tercera aplicación de la comprensión es la instrucción y orientación del software, que resulta de interés para aprender importantes algoritmos, estructuras de datos o técnicas de programación. Con las herramientas de comprensión de programas las personas afectadas pueden beneficiarse por la información provista por ellas, antes de consultar el código fuente.

### **Antecedentes de la visualización de software.**

La preocupación frente a la dificultad en la comprensión del software ha sido constante. Entre las distintas mejoras se pueden mencionar las orientadas hacia la estructura lógica y expresiva del estilo de los programas, tal como el diseño top down y el refinamiento paso a paso [14], la programación estructurada [15], la modularidad [16], y las herramientas de software [17]; en la claridad y potencia expresiva de los lenguajes de programación [18], y en el desarrollo de aproximaciones orientadas a objeto para diseño y desarrollo [19].

El embellecimiento de la escritura, el uso de espaciado, indentación, y disposición para que el código fuente fuera más fácil de leer son algunas de las primeras ideas de visualización que surgieron en un principio. Los Prettyprinters son programas que sistemáticamente indentan el código fuente de un programa de acuerdo a su estructura sintáctica. SEE Program Visualizer [3] es un sistema que automáticamente establece un programa en C de acuerdo a una guía de estilo basada sobre principios de diseño. El sistema WEB mejora la escritura de programas combinando el texto del programa fuente con la documentación en una sola publicación usando un lenguaje sofisticado [2].

A medida que se progresa en la capacidad de representación gráfica de las computadoras, se desarrolla software que permite visualizar el comportamiento de los programas. El primer trabajo

que ilustra la ejecución de un programa es el que presenta Baecker a mediados de los 70 [3]. El sistema Balsa es uno de los primeros que permite la creación de las visualizaciones de algoritmos genéricos en Pascal [4]. El sistema agrega al código primitivas de animación las cuales activan las funciones de representación gráfica que ofrece el sistema. A finales de los años 80 y en la década de los 90 surgen la gran mayoría de los sistemas que contemplan ahora la visualización de programas concurrentes o paralelos como los sistemas Voyeur [5] o Belvedere [6].

Como sucesores del sistema Balsa, surgen los sistemas Tango [7], Zeus [8] y [9]. Estos dos últimos permiten la visualización de ejecuciones de programas concurrentes. El sistema Pavane [10] permite la visualización de programas declarativos.

### **Visualización de algoritmos.**

La Visualización de Algoritmos presenta un aspecto distintivo de los otros tipos de visualizaciones ya que no existen entidades tangibles a las cuales asirse para diseñar la visualización. Se debe decidir lo que se quiere visualizar y cómo se va a obtener la información necesaria para hacerlo [23].

En las visualizaciones de algoritmos, en las que se utilizan grandes niveles de abstracción, puede resultar difícil especificar la apariencia visual de los objetos lógicos presentes. En este caso el sistema de visualización debe ofrecer un conjunto de transformaciones y entidades gráficas para realizar la especificación.

La visualización del algoritmo en cualquiera de sus dos formas, dinámica o estática, puede ser muy difícil de relacionar con las construcciones del programa que la codifican y que constituyen las especificaciones de su comportamiento. Es decir, se presenta allí una brecha entre la representación del algoritmo y la especificación del programa subyacente [4].

### **Contribución pedagógica de la Visualización de Algoritmos**

Se debe disponer de sistemas amigables con el usuario (facilidad de interacción hombre-máquina, en cuyo diseño se debe poner especial interés en la percepción humana). Pero, además, la experiencia de los usuarios con el lenguaje de programación y su nivel de familiaridad con la propia visualización de software resultante, influyen sobre la calidad del sistema de visualización.

La Visualización de Algoritmos presenta un medio audio-visual interactivo multimedial cuyas bondades en el campo de la psicopedagogía han sido ampliamente comprobadas. La Visualización de Algoritmos es una disciplina joven y, por ello, tanto la experiencia en sí como la evaluación de los resultados deben ajustarse a ella. Las animaciones de algoritmos presentan importantes beneficios educativos [24] [25] [26]:

- Logran un incremento de la motivación.
- Facilitan el desarrollo de destrezas.
- Asisten en el desarrollo de habilidades analíticas.
- Ofrecen un buen soporte al docente.
- Permiten la exploración, de las peculiaridades de un algoritmo, jugando interactivamente.

Distintos centros educativos emplean estos sistemas como apoyo al aprendizaje [13] [24].

### **Sistema de Visualización de Algoritmos propuesto**

Se propone un Sistema de Visualización Interactiva de Algoritmos orientado a una programación estructurada y modular.

El sistema debe proveer multiventanas en las que se debe exhibir la estructura estática del algoritmo, la estructura estática de los datos, el menú de selección de la representación de datos y aquellas ventanas que permiten el ingreso de los valores de los datos y de su representación con los cuales se realizará la corrida del programa.

Durante la ejecución del programa, el sistema debe permitir la visualización del dinamismo de la activación de los distintos módulos en ejecución, la historia del paso por los distintos módulos y fundamentalmente, el comportamiento del algoritmo a través de la animación de los datos. Ventanas adicionales, deben estar activas para permitir la interacción del usuario con la visualización de manera que pueda interrumpirla, fijar la imagen, modificar su velocidad de presentación, modificar los valores de entrada, modificar la representación de los datos y sus colores, activar la audición ya sea de errores o de pasos claves. La audición ofrece algunas ventajas respecto de la observación permanente de las ventanas activas. Escuchar un sonido determinado ayuda al usuario a detectar una acción cuya ejecución se está llevando a cabo en ese momento, sin necesidad de observar la pantalla.

## Conclusión

La animación de algoritmos y la visualización de programas ayudan a los estudiantes a comprender los conceptos de software y también a los docentes en su tarea de enseñar dichos conceptos.

Independientemente de la calidad de un sistema de visualización se deben tener en cuenta algunos factores importantes que influyen sobre la efectividad del mismo, como son la experiencia de los usuarios con el lenguaje de programación y su nivel de familiaridad con la propia visualización de software resultante.

Actualmente se están estableciendo los lineamientos básicos de un sistema de visualización de algoritmos basados en un ambiente estructurado y modular, que constituya un entorno de aprendizaje efectivo para ayudar a los estudiantes en la comprensión y desarrollo de algoritmos y programas.

## Bibliografía

- [1] Jeffery, Clinton L. *Program Monitoring and Visualization*. Springer-Verlag. 1999.
- [2] Knuth, D. *Computer-drawn Flowcharts*. Communication of the ACM. 1963.
- [3] Baecker, R. *Two systems Which Produce Animated Representations of the Execution of Computer Programs*. 1975. ACM SIGCSE Bulletin.
- [4] Brown y Sedgewick. *A system for Algorithm Animation*. ACM Computer Graphics (Proc of SIGGRAPH'84) 1984-1985.
- [5] Socha, D. Bailey, M. Y Notkin, D. *Voyeur: Graphical Views of Parallel Programs*. Conf. On Parallel Processing, 1987.
- [6] Hough, A. y Cuny, J. *Initial Experiences with a Pattern-Oriented Parallel Debugger*. Int. Conf. On Parallel Processing. 1987.
- [7] Stasko, J. *Tango : A framework and system for Algorithm Animation*. IEEE Computer. 1990.
- [8] Brown, M. *Zeus: A System for Algorithm Animation and Multi-view Editing*. Technical report SRC-75, Digital- Systems Research Center. 1992]
- [9] Stasko, J y Kraemer, E. *A Methodology for Building Applications Specific Visualizations of Parallel Programs*. Journal of Parallel and Distributed Computing 1992.
- [10] Roman, G. Cox, K, Wilcox, C, y Plun J. *Pavane: a System for Declarative Visualization of Concurrent Computations*. Journal of Visual Languages and Computing. 1992.
- [12] Stasko, J., Domingue, J., Brown, M., Price, B. *Software Visualization: Programming as a Multimedia Experience*. MIT Press, 1998.
- [13] Bazik, J. Tamassia, R., Reiss, S y van Dam, A. *Software Visualization in Teaching at Brown University*. 1996.
- [14] Wirth N. *Program Development by Stepwise Refinement*. Communications of the ACM. 1971.
- [15] Conroy K. y Smith R., *NEATER2: A AP/I Source Statement Reformatter*. Communications of the ACM. 1972.
- [16] Parnas D. y Weiss, D. *Active Design Reviews: Principles and Practices*. Proceedings of the 8<sup>th</sup> International Conference on software Engineering. 1985.
- [17] Kernighan B y Plauger P. *Software Tools*. Addison-Wesley, Reading, MA, 1976.

- [18] Wirth N. *Modula: A Language for Modular Multiprogramming*. *Software- Practice and experience*. 1977.
- [19] Booch G. *Object Oriented Design with Applications*, Benjamin/Cummings, 1984.
- [20] Wasserman A. *Tutorial: Software Development Environments*. IEEE Computer Society Press, 1981.
- [21] Dart, S.; Ellison, R.; Freiler, P y Habermann. *A Software Development Environments*, Computer 1987.
- [22] Chifosky, E. y Rubenstein, B. *Categorisation and representation of physics problems by experts and novices*. *Cognitive Science*, 1981.
- [23] Miller, G. *What to Draw? When to Draw? An Eassy on Parallel Program Visualisation*. *Journal of Parallel and Distributed Computing*. 1993.
- [24] Lawrence, A., Brade, A., Stasko, J., *Empirically Evaluating the Use of Animations to Teach Algoritms*. Technical Report GIT-GVU-94-07, Graphics, Visualisation, and Ussability Center, College of Computing. Georgia Institute of Technology. 1994.
- [25] Price, B.; Beacker, R; y Small, I. *An Introduction to Software Visualisation*. *Software Visualisation*. MIT Press. 1998.
- [26] Merril, P. at all . *Computers in education*. Allyn & Bacon. 1996.
- [27] Álvarez Méndez, J. *Didáctica de la lengua materna*. Madrid. Akal. 1987.
- [28] Contreras, D. *Enseñanza, profesorado y curriculum*. Madrid. Akal. 1994.

## A SYSTEM FOR CONTROL OF USE OF DIDACTIC MATERIAL AT THE DISTANCE

**M.Sc. Marcelo Iserhardt Ritzel**

Centro Universitário Feevale  
Rua Emílio Hauschild, 70 – Vila Nova, Novo  
Hamburgo/RS, Brasil – Caixa Postal 2121  
ritzel@feevale.br

**Ph.D. José Valdeni de Lima**

Universidade Federal do Rio Grande do Sul  
Av. Bento Gonçalves, 9500 - Agronomia, Porto  
Alegre/RS Brasil –Caixa Postal 15064  
valdeni@inf.ufrgs.br

### ABSTRACT

The increases checked in the processing capacity and in the power of communication of the nets they allow to propose didactic systems with the integration of new technologies to the education, such as audio and video.

Today, it can observe an expressive increase of digital information published through the Internet. This research emphasizes the use of the computer and of the Internet as parts of the teaching atmosphere at the distance. The implemented system allows to establish (i) measures of control of use of the available didactic material in the Internet, as well as the application of (ii) evaluation techniques for, finally, (iii) to specify concrete approaches to compose the evaluation process of the learning at the distance. For so much, it was projected and implemented a software for this atmosphere type with the objective of checking that the ideas proposals are viable.

A direct consequence of the use of this atmosphere type is the need of a larger training so much of teachers as of students, for demanding a change of the habitual teaching-learning process, such as (i) the operation of the atmosphere of computerized teaching, (ii) the softwares involved for sailing and responsibility, (iii) notions of nets of computers and (iv) the introduction of evaluation approaches in the system.

### 1 INTRODUCTION

A lot of considerations should be made so that there is possibility of implantation of this " educational " method, such as: the appropriate and attractive format of the didactic material to be published; its use (with relationship the interface); the accompaniment of the use of this material; the control about the activities be she exercised by the students; the evaluation of the reached learning; the physical structure to assist in a satisfactory way its implantation and, mainly, its use.

Still, considering that one of the great advantages of the use of the teaching at the distance is to personalize the own teaching of a same content in different speeds, of agreement the capacity of the students' learning, and that to establish a control form he becomes a facilitative element in the definition of the teaching strategies to offer these contents different to the students, its different second and respective objectives, with views to look for the most efficient and more productive work, this work comes to contribute in a quite significant way for this proposition.

### 2 PROPOSAL OF THE SYSTEM

To have the student's control in this teaching process at the distance is quite important, once it contributes to the individual learning [1] [2]. Some factors stand out in what he/she refers to personalize this process, as the (i) several existent sailing processes [3], the (ii) specific objectives that each class possesses and the (iii) interests that each one has with relationship to the provided content. The student's of the objectives escape proposed by the course, that usually happens in function of the new interface or simply for the atmosphere of differentiated teaching, it is minimized through this efficient control, confirming the success of this process.

As well as the traditional teaching, this individual, personalized attendance, he/she can make the necessary difference in the search of the best learning, resulting in the improvement of the teaching.

Like this being, the measures of use control are fundamental for the proposition of a model of evaluation of the learning in this it sets of teaching at the distance. Such examples as the statistics accomplished on the traveled nodes of the didactic material, the sequence of nodes (links) traveled (effective sailing) [3], the time of permanence in each node (page HTML) and the not visited nodes, referring to the appraised subject, they can be part of the evaluation as a completely. It is known that these measured certainly cannot evaluate the learning directly, but yes, to cooperate as part of this evaluation process.

## 2.1 DESCRIPTION OF THE PROJECT

This project seeks to specify a proposal that allows the use control and the accompaniment of all the activities exercised on the available didactic material in the Internet, as well as to use the acquired results of this use for the assembly of a base of data that, together with specifications done by the teacher and/or assessor, he can be part of the composition of the process of evaluation of the learning.

The proposed model bases on the architecture customer-server used in the Internet, where the "didactic" material represents the illustration of the server, of the type Web, while the students represent the customers, through its navigators.

The system should foresee the use of two basic modules. A first one, responsible for the control/accompaniment of the user's/student sailing and generation of the log files, where it consists all the activities accomplished by this student in the site (didactic material properly says). Para that the proposed model can accomplish that accompaniment and generation of the logs, three stages should be accomplished:

- A registration process: The students / users that will have access to the didactic material published at the distance, will owe, initially, to register in the respective site. The same can be accomplished automatically (at the distance), being filled the available form in the initial page of the course. This registration process enables the access to the whole didactic material, after having made the logon in the course.
- The student's initial access with its identification: species of book virtual point, the corresponding to doing the call " in the traditional teaching. It should register every moment of the student's entrance / user in referred him course, what points for the beginning of its study.
- The sailing properly said in the didactic material, the accompaniment process: all the activities accomplished by the user/student on the site (on the didactic material/course, used as synonyms in this work) they are stored in the log files. It is important that the student has science that is being monitored in whole site, that its " steps " are being stored in log files. First, because he/she happens more credibility of the course when, since its beginning, they are shown whole the rules and its operation and, in second place, the students, in a general way, take its school activities more seriously (academic) when they know that are being observed by the teacher (monitored by the system). of course this last one is more evident when the resource of the teaching at the distance is used mainly for complementary activities of study. It is notorious that the decisive factor of the student's success, with this or other systematic of teaching any, is its degree of interest and dedication to the specific activities of the course.
- Already in the second module, once established the control of the sailing and generated the respective log files, there is the need to generate a base of data with all those information. That base is taken for another atmosphere with the purpose of to analyze and to refine everybody these data.

Already in the second module, once established the control of the sailing and generated the respective log files, there is the need to generate a base of data with all those information. That base is taken for another atmosphere with the purpose of to analyze and to refine everybody these data.

Besides, the teacher and/or assessor establishes a group of specifications regarding the didactic material, composing them in approaches that will be part of the process of evaluation of the learning. They are defined the important judged approaches for the proposed course, specifying for them a group of values and/or respective indexes. Based on these approaches filters and statistical and graphic reports they are accomplished on this base of data, in agreement with the established specifications, so that it cannot him, like this, to generate notes for the students in each one of these approaches, and still, it can be identified a profile for this student or group.

The main objective of this module is, starting from the specifications done by the teacher and / or assessor for the proposed course, together with the integral information of the generated base of data, to propose some approaches (in agreement with the teacher and / or assessor) to compose the process of evaluation of the learning. These approaches refer, mainly, the characteristics observed in the traditional teaching that they continue to be used in this methodology, even so, generated by different mechanisms. Still, for each used approach a note is generated, generated by the own module, that can still be used so that he/she/it can identify a specific profile generated for the students or for the group as a completely.

## 2.2 Module II

In the current version, the prototype fastens the whole group of rules regarding the approaches proposed to compose the process of evaluation of the learning, in way to establish a group of degrees, of notes. The numeric generation of the degrees is made, basically, starting from percentile established in the comparison and / or analysis done among the suggested information and esteemed by the teacher and / or assessor, with those generated again by the student during the process beginning of the course. This way, for example, in that module a percentile one is established equivalent at 100% for those students that navigated in all the pages of the didactic material, of 50% for those that navigated equivalent amount to the minimum of dear pages and, starting from there, in a decreasing way the amount of navigated pages arrives at 0% of the total. The more appropriate judged form, in agreement with the established in the conceptual model of the proposal done for the system, it is that those percentile ones can vary in agreement with the judgement and the teacher's approaches or assessor, because certainly it is found a diversity of approaches and different teachers' positions, related to a same module, subject or study area.

## 2.3 Approaches for evaluation of the learning

The model should propose, starting from its module II, a group of approaches to aid in the composition of the evaluation of the learning, or, in some cases, until executing it. The main source for formation of this base of data resides in the log files generated by the module I.

Besides this proposition, a series of information can be modeled starting from data maid's base, and that it can vary in agreement with the objectives proposed by the different courses at the distance, or in agreement with the work profile exercised by the teacher, author and / or assessor of the didactic material, and still, for the several characteristics that the students can have, be fruit of its origin diversity, formation or experience in the respective area of interest of the course in subject.

Independent of these items it was established a minimum group of information that, analyzed individually and in group, it is believed they be useful in the specification of the approaches that one wants to propose in the composition of the evaluation of the learning. The sailing sequence generated by the student during the accomplishment of the course at the distance. A degree for this sailing is generated starting from the comparison of the chain of characterses of the sailing sequence

suggested with that generated by the student. He suggests himself that the percentile of the found similarity corresponds to east degree, in agreement with the structure specified for this field in the table. Still, it can identify more than a suggestion for this suggested sequence, being used as degree the found percentile best of similarity.

◆ The total time used by the student to accomplish the course at the distance. They are added all the times of the student's permanence in each one of the monitored pages, and compared with the time suggested as ideal to accomplish the whole proposed course in a satisfactory way to the proposed objectives. The degree destined at the time is resulting of the analysis of the total time added for the student with that esteemed (suggested), relative to the percentile of proximity of these times: the same (100%), limit superior of twice the dear time (0%), limit inferior zero of time (0%) and limit medium suggested minimum.

◆ The date at the beginning and the date of end of the accomplishment of the course, referring to the first and last access (I liberate virtual point, LogOn) accomplished in the didactic material. A lot of times, in a traditional way, it can be noticed the degree of the student's interest by its participation in classroom, during the accomplishment of a given discipline in a semester, for example. It would be possessor of a larger " consideration " that student that of the beginning to the end of the classes was shown interested, and not just in certain period, in agreement with its interests, or even comfort. This referring items the dates at the beginning and end has for objective to verify the period of the student's participation in the accomplishment of the course at the distance. He suggests himself that the number of days in that accomplished the course it is compared with that esteemed, fruit of the difference of time among the initial and final date, discounting of this last one 10%, as margin of safety (for those that contained it previously the foreseen date, after accomplishing all the activities proposals for the course). Like this, the number of discounted foreseen days the 10% would indicate the 100% of this approach, decreasing starting from there until zero, the degree regarding the period.

◆ The accomplished accesses can be considered (LogOn) on the material as indicative of participation, what corresponds the presence in classroom to each present class, in a teaching method typically traditional. It can be foreseen, for example, that once a day, in the minimum, and three, at the most (for not decreasing the granulation a lot), the student accomplishes accesses to the material (virtual point book), and these accesses correspond to a degree of the student's participation in the course. This item can be shared with the previous, that is, analyzed jointly with referred them dates at the beginning and end.

◆ The amount of pages visited by the student is an approach that corresponds, in a similar way, to the number of pages certain book read by the student in classroom. On the accomplished control, the total number of pages is counted visited by the student, and for that case, he/she is considered valid the page that obtained time of superior permanence to thirty seconds. It is established the total amount of pages that they integrate the didactic material (those that one wants to monitor), as parameter of 100%, and a minimum amount of pages that they should be visited by the student, suggested by the teacher and/or assessor, so that he assists to the minimum requirements proposed for the course in subject, being indicated by the attribute Aval\_Indicad.PercPagMin. Starting from this point, there is a proportional decreasing until the zero, for the respective degree of this approach.

◆ Electronic Subjects distributed along the didactic material can be useful for the sum or profile of the evaluation. The correction, also accomplished in an automated way in the same ambient Web, generates degrees when being compared with the correct solutions proposed by its teacher and / or author. The average generated for these degrees, in agreement with the amount of subjects, or groups of existent subjects, generates a degree of electronic subjects. A quite interesting item, because besides forming more an approach to the evaluation, he can serve as reference, and even of requirement, for the finish or passage among modules inside of the same didactic material

(levels of difficulty, of progression). Still, the degrees obtained in certain subject of the material can be analyzed together with its respective times of study (sailing) obtained.

Many of the approaches can assume a high subjectivity degree if taken separately for the evaluation of the learning. For example, a student that stays an excessive judged time to travel certain chapter or subject of the didactic material, it cannot, if studied in an isolated way, to mean a high degree of interest and effort so much by virtue of the difficulty, as well as, of interest and effort by virtue of the easiness, identification and personal taste. For so much, this dice could be better measured analyzed jointly with the worn-out total time for the whole didactic material and the average for chapter or this same student's subject.

It is worth to stand out, that the own education at the distance possesses a very peculiar group of own characteristics, as well as the traditional form of teaching. Many of these characteristics were brought of the "old school", and that here generated in a different way, they continue to exercise the same importance in this new methodology. It is, simply, a new proposition. The substitution intention is not had, but, of intending an alternative in the attempt of reversion of the current teaching picture, mainly of Brazil, as well as of extending this modality for several and different applications.

### **3 ENVIRONMENTS PUBLIC OBJECTIVE OF THE STUDY OF CASE**

The case study was accomplished with students of the 1st series of the Medium Teaching (6 groups with 150 students, like this distributed: groups 101, 25 students' 105 and 106; groups 102 and 103 of 24 students; and 27 students' group 104) of a public school, in the period understood between 6 and September 20, 1999, it leaves of the 3rd Two months, in the discipline of History. To 1st series it is composed for.

The chosen course went it remains to Batalha dos Guararapes, whose didactic unit makes part of the Plan of Matters of this discipline and series.

#### **3.1 Applied methodology**

The accomplished work was differentiated among the 6 groups. Two groups, 102 and 104, they studied the subject Guararapes just using the available didactic material in the Internet; other two, 103 and 105, besides having studied the material through Internet, the likeness of the above exposed, he/she had in class room, in a traditional way, the classes worked by the teacher of History on that subject; and the two remaining groups, 101 and 106, they had the classes you just witness worked by the teacher in class room. This way, the whole group of students of the 1st series would have access the information on the Battle of Guararapes, one at the distance, others in a traditional way, but with the same subject, in volume and quality fellow creatures, just differentiated by the peculiar characteristics of each involved environment.

At the end of the course, in both teaching forms, a same local test was executed with the 6 groups. Starting from there, of ownership of the all the students' results in the test, together with the logs generated in the participant students' accompaniment of the course at the distance, it could be arrived to relative conclusions, or the simple indications, that the approaches initially judged important to compose a process of evaluation of the learning they are, really, significant, and like them they behave when being compared with those obtained in a traditional way.

### **4 CONCLUSIONS**

The teaching at the distance is an important alternative in the search of the basic and complemental formation of the learning, which, in a main way, it motivated the study of a model that could not in an efficient way to control and to monitor the accesses to an available didactic material at the distance, in ambient Internet, and that could contribute with important approaches to the evaluation of the learning, that comes to contain the whole process.

The developed model proposes an efficient and viable alternative for this implication, in way to reach the proposed objectives. He generates a (i) base of data originating from of the control accomplished on top of all the activities exercised on the didactic material, without the need of any alteration in the natural form of I handle of this teaching atmosphere, and it creates a (ii) base of information, originating from of the module I and of the author's interactive (teacher or assessor).

The study fact could show that (i) the teaching at the distance is a much more difficult methodology than the traditional form, mainly in what he/she says respect at the time I spend properly for the material elaboration and accompaniment of the course said; that (ii) it can have conflict of the chosen approaches and mensurated with the didactic material and, for consequence, with the accompaniment process (objectives and proposed teaching strategy); and that the (iii) historical of the accompaniment it can aid in the identification of problems in the interface, allowing, still, (iv) the association of a profile individualized to each participant user of the process, with a series of relative information the person, to the didactic material, to the course, to the exercise of the course, and to the crossing relationships among these information.

The teaching at the distance is an important alternative in the search of the basic and complemental formation of the learning, which, in a main way, it motivated the study of a model that could not in an efficient way to control and to monitor the accesses to an available didactic material at the distance, in ambient Internet, and that could contribute with approaches reliable to the evaluation of the learning, that comes to contain the whole process.

## BIBLIOGRAPHY

- [1] BARRETO, L. S.; MAGALHÃES, M. R. de A.; NASCIMENTO, M. E. M. Aplicação de multimídia interativa em um courseware. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 14., 1996, Fortaleza. **Anais ...** Fortaleza: UFC, 1996.
- [2] SALOMON, G. **What is learned and learner in media and symbols:** The forms of expression, communication, and education. Chicago: [s.n.], p. 383-408, 1974.
- [3] GOTTSCHALK, Tania H. **Distance Education at a Glance.** Disponível por WWW em <http://www.uidaho.edu/evo/distglan.html> (17 fev. 1998).
- [4] CENTER for Advanced Educational Services. MIT. Disponível por WWW em <http://www-caes.mit.edu> (03 ago. 1998).
- [5] CESAR, H. Lenz. Utilização do modelo multi-instrumental ao ensino à distância. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES, 14., 1996, Fortaleza. **Anais ...** Fortaleza: UFC, 1996.
- [6] GLATZ, Phil. Knee-Deep in Log Files. **Web Techniques**, San Francisco, v.4, n.7, p. 79-85, July 1999.
- [7] GONÇALVES, C. T. Fernandez. Quem tem medo do Ensino à Distância. **Revista Educação à Distância INED/IBASE**, n. 7/8, jul./ago. 1996.
- [8] HANSOO, Bae; HYUNGSUK, Kim. A model of remote education system using MFS. In: ED-MEDIA & ED-TELECOM 97, 1997, Virginia. **Proceedings ...** Virginia: Multimedia Research Lab, R&D Group, Korea Telecom., 1997.
- [9] KOSMA, R. B. Learning with media. **Review of educational research**. v. 61, n. 2, p. 179-211, 1991.
- [10] MEDINA, Nelkis de La O.; FARINES, Jean-Marie. Um suporte para aplicações cooperativas multimídias sobre a Internet. In: WORKSHOP EM SISTEMAS MULTIMÍDIA E HIPERMÍDIA, 3., 1997, São Carlos. **Anais ...** São Carlos: USP, 1997.
- [11] SHERRY, Lorraine. **Questões sobre educação à distância, baseado em "Issues in distance learning"**. Disponível por WWW em <http://penta.ufrgs.br/edu/edu1.html> (15 out. 1998).

## CONSTRUCCIÓN DE TUTORIALES BASADOS EN COMPONENTES REUSABLES

Mg. Ing. Zulema Beatriz Rosanigo  
Ing. Pedro Bramati  
A.P.U. Alicia Beatriz Paur

[pbrsa@satlink.com](mailto:pbrsa@satlink.com)  
[bramati@infovia.com.ar](mailto:bramati@infovia.com.ar)  
[apaur@topmail.com.ar](mailto:apaur@topmail.com.ar)

Facultad de Ingeniería – Universidad Nacional de la Patagonia San Juan Bosco – Sede Trelew  
Roca y Belgrano – (9100) Trelew –Chubut – Argentina  
TE-Fax: 02965 42 84 02 e-mail: [fi.trelew@unp.edu.ar](mailto:fi.trelew@unp.edu.ar)

### Resumen

El presente trabajo es parte del proyecto de investigación PI 383 en el área de informática educativa denominado "Construcción de tutoriales basados en componentes reusables" que tiene por objeto facilitar el desarrollo de material didáctico en general, y de tutoriales en particular, con capacidad de ser utilizados a través de Internet y basados en componentes reusables que soporten un conjunto de comportamientos estándares, independientemente de la funcionalidad específica para la que han sido diseñados.

Se trata de diseñar una herramienta, que a partir de algunos componentes –desarrollados especialmente para el dominio de aplicación– el docente pueda construir tutoriales del tipo enseñanza paso a paso, para que el alumno pueda ejercitarse tanto como lo necesite.

Este tutorial que acaba de crear, se convierte en un nuevo componente del dominio y por lo tanto puede ser utilizado para la creación de otro, debiéndose adaptar en forma inteligente al nuevo contexto.

### Introducción y motivación

En los últimos años, los investigadores tanto de áreas tecnológicas, como de las educativas, han trabajado intensamente con psicólogos, sociólogos, pedagogos, diseñadores gráficos, etc., en el desarrollo de estrategias que ayuden a mejorar la calidad de la educación, apoyándose en las capacidades que hoy brinda la tecnología.

Entre los requerimientos básicos que todo curso de educación superior por ordenador necesita para cubrir adecuadamente sus objetivos, podemos considerar:

- gran nivel de interactividad
- entorno de aprendizaje agradable
- secuencias animadas de determinados procesos
- posibilidad de simulaciones
- facilidad de actualización y utilización
- capacidad de adaptación
- capacidad de motivación y orientación para una mayor especialización.

La red WWW (World Wide Web) se ha consolidado como un recurso muy valioso de aprendizaje y educación al permitir exponer a un amplio número de individuos una información variada,

fácilmente actualizable y accesible, disponible independientemente de la plataforma utilizada.

La principal ventaja de Internet sobre otros medios de comunicación es la interactividad y uno de los recursos que permite explorar mejor la interactividad en la web es el lenguaje Java, a través de aplicaciones pequeñas y seguras (applets). Con Java se pretende buscar una solución de alta calidad del producto final pero manteniendo un costo de desarrollo razonable. Una manera de ayudar a cumplir con este objetivo es maximizar el reuso y posibilidad de evolución. El reuso produce reducción de tiempo y costo e incrementos de calidad, en la medida que el desarrollador pueda encontrar, utilizar y adaptar al nuevo contexto, aquellas soluciones que ya han sido probadas y usadas exitosamente.

Por ello, este proyecto busca encontrar mecanismos que permitan integrar esta interactividad en el proceso de enseñanza-aprendizaje, y facilitar la construcción de material didáctico y herramientas educativas, aplicando los conceptos modernos de la ingeniería de software y de la tecnología orientada a objetos, que pone énfasis en la reusabilidad y flexibilidad de las soluciones.

En nuestro caso el interés se centra en la generación y utilización de tutoriales y entrenadores como apoyo y complemento a la enseñanza universitaria, en pos de solventar dificultades que se plantean en el aprendizaje de algunas materias, como es el caso de Dibujo técnico o Sistemas de Representación, que requieren del desarrollo de ciertas habilidades por parte de los educandos y disponen de escaso tiempo para alcanzarlas.

La insuficiente formación del alumnado al iniciar estudios universitarios y la disminución de tiempos en los nuevos planes de estudio ofrece un problema adicional para su desarrollo. Un tutorial ayudará en el proceso de enseñanza-aprendizaje mostrando, por ejemplo, la resolución de problemas o ejercicios teóricos prácticos paso a paso y permitiendo aumentar las posibilidades de ejercitación de los alumnos en clases y extraclases.

Se ha elegido realizarlo sobre contenidos de la asignatura "Sistemas de Representación" y así brindar una herramienta educativa complementaria que permita disminuir los tiempos de impartición sin afectar sus contenidos, habida cuenta de la disminución de horas lectivas que esta asignatura tiene en los nuevos planes de estudio y de la necesidad de otorgar a los alumnos una formación que tienda al desarrollo y asimilación de la concepción espacial con el sentido de solucionar proyectos concretos. De esta manera no se sacrifican tiempos de la temática de desarrollo adoptada, que es en definitiva la *interrelacion CMA+CAD* (Croquizado a Mano Alzada y Diseño Asistido por Computadora), sino que por el contrario, este complemento involucra un conjunto de actitudes cognoscitivas básicas para el correcto entendimiento de la problemática espacial, su representación proyectual y las aplicaciones correspondientes apoyados en la interactividad que otorgan sus contenidos que son inherentemente gráficos y visuales.

La utilización de tutoriales basados en multimedia es una herramienta a tener en cuenta en todas las materias de la currícula de cualquier carrera universitaria. El profesor ha de ser consciente de que su tarea respecto a los medios no es conocer las rutinas de su uso o aplicación. Su incorporación al proceso de enseñanza-aprendizaje exige ingenio y creatividad, y sobre todo, contemplar "de otro modo" ese proceso, en la medida en que afectará a todos los elementos que lo componen.

Nuestro objetivo es facilitarle al docente la construcción de nuevos tutoriales, diseñando una herramienta, que a partir de algunos componentes desarrollados especialmente para el dominio de aplicación, el propio docente pueda construir tutoriales del tipo enseñanza paso a paso, para que el alumno pueda ejercitarse tanto como lo necesite.

Con la utilización de este tipo de tutorial, se espera resolver una de las dificultades que se plantea en el aprendizaje tradicional de algunas materias demasiado técnicas, donde en forma usual el alumno se encuentra con la resolución impresa en un solo paso, con todos los procesos intermedios obviados. Al poder seguir el proceso "paso a paso", se eliminará la dificultad antes expuesta.

**Algunos requisitos funcionales:**

- Dos modalidades de uso bien definidas: creación de tutorial y uso de un tutorial.
- El trabajo de creación de un tutorial mediante la herramienta, debe resultarle al docente tan natural como si lo desarrollara manualmente, y en general, mucho más sencillo. Esto implica que muchas tareas pueden y deben estar automatizadas.
- El tutorial que acaba de crear debe poder convertirse en un nuevo componente del dominio y por lo tanto, podrá ser utilizado para la creación de otro, debiéndose adaptar en forma *inteligente*, al nuevo contexto.
- En modalidad de ejecución de un tutorial, puede ser practicado individual mente tantas veces como se quiera, en forma completa o parcial, permitiendo volver atrás cuando el alumno lo requiera.

**Estado del proyecto:**

Con el objeto de adquirir conocimientos avanzados en el uso de Java y diseño conducido por patrones, los integrantes del área de desarrollo de software tomaron cursos específicos e intensivos.

Para lograr un lenguaje común entre los integrantes de la unidad ejecutora y comprender los contenidos de "Sistemas de Representación", se realizaron un taller coordinado por el especialista en el tema.

Los aspectos relevantes del dominio de aplicación quedaron claramente caracterizados al finalizar el taller de Sistemas de Representación.

De este taller, surgió la idea de tomar un tema elemental como el "trazado de perpendiculares" para poder ejemplificar, probar y visualizar las posibles soluciones de diseño que se van proponiendo. Para ello se desarrollaron en Java los componentes necesarios sobre los cuales se va probando y evaluando diferentes técnicas y soluciones.

El relevamiento de necesidades y preferencias se extendió hacia otras áreas de aplicación además de Sistemas de Representación, de manera de contemplar en el diseño, soluciones más versátiles.

En forma casi permanente se observan y evalúan productos educativos en la Web buscando "descubrir" patrones de diseño que contribuyan a soluciones flexibles.

En función de definir y generar componentes educativas que integran el tutorial, se establecieron los aspectos que tienen mayor impacto en el diseño, y las primitivas propias del dominio de aplicación.

Se ha definido una arquitectura flexible para generación de tutoriales, y se están haciendo pruebas.

Actualmente se está trabajando sobre el aspecto de *persistencia de pasos inteligentes*: Puesto que un paso de un tutorial a crear puede ser tanto un componente primitivo (ya desarrollado) como un tutorial creado con anterioridad por el docente, nos enfrentamos a la situación en que no sólo el alumno debe aprender, sino que el sistema que estamos diseñando también debe "aprender" a utilizar los tutoriales creados, de la misma manera que sabe utilizar las primitivas. Esto implica que:

1. *cada tutorial debe tener un mecanismo de persistencia.*
2. *cada tutorial debe autoejecutar los pasos en función de las entradas del nuevo contexto en que se ejecuta y las salidas de los pasos anteriores.*

## Conclusiones

Nuestro objetivo es facilitarle al docente la construcción de nuevos tutoriales, diseñando una herramienta, que a partir de algunos componentes desarrollados especialmente para el dominio de aplicación, el propio docente pueda construir tutoriales del tipo enseñanza paso a paso, para que el alumno pueda ejercitarse tanto como lo necesite.

Con la aplicación sistemática de patrones de diseño y procurando diseñar soluciones flexibles y reusables, creemos que la herramienta que proporcionaremos para crear tutoriales sobre temas de *Sistemas de Representación*, podrá fácilmente convertirse en un framework de generación de tutoriales, de modo tal que, desarrollando las primitivas propias de cada dominio, pueda ser usada en él.

Con este tipo de herramienta, el docente podrá crear tutoriales en forma amigable y sencilla, y el alumno, podrá practicar o aprender paso a paso, a su propio ritmo y necesidad.

## GENERALIZACIÓN DEL MODELO DE HIPERMEDIA PARA LA CONSTRUCCIÓN DE REDES CONCEPTUALES

Mg. Ing. Zulema Beatriz Rosanigo  
A.P.U. Alicia Beatriz Paur  
Ing. Pedro Bramati

[pbrsa@satlink.com](mailto:pbrsa@satlink.com)  
[apaur@topmail.com.ar](mailto:apaur@topmail.com.ar)  
[bramati@infovia.com.ar](mailto:bramati@infovia.com.ar)

Facultad de Ingeniería – Universidad Nacional de la Patagonia San Juan Bosco – Sede Trelew  
Roca y Belgrano – (9100) Trelew – Chubut – Argentina  
TE-Fax: 02965 42 84 02 e-mail: [fi.trelew@unp.edu.ar](mailto:fi.trelew@unp.edu.ar)

### Resumen

Este trabajo de investigación (PI 253) se enmarca en la línea de informática educativa. Propone una metodología para el desarrollo de hipermedios educativos que respeta los lineamientos actuales de la Ingeniería de Software para el desarrollo y construcción de productos de calidad, y contempla los principios y objetivos de la teoría educativa subyacente.

Esta metodología se ha ido aplicando en la construcción de una herramienta educativa hipermedial que cubre diversos aspectos de la materia "Construcciones e Instalaciones en Edificios" de la carrera de Ingeniería Civil.

Los resultados del proyecto pueden resumirse en:

Desde el punto de vista de formación de recursos humanos:

- Motivar a los integrantes y a otros docentes a participar de jornadas y eventos científicos.
- Interesar a otros docentes y alumnos en participar en actividades de investigación.
- Formar un grupo de trabajo que continuará con esta línea de investigación.

Desde el punto de vista de la transferencia tecnológica:

- Brindar una metodología adecuada para la generación de herramientas educativas hipermediales.
- Proporcionar un Software Educativo para la Cátedra de Instalaciones en Edificios.

### DISCUSION

Las posibilidades que ofrecen las nuevas tecnologías adaptadas a las necesidades educativas permiten disponer de un material didáctico elaborado específicamente para su empleo en computadoras y que pueda utilizarse tanto en puestos independientes como conectados en red empleando una arquitectura cliente-servidor. El objetivo último es intentar mejorar el proceso enseñanza-aprendizaje, promoviendo el protagonismo del sujeto y facilitando el trabajo que para alumno y profesor supone la tarea de formación.

El aprendizaje es un proceso de desarrollo de estructuras significativas. La formación y desarrollo de estas estructuras, depende del modo como percibe una persona los aspectos psicológicos del mundo personal, físico y social.

Para aprender significativamente una persona debe tratar de relacionar los nuevos conocimientos con los conceptos y las proposiciones relevantes que ya conoce. De esta manera construye su propio conocimiento y además está interesada y decidida a aprender, es un proceso activo y personal.

Un software educativo (SE) debe poner el énfasis en lograr aprendizajes significativos, brindando posibilidades de vincular los nuevos conceptos con los que se tienen adquiridos y estableciendo relaciones no arbitrarias entre ellos. Debe tener en cuenta los objetivos de la unidad, además de los contenidos y sus relaciones. También debe brindar mecanismos de evaluación, tanto para el alumno como para el docente.

Cada alumno tiene sus propias necesidades, motivaciones, deseos, aspiraciones, las cuales dependen de su estructura cognitiva y varían por medio del aprendizaje. En el diseño de un SE, también es muy importante tener en cuenta las diferentes motivaciones con que el alumno puede acercarse a él: aprendizaje básico de un tema, aprendizaje detallado, repaso de conocimientos, búsqueda de información “en profundidad” o “en amplitud”, autoevaluación.

La tecnología hipermedial es un valioso recurso para la construcción de un SE al permitirnos:

- Reunir en un mismo material distintos recursos: video, animación, sonido, fotografía, texto, etc., permitiendo un aprendizaje multisensorial.
- Representar información no-lineal, permitiendo asociar distintos componentes de una aplicación.
- Materializar las redes de asociaciones conceptuales que vinculan los contenidos de una unidad didáctica y sus relaciones primarias.
- Manejar grandes unidades de información a ser compartida por grupos que trabajan en forma colaborativa.

Ya que la hipermedia permite manejar con total libertad la red conceptual que se forma, es necesario tener en cuenta el modelo de enseñanza-aprendizaje elegido y adoptar una metodología que asegure aprendizaje significativo.

La metodología propuesta para el diseño de SE hipermedial, que surge de esta investigación, combina reglas existentes en el campo de la Ingeniería de Software y en el de las Ciencias de la Educación, incorporando aspectos educativos en etapas tempranas del ciclo de vida del software.

Consiste en la definición de las etapas necesarias para lograr el producto, considerando las herramientas, técnicas y recursos necesarios en cada una de las mismas. Distingue en el proceso de diseño tres etapas: *Diseño de la información*, *Diseño de la interacción* y *Diseño de la presentación*, resaltando la importancia de la organización y estructuración de los contenidos para lograr redes conceptuales coherentes y significativas.

Esta coherencia interna, se logra mediante un desarrollo metódico, que permite realizar las conexiones lógicas y conceptuales entre los elementos. Nuestra propuesta sugiere la utilización de redes conceptuales hipermediales (RCH), basadas en el modelo de aprendizaje significativo de Ausubel y mapas conceptuales de Novak [Novak+88], y enriquecidas con algunas características de las redes semánticas y mapas conceptuales hipermediales de Señas [Señas +97].

Los principales aportes de este modelo son:

- La incorporación y contemplación de aspectos pedagógicos, educativos y comunicacionales dentro de cada etapa del ciclo de vida.
- La estructuración de las actividades a realizar para el diseño y desarrollo del software.
- La pronta incorporación de un prototipo que permite evaluaciones tempranas del producto y va evolucionando constantemente hasta convertirse en el producto deseado.

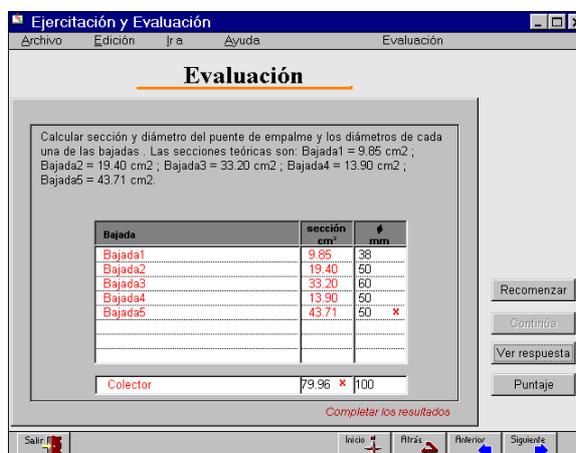
Esta metodología se ha ido aplicando en la construcción de una herramienta educativa hipermedial que cubre diversos aspectos de la materia "Construcción de Edificios e Instalaciones" de la carrera de Ingeniería Civil. El software, denominado SEDIE (Software Educativo para Instalaciones en Edificios), tiene como funcionalidad pretendida, guiar el aprendizaje como apoyatura a la



Incorpora un módulo para autoevaluación, el estudiante puede elegir si la realiza o no. Puede ver las respuestas correctas y el grado de efectividad logrado. Los tipos de ejercicios contemplados son:

- múltiple-choice (una pregunta y un conjunto de respuestas entre las que debe seleccionar la o las correctas),
- verdadero - falso o de respuesta si - no
- problemas con una o varias respuestas numéricas calculadas

Si el alumno consulta la respuesta, el programa muestra los resultados correctos y marca las respuestas incorrectas. En este ejemplo de problemas con varios ítems a calcular, la respuesta es mostrada en ventana popup (no se muestra en la figura) y marca con una cruz los valores que no coinciden.



El docente puede cambiar las preguntas y sus respuestas sin modificar el software: editando el archivo de texto correspondiente. Cada tipo de problema debe respetar un formato sencillo de escritura para formular la ejercitación, así el programa lo interpreta y puede ajustar automáticamente su presentación.

## CONCLUSIONES

Sobre la propuesta metodológica:

- El desarrollo de hipermedios educativos requiere de métodos específicos, que incorporen aspectos educativos en etapas tempranas del ciclo de vida del software.
- La utilización de las RCH como modo de organización de contenidos, facilita el traspaso de los mismos al escenario hipermedial de un SE. Si bien la elaboración de las redes no es una tarea sencilla, su correcta concepción, garantiza la eficiencia del SE en la construcción del conocimiento y logro de aprendizaje significativo.
- Tan importante como la organización de la información mediante las RCH, es cuidar la calidad de los contenidos y la apariencia que toman en su representación. El tiempo invertido en ello se convierte en beneficios futuros.
- La integración, evaluación y testeo temprano de las tres etapas del diseño previamente mencionadas, permite detectar y corregir defectos antes que se vuelvan problemáticos, decidiendo a tiempo, la conveniencia o no del uso de determinados recursos.

Sobre el software obtenido y su utilización:

- El simple uso de hipermedios didácticos, no hace de los estudiantes mejores aprendices; su incorporación requiere modelos muy claros de manera que permitan la apropiación de los contenidos presentados. Las actitudes y creencias que tenemos hacia ellos determinan la forma en que interactuaremos y, en consecuencia, los resultados que se obtengan.
- Hasta la fecha los resultados de SEDIE han sido favorables, la respuesta por parte de los alumnos ha sido positiva y el proyecto ha despertado interés entre los docentes.
- El uso de SEDIE permite orientar las clases presenciales de los alumnos de forma que se impartan los contenidos haciendo especial hincapié en los aspectos más conceptuales y más difíciles de entender.

- La facilidad con que se confeccionan las evaluaciones y se incorporan automáticamente en el módulo de Evaluación, es una de las características más apreciadas por parte de quienes lo utilizan, tanto para el alumno que desea autoevaluarse como para el docente que desea tener indicadores acerca de la comprensión de los temas por parte de los educandos.
- Si bien existe la posibilidad de cambiar contenidos en los módulos temáticos o agregar nuevos, no resulta tan automática como para con la evaluación. En general, las páginas de contenidos requieren ser revisadas y acomodadas para su presentación definitiva.
- Con la utilización del SEDIE como apoyo a la docencia, se ha logrado un beneficio notorio tanto en el desenvolvimiento de las actividades docentes específicas de la cátedra, como en el desarrollo de las actividades curriculares de los alumnos.
  - Además, al estar el alumno desarrollando un proyecto de Instalaciones bajo algún sistema CAD, y disponer de la facilidad de interconexión con el SEDIE, y consecuentemente obtener rápidamente la satisfacción a sus inquietudes, el tiempo de ejecución de sus labores se ve altamente disminuído.
  - Por otro lado, la cátedra pudo abreviar el tiempo de dictado en un cuarenta por ciento, al apoyarse en esta herramienta educativa, en donde está concentrada toda la información temática requerida.

Nuestra línea de trabajo e investigación inmediata, busca encontrar mecanismos que permitan integrar la interactividad en el proceso de enseñanza-aprendizaje, y facilitar la construcción de material didáctico y herramientas educativas, poniendo énfasis en la *reusabilidad y flexibilidad de las soluciones* sin que por ello se pierda el objetivo educativo que se pretende alcanzar.

## REFERENCIAS BIBLIOGRÁFICAS

- [Cejudo97] *Proyecto Innova: Redes conceptuales -Memoria del seminario de D. Ginés Delgado Cejudo – 1997-* <http://www.innova.ulpgc.es/doc/redes/index.htm>
- [Hinostroza+97] Hinostroza E., Hepp P., Straub P. - *Método Desarrollo Software – 1997 –* <http://www.enlaces.el/documentos/metododesarrollo/metodo.html>
- [Kristof+98] Kristof, Ray –Satran, Amy. *Diseño interactivo*. Ediciones Anaya Multimedia 1998.
- [Marqués99] Marqués, Pere – *Programas Didácticos: Diseño y Evaluación – 1999 -* <http://www.xtec.es/~pmarques/edusoft.htm>
- [Novak+88] Novak, J., Gowin, D - *Aprendiendo a aprender -* Ed. Martínez Roca. 1988 (Traducción de Learning how to learn (1984) Cambridge University Press)
- [Ontoria+97] Ontoria, A. , Ballesteros A., Cuevas C., Giraldo L., Martín I., Molina A., Rodríguez A., Vélez U. - *Mapas Conceptuales: Una técnica para aprender*. Narcea – 7ma edición 1997
- [Rosanigo+00] Rosanigo, Z.B., Paur, A., Bramati, P. *Metodología de desarrollo de software educativo*. Actas de VI Congreso Internacional de Ingeniería Informática ICIEY2K Fac. de Ingeniería, U.B.A. - Buenos Aires - 2000
- [Señas+97] Señas, P. , Moroni, *Mapas conceptuales hipermediales: una herramienta para el desarrollo de software educativo – actas CACIC 97 –UNLP - La Plata 1997*

## INFORMÁTICA Y EDUCACIÓN

Perla Señas

Laboratorio de Investigación y Desarrollo en Informática y Educación (LIDInE)

Instituto de Investigación en Ciencias y Tecnología Informática (IICTI)

Departamento de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur - Bahía Blanca

pseñas@cs.uns.edu.ar - T: 0291-4595101 interno 2614

Argentina

### Introducción

La investigación en Informática Educativa puede ser abordada desde dos perspectivas diferentes, desde las Ciencias de la Educación y desde las Ciencias de la Computación. La primera centra su atención en la influencia que tiene el uso de la tecnología computacional en los procesos de enseñanza y de aprendizaje, es decir en el valor de dicha tecnología desde lo pedagógico y lo didáctico. La segunda perspectiva abarca dos áreas visiblemente diferenciadas, una que se refiere al desarrollo de tecnología computacional orientada a la educación, claramente relacionada con las restantes áreas de Ciencias de la Computación y otra conformada por los temas propios de Didáctica de la Computación.

En el Laboratorio de Investigación y Desarrollo en Informática y Educación (LIDInE) del Departamento de Ciencias e Ingeniería de la Computación de la Universidad Nacional del Sur se ha capitalizado todo el trabajo que ha desarrollado el grupo InE en temas de Informática Educativa en los últimos once años. Las investigaciones se centran en modelos de aprendizaje constructivistas y en la incorporación de la computadora como herramienta colaboradora en los procesos de enseñanza y de aprendizaje en tanto esté identificada con el desarrollo de capacidades de pensamiento. Participan investigadores de Ciencias de la Computación y de Ciencias de la Educación cuyo trabajo comprende todas las áreas mencionadas. Actualmente se encuentran vigentes las siguientes líneas de investigación:

#### 1. Diseño y desarrollo de Ambientes de Aprendizaje

En el marco de las aplicaciones de la tecnología computacional a la educación se destaca de manera especial el trabajo sobre Ambientes de Aprendizaje. Las investigaciones que se realizan en el LIDInE se centran en modelos de aprendizaje constructivistas y en la incorporación de la computadora como herramienta colaboradora en los procesos de enseñanza y de aprendizaje. Para promover situaciones de aprendizaje constructivista los ambientes deben poseer facilidades para la construcción de conocimiento y para la participación activa del alumno en la ejecución de ese proceso. Con ese propósito se han creado los Mapas Conceptuales Hipermediales (MCH) [Señ96], un conjunto interesante de herramientas y estrategias visuales que los complementan, metodologías de desarrollo y plataformas interactivas para trabajar con todos estos recursos en Ambientes de Aprendizaje de diferentes tipos [Mor96], [Señ00]. Se ha estudiado también el problema de la interconexión de mapas y se han realizado y evaluado experiencias educativas en Ambientes de Aprendizaje MCH.

Se trabaja actualmente en una nuevas versiones de Plataformas MCH capaces de soportar los siguientes aspectos:

- Grafo Integrador Anidado ( $GIA_{MCH}$ )
- Mapa de vistas, [Mar01]
- Interconexión de mapas, [Mor00].
- MCH Multidimensionales, [Mor00].

Se continúa también investigando sobre herramientas y estrategias visuales para los sistemas de aprendizaje, en particular, para los ambientes MCH. Se busca potenciar la información representada en los mismos, ofreciendo acceso a la bibliografía, permitiendo un paso desde el esquema visual a documentos en los que se desarrollan y amplían los conceptos representados en las vistas de los MCH. Desde lo cognitivo se trata de proveer un mecanismo de adquisición de información dentro de un marco contextual, el brindado por el MCH.

## **2. Entornos para el aprendizaje de la programación**

Un capítulo importante de la Didáctica de la Computación, es el que se ocupa de los aspectos relacionados con la enseñanza y el aprendizaje de la programación. La programación es una actividad muy compleja que suscita interés desde lo psicopedagógico debido a sus efectos sobre las capacidades cognitivas. Está vinculada al análisis de los problemas o situaciones, a la planificación de acciones, a la reflexión previa al resolver y al actuar, y al aprendizaje de la lógica aplicada a circunstancias familiares y cercanas. Las nuevas estrategias metodológicas para la enseñanza de la programación mantienen el uso de la computadora como un recurso natural para tal fin, en todas sus etapas [Mor98]. En tal sentido, se sigue estudiando sobre el diseño y desarrollo de entornos interactivos específicos para el aprendizaje de la programación, siempre desde una perspectiva constructivista.

Se investiga actualmente sobre la “lectura comprensiva de algoritmos” y para ello se continúa trabajando en aplicaciones de visualización de software para entornos de programación estructurada y modular. Actualmente se analizan distintas alternativas para la creación de un Sistema de Visualización Interactiva de Algoritmos. Se debe decidir qué es lo que se quiere visualizar y cómo se va a obtener la información necesaria para hacerlo. En las visualizaciones de algoritmos, que requieren mayores niveles de abstracción, la descripción de la apariencia visual de los objetos lógicos está sujeta a un análisis profundo de la situación. En este caso el sistema de visualización debe ofrecer un conjunto de transformaciones y entidades gráficas para realizar la especificación correspondiente.

## **3. Representación de conocimiento en agentes pedagógicos**

Existe un tipo especial de agentes de software llamados Agentes Pedagógicos. Forman parte de sistemas donde colaboran agentes humanos y de software, integrando acción con instrucción. Son agentes autónomos que apoyan el aprendizaje humano integrando junto con estudiantes diferentes Ambientes de Aprendizaje Interactivos. Ellos tienen capacidad para mantener un espectro amplio de interacciones instruccionales efectivas con los alumnos que componen el entorno de aprendizaje. En estos sistemas la representación de conocimiento tiene un doble propósito, permitir hacer razonamiento automatizado y ser un recurso pedagógico eficaz para la construcción del conocimiento en los seres humanos. Es así como cobran gran interés los sistemas inteligentes dedicados al aprendizaje, en particular aquellos diseñados como Sistemas Multiagentes Mixtos.

Todos los Agentes Pedagógicos requieren algún tipo de representación de conocimiento que describa el tema de instrucción. Se ha estudiado la posibilidad de tomar a los MCH como estructura de representación de conocimiento en Sistemas Mixtos que incluyen Agentes Pedagógicos, se ha realizado una extensión de los MCH incorporando elementos de los Grafos Conceptuales de Sowa, definiendo así, los MC<sup>S</sup> y una arquitectura para la representación de la base de conocimiento de un Agente Pedagógico genérico [Señ99] y se ha propuesto una extensión a la plataforma MCH [Señ00b].

Actualmente se continúa investigando sobre la representación de conocimiento en Agentes Pedagógicos que integran Ambientes de Aprendizaje adecuados para hacer posible el trabajo de

autoría y se está estudiando el diseño de un motor pedagógico y la extensión de la plataforma MCH a los MC<sup>S</sup> para su uso en la creación de las bases de conocimiento temáticas. Con estos desarrollos puede hacerse un aporte significativo desde el área de Ciencias de la Computación al área de Educación que vaya más allá de lo meramente operacional.

#### 4. Aplicaciones MCH

Como esquemas de representación de conocimiento, los MCH pueden resultar provechosos en distintas aplicaciones. En tal sentido se continúa trabajando en los siguientes temas:

- Utilización de los MCH como medio para lograr descripciones concisas y efectivas de documentos.
- MCH para diseños curriculares incrementales [Vit99] [Mal00].
- Uso de los MCH para el aprendizaje de lectura comprensiva de textos. Los MCH son un medio para modelar el esquema de relaciones conceptuales que se elaboran con la adquisición de nuevos conocimientos. Bajo esta premisa encontramos valiosa su aplicación en los cursos de Lectura Comprensiva de Textos en Inglés que se dictan para alumnos universitarios. Para ello se trabajó en la elaboración de una metodología específica para la creación de estos mapas en el contexto de la aplicación mencionada y en el diseño de una experiencia educativa programada para poder evaluar los resultados [Vit02]. Se continúa investigando sobre la correspondencia entre patrones de MCH con patrones de texto ya definidos.

#### Bibliografía

- [Aus78] Ausubel, D. P., Novak J. D. "Educational Psychology: A Cognitive View". 2nd Ed. New York: Holt, Rinerhart and Winston. 1978.
- [Baz96] Bazik, j. Tamassia, R., Reiss, S y van Dam, A. Software Visualization in Teaching at Brown University. 1996.
- [Con 94] Contreras, D. "Enseñanza, profesorado y curriculum". Madrid. Akal. 1994
- [Dib99] Di Battista, P. Eades, G. Tamassia, R. y Tollis, I. "Graph Drawing: algorithms for the visualization of graphs", Prentice Hall, 1999.
- [Ead94] Eades, P. and Wormald, N. "Edge crossings in drawings of bipartite graphs". Algorithmica, 11, 1994.
- [Gri95] Grinstein G - Levkowits H. "Perceptual Issues in Visualization", Springer-Verlag, 1995.
- [Jün97] Jünger, M. and Mutzel, P. "2-Layer Straightline Crossing Minimization: Performance of exact and heuristics algorithms". JGAA, 1, n. 1, 1997.
- [Laj93] Lajoie, S. "Computer Environments as Cognitive Tools for Enhancing Learning". 1993. McGill University.
- [Law94] Lawrence, A., Brade, A., Stasko, J., Empirically Evaluating the Use of Animations to Teach Algorithms. Technical Report GIT-GVU-94-07, Graphics, Visualisation, and Usability Center, College of Computing. Georgia Institute of Technology. 1994.
- [Leh93] Lehrer, R. "Authors of knowledge: Patterns of Hypermedia Design". 1993. University of Wisconsin-Madison.
- [Mal00] Malet, A. y Señas, P. "Los MCH como recurso interactivo del currículo". III Jornadas de Investigación Educativa. La Plata. 2000.
- [Mal02] Malet, A. y Señas, P. "Mapas Conceptuales Hipermediales para el diseño curricular". INTERTECH 2002. Brasil.
- [Mar00] Martig, S. y Señas, P. "Herramientas para la construcción de conocimiento en ambientes de aprendizaje abiertos: Construcción y Visualización del Grafo Integrador de un MCH". VI CACIC. Argentina. 2000.

- [Mar01] Martig, S. y Señas, P. "Información contextual en ambientes MCH". VII CACIC. Argentina. 2001.
- [Mar02] Martig, S. y Señas, P. "Grafo de Vistas de un Mapa Conceptual Hipermedial: Foco + Contexto". INTERTECH 2002. Brasil.
- [Mor96] Moroni, N. - Vitturini, M. - Zanconi, M. - Señas, P. "Una plataforma para el desarrollo de mapas conceptuales hipermediales". Taller de Software Educativo - IV Jornadas Chilenas de Computación. Valdivia. 1996.
- [Mor00] Moroni, N. y Señas, P. "Herramienta Computacional para el logro de meta-aprendizajes". VI CACIC. Argentina. 2000.
- [Mut97] Mutzel, P. "An Alternative Method to Crossing Minimization on Hierarchical Graphs".
- [Nov84] Novak, J. y Gowin, D. "Learning how to learn". Cambridge University Press. 1984.
- [Nov85] Novak, J. "Metalearning and metaknowledge strategies to help students learn how to learn. Cognitive Structure and Conceptual Change". New York. Academic Press. 1985.
- [Señ96] Señas, P., Moroni, N., Vitturini, M. y Zanconi, M.: "Hypermedial Conceptual Mapping: A Development Methodology". 13th International Conference on Technology and Education. University of Texas at Arlington, Department of Computer Science and Engineering. New Orleans 1996.
- [Señ98] Señas, P., Moroni. "Herramientas no convencionales para el aprendizaje de la programación". IV CACIC. Argentina. 1998.
- [Señ99] Señas, P. Tesis de Magíster: "MCH como herramienta para la Representación de Conocimiento en Agentes Inteligentes". Universidad Nacional del Sur. 1999.
- [Señ00a] Señas, P., Moroni, N. "Computing Environments for metalearning. Interconnecting Hypermedia Concept Maps". ED-MEDIA 2000. Montreal. Canada. 2000.
- [Señ00b] Señas, P., Moroni, N. "Plataforma para el uso de Mapas Conceptuales para la Representación de Conocimiento". VI CACIC. Argentina. 2000.
- [Sow00] Sowa, J. "Knowledge representation". Brooks Cole. 2000.
- [Sta98] Stasko, J., Domingue, J., Brown, M., Price, B. Software Visualization: Programming as a Multimedia Experience. MIT Press, 1998.
- [Vit99] Vitturini, M. y Zanconi, M. "Mapas Conceptuales Hipermediales para describir un Curriculum". V Conferencia Internacional de Ciencias de la Educación. Cuba. 1999.
- [Vit02] Vitturini, M., Benedetti, L. and Señas, P. "Hypermedia Conceptual Mapping for foreign language learning". Aprobado para su publicación en The 5th IASTED International MultiConference: Computers and Advanced Technology in Education. México. 2002
- [Zan98] Zanconi, M., Moroni, N., Vitturini, M., Malet, A., Borel, C. y Señas, P. Tecnología computacional y meta-aprendizajes. RIBIE-98. 1998.



Universidad Nacional de la Patagonia San Juan Bosco  
Facultad de Ingeniería

## Mapas Conceptuales: una herramienta para el aprendizaje de Estructuras de Datos

C.C. Patricia Ruth Uviña <sup>1</sup>

Ing. Mabel Angélica Bertolami <sup>2</sup>

Ing. María Elena Centeno <sup>3</sup>

A.P.U. Gabriela Carmen Oriana <sup>4</sup>

Departamento de Informática

Facultad de Ingeniería

Universidad Nacional de la Patagonia "San Juan Bosco"

### Resumen

La función de los Mapas Conceptuales(MC) consiste en ayudar a la comprensión de los conocimientos, que el alumno tiene que aprender, y a relacionarlos entre sí o con otros que ya posee. Los MC están dentro de las estrategias que pretenden organizar los nuevos conocimientos. Se apoyan en el criterio de la Jerarquización, que es análogo a la técnica de Refinamientos Sucesivos, aplicada a la Programación Estructurada. La utilización de MC fomenta el pensamiento reflexivo, la creatividad y el espíritu crítico, conductas imprescindibles en la formación y el desenvolvimiento profesional. Este proyecto consiste en desarrollar una herramienta de enseñanza-aprendizaje, para utilizar distintos MC.

### Introducción y Presentación

La función de los Mapas Conceptuales(MC) consiste en ayudar a la comprensión de los conocimientos, que el alumno tiene que aprender, y a relacionarlos entre sí o con otros que ya posee. Los MC están dentro de las estrategias que pretenden organizar los nuevos conocimientos. Se apoyan en el criterio de la Jerarquización, que es análogo a la técnica de Refinamientos Sucesivos, aplicada a la Programación Estructurada. La utilización de MC fomenta el pensamiento reflexivo, la creatividad y el espíritu crítico, conductas imprescindibles en la formación y el desenvolvimiento profesional.

Este proyecto consiste en desarrollar una herramienta de enseñanza-aprendizaje, para utilizar distintos MC. Los mismos permitirán representar en diversos niveles de estudio las Estructuras de Datos, lo que favorecerá al alumno la profundización gradual de sus propios juicios.

Como consecuencia de la aplicación de este producto, se pretende

- Mejorar la producción de los alumnos de Primero y Segundo año, de las carreras de **Analista Programador Universitario** y **Licenciatura en Informática**

---

<sup>1</sup> patricia@unpbib.edu.ar

<sup>2</sup> mbertolami@gmx.net

<sup>3</sup> malenac@sinectis.com.ar

<sup>4</sup> orianag@arnet.com.ar



*Universidad Nacional de la Patagonia San Juan Bosco*  
**Facultad de Ingeniería**

- Favorecer la auténtica reorganización cognoscitiva del alumno, en el dominio de las estructuras de Datos.
- Brindar los elementos conceptuales que vinculen las asignaturas *Algorítmica y Programación y Estructuras de Datos y Algoritmos*.

Para ello se debe:

- Desarrollar un software de aplicación, para la representación y el manejo de mapas conceptuales, en forma integrada, aplicando el modelo de hipermmedia, en el que se combinen gráficos, textos, imágenes, animaciones y sonidos.
- Integrar la herramienta educativa **Mapas Conceptuales** como recurso pedagógico para la enseñanza de distintos contenidos en *Algorítmica y Programación y Estructuras de Datos y Algoritmos*
- Integrar un equipo de trabajo capaz de transmitir a los estudiantes los conocimientos y experiencias adquiridas durante el desarrollo del Proyecto.

La metodología que utilizaremos requiere :

- Capacitación en el uso de los Mapas Conceptuales, a partir de una investigación bibliográfica y del asesoramiento de un profesional en Ciencias de la Educación.
- Aplicar los MC para representar las Estructuras de Datos en sus distintos niveles de estudio.
- Realizar las etapas de Análisis, Diseño y Codificación del Software que refleje el modelo en estudio.
- Intensificar el uso de la plataforma de desarrollo DELPHI, para la incorporación de sonidos, la construcción de animaciones y la simulación del comportamiento de las Estructuras de Datos

Los recursos necesarios para estos pasos son:

- Capacitación de los integrantes del Proyecto
- Bibliografía
- Herramientas de Software
- Equipamiento para desarrollo (disponible en el Departamento de Informática)

### **Breve descripción de avance del Proyecto**

Dentro de la metodología de trabajo propuesta se hicieron avances en el proyecto que significan cierto adelanto respecto de cronogramas establecidos.

Se han desarrollado un conjunto de mapas conceptuales contando con la asistencia de una especialista en el tema, a quien se le pidió que los evaluara, de esta manera se efectuaron correcciones a los mismos.

Dado que la idea es explotar los conceptos en un cuadro de texto y/o mediante un hipervínculo, acceder al algoritmo de las operaciones, se ha generado un “Diccionario de Conceptos” en el cual se especifica en qué mapa se utiliza cada concepto. Hay algunos que se utilizan en casi todos los mapas, como p.e. *colección*, y hay otros que sólo se utilizan en un único mapa, como p.e. *lista de adyacencia*.



Universidad Nacional de la Patagonia San Juan Bosco  
Facultad de Ingeniería

Los próximos pasos se encaminan hacia el desarrollo del software.

### Bibliografía

- Guarro, A. "Un modelo de análisis y representación de la estructura del contenido". Enseñanza (1988). Anuario interuniversitario de Didáctica, Num. 3.(237-267).
- Novak, J. D. y Gowin, D. B.: "Aprendiendo a aprender".(1988) Ed. Martínez Roca. Barcelona .
- Ontoria, A. y otros.: "Mapas Conceptuales: una técnica para aprender".(1992) Ed.Narcea.
- Ontoria, A. y otros.: "Los mapas conceptuales en el aula".(1996) Ed.Lumen.
- Charte, F. : "Programación con Delphi 4" (1998) Ed. Anaya
- Charte, F. : "Delphi 4 Guía Práctica" (1998) Ed. Anaya
- Boggino, N: "Cómo elaborar mapas conceptuales en la escuela" (2000) Ed. Homo Sapiens
- D.E. Knuth : "Algoritmos Fundamentales. Volumen I"(1980) Reverté
- M.E.Loomis: "Estructura de Datos y Organización de Archivos"(1991)Prentice Hall
- A. Tenenbaum, M. Augenstein "Estructura de Datos en Pascal"(1985) Prentice Hall
- N.Ziviani: "Algoritmos e Estructura de Datos" (1986) E.B.A.I.
- D.E. Knuth : "Sorting and Searching.D.E."(1973) Ed. Reverté
- N. Wirth: "Algoritmos + Estructura de Datos = Programas"(1986) Prentice Hall
- A.V.Aho, J.Ullman, J.E.Hopcroft: "Data Structures and Algorithms"(1988) ADDISON WESLEY
- M.A. Weiss: "Estructuras de datos y Algoritmos"(1995) ADDISON WESLEY
- R.Pressman: "Ingeniería del Software. Un enfoque práctico"(1993) Mc Graw Hill
- N. Dale, S.Lilly: "Pascal y Estructuras de Datos" (1991) Mc Graw Hill
- Welsh-Elder: "Pascal Introducción"(1984) Prentice Hall
- Jorge Boria: "Ingeniería de Software" (1986) Kapeluz
- <http://www.geocities.com/Athens/Olympus/3232/>
- <http://www.conceptmaps.it/default-esp.htm>
- Existe en Internet gran cantidad de información en línea, y referencias o documentos sobre mapas conceptuales y temas relacionados, entre otros, [http://starbuck.ced.appstate.edu/rc/math/k4m\\_connect.htm](http://starbuck.ced.appstate.edu/rc/math/k4m_connect.htm) donde se hallan ejemplos de mapas hechos por niños; <http://www.skemp.com>, donde se continúa el trabajo de Richard Skemp (fallecido en 1995) en el programa SAIL, donde juegan un papel determinante los mapas conceptuales; <http://trochim.human.cornell.edu/kb/conmap.htm>, con aplicaciones de los mapas conceptuales a la evaluación de proyectos; <http://www.gold.net/Buzan>; en relación a un tema afín, el de los mapas mentales y [http://www.to.utwente.nl/user/ism/lanzing/cm\\_bibli.htm](http://www.to.utwente.nl/user/ism/lanzing/cm_bibli.htm), con bibliografía adicional sobre el tema.

# MAPAS CONCEPTUALES HIPERMEDIALES: SU APLICACIÓN EN CURSOS DE LECTURA COMPRENSIVA

Mercedes Vitturini - Laura Benedetti - Perla Señas

Laboratorio de Investigación y Desarrollo en Informática y Educación (LIDInE)

Instituto de Investigación en Ciencias y Tecnología Informática (IICyTI)

Departamento de Ciencias de la Computación

Universidad Nacional del Sur – Bahía Blanca - Argentina

[mvitturi@cs.uns.edu.ar] [benedett@criba.edu.ar] [psenas@cs.uns.edu.ar]

Palabras Claves: MCH – Informática Educativa – Comprensión de Textos

## 1. Resumen

Los Mapas Conceptuales Hipermediales (MCH) son un medio para modelar el esquema de relaciones conceptuales que se elaboran con la adquisición de nuevos conocimientos. Bajo esta premisa encontramos valiosa su aplicación en los cursos de Lectura Comprensiva de Textos en Inglés que se dictan para alumnos universitarios. A partir del trabajo interdisciplinario entre docentes del área de Informática Educativa y docentes del área de Idiomas Extranjeros se abrió una nueva línea de investigación en el LIDInE. Se trabajó en la elaboración de una metodología específica para la creación de MCH en el contexto de la aplicación mencionada y en el diseño de una experiencia educativa programada que permitió evaluar los resultados. Actualmente se está trabajando en la definición de patrones de mapas que se correspondan con los patrones de textos existentes.

## 2. Introducción

Los MCH, basados en los Mapas Conceptuales presentados por Novak [10], constituyen una valiosa herramienta que favorece los procesos de aprendizaje significativo. En particular los MCH con su componente hipermedial, profundizan la tarea de selección y jerarquización de conceptos [17]. El uso de los MCH es válido en la enseñanza de cualquier tema y en particular en los cursos de Lectura Comprensiva de Textos en Idioma Extranjero. En esta disciplina, se han podido insertar naturalmente en el proceso de enseñanza a lo largo de todo el curso incluyendo la evaluación final.

La elaboración de un MCH sobre un tema no es una tarea trivial y requiere de una clara comprensión del mismo. Las experiencias en el aula muestran que los alumnos en general no tienen problemas con el aprendizaje de la técnica sino en lograr MCH's semánticamente correctos [20]. Las dificultades en la construcción del mapa dan nota de que todavía no se ha logrado una comprensión del tema bajo estudio. Si para un texto dado, no se pueden determinar los conceptos fundamentales, jerarquizarlos y plantear las proposiciones que los vinculen, básicamente significa que no se ha comprendido el tema, esto es, no se ha logrado un aprendizaje significativo. Todas estas decisiones que un alumno tiene que tomar para construir un MCH lo llevan a analizar el texto más profundamente. El aporte de los MCH en la tarea del docente también es importante, el mapa que pueda hacer un alumno sobre un texto da una idea clara de la comprensión que ha logrado sobre el mismo.

## 3. Mapas Conceptuales Hipermediales

Un MCH es un Mapa Conceptual enriquecido por el aporte de tecnología hipermedial. Los elementos básicos que componen los MCH son: conceptos, relaciones vinculares y vistas. Un concepto es “una regularidad en los acontecimientos o en los objetos que se designa mediante algún

término”. Las relaciones quedan determinadas por palabras enlace que unen dos conceptos formando una unidad con significado semántico que tiene valor de verdad. Estas unidades semánticas se denominan proposiciones. [Ont92]

Los MCH heredan los constructores propios de los Mapas Conceptuales e incorporan otros nuevos necesarios para mantener la estructura hipermedial. En la tabla 1 se muestra la representación gráfica de cada uno de ellos [9].

CONCEPTOS		RELACIONES		VISTAS
Terminal		Internas		
No Terminal		Externas		
Externo				

Tabla 1: Representación Gráfica de los Constructores del MCH

#### 4. Diseño de la Experiencia

Para comprobar y medir resultados se llevó adelante una experiencia interdisciplinaria en el que participaron docentes-investigadores de los Departamentos de Ciencias e Ingeniería de la Computación y de Humanidades. El plan de trabajo consistió en elaborar una metodología que incluyera la utilización de MCH. Dicha metodología se aplicó durante el desarrollo curricular y evaluación de la materia “Curso de Lectura Comprensiva de Textos en Inglés – Segundo Nivel”. Se trata de un curso que dura un cuatrimestre, no pertenece al currículum de una carrera y se ofrece como servicio a los alumnos en general. Los estudiantes que toman estos cursos son de diferentes carreras y niveles.

En un curso de lectura comprensiva, en este caso particular de idioma Inglés, el objetivo es que los estudiantes desarrollen habilidades para la comprensión de textos en idioma extranjero aún sin tener un dominio fluido del mismo [12]. Las producciones (resúmenes, lecturas y debates) son en la lengua madre. Los textos seleccionados son del tipo de los textos académicos de estudio. Como meta más general lo que se busca con este tipo de cursos, es enseñar a los alumnos técnicas de estudio que puedan aplicar a cualquier tema, independientemente del idioma en que se presenten. Bajo estos objetivos, encontramos que los MCH y la plataforma para su creación, refinamiento y lectura, se podían incluir naturalmente en el desarrollo del curso.

Para el desarrollo de la experiencia se siguieron los siguientes pasos:

- Capacitación de los docentes del área de Idiomas Extranjeros con la técnica MCH.
- Desarrollo conjunto de la metodología.

- Capacitación de los alumnos del curso sobre la plataforma y la metodología de elaboración de MCH.
- Realización de lecturas comprensivas usando MCH.
- Evaluación de los alumnos.
- Evaluación de los resultados.

#### 4.1 Metodología para la Construcción de MCH

Se diseñó una metodología apta para guiar a los estudiantes en la construcción del mapa y que además permite a los docentes evaluar la comprensión del texto por parte de los alumnos. Un punto importante a tener en cuenta es que se debieron respetar las técnicas que se venían aplicando en el dictado del curso, como por ejemplo reconocimiento de patrones. El conjunto de pasos propuestos incluye:

- 1) Tareas a realizar antes de la lectura:
  - a) Plantear el propósito de la lectura: en este caso lograr la comprensión del texto completo.
  - b) Hacer predicción: esta técnica se utiliza en la mayoría de los ejercicios de lectura, se trata de iniciar la lectura con algunas ideas en mente. Resulta de mucha utilidad para la comprensión.
- 2) Tareas a realizar durante la lectura:
  - a) Seleccionar las ideas principales de cada párrafo.
  - b) Reconocer patrones.
  - c) Inferir el vocabulario no conocido.
- 3) Tareas a realizar después de la lectura:
  - a) Crear un MCH que refleje una comprensión acabada del texto.
    - i. A partir de las ideas principales, seleccionar los conceptos más relevantes que se desarrollan en el texto.
    - ii. A través del uso de la plataforma definir una vista auxiliar con todos los conceptos elegidos.
    - iii. Estimar el número de vistas a usar. La cantidad de conceptos definidos dará una idea general del número de vistas necesarias. Como regla práctica se aconseja entre  $7 \pm 2$  conceptos por vista.
    - iv. Jerarquizar los conceptos y clasificarlos en conceptos botones no terminales y conceptos terminales.
    - v. Distribuir los conceptos en las distintas vistas definidas.
    - vi. Comenzar la construcción del MCH con los conceptos y según las decisiones tomadas.
    - vii. Establecer las relaciones entre los conceptos de una misma vista.
    - viii. Mapear patrones de texto en patrones de mapas.
    - ix. Analizar las posibles relaciones cruzadas. Una vez determinadas, si existen, importar los conceptos necesarios en forma de conceptos botones externos.
  - b) Refinar el mapa con el texto.

#### 4.2 Evaluación de los MCH

Se realizó una evaluación de producto terminado; se lo hizo desde las perspectivas sintáctica y semántica y se cuantificaron los siguientes tópicos:

- Número de vistas
- Número de conceptos
- Correctitud de relaciones internas
- Existencia de relaciones externas
- Correctitud de las proposiciones que se obtienen a partir del mapa

## 5. Conclusiones

De la experiencia sobre el uso de MCH en cursos de lectura comprensiva pudimos comprobar que no se presentan dificultades en la comprensión de la metodología, sino que resulta motivadora, clarificadora y complementaria de las técnicas que se venían empleando en el curso, tanto para los alumnos, como para los docentes.

También se obtuvieron resultados que aportan a la investigación. Los errores de construcción más comunes que se pudieron detectar, motivo a diseñar para una nueva versión de la plataforma, herramientas de testeo que permitan al autor validar su mapa. Actualmente se está trabajando en este sentido con relación a la herramienta. Por otra parte, resultó muy enriquecedor la definición e inclusión de los pasos previos a la construcción del mapa y la definición de algunos patrones de MCH para los patrones de textos existentes, se pudo verificar el aporte de los mismos en la construcción. La información estadística permite afirmar que los resultados finales fueron superiores a los logrados en experiencias anteriores, donde se había trabajado sin la metodología propuesta en 4.1.

## 6. Bibliografía

- [1] D. P. Ausubel, J. D. Novak. *Educational Psychology: A Cognitive View*. 2<sup>nd</sup> Ed. New York: Holt, Rinerhart and Winston. 1978.
- [2] Forman, D., Donoghue, F., Abbey, S., Cruden, B. and Kidd, I. *Campus English A study skills course for university students*. MacMillan Publishers. 1990
- [3] Garzzoto, Mainetti and Paolini: *Hypermedia Design, Analysis and Evaluation Issues*. Communications ACM. August 1995.
- [4] Isakowitz, T. , Stohr, E. and Balasubramanian, P. *RMM: A methodology for Structured Hypermedia Design*. Communications ACM. August 1995.
- [5] Malet, A. y Señas P. *Los Mapas Conceptuales Hipermediales y la construcción de conocimiento*. V Conferencia Internacional de Ciencias de la Educación. Cuba. 1999.
- [6] Marchionini, Gary - Crane, Gregory. *Evaluating Hypermedia and Learning: Methods and Results from the Perseus Project*. ACM Transactions on Information Systems. Vol 12. N1. Enero 1994, pp 5-34.
- [7] Martig, S. Y Señas, P. *Herramientas para la construcción de conocimiento en ambientes de aprendizaje abiertos: Construcción y Visualización del Grafo Integrador de un MCH*. CACICC-2000. Argentina.2000.Sergio R. Perla Señas
- [8] Meeting of the National Association for Research in Science Teaching. Atlanta. USA. 1993.
- [9] Moroni, N. - Vitturini, M. - Zanconi, M. - Señas, P. *Una Plataforma para el desarrollo de mapas conceptuales hipermediales*. Taller de Software Educativo - IV Jornadas Chilenas de Computación. Valdivia. 1996.
- [10] Novak, J. And Gowin, D. *Learning how to learn*. New York. Cambridge University Press. 1984.
- [11] Novak, J. *Metalearning and metaknowledge strategies to help students learn how to learn. Cognitive Structure and Conceptual Change*. New York. Academic Press.1985.
- [12] Nuttall, C. "Teaching Reading Skills in a Foreign language" Heinemann. New Edition. 1996.
- [13] A. Ontoria. *Mapas Conceptuales: Una técnica para Aprender*. Narcea S.A. de Ediciones. Madrid. 1992.
- [14] Perkins, David. *La escuela inteligente*. Barcelona. Gedisa. 1995.
- [15] Sanchez J. *Concept mapping and educational software production*. 66<sup>th</sup>. Annual

- [16] Sanchez J. *Metalearning and metaknowledge strategies to produce educational software*. Amsterdam Elsevier Science Publishers B. V. 1993.
- [17] Señas, P., Moroni, N., Vitturini, M. y Zanconi, M.: *Hypermedial Conceptual Mapping: A Development Methodology*. 13th International Conference on Technology and Education. University of Texas at Arlington, Department of Computer Science and Engineering. New Orleans 1996.
- [18] Señas, P. y Moroni, M. *Computing Environments For Metalearning: Interconnecting Hypermedia Concept Maps*. ED-MEDIA2000. Canada. 2000.
- [19] Wallace, C. *Reading* Oxford University Press. 1992.
- [20] Zanconi, M., Moroni, N., Vitturini, M., Malet, A., Borel, C. y Señas, P. *Tecnología computacional y meta-aprendizajes*. RIBIE-98. Brasil. 1998.

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

## WICC 2002

*16 y 17 de mayo de 2002*

### Ingeniería de Software y Base de Datos



## INCA RUCA QUIMN<sup>1</sup>: Proyecto de automatización de oficinas

Acosta Nelson, Simonelli Daniel y Tosini Marcelo  
INCA / INTIA – Departamento de Computación y Sistemas  
Facultad de Ciencias Exactas – UNCPBA – TANDIL  
Campus Universitario – Paraje Arroyo Seco s/n  
TE: +54 - 2293 - 432466 – FAX: +54 - 2293 - 444431  
Email: nacosta@exa.unicen.edu.ar

### Resumen

Este artículo describe una aplicación de automatización de edificios del grupo INCA/INTIA, en el marco del proyecto de investigación ALIASEMEH (ALgoritmos de Inteligencia Artificial para Sistemas Empotrados: METodologías y Herramientas) financiado por la UNCPBA.

### 1) INTRODUCCIÓN

Este proyecto presenta una primera aproximación a la automatización de edificios llevada a cabo en las instalaciones del grupo de investigación INCA. Las instalaciones edilicias del grupo cumplen con los requisitos mínimos y además proveen ciertas ventajas: a) factibilidad de cableado sin perjuicios estéticos ni funcionales; b) rápida implementación; c) reducción de costos e incremento de confort para el grupo; d) optimización del gasto de energía.

Dados los condicionamientos económicos que presenta el país en este momento y a las facilidades tecnológicas disponibles se pretende lograr, por lo menos, los siguientes tres objetivos principales: a) afianzar el funcionamiento del grupo a través de la materialización de un proyecto ambicioso, técnicamente complejo y que, a la vez, ofrece un gran potencial industrial y comercial; b) obtener experiencia en proyectos insertos en entornos comerciales o industriales concretos para encarar los cada vez más crecientes requerimientos del medio; c) transferir resultados obtenidos en el proyecto institucional ALIASEMEH (ALgoritmos de Inteligencia Artificial para Sistemas Empotrados: METodologías y Herramientas).

### 2) GRANDES BLOQUES DEL SISTEMA

#### 2.1 – Central de mando a periféricos (Remote-X).

Remote-X provee un soporte adaptable para construir, en forma sencilla, aplicaciones especializadas, orientadas a la realización de actividades sobre dispositivos electrónicos remotos. Para permitir que estas aplicaciones puedan ser implementadas en distintos lenguajes de alto nivel, la herramienta provee un componente ActiveX. La funcionalidad de la herramienta se puede resumir en las siguientes características: a) adaptación y flexibilidad en la construcción de aplicaciones para realizar actividades sobre dispositivos remotos; b) atención concurrente, eficiente y priorizada de las aplicaciones; c) administración centralizada y eficaz de los dispositivos; d) monitoreo de la actividad general que lleva a cabo la herramienta; e) control y manipulación de los dispositivos remotos; f) detección y reporte de las posibles fallas.

Remote-X distribuye la funcionalidad en el conjunto de componentes siguiente: a) *RxRemoteBoard*: interactúa directamente con los dispositivos electrónicos, tiene la capacidad para obtener información, responder a los requerimientos de sus dispositivos, controlar los valores y realizar acciones sobre dichos dispositivos electrónicos; b) *RxClient*: brinda facilidad y flexibilidad a los programadores para desarrollar aplicaciones; c) *RxServer*: núcleo de la herramienta, administra la información que envían las aplicaciones creadas con un *RxClient* y manipula los puertos series para

---

<sup>1</sup> Ruca Quimn (Mapuche) : casa inteligente.

establecer la comunicación con los componentes *RxRemoteBoard*; d) *RxSupervisorClient*: permite ver el estado general en que se encuentra la herramienta.

## **2.2 - WERKEN KELLU<sup>2</sup> (Mensajero Ayudante) – Cadete electrónico.**

Vehículo autoguiado de uso interno para enviar recados entre oficinas: cadete electrónico. Se requiere su servicio a través de la web, usando placas por radio el robot está en la red. Trabaja por conocimiento del escenario, pero igualmente necesita análisis de imágenes para adaptarse a los cambios. El navegante determina los caminos alternativos para llegar a destino con la información del Remote-X.

Principales características: a) cinco sensores de distancia, tres delante y dos en la parte trasera; b) sensores de choque alrededor; c) una cámara digital para capturar imágenes del camino; d) codificadores angulares en el eje tractor para determinar la posición; e) un servo eléctrico para controlar la rueda de dirección; f) un paquete de 4 baterías que brindan una autonomía de 4 horas; g) medidor de estado de baterías; h) placa de red por radio; i) procesador Pentium 100 con 20 MBytes de RAM; j) Disco rígido de 2 GBytes; k) Salida de televisión para ver el estado de la consola; l) tres ruedas, dos tractoras traseras y la directriz adelante; m) 50 ó 75 cm de altura y 50 cm de diámetro.

## **2.3 - INCA PUNKU<sup>3</sup> (Puerta del INCA) – Portero visor.**

Portero visor por la web. Diseño e implementación de una arquitectura basada en la Web con las siguientes características: a) realización de comunicaciones audiovisuales entre computadoras con diferentes ubicaciones físicas; b) soporte multiplataforma; c) acceso a través de un navegador estándar (Netscape, Mozilla, Opera, IExplorer, etc.); d) manejo de dispositivos remotos por medio de computadoras cliente situadas en cualquier lugar del mundo.

El sistema administra una o más consolas externas a través de las cuales los visitantes pueden localizar y contactar al personal de grupo INCA y mantener una conferencia audiovisual. Además, los usuarios internos pueden, no sólo atender requerimientos externos sino, también, realizar video comunicaciones entre miembros del grupo y manejar dispositivos remotos (apertura de puertas de acceso).

## **2.4 - PIN LIGHUEN<sup>2</sup> (Mandar luminosidad) – Control de luces y persianas.**

Se debe tener control manual y automático tanto de todas las luces como de las persianas del edificio. La motorización puede ser artesanal automatizando los rodillos actuales o utilizando rodillos eléctricos. En caso que manualmente se cambie el estado, se envía dicha información al sistema de control general. Las luces se controlan teniendo en cuenta: a) la entrada y salida de los integrantes; b) los parámetros definidos en la configuración; c) las condiciones lumínicas de la oficina y estación del año; d) la programación del encendido o apagado por horario o por evento de sensores (la intensidad de luz es más baja que determinado rango); e) apagado automático al detectar la ausencia de personas; f) simulación de presencia al estar el edificio solo (también controla equipos de audio, televisión, sonidos sintetizados, etc.).

Para el control de las persianas se tienen en cuenta también las condiciones climáticas y la temperatura de la oficina.

## **2.6 - INCA PEUTU<sup>4</sup> (Vigilante de INCA) – Seguridad y alarma.**

Se activa por teclado o automáticamente cuando se retira la última persona del edificio. Cada perfil puede tener acceso sólo a un área, en caso de salirse del área permitida puede ser una condición de alarma. En caso de detección de ingresos no autorizados, primero los verifica para luego: a) genera

---

<sup>2</sup> Mapuche.

<sup>3</sup> Inca.

<sup>4</sup> Mapuche.

señales de presencia a la alarma general; b) aviso 'mudo' a una lista de teléfonos por defecto; c) captura imágenes de los intrusos y las guarda en un servidor externo por la red y las manda por email a una lista de direcciones.

### **2.7 - LULUN ARE<sup>4</sup> (Frío Calor) – Aire acondicionado y calefacción.**

Se tiene control manual y/o automático de la temperatura de cada oficina del edificio. Las preferencias de cada oficina son especificadas por medio de reglas. Los parámetros de entrada al sistema son: a) la entrada y salida del personal; b) la agenda particular de las personas; c) los parámetros definidos en la configuración por defecto; d) los sensores de temperatura; e) por las condiciones climáticas externas.

### **2.8 - SUGUN ALDU PU<sup>4</sup> (Hablar a otro lejos) – Telefonista electrónica.**

Una computadora dedicada a la atención de todas las llamadas telefónicas externas: “una telefonista electrónica”. Las principales características son: a) De acuerdo a un menú se direcciona a la oficina donde esté el destinatario (no siempre a la por defecto); b) Si no está en ese momento debe avisar que no está y opcionalmente tomar el mensaje; c) Permite consultar por TE estados del sistema (temperatura, cuantas personas hay, donde estan, ...) por medio de voz o modem; d) Hacer llamadas por condiciones especificadas por reglas (alarmas, sensores, etc.) por sonido o PC; e) Registro completo de llamadas de todos a todos lados, con todos los datos; f) Permite derivar llamadas telefónicas a partir del número del llamador (necesita el caller ID) y de quienes están en el edificio (especificado por reglas); g) Permite controlar (accesos, luces, persianas, ...) o monitorear alguna información por teléfono externo (quienes están en el edificio, estado de luces, ...); h) Llamadas programadas en el tiempo (los lunes a las 8 llama a alguien) o por eventos (se abre 1 puerta y me avisa a tal teléfono).

### **2.9 –INAPUNON CHE<sup>4</sup> (Seguir el rastro a personas) – Seguimiento de personas.**

Sistema de seguimiento e identificación de personas dentro de las instalaciones. Se hace una identificación visual diaria (login) donde se identifican parámetros basándose en la composición de colores y algunas otras características. Utilizando cámaras se realiza una búsqueda, seguimiento e identificación de las personas de acuerdo a la mayor probabilidad. Se aplican técnicas de análisis de imágenes para obtener los parámetros, y con esos parámetros se consulta una base de datos aplicando lógica difusa para determinar quien es dicha persona.

## **3) TEMAS INVOLUCRADOS EN EL PROYECTO**

Las principales áreas involucradas en el proyecto son las siguientes: a) protocolo X10, aparatos, moduladores y (de)moduladores, condiciones de uso, factores legales; b) placas de entrada-salida digital y analógica; c) comunicación inalámbrica; d) circuitos para el control de relés, (opto)acopladores, motores, servos, cerradura electrónica; e) circuitos de: sensores de presencia, llaves (con tratamiento de rebotes), sensores de temperatura-luminosidad, resistencias variables; f) manejo de teclados externos membranas, teclas (mecánicas, capacitivas, ópticas) o pantallas sensibles; g) análisis de imágenes; h) telefonía por computadora; i) desarrollo con Linux y TCL/TK; j) comunicación por web (imagen y sonido); k) microcontroladores (pic, hc08, hc11, 8051).

## **4) CONCLUSIONES**

Los módulos Remote-X e INCA PUNKU, únicos módulos sin nombre Mapuche, han sido instalados y están funcionando desde principios y fines de 2001 respectivamente.

- **Remote-X** tiene como responsabilidad el acceso a todos los dispositivos y controladores remotos vía la web; se utiliza para medir el estado de las persianas, ventanas y ángulo de apertura de las puertas. También se utilizará para el control de las luces.

- **INCA PUNKU**, es el video portero que actualmente se está utilizando en el grupo para la atención de los visitantes. El servidor y las aplicaciones se realizaron en java, el control de dispositivos en Linux, teclado óptico interno que captura los eventos del exterior, video a 9 fps y audio a 44100bps sobre una red de 100Mbps, y mantiene un log-file con las fotos de los visitantes. Actualmente no funciona en Linux porque la librería de multimedia de java no está terminada.
- El cadete electrónico **WERKEN KELLU** está en fase de prueba de los sistemas electrónicos y mecánicos. El sistema de control está completo, actualmente se está trabajando en el navegador.

Los módulos restantes, PIN LIGHUEN<sup>5</sup>, INCA PEUTU<sup>6</sup>, LULUN ARE<sup>7</sup>, SUGUN ALDU PU<sup>8</sup> e INAPUNON CHE<sup>9</sup> están actualmente en fase de diseño. Se estima que en dos años estarán operativos.

### ALGUNAS REFERENCIAS

1. Amor, M., "MultiTEL, Arquitectura para la coordinación de aplicaciones distribuidas en la Web", Proyecto fin de carrera, Univ. De Málaga, julio 1998,
2. Berson, Alex. "Client/Server Architecture", 2da. Ed. New York, McGraw-Hill, 1996, 71-154.
3. Dorsey T., "Cu-SeeMe Desktop Videoconferencing Software", Connections, Volume 9, No. 3, march 1995.
4. Fredrick M. Cady, "Software and Hardware Engineering Motorola M68HC11", New York, OXFORD UNIVERSITY PRESS, 1997.
5. Heller, Martin. "Tips and Tricks on Developing Killer Server Applications for Windows NT", Microsoft Systems Journal 10 (August 1995). (MSDN Library, Periodicals).
6. Jong Jin Kim, Jin Kim Jong, "Intelligent Buildings", 1998, Butterworth-Heinemann Publisher.
7. Olsina L., Echeverría E., Ballarini T., Molina H., Wainerman E., "Arquitectura Centrada en la Web para el control y Monitoreo de Funcionalidad Domótica", CACIC 2000, Calafate, Argentina.
8. Quinteros González J., Lamas Graziani J., Sandoval J., "Sistema de Control para Viviendas y Edificios: Domótica", 1999, Ed. Paraninfo, Madrid España.
9. Razquin Mauricio, Rodríguez Maximiliano, "Remote-X, Prototipo de una Herramienta para el Control de Dispositivos Electrónicos Remotos", INCA, UNCPBA, 2001.
10. Robert Orfali, Dan Harkey, Jeri Edwards. "Client/Server Survival Guide". 3ra. Edición. New York, John Wiley & Sons Inc, 1999, 2-34, 80-122, 157-177.
11. Vert, John. "Writing Scalable Applications for Windows NT" June 1995. (MSDN Library, Technical Articles).

<sup>5</sup> Control de luminosidad (Mapuche): Control de luces y persianas.

<sup>6</sup> Vigilante del INCA (Mapuche): Sistema de alarma.

<sup>7</sup> Frio/Calor (Mapuche): Control de aire acondicionado y calefacción..

<sup>8</sup> Hablar a otro lejos (Mapuche): Telefonista electrónica.

<sup>9</sup> Seguir el rastro a personas (Mapuche): Identificación y seguimiento de personas.

## Interoperabilidad de Componentes Software mediante Lenguajes de Coordinación\*

Silvia Amaro, Nadina Martinez Carod  
*Departamento de Informática y Estadística, Universidad Nacional del Comahue,  
Buenos Aires 1400, Neuquén, Argentina  
TE: (0299) 4490312 - Fax:(0299) 4490313  
Email: samaro, namartin@uncoma.edu.ar*

Ernesto Pimentel  
*Departamento de Lenguajes y Ciencias de la Computación, Universidad de Málaga,  
Campus Universitario, Málaga, España  
Email: ernesto@lcc.uma.es*

**Resumen** El desarrollo de software basado en componentes es una disciplina en continuo crecimiento dentro del campo de la Ingeniería de software. Con las primeras plataformas de componentes del mercado surgió el desarrollo de los lenguajes de descripción de interfaces con la idea de detectar problemas de compatibilidad entre componentes en el desarrollo de aplicaciones distribuidas y abiertas. Sin embargo, al describir los servicios que ofrecen los objetos, los lenguajes de descripción de interfaces sólo consideran la signatura de los métodos, obviando el problema principal de la composición de componentes respecto al protocolo de interacción que les permite comunicarse y cooperar. Nuestro trabajo actual se enfoca en el análisis del uso de Manifold, un modelo de coordinación orientado a control, como base para un formalismo en cuyo contexto definiremos relaciones que permitan analizar las propiedades para que el comportamiento interactivo de componentes software sea seguro.

### 1. Introducción

Para responder a la creciente demanda de nuevas tecnologías para la construcción de aplicaciones abiertas y distribuidas, la Ingeniería de Software Basada en Componentes (CBSE) propone la creación de colecciones de componentes de software reutilizables que puedan ser adaptadas e interconectadas en forma dinámica en el desarrollo de nuevas aplicaciones. El reuso de componentes permite adaptar aplicaciones cambiando componentes existentes o introduciendo nuevas. Para que la búsqueda y recuperación de componentes para su ensamblado en la creación de nuevas aplicaciones tenga éxito es necesario verificar que el comportamiento de las componentes es el apropiado para que puedan interoperar, es decir “comunicarse y cooperar a pesar de las diferencias en los lenguajes de implementación, el ambiente de ejecución o la abstracción del modelo” [15].

Las plataformas de componentes actuales, como CORBA, EJB, COM, .NET facilitan el desarrollo de aplicaciones concurrentes y abiertas. Mediante los lenguajes de descripción de interfaces (IDL) que proveen, se generan descripciones estáticas de los servicios ofrecidos por las componentes. Sin embargo, los únicos mecanismos que pueden ofrecer para determinar la conveniencia o no de integrar una componente ya existente en un contexto determinado es la comprobación de que el nombre de los servicios requeridos por parte del sistema coincide con los que son proporcionados por la componente localizada, pero nada se puede decir de la forma en que esos servicios deben ser utilizados.

Actualmente existen varias propuestas de extensiones a los IDL tradicionales. Se han utilizado álgebras de procesos para describir los protocolos de interacción y se han definido formalismos que establecen relaciones para medir el nivel de compatibilidad entre componentes, y como consecuencia la posibilidad de reemplazar unas componentes por otras [7, 10, 13]. Otros trabajos [11] dan una vista

---

\* Este trabajo es financiado parcialmente por CYTED (Ciencia y Tecnología para el desarrollo), proyecto VII\_J\_RITOS2 (Red Iberoamericana de Tecnologías de Software para la década del 2000)

mas práctica presentando la aplicación de álgebras de procesos a plataformas de componentes ampliamente difundidas. La próxima sección introduce brevemente los modelos de coordinación y sus posibilidades en la interoperabilidad de componentes. Finalmente se dan algunas conclusiones

## **2. Modelos de coordinación para la interacción de componentes software**

Uno de los principales objetivos a lograr al trabajar con componentes software heterogéneas es conseguir una separación clara entre los aspectos de interacción y de computación, para así favorecer la reusabilidad de componentes, facilitando el análisis global de la aplicación [2]. En este sentido, la alternativa ofrecida por los lenguajes y modelos de coordinación (Linda [12] y Manifold [1], por ejemplo) resulta prometedora.

### **2.1 Lenguaje de coordinación Linda**

Linda es un modelo de coordinación en el que el sistema está formado por agentes que desarrollan sus actividades de forma independiente y cuya comunicación se logra de alguna manera asincrónica a través de un medio compartido, referido como espacio de tuplas. Su modelo de comunicación posee una gran potencia expresiva para especificar sistemas distribuidos y concurrentes [9].

Actualmente se está explorando la aplicación de Linda en el terreno de la interoperabilidad de componentes software. En [8] se propone Linda para describir el comportamiento interactivo de componentes software, junto con una relación de compatibilidad -definida a partir de un álgebra de procesos basado en el modelo- que permite establecer comprobaciones dinámicas de la compatibilidad de interfaces. Dado que la evolución del espacio de tuplas gobierna cada paso del cómputo basado en Linda, la referencia al entorno en la relación de compatibilidad definida es esencial, ya que proporciona información relevante respecto al estado del sistema, representado por el conjunto de tuplas incluidas en el espacio de datos. La composición segura de componentes sensible al entorno permite comprobar la compatibilidad de un componente y un sistema en ejecución con respecto al entorno actual, y puede condicionar la incorporación de una componente dada a un sistema en ejecución hasta que el espacio de tuplas alcance un estado adecuado con el objeto de asegurar el éxito del sistema globalmente.

### **2.2 Manifold**

Manifold es un lenguaje de coordinación orientado a control en el cual la comunicación entre procesos se realiza por medio de conexiones de flujos (*streams*) punto a punto. Los procesos que intervienen en un patrón de coordinación se definen por medio de un estado formado por el conjunto de conexiones de flujos observables y reaccionan ante la presencia de eventos [1]. Su modelo conceptual se basa en el modelo IWIM [3], según el cual existen procesos que juegan el rol de *workers*, y procesos *manager*, coordinadores en el caso de Manifold.

Posee características interesantes -composicionalidad, comunicación anónima, evolución de la coordinación en reacción a eventos, separación de coordinación y computación- que lo disponen como un lenguaje apropiado para desarrollar configuraciones de componentes que evolucionan dinámicamente [4, 14].

Las capacidades del lenguaje no han sido suficientemente exploradas aún. Nuestro primer objetivo es utilizar Manifold para describir el protocolo de componentes y proponer una extensión que permita analizar distintas componentes y determinar si su composición paralela estará libre de bloqueos. Para ello se debe definir una relación de compatibilidad que sea decidible, ya que nuestro segundo objetivo es implementar una herramienta que compruebe automáticamente dicha compatibilidad. Por último nos interesa además, por medio de alguna relación de sustitutabilidad decidible, proponer condiciones

suficientes para garantizar, a partir de la descripción del protocolo de una componente, que dicha componente podrá sustituir a otra existente, manteniendo la compatibilidad con el resto del sistema.

### Conclusiones y trabajo futuro

Los distintos lenguajes de coordinación comparten características interesantes, tales como desacople de tiempo y espacio, y una gran potencia expresiva para especificar sistemas distribuidos y concurrentes. Motivados por ello, una vez logrado nuestro objetivo respecto a la interoperabilidad de componentes utilizando Manifold, nuestro trabajo estará orientado a obtener un resultado aplicable a los lenguajes de coordinación en general. Para lograrlo estamos planeando combinar los resultados obtenidos con nuestro trabajo en Manifold, los resultados presentados en [8] respecto a la interoperabilidad de componentes en Linda, y las relaciones entre los distintos estilos arquitectónicos y las diversas familias de lenguajes de coordinación presentados en [5, 6].

### Referencias

1. F. Arbab, *Manifold Version 2.0*. 2000
2. F. Arbab, *What do you mean, coordination?*, Bulletin of the Dutch Association for theoretical Computer Science, NVTI, pages 11-22, 1998.
3. F. Arbad, *The IWIM Model for Coordination of Concurrent Activities*, Proceedings First International Conference on Coordination Models, Languages and Applications (Coordination'96), pp 34-56.
4. F. Arbab, M.M. Bonsangue, F.S. de Boer, *A coordination Language for Mobile Components*, Proceedings of SAC 2000, ACM Press, pages 166-173, 2000.
5. M.M. Bonsangue, J.N. Kok, G. Zavattaro, *Comparing Software Architectures for Coordination Languages*, Proceedings of Coordination 99, Vol 1594 of Lecture Notes in Computer Science, pages 150-165, 1999.
6. M.M. Bonsangue, J.N. Kok, G. Zavattaro, *Comparing Coordination Models and Architectures using embeddings*, Reporte SEN-R0025, Centrum voor Wiskunde en Informatica (CWI), 2000
7. A. Bracciali, A. Brogi, F. Turini, *Interaction patterns*, V Jornadas Iberoamericanas de Ingeniería de Software, 2001.
8. A. Brogi, E. Pimentel, A.M. Roldán, *Interoperabilidad de Componentes Software en Linda*, IDEAS 2002
9. A. Brogi, J.M. Jacquet, *On the expressiveness of coordination models*, Coordination Languages and Models:3<sup>rd</sup>. International Conference, Vol. 1594 of Lecture Notes, pages 134-149, 1999.
10. C. Canal, *Un lenguaje para la especificación y validación de arquitecturas de software*, Ph.D. thesis, Depto Lenguajes y Ciencias de la Computación, Universidad de Málaga, 2001.
11. C. Canal, L. Fuentes, J.M. Troya, and A. Vallecillo, *Extending CORBA interfaces with pi-calculus for protocol compatibility*, Proceedings of the Technology of Object-Oriented Languages and Systems - TOOLS Europe 2000, IEEE Press 2000, pp. 208-225
12. D. Gelernter and N. Carriero, *Coordination Languages and their significance*, Communications of the ACM, No. 35, pp. 97-107. 1992
13. G.T. Leavens, M. Staman, *Foundations of component based systems*, Cambridge University Press, 2000.
14. G. Papadopoulos and F. Arbab, *Dynamic Reconfiguration in Coordination Languages*, Advances in Computers 46, Marvin V. Zelkowitz, Academic Press, 1998, pp. 329-400
15. P. Wegner, *Interoperability*, ACM Computing Surveys, Vol. 28, No. 1, March 1996.

## Tool Support for Verifying Applications using Object-Oriented Patterns

Gabriela Aranda, Andrés Flores, Agustina Buccella and Luis Reynoso  
*Departamento de Informática y Estadística, Universidad Nacional del Comahue,  
Buenos Aires 1400, Neuquén, Argentina*  
*phone: (54) 299 - 4490312, fax: (54) 299 4490313*  
*Email: garanda,aflores,agusb,lreynoso@uncoma.edu.ar*

**Abstract.** Applying design patterns is considered a helpful technique for designing software systems. Patterns description, however, results not sufficiently precise providing a weak understanding and making it difficult to be certain when a pattern is being used correctly. We have formally specified properties of patterns and a formal basis for object-oriented design. In the present work, our formal basis is used as a support to an automatic tool for verifying proper applications of patterns. The usage of this tool is mainly focused on the learning process about patterns applications. Through a better understanding of patterns, the designer may certify when and how a pattern is being appropriately applied to solve a specific design problem. Furthermore, the whole design process could be improved by using a precise technique supported by an automatic tool for verification.

### 1. Introduction

Designing software systems can be assisted by different design techniques such as applying design patterns, which is one of the most useful techniques because their widely recognised advantages in reinforcing several quality attributes such as high modifiability and reusability. Patterns are abstractions of concrete design problems which recur in a range of different contexts [1], and they describe a generic solution to these problems, which can be used to obtain a solution to a specific problem [2,3]. GoF catalogue [4] presents a well-known group of design patterns which uses object-oriented notation to capture the experience of several experts in software design.

Patterns are described by means of natural language narrative and graphical notation, which gives a sort of abstraction allowing a wide range of usage. However, this kind of notation results not sufficiently precise providing ambiguities and inconsistencies which leads a weak understanding and makes it difficult to be certain when a pattern is being used correctly. Following the goal of Pattern-Based Design, there should be applied the design principle of *rigour and formality*, in order to state a solution and to enhance such quality attributes as well. With this in mind, we have developed in the RAISE Specification Language RSL [5], a formal basis for object-oriented design where patterns may be applied [6] and formally specified the properties of each pattern in the GoF catalogue [7,8,9]. Thus, providing a more precise notation can improve understanding about patterns and designers can grow in the knowledge of when and how a pattern is being applied appropriately.

An enhancement concerning flexibility might be added to the process of pattern-based design. For this, we intent to build an automatic tool for modelling object-oriented design with patterns. The formal basis gives the precise format in which a design model can be represented with the aim of providing facilities for verifying a proper application of a pattern. The designer can graphically model a design specifying a particular pattern that suits with the problem. Thereafter, verification activities are performed on the representation of the graphical model.

Since our formal model was specified at an abstract level, it should be translated into a more concrete one in order to be closer to the codification phase. Thus, we have decided to translate the RSL Specification to an object-oriented model maintaining the structures and semantic of the building blocks. The representation of the graphical model in the tool is translated to an object structure according to the building blocks of the formal model in order to run the verification tasks.

The usage of this tool is also focused on the learning process about pattern application. Through a better understanding of patterns, the designer may certify when and how a pattern is being appropriately applied to solve a specific design problem. Furthermore, the design process may be improved with a precise technique supported by an automatic tool for verification.

We briefly present the formal basis in section 2 and its translation into an object-oriented model in section 3. Then, in section 4, the structure for the automatic tool is presented. We discuss future work and conclusion afterwards.

## 2. RAISE Model of a Pattern-based Design

Pattern-based design involves the binding of pattern elements to elements of the design [10]. A subset of the classes and relations in a design then conforms to a specific pattern if their properties are the same as those of the counterparts in the pattern. Our formal model was specified according to a Bottom-Up approach such that, given a (subset of a) design, find a pattern that matches it or check if a given pattern matches it [7]. For its definition the properties of a general object-oriented design were abstracted out. It includes the components of OMT-extended notation: classes, methods, variables, and relations. In addition, we have specified the meaning behind the notation: hierarchical properties, meaningful relationships between variables and relation names, and description of method's functionality.

Building blocks of the formal model were specified by using the RAISE specification language RSL[5]. Following we briefly explain its main constituents. See [6] for details and [11] for full specification.

- **Design Structure:** A design consists of a collection of classes and a collection of relations.

Design\_Structure = C.Classes x R.Wf\_Relations

- **Classes:** Each class has a name, which is unique in the design, a set of methods, a state which is represented as a set of variables, and a type which may be concrete or abstract.

Design\_Class :: class\_state : G.State  
                   class\_methods : M.Class\_Method  
                   class\_type : G.Class\_Type

- **Methods:** Every method has a name and a list of parameters. Also a pseudocode annotation attached to the method indicating the actions it will perform. Generic actions can be included in an annotation. They are mainly: Invocation, Instantiation and Conditional structure. See [11] for a detailed specification.

- **Relations:** A relation is basically determined by the classes it links and its type, which may be inheritance, association, aggregation, or instantiation. All relations are represented in the model as binary relations. We have modelled source and sink classes as well as their cardinality.

Design\_Relation :: relation\_type : Relation\_Type  
                   source\_class : G.Class\_Name  
                   sink\_class : G.Class\_Name,

Ref ::  
     relation\_name : G.Wf\_Vble\_Name  
     sink\_card : G.Card  
     source\_card : G.Card,

Relation\_Type ==  
     inheritance | association(as\_ref: Ref) |  
     aggregation(ag\_ref : Ref) | instantiation,

- **Design-Pattern Binding:** In order to link a subset of a design model to a specific pattern there is a *renaming map*, which associates design entities (classes, state variables and methods) with corresponding entities in the pattern. This mapping indicates the *role* each entity in the design plays in the pattern. Then a design matches a particular pattern if all the entities in the design

playing a pattern role satisfy its properties. Pattern properties were specified as Boolean-valued functions (see [9,11,12,13,14] for examples).

```

ClassRenaming ::                               Renaming =
  classname : G.Class_Name                      G.Class_Name -m-> ClassRenaming-set,
  methodRenaming :
    Method_and_Parameter_Renaming
  varRenaming : VariableRenaming,

```

The formal model as a whole then includes the *Design Structure* and the *Renaming* mapping, in such a way that given a design and a renaming to a specific pattern it is possible to verify if there is a matching.

Design\_Renaming = DS.Wf\_Design\_Structure x Wf\_Renaming,

### 3. Moving from RAISE Specification to an Object-Oriented Model

Every building block in our formal model will have a counterpart in the object-oriented model. Figure 1 presents an example: the *Wf\_Design\_Structure* type will be represented by a class called by a similar name. Since the *Design\_Structure* type comprises a collection of *Classes* and a collection of *Relations (Wf\_Relations)*, there will be two *state variables* in that class, which will be used to relate this class to others, with appropriate names, representing the collection of *Classes* and the collection of *Relations*.

```

Design_Structure = C.Classes x R.Wf_Relations,
Wf_Design_Structure = {| ds : Design_Structure :- is_wf_design_structure(ds) |}

```

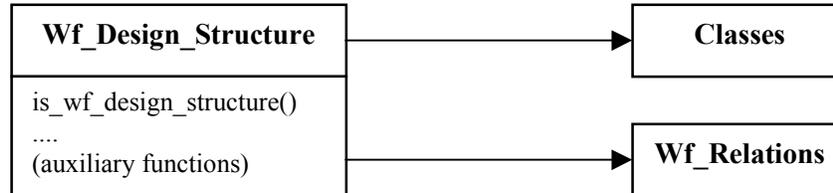


Figure 1: Translation of a RAISE type to OOModel

Since the principal goal of this work is to verify correctness in a design model there is a function in every building block of the formal model whose purpose is checking well-formedness. For example for the *Design\_Structure* to be checked if it is well-formed there is a function called *is\_wf\_design\_structure*. This function is represented as a method in the class *Wf\_Design\_Structure* of the object-oriented model. Functions called by the previous one are represented as private methods in the same class. For every building block there are “auxiliary functions” allowing verification but mainly supplying a shortcut through the whole structure. Those functions are methods included in the interface of the class representing that building block. The same process is applied for the specification of every GoF pattern.

### 4. Verification Tool for OOD using GoF Patterns

Basically the tool is divided into two layers: the modelling layer, whose result is a specification of an object-oriented design model provided by a graphical component; and the verification layer, which carries out the process of checking the correctness of the design model and also if the design subset related to a specific pattern satisfies the pattern properties.

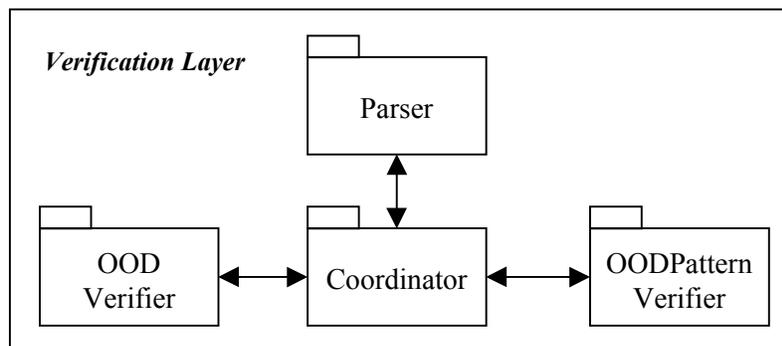
#### 4.1. Specification Layer

Since many tools providing a graphical component for modelling object-oriented design have already been developed with a proved success, we have decided to choose one of them, and thus concentrate the major effort in the field where less work has been delivered. After studying some tools available in the market we have selected a non-commercial tool called *FUJABA* [15]. This tool was developed in Java and it may produce a Java specification of the object-oriented design that was modelled by using it.

The functionality of *FUJABA* is quite similar to our expectations. However, according to our formal basis, a new behaviour needs to be developed in order to be able to represent an entire object-oriented model in which a pattern has been applied.

The extended functionality is related, for example, to the *annotations* attached to methods of classes. Annotations are used to express how collaborations between classes are carried out. Since our formal model expresses collaborations in a static way, annotations come to fulfil this subject. Other important new behaviour concerns the possibility of selecting a particular pattern from a pattern repository and setting which pattern roles are played by different entities at design level.

Some changes are also necessary, mainly in the notation of the object-oriented model, given by a Java specification. An example is the simplification needed when an *aggregation relation* is represented, thus complexity of the grammar for a Parser is not increased. Pattern roles and other design or pattern elements will be expressed adding “*comments*” in the Java specification.



*Figure 2: Verification Layer Components*

#### 4.2. Verification Layer

This Layer is composed of some components co-ordinated by a *Coordinator* module to deal with verification activities. The rest of components are: a *Parser*, a *OOD Verifier*, and a *OODPattern Verifier* as is showed in Figure 2.

Basically the Java specification obtained from the graphical component will be the input for the *Parser* component, which will create the representation of the model according to our formal model of an object-oriented design. As a result, an object structure, will be passed to the *OOD Verifier* component. This component will verify, by using the appropriate methods of the object structure, the correctness at design level of the whole model. In case an error is detected, then this component will return where the error was found in such a way it can be fixed in the graphical component.

If the result of the *OOD Verifier* is successful, then the *Coordinator* will pass the object structure to the *OODPattern Verifier*. This component will check if the properties of the selected pattern (in the graphical tool) are in fact satisfied. Once again, if an error is found the result will tell the designer what the property not satisfied is.

## 5. Conclusions

The main goal of our work is to accomplish a more precise but flexible Pattern-Based Design process. For this, we have developed, by means of RAISE [5], a formal basis of an object-oriented design and GoF design patterns [4], which was presented in section 2. In order to provide flexibility we are developing an automatic tool for verifying a correct application of patterns. With this in mind, the abstract formal model, giving the structure for representing a verifiable design model, needs to be concretised. Thus, a translation from the RAISE specification to an Object-Oriented Model is being carried out (see section 3). The main aspects concerning the automatic tool including its structure and its main components have been introduced in Section 4. Very often more than one pattern is needed to solve a design problem. If indeed our tool verifies a single pattern at a time, a piece of work has been already done concerning composition among patterns in [14]. Our current work is intended to be applied, in future, to improve a component development process.

## 6. References

1. Brad Appleton. *Patterns and Software: Essential Concepts and Terminology*. <http://www.enteract.com/~bradapp>, November 1997.
2. Doug Lea. *Patterns-Discussion FAQ*. <http://g.oswego.edu/dl/pd-FAQ/pd-FAQ.html>, December 1999.
3. Robert Zubeck. *Much Ado about Patterns*. <http://www.acm.org/crossroads/xrds5-1/patterns.html>, March 2000.
4. Gamma E., Helm R., Johnson R. and Vlissides J. *Design Patterns - Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
5. The RAISE Language Group. *The RAISE Specification Language*. BCS Practitioner Series, Prentice Hall, 1992.
6. Andres Flores, Luis Reynoso and Richard Moore. *A Formal Model of Object Oriented Design and GoF Design Patterns*. In proceedings of the FME 2001, Formal Methods Europe, Berlin, Germany, LNCS 2021, Springer Verlag 2001, 12-16 March 2001, pp. 223-241.
7. Andres Flores and Richard Moore. *Analisis and Specification of GoF Structural Patterns*. In proceedings of 19<sup>th</sup> IASTED, International Conference on Applied Informatics (AI 2001), Innsbruck, Austria, 19-22 February 2001, pp. 625-630.
8. Luis Reynoso and Richard Moore. *A Precise Specification of GoF Behavioural Patterns*. In proceedings of SNPD'01, 2<sup>nd</sup> International Conference on Software Engineering, Artificial Intelligence, Networking & Parallel/Distributed Computing, Nagoya, Japan, 20-22 August 2001, pp. 262-270.
9. Gabriela Aranda and Richard Moore. *GoF Creational Patterns: A Formal Specification*. Technical Report 224, UNU/IIST, P.O. Box 3058, Macau, 2000.
10. Marco Meijers, *Tool Support for Object-Oriented Design Patterns*. Master Thesis. Department of Computer Science, Utrecht University, The Netherlands, <http://www.serc.nl/people/florijn/work/patterns.html>, August 1996.
11. Andres Flores, Luis Reynoso and Richard Moore. *A Formal Model of Object Oriented Design and GoF Design Patterns*. Technical Report 200, UNU/IIST, P.O. Box 3058, Macau, July 2000.
12. Andres Flores and Richard Moore. *GoF Structural Patterns: A Formal Specification*. Technical Report 207. UNU/IIST, P.O. Box 3058, Macau, August 2000.
13. Luis Reynoso and Richard Moore. *GoF Behavioural Patterns: A Formal Specification*. Technical Report 201, UNU/IIST, P.O. Box 3058, Macau, May 2000.
14. Gabriela Aranda and Richard Moore. *Formally Modelling Compound Design Patterns*. Technical Report 225, UNU/IIST, P.O. Box 3058, Macau, December 2000.
15. Software Engineering Group. *FUJABA (From UML to Java And Back Again)*. University of Paderborn, Germany, <http://www.uni-paderborn.de/cs/fujaba>.

## Métricas de Performance en Administración de BDD en redes LAN y WAN

Lic. Rodolfo Bertone<sup>1</sup>  
Ing. Armando De Giusti<sup>2</sup>

*L.I.D.I.* (Laboratorio de Investigación y Desarrollo en informática)

### Resumen

En este proyecto se trata de investigar las características de replicación y fragmentación de la información, en particular la propagación y regulación de las actualizaciones de datos. El esquema de replicación planteado en este estudio se considera estático y preestablecido, interesando el comportamiento de los métodos, en cuanto al tratamiento de lecturas y modificaciones de los datos. El soporte de simulación elegido es Java, dado que permite una mayor versatilidad en las construcciones de aplicaciones.

Las métricas propuestas están relacionadas, básicamente, con el tiempo de respuesta ante operaciones de ABM-C, bajo diferentes entornos de replicación y actualización. Los resultados dependen, además, del entorno de simulación definido. Esto es, del tasa de comunicación de cada localidad, el volumen de información que contiene, el porcentaje de cada tipo de operación (alta, baja, modificación o consulta) etc. A partir de los parámetros definidos y los resultados obtenidos es posible evaluar el comportamiento del entorno distribuido según el esquema de replicación y actualización seleccionado.

Además, la investigación se combina con estudios realizados sobre protocolos de integridad para BDD (protocolos de cometido), para integrar la solución al esquema de replicación con otro esquema que asegure integridad de los datos. De esta forma se podrá incorporar a las mediciones obtenidas, la “distorsión (overhead) de tiempo de procesamiento” que produce el proceso de aseguramiento de integridad de datos, en conjunto con las operaciones que recuperan la BD ante errores de ejecución.

### Palabras Clave

Bases de Datos Distribuidas – Replicación de Información – Métricas de procesos

---

<sup>1</sup> Profesor Adjunto Dedicación Exclusiva – Facultad de Informática – UNLP – pbertone@lidi.info.unlp.edu.ar

<sup>2</sup> Profesor Titular Dedicación Exclusiva, Investigador Principal CONICET – Facultad de Informática - UNLP – degiusti@lidi.info.unlp.edu.ar

## Introducción

El procesamiento de bases de datos distribuidas (BDD) consiste en trabajar con información en la cual la ejecución de transacciones y la recuperación y actualización de los datos se produce en dos o más computadoras independientes, por lo general separadas geográficamente.

La utilización de bases de datos distribuidas (BDD) representa una solución viable para los usuarios cuando deben optar para la generación de sus sistemas de información. La utilización de estas bases de datos para el mantenimiento de la información requiere el estudio de una gran cantidad de casos particulares, a fin de determinar las mejores condiciones de trabajo para cada problema real.

La replicación de datos es uno de los casos particulares que se presentan cuando se estudia la utilización de BDD. Esta replicación es mucho más que simplemente copiar el dato a lo largo de varios nodos de la red. Está acompañado de análisis, diseño, implementación, administración y monitoreo de un servicio que garantiza la consistencia de la información a través de múltiples localidades en el ambiente distribuido.

El propósito de distribuir datos sobre múltiples sitios es aumentar la confiabilidad, disponibilidad y eficiencia, entre otros. En estas aplicaciones, a menudo, el acceso concurrente de una gran cantidad de usuarios al sistema de base de datos, demanda alta disponibilidad y rápidos tiempos de respuesta. La replicación es el factor clave para una mayor disponibilidad (confiabilidad) y eficiencia. Los datos replicados son almacenados en múltiples sitios para que estos puedan ser accedidos por los usuarios cuando alguna de esas copias no está disponible debido a fallas en los sitios, proporcionando mayor disponibilidad. También la performance de los sistemas de base de datos distribuidas es mejorada ubicando los datos en los lugares próximos a donde se usan y poder de esta manera acceder a copias locales, reduciendo la necesidad de costosos accesos remotos.

## Objetivos

El procesamiento sobre bases de datos distribuidas continúa en evolución, por lo tanto no es posible hablar de una disciplina en plena madurez, donde para todos los problemas que se presentan existe una solución óptima y efectiva; y donde existe una caracterización única que represente una solución eficiente para aplicar en todo caso. El control de concurrencia distribuido, en comparación con uno centralizado está en pleno desarrollo. Se han propuesto una serie de algoritmos para su tratamiento y algunos de ellos cuenta con implementaciones que, si bien se adaptan a problemas específicos del mundo real, distan de ser el óptimo aplicable en todas las casos.

Además, existen problemas identificados con BDD, pero con soluciones limitadas y con falta de eficiencia. Algunos resultados teóricos son difíciles de utilizar en aplicaciones reales de BDD.

El presente trabajo evoluciona en el estudio de las principales características que debe seguir un esquema de replicación de datos. A partir de cada una de las características en juego, se definió un modelo de simulación que permite evaluar el comportamiento posible del modelo de datos y su esquema de replicación. Con los resultados obtenidos es posible evaluar distintas alternativas de solución y, de esa forma, aproximarse al esquema que mejor se adecue para el problema que se está estudiando. Básicamente, se mide el tiempo de respuesta de cada traza de ejecución en función de la configuración de replicación y actualización definida. Cada traza de ejecución se define en función de parámetros como: porcentaje de replicación de datos, porcentaje de fragmentación de datos, número de localidades intervinientes, "costo" de comunicación de cada localidad, tamaño de la traza (cantidad de transacciones involucradas),

porcentaje de operaciones de la traza (altas, bajas, modificaciones y consultas), etc. Posteriormente se pretende agregar a esta definición los parámetros que definen el porcentaje de transacciones con error, cuando se integre el trabajo con los estudios realizados sobre los protocolos de cometido.

Los objetivos específicos de la primer etapa, utilizando el ambiente desarrollado en el LID [Len 2001], son:

- ✎ Implementar técnicas de replicación basadas en diferentes esquemas de regulación y propagación de actualizaciones sobre las réplicas.
- ✎ Medir las características de performance de dichas técnicas.
- ✎ Comparar las técnicas evaluando las mediciones de los resultados obtenidos.
- ✎ Evaluar los beneficios e inconvenientes de la replicación de datos en general.
- ✎ Combinar los desarrollos con los protocolos de integridad
- ✎ Reevaluar los resultados obtenidos

## Bibliografía

- [Bell et al, 1992] *Distributed Database Systems*. Davil Bell, Jane Grimson. Editorial: Addison Wesley. 1992.
- [Bhargava, 1987] *Concurrency and Reliability in Distributed Database Systems*. B. Bhargava (editor). Van Nostrand Reinhold, 1987.
- [Buretta 1997] *Data Replication. Tools and techniques for managing distributed information*. Marie Buretta. Editorial: John Wiley & Sons, Inc. 1997.
- [Carey et al, 1991] *Conflict Detection Tradeoff for Replicated Data*. Michael Carey, Miron Livny. ACM Transactions on Database Systems, Vol 16, 4 December 1991, pp 703-746
- [Di Paolo et al., 1999] *Ambiente Experimental para Evaluación de Bases de Datos Distribuidas*. Monica Di Paolo, Rodolfo Bertone, Armando De Giusti, Anales ICIE 99. International Congress of Information Engineering. Buenos Aires 18-20/08/99. pp 729-743
- [Di Paolo 1999] *BDD. Estudio de consistencia de transacciones con dos modelos de subsistemas de comunicaciones*. Mónica Di Paolo. Tesina de Grado. Facultad de Ciencias Exactas. 1999. Director: Rodolfo Bertone
- [Hoffer 2002] *Modern Database Management*. Jeffrey Hoffer, Mary Prescott, Fred McFadden. Editorial: Prentice Hall. 2002
- [Gray et al., 1996] *The dangers of replication and a solution*. Jim Gray, Pat Helland, Patrick O'Neil, Dennis Shasha. SIGMOD '96. 6/96. Montreal, Canada., pp 173-182.
- [Len et al., 2001] *Estudio de actualización de Réplicas de datos en entornos de BDD*. Sergio Len. Rodolfo Bertone. Sebastián Ruscuni. Anales: Cacic 2001. Congreso Argentino de Ciencias de la Computación. El Calafate. Octubre 2001. pp 695-706.
- [Len 2001] *Actualización de Réplicas de datos en entornos de BDD*. Sergio Len. Tesina de Grado. Facultad de Informática. 2000. Director: Rodolfo Bertone.
- [Miaton et al., 1998] *Expediencias en el Análisis de Fallas en Bases de Datos Distribuidas*. Ivana Miatón, Sebastián Ruscuni, Rodolfo Bertone, Armando De Giusti. Anales: Cacic 98. Congreso Argentino de Ciencias de la Computación. Neuquen. Octubre 1998. pp 265-276
- [Miaton et al., 1999] *Ambiente de simulación para la Recuperación en un Entorno con BDD*. Ivana Miatón, Sebastián Ruscuni, Rodolfo Bertone, Armando De Giusti. Anales: Cacic 99. Congreso Argentino de Ciencias de la Computación. Tandil. Octubre 1998.
- [Özsu et al., 1991] *Principles of Distributed Database Systems*. Second Edition. M.Tamer Özsu, Patrick Valduriez. Prentice Hall. 1991.

- [Ruscuni et al., 2000] *Evaluación de Replicación y Consistencia en Bases de Datos Distribuidas*. Sebastián Ruscuni, Rodolfo Bertone. Cacic 2000. Congreso Argentino de Ciencias de la Computación. Ushuaia, Octubre 2000. pp 145-158
- [Ruscuni 2000] *Estudio de Recuperación de errores en BDD*. Sebastián Ruscuni. Tesina de Grado. Facultad de Informática. 2000. Director: Rodolfo Bertone.
- [Silberschatz et al, 1998] *Fundamentos de Bases de Datos*. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. Editorial: Mc Graw Hill. 1998. Tercera Edición
- [Simon 96] *Distributed Information Systems. From C/S to distributed multimedia*. Errol Simon. Editorial: Mc Graw Hill. 1996.
- [Stamos et al., 1993] *Coordinator log transation execution Protocol*. J. Stamos, F. Cristian. Journal of Distributed and Parallel Databases. 1993.
- [Triantafillou et al., 1995] *The location-Based Paradigm for replication: Achieving Efficiency and Availability in Distributed Systems*. Peter Triantafillou, David Taylor. IEEE Transaction on Software Engieneering, Vol 21, No. 1, January 1995.

# Methods for Measurement-Based COTS Assessments and Selection \*

Alejandra Cechich<sup>1</sup> and Mario Piattini<sup>2</sup>

<sup>1</sup>*Departamento de Informática y Estadística – Universidad Nacional del Comahue  
Buenos Aires 1400, Neuquén, Argentina  
phone: (54) 299 - 4490312, fax: (54) 299 4490313  
E-mail: acechich@uncoma.edu.ar*

<sup>2</sup>*Escuela Superior de Informática - Universidad de Castilla-La Mancha  
Paseo de la Universidad 4, Ciudad Real, España  
phone: (34) 926 295300, fax: (34) 926 295354  
E-mail: Mario.Piattini@uclm.es*

**Abstract.** During recent years, new software engineering paradigms like component-based software engineering and COTS-based development have emerged. Both paradigms are concerned with reuse and customisation of existing components. The use of COTS software has become more and more important in state-of-the-art and state-of-the-practice software and system development. Using COTS software promises faster time-to-market, which can yield substantial advantages over competitors with regards to earlier placement of a new product on a market. At the same time, COTS software introduces risks such as unknown quality properties of the COTS software in use that can inject harmful side effects into the final product. In this paper, we present a brief comparison of well-known COTS evaluation methods. Our work is focused on identifying the main goals and features of the methods in order to extend their uses on evaluating COTS products as well as COTS development processes.

## 1. Introduction

Typically, COTS-based development - in parallel to traditional development process models, e.g. waterfall, spiral - consist of four phases: (1) COTS assessment and selection, (2) COTS tailoring, (3) COTS integration, and (4) maintenance of COTS and non-COTS parts of the system. The first phase — COTS assessment and selection — is the most crucial phase in the COTS-based cycle. Here long-term decisions on which COTS will be used in a software system are made. Non-optimal COTS software used in the development of a system can become extremely costly for a software organisation. Apart from the general reuse problems (selection, integration, maintenance, etc.), COTS products have by their own specific problems [1]:

- **Incompatibility:** COTS components may not have the exact functionality required; moreover, a COTS product may not be compatible with in-house software or other COTS products.
- **Inflexibility:** usually the source code of COTS software is not provided, so it cannot be modified.
- **Complexity:** COTS products can be too complex to learn and to use imposing significant additional effort.
- **Maintainability:** different versions of the same COTS product may not be compatible, causing more problems for developers.

---

\* This work is partially supported by the CYTED (Ciencia y Tecnología para el Desarrollo) project VII.J-RITOS2 (Red Iberoamericana de Tecnologías de Software para la década del 2000).

Therefore, repeatable and systematic methods to assess and select COTS software are an important issue in COTS-based software engineering. Currently, there exist several solutions for COTS selection and the following section gives a brief summary of existing methods to COTS assessment and selection.

## 2. Methods for COTS Assessment and Selection

The **OTSO (Off-The-Shelf-Option)** method and some experiences of its use are presented in [2, 3]. The method supports the search, evaluation and selection of reusable software and provides specific techniques for defining evaluation criteria and comparing the costs and benefits of product alternatives. In OTSO method, the evaluation criteria are gradually defined as selection process progresses. The evaluation criteria are derived from reuse goals and factors that influence these goals [4]. The evaluation criteria definition process in OTSO essentially decomposes the requirements for the COTS software into a hierarchical criteria set. Each branch in this hierarchy ends in an *evaluation attribute*: a well-defined measurement or a piece of information that will be determined during evaluation. This hierarchical decomposition principle has been derived from Basili's GQM [5, 6] and Saaty's approach [7].

The *search* in the selection process attempts to identify and find all potential candidates for reuse. The objective of the *screening* process is to decide which alternatives should be selected for more detailed evaluation. The objective of the *evaluation* process is to evaluate the selected alternatives by the evaluation criteria and document evaluation results. Evaluation produces data on how well each alternative meets the criteria defined. As the COTS alternatives have been evaluated the evaluation data needs to be used for making a decision. The analysis of results relies on the use of the Analytic Hierarchy Process (AHP) for consolidating the evaluation data for decision-making purposes [7].

The **CAP (COTS Acquisition Process)** method [9, 10] consists of three components: the CAP Initialisation Component (CAP-IC), the CAP Execution Component (CAP-EC), and the CAP Reuse Component (CAP-RC).

The first step in CAP-IC is the identification of criteria against which candidate COTS software alternatives must be evaluated (CAP activity "Tailor & Weight Taxonomy"). In this activity the requirements are translated into taxonomy of evaluation criteria and prioritised (or weighted) according to the Analytic Hierarchy Process (AHP) under incorporation of multiple stakeholder interests.

The second step is to estimate how much effort will probably be needed to actually apply all evaluation criteria to all COTS software candidates (CAP activity "Estimate Measurement Effort"). The third step is to set up the measurement plan according to which all evaluation activities will be conducted (CAP activity "Elaborate Measurement Plan"). The measurement plan is either designed straightforward from the taxonomy of evaluation criteria - in the case the measurement effort estimates for measurement satisfy the budget and resource constraints. Alternatively, the measurement plan is constructed by employing optimisation algorithms with the objective to maximise priority coverage in the measurement plan. Finally, a review step certifies that all CAP-IC activities have been conducted correctly (CAP activity "IC-Review").

**PORE (Procurement-Oriented Requirements Engineering)** [11] uses an iterative process of requirements acquisition and product evaluation/selection as its main approach for Component-based Systems Engineering (CBSE). The basic PORE life-cycle process model has six generic processes. They are defined at 3 levels according to Humphrey's process model; hence, the heart of PORE is an iterative and parallel process of requirements acquisition and product evaluation and

selection. Within this iterative process, the PORE method integrates different methods, techniques and tools for requirements acquisition and product identification and evaluation with process guidance for choosing and using each technique such as feature analysis techniques, MCDM (Multi-Criteria Decision Making) techniques, argumentation techniques, requirements acquisition techniques, requirements engineering methods, etc. As well as integrating these techniques, PORE also provides guidelines for designing product evaluation test cases and for organising evaluation sessions.

The PORE method uses the requirement model to both acquire and elaborate the requirements statements and to check requirement-product compliance during the product evaluation and selection process. In the CBSE development process, this view is even taken further to include information about product suppliers such as their technical capabilities, application domain experience, ISO standard certification, CMM level, etc. and legal issues involved in product procurement such as negotiation contract terms and conditions, licensing arrangements, etc.

**CEP (Component Evaluation Process)** [12,13] defines a tailorable process and associated methods to evaluate components for inclusion into a system. At the start of a project, the Component Capability Database is searched for any components that already exist in the system, components that have been dictated or preselected for use in the system, or known candidate components for the system. There are five primary activities in the Component Evaluation Process, a scope of the activities to be performed, a search and screening of components that are candidates for inclusion into the system, the definition of the evaluation criteria, the evaluation itself, and, the analysis of the evaluation results. Decision support methods have been adapted for use with the Component Evaluation Process and may be used in activities ranging from search and screening of candidate components to analysing the evaluation results.

The Evaluate Component activity in the decomposition contains the following activities:

- *Scope Evaluation Efforts*, determines the scope for the activities involved in the Component Evaluation Process.
- *Search and Screen Candidates*, conducts a search of component sources, both internal and external, for all potential candidate components to be considered for inclusion in the system. This includes components that already exist in the system, have been preselected for the system, or dictated for use in the system.
- *Define Evaluation Criteria*, uses the initial search criteria, including several characteristics to produce detailed evaluation criteria necessary to conduct the evaluations of the component alternatives.
- *Evaluate Component Alternatives*, uses the detailed evaluation criteria to perform the evaluation on the selected component alternatives.
- *Analyse Evaluation Results*, performs the analysis of the evaluation results for the evaluations performed on the selected component alternatives.

### 3. Major issues

The OTSO method starts from the same input as CAP but appeared too much consuming effort. Furthermore, OTSO does not allow for a smooth and dedicated change of COTS evaluation depths. The OTSO method supports systematic evaluation of COTS alternatives and considers both financial and qualitative aspects of the selection process. The experiences in several cases [3, 8] indicate that the method is feasible in operational context although more formal experiments are required to validate the method.

PORE method focuses on requirements engineering and COTS procurement at the same time, thus its scope is broader than the other methods. One of the problems of the PORE method is that the

iterative process of requirements acquisition and product evaluation/selection is very complex. At any point, a large number of possible situations can arise, or a large number of processes and techniques to use in a single situation can be recommended. To handle this scale of complexity, a prototype tool known as PORE Process Advisor is being developed. The main components of the tool are a process engine which analyses the current set of goals to be achieved, model properties (inferred by the situation inference engine) and instructions from the requirements engineering team to recommend process advise.

CEP is an advancement of OTSO. There are several enhancements, e.g. planning facility for evaluating the available component alternatives, an activity for developing, executing, analysing evaluation scenarios, and the use of an additional mathematical method for evaluation data analysis. The Component Evaluation Process is a standalone process; however, it also may be used in conjunction with a higher-level development process.

CAP is also based on OTSO, but strictly Measurement-oriented, allowing for optimisation of the evaluation towards cost-efficiency, systematic changes of evaluation depth, and spiral model-like enactment. CAP defines a procedure for evaluating and selecting COTS software. The starting point for this decision procedure is an initial set of COTS software alternatives and a measurement plan that defines which criteria have to be measured and evaluated in order to identify the most suitable COTS software. However, simply measuring all applicable criteria on all COTS software alternatives can be expensive since (i) many COTS software alternatives might be available, (ii) the set of evaluation criteria could be quite large, and (iii) some of the criteria might be very difficult or expensive to measure, e.g. reliability. The effectiveness of CAP depends on the expressiveness of the criteria selected for evaluation. A trade-off between the effectiveness of the evaluation criteria and the cost, time, and resource allocation of the criteria while measuring must be reached.

All methods form a more or less waterfall-like process for reusable component evaluation and selection. It is based in all cases on general-purpose criteria without going into deeper characterisations.

## Conclusions and Future Work

We have presented brief comparison of existing methods for assessment and selection of COTS components. We know that a late recognition that non-optimal COTS software was used in the development of the system can become extremely costly for a software organization. However, existing methods are specially focused on general properties of components instead of domain-oriented properties. A domain-oriented focusing could produce a more accurate evaluation since properties of components could be more precisely defined. In the next stage of our work, we are adapting traditional COTS evaluation methods, such as the OTSO method or the CAP method, to include characteristics of products in e-commerce applications. Our extension will consider aspects according to ISO standard 9126 on Product Quality [14] applied to COTS [15] including detailed metrics as well as other specific proposals for e-commerce.

## References

- [1] Buschmann R. Meunier, H. Rohnert, P. Sommerlad, M. Stal: Pattern-Oriented Software Architecture - A system of Patterns, John Wiley & Sons, 1996
- [2] Kontio Jyrki.: OTSO: A Systematic Process for Reusable Software Component Selection, University of Maryland, Technical Report UMIACS-TR-95-63, 1995.
- [3] Kontio J., Chen S., and Limperos K.: A COTS Selection Method and Experiences of its Use, 20th Annual Software Engineering Workshop, NASA Software Engineering Laboratory, Greenbelt, MD, 1995.
- [4] Kontio J., Caldiera G., and Basili V.: Defining Factors, Goals and Criteria for Reusable Component Evaluation, Proceedings of CASCON'96 Conference, Toronto, Canada, 1996.

- [5] Basili V.: Software Modelling and Measurement: The Goal/Question/Metric Paradigm, University of Maryland, Technical Report CS-TR-2956, 1992.
- [6] Basili V. and Rombach H.: Tailoring the Software Process to Project Goals and Environments, Proceedings of the 9th International Conference on Software Engineering, IEEE Computer Society Press, 1987, pp. 345-357.
- [7] Saaty T. L.: The Analytic Hierarchy Process, McGraw-Hill, 1990.
- [8] Kontio Jyrki: A Case Study in Applying a Systematic Method for COTS Selection, Proceedings of the 18th International Conference on Software Engineering, Germany, 1996.
- [9] Ochs M., Pfahl D., Chrobok-Diening G., and Nothhelfer-Kolb: A Method for Efficient Measurement-based COTS Assessment and Selection - Method Description and Evaluation Results, Fraunhofer Institut Experimentelles Software Engineering, IESE-Report No. 055.00/E, Version 1.0, 2000.
- [10] Ochs M., Pfahl D., Chrobok-Diening G., and Nothhelfer-Kolb: A COTS Acquisition Process: Definition and Application Experience, Fraunhofer Institut Experimentelles Software Engineering, IESE-Report No. 002.00/E, 2000.
- [11] Ncube C. and Maiden A.: PORE: Procurement-Oriented Requirements Engineering Method for the Component-Based Systems Engineering Development Paradigm, <http://www.sei.cmu.edu/cbs/icse99/papers/11/11.htm>
- [12] Polen S., Rosen L., and Phillips B.: Component Evaluation Process, Software Productivity Consortium, Report SPC-98091-CMC, Version 01.00.02, May 1999.
- [13] Polen S., Rosen L., and Phillips B.: Component Evaluation Process, Appendix G: CEP in ETVX Notation, Software Productivity Consortium, Report SPC-98091-CMC, Version 01.00.02, March 2000.
- [14] ISO/IEC 9126 Standard, Information Technology - Software Product Evaluation - Quality Characteristics and Guidelines for their Use, International Organization for Standardization, (Ed.) Geneva, 1991
- [15] Bertoa M. and Vallecillo A., Atributos de Calidad para Componentes COTS, V Workshop Iberoamericano de Requisitos y Ambientes de Software, La Habana, Cuba, 23-26 April 2002.

# An extension to EMTPL

María Laura Cobo      Marcelo Alejandro Falappa  
Department of Computer Science and Engineering \*  
Universidad Nacional del Sur  
Bahía Blanca - Argentina  
fax: +54 291 4595136  
[mlcobo, mfalappa] @cs.uns.edu.ar

## Abstract

The development of languages allowing a proper handling of time is important in many areas of computer science because the capability to deal with the notions of change and time are essential to solve their problems.

In the development of EMTPL [CA99a] we focused the research work in making the languages useful in the area of databases, where some problems demand the possibility to represent and use temporal information. We achieve that goal, although we managed the negation in a way that causes the lost of some properties of the underlying logic. This logic was deeply studied and it is one of the most famous metric temporal logics [Pri67a]. The main idea of this work is to extend EMTPL by the addition of a new operator, and also grant the negation with its logic meaning.

**Key words:** Programming Languages, Metric Temporal Logic, Deductive Databases, Negation.

## 1 Introduction

The database research field provides solutions for the problem of storing and retrieving information. One of the possible ways to achieve this goal is by using deductive databases. They use logic as a mean to represent information in a declarative way and an inference procedure to answer queries. Some problems are involved with the notions of change and time. For example we could be interested on keeping record of the different positions of the employees in some factory, evolution of patients in a hospital, transactions in a business or events in an industry environment.

Some implementations are available for languages with capability to handle temporal concepts, for example, *Temporal Prolog* in [CA99b] based on the proposal of [Gab87]. The Temporal Prolog's proposal was improved in [CA99a] where we had presented a programming language, *Metric Temporal Programming Language* (MTPL), with the capability of dealing with more precise temporal information; this language is based on a deeply studied logic proposed by Prior ([Pri67a] and [Pri67b]) and follows one of the most known ways of seeing time, through A-series (past, present and future). There are other logics, and some of them formalize the other view of time, as B-series (before, after), where we use explicit references of time. An example of logic of this later type has been presented by Rescher [RU71]. However, there are many circumstances when it is very useful if not mandatory to have both ways to use and refer time.

To preserve all the properties and benefits of the underlying logic is important to associate the logic meaning to the negation operator. In the first version of EMTPL, we use negation as

---

\*Partially supported by the Secretary of Science and Technology (Universidad Nacional del Sur).

failure instead of classical negation. Because of this the manipulation of negative information is not always appropriate, we could think about the need of counting with an operator that offers to the language some kind of temporal negation. This new operator could be interesting, because we will count with an operator that provides a way to manage that negative information that is vital in some contexts or problems.

EMTPL [CA99a] was designed to manipulate temporal references of both kinds, making the language appropriate for Databases. In section 2 we present a brief presentation of the programming language EMTPL. In section 3 we mention the principal aspects of the extension we have in mind. Conclusions and future work suggestions will be presented in section 4.

## 2 Brief revision to EMTPL

### 2.1 Syntax and semantics

The syntax of the EMTPL's operators as follows:

$$\phi = p | \neg\phi | \phi_1 \wedge \phi_2 | AT_n\phi | \diamond_n \phi | \exists n \diamond_n \phi | \forall n \diamond_n \phi | \diamond_n \phi | \exists n \diamond_n \phi | \forall n \diamond_n \phi$$

Disjunction and implication could be defined as usual. More temporal operators are definable as specified below.

The intuitive meaning of the temporal operators  $\diamond$  and  $\exists$  is as follows:

- $\diamond \phi$  sometime in the future  $\phi$  or  $\phi$  will be true in some moment of the future.
- $\exists \phi$  sometime in the past  $\phi$  or  $\phi$  was true in some moment of the past.

With this intuitive idea we can now give the formal semantic the operators. We assume, then, a set of instants  $\dots, i_n, i_{n+1}, i_{n+2}, \dots$  that represents an unbounded and linear temporal structure. The information in the system could be seen as a list of sets,  $\sigma$ , that contains all the truths we have in our theory. Each member of the list is a set denoted by  $\sigma(j)$ . Then  $\sigma(j)$  represents the set of truths at an instant  $j$ . We follow with an inductive definition for "a temporal proposition true at an instant  $j$ ". We start specifying that  $p \in \sigma(j)$  means " $p$  is true at instant  $j$ ".

It is important to point out that the quantification is applied only over the members of the temporal structure, *i.e.*, the set of numbers we choose to represent time. We claim this set of numbers to be a metric space.

$$\begin{aligned} (\sigma, j) \models p &\text{ iff } p \in \sigma(j) \\ (\sigma, j) \models \neg p &\text{ iff } (\sigma, j) \not\models p \\ (\sigma, j) \models p \wedge q &\text{ iff } (\sigma, j) \models p \text{ and } (\sigma, j) \models q \\ (\sigma, j) \models \diamond_c p &\text{ iff } (\sigma, j + c) \models p \\ (\sigma, j) \models \exists_c p &\text{ iff } (\sigma, j - c) \models p \\ (\sigma, j) \models \exists n \diamond_n p &\text{ iff exists an instant } n, n > 0 \text{ such that } (\sigma, j + n) \models p \\ (\sigma, j) \models \exists n \exists_n p &\text{ iff exists an instant } n, n > 0 \text{ such that } (\sigma, j - n) \models p \\ (\sigma, j) \models \forall n \boxplus_n p &\text{ iff for all instant } n, n > 0 \text{ we have } (\sigma, j + n) \models p \\ (\sigma, j) \models \forall n \boxminus_n p &\text{ iff for all instant } n, n > 0 \text{ we have } (\sigma, j - n) \models p \\ (\sigma, j) \models AT_c p &\text{ iff } (\sigma, c) \models p \end{aligned}$$

We can define some other nice operators from this ones [CA00, CA99a]

## 2.2 Definition of the language

The language is based on Prior's logic [Pri67a, Pri67b] plus an extension to that logic by the introduction of the  $AT$ . This extension consists mainly, on introducing the  $AT$  operator in Prior's axiomatization [CA99a].

In what respects to the language based on this extended logic, *Extended Metric Temporal Programming Language* (EMTPL), we define it in this way:

### DEFINITION 1 (EMTPL)

Let us consider a language with propositional atoms, the logic connectives:  $\wedge, \vee, \rightarrow, \neg$  and the temporal operators:  $\diamond_n, \diamond_n^1, \diamond_n^2, \boxplus, \boxminus$ . We define the notion of EMTP's *program, clause, always clause, ordinary clause, head, body* and *goal* as follows:

- A *program* is a set of "clauses".
- A *clause* is  $AT_i C$  or  $C$  where  $C$  is an "ordinary clause".
- An *ordinary clause* is a "head" or a  $B \rightarrow H$ , where  $B$  is a "body" and  $H$  is a "head".
- A *head* is an atomic formula,  $\diamond_n A$  or  $\diamond_n^1 A$  or  $\boxplus A$  or  $\boxminus A$  where  $A$  is a conjunction of "ordinary clauses".
- A *body* is an atomic formula, a conjunction of bodies,  $\diamond_n B$  or  $\diamond_n^1 B$  or  $\boxplus B$  or  $\boxminus B$  or  $\neg B$  where  $B$  is a body.
- A *goal* is  $AT_n B$  or  $B$  where  $B$  is any body or a disjunction of bodies. ■

It is important to point out that the facts or rules (*Prop*) that are not affected with an  $AT_n$  operator are interpreted as facts or rules that happen in the present moment, *i.e.*, we are going to put them in the base as  $AT_m Prop$ , where  $m$  means "now". The reader can find more details of this language in [CA99a].

## 3 Motivation for a negation extension

The definition of EMTPL includes the operator  $\neg$ , this operator can be part of a clause, more specifically it can form part of the body of an EMTPL clause. We have previously seen that the semantic of  $\neg$  operator is:

$$(\sigma, j) \models \neg p \text{ iff } (\sigma, j) \not\models p$$

Intuitively this is the negation as failure semantic, because it means that some formula  $\neg A$  is **true** if and only if the system can not prove the truth of  $A$ . This semantic is quite different from the classical negation semantic, in which the truth of  $\neg A$  depends only in the proof of the negated formula. The decision of which semantic choose is quite important, because no matter which one the language support, it causes some effects on it. If we choose negation as failure, the system loses some properties of the underlying logic and a considerable set of theorems. Another disadvantage of this way of conceive negation is the restriction to manipulate negative information. That means that we can not have in the base any negative information, this can be seen in the fact that the negation operator can be only part of a body clause and that body must be proven in the system. On the other hand since databases are usually systems with incomplete information, classical negation will not always has an positive or negative answer

---

<sup>1</sup> $\diamond \phi =_{Def} \exists n \diamond_n \phi$

<sup>2</sup> $\boxplus \phi =_{Def} \forall n \diamond_n \phi$

to all possible queries. Instead the system could answer **true**, **false** and **unknown**. In the development of EMTPL we thought of critical data bases where the answer **unknown** is quite dangerous.

Considering this we can think in the possibility of adding a new negation operator to EMTPL language, *i.e.* the language keep having negation as failure, but provides some kind of temporal negation, that allows manage negative information.

### 3.1 Extension 1: $EMTPL_{\nabla}$

$EMTPL_{\nabla}$  is like EMTPL, except for the new operator  $\nabla$ . The syntax of the new language  $EMTPL_{\nabla}$  is as follows:

$$\phi = p | \neg\phi | \phi_1 \wedge \phi_2 | AT_n\phi | \nabla\phi | \diamond_n\phi | \exists n\ \diamond_n\phi | \forall n\ \diamond_n\phi | \diamond_n\phi | \exists n\ \diamond_n\phi | \forall n\ \diamond_n\phi$$

Assuming a temporal structure like EMTPL's one, the semantic of the new operator is:

$$(\sigma, j) \models \nabla p \text{ iff } \nabla p \in \sigma(j) \text{ or } (\sigma, j) \models \neg p$$

The intuitive meaning of this semantic is: “ $\nabla p$  is true only if we have  $\nabla p$  in the set of proven facts, or we can not prove  $p$ ” In order to avoid the *or-problems* of logic programming we only are going to allow the negation of atomic formulas.

As a consequence of the definition we have the following definition of  $EMTPL_{\nabla}$

#### DEFINITION 2 ( $EMTPL_{\nabla}$ )

Let us consider a language with propositional atoms, the logic connectives:  $\wedge, \vee, \rightarrow, \neg, \nabla$  and the temporal operators:  $\diamond_n, \diamond_n, \diamond, \diamond, \boxplus, \boxminus$ . We define the notion of  $EMTPL_{\nabla}$ 's *program, clause, always clause, ordinary clause, head, body* and *goal* as follows:

- A *program* is a set of “clauses”.
- A *clause* is  $AT_i C$  or  $C$  where  $C$  is an “ordinary clause”.
- An *ordinary clause* is a “head” or a  $B \rightarrow H$ , where  $B$  is a “body” and  $H$  is a “head”.
- An *atomic formula* is an atom or  $\nabla A$  where  $A$  is an atom.
- A *head* is an atomic formula,  $\diamond_n A$  or  $\diamond_n A$  or  $\boxplus A$  or  $\boxminus A$  where  $A$  is a conjunction of “ordinary clauses”.
- A *body* is an atomic formula, a conjunction of bodies,  $\diamond_n B$  or  $\diamond_n B$  or  $\boxplus A$  or  $\boxminus A$  or  $\neg B$  where  $B$  is a body.
- A *goal* is  $AT_n B$  or  $B$  where  $B$  is any body or a disjunction of bodies. ■

For example we can now infer negative information like in the following rule:

$$in\_way(Car, Street) \wedge move(Car) \rightarrow \nabla free\_way(Street)$$

In EMTPL, the previous rule is not allowed because it is meaningless to infer something whose value depends on the failure of some other proof. Also, we can consider a short example where the use of  $\nabla$  is different form  $\neg$ .

$$\mathbf{op} \ move(Car) \wedge in\_way(Car, Street2) \wedge lateral\_way(Street, Street2) \rightarrow free\_way(Street)$$

In the previous rule if there are negative information the meaning change if we replace **op** with  $\neg$  or with  $\nabla$ . That is because it is not the same infer that a car is not moving because we just know that the car is not moving.

### 3.2 Extension 2: $EMTPL_{\nabla_c}$

The new operator could be used as the realization operand  $AT$  or like  $\diamond_c$  but for negative information. In this sense we can analyze what happens with an operator of this kind.

The syntax of the language is quite similar to the corresponding to  $EMTPL_{\nabla}$  the only modification is that we replace  $\nabla p$  by  $\nabla_c p$

Let see the semantic of  $\nabla_c$ , again we assume a temporal structure like EMTPL's one.

$$(\sigma, j) \models \nabla_c p \text{ iff } (\sigma, j + c) \models \neg p$$

With this semantic we do not manage negative information in the same way as in 3.1, because the manipulation of that kind of information is, in some way, virtual.

We can think in having both operators, but it seems to be not very attractive because at first sight we can think that:

$$\nabla_c p = \diamond_c \nabla p$$

Although it can be interesting to analyze what happens with this kind of operators more carefully.

## 4 Conclusions and future work

We show some ways of extend the language EMTPL to manage negative information. The idea behind the first one is to combine classical negation and negation as failure, so we can have negative information and also keep a two-valuate system. For this alternative we believe that can be interesting to find a way to make the operator  $\nabla$  more flexible, *i.e.*, analyze what happens with the system if we extend its application to any formula instead of only atomic ones. The idea of the second extension is to provide some kind of temporal negation, this can lead to some interesting operator.

In both cases it should be analyzed how this extensions affects the expressiveness of the language, and at the same time take a look to the underlying logic part. Another aspect to consider is how to introduce these operators in an interpreter of our language.

## References

- [CA99a] María Laura Cobo and Juan Carlos Augusto. EMTPL: A Programming Language for Temporal Deductive Data Bases. In *Proceedings de la XIX International Conference of the Chilean Computer Science Society*, pages 170–178, Talca, Chile, 1999.
- [CA99b] María Laura Cobo and Juan Carlos Augusto. Fundamentos lógicos e implementación de una extensión a temporal prolog. *The Journal of Computer Science and Technology (JCS&T), sponsored by ISTECS (Iberoamerican Science & Technology Education Consortium)*, 1(2):22–36, 1999.
- [CA00] María Laura Cobo and Juan Carlos Augusto. Towards a Programming Language Based on Prior's Metric Temporal Operators. In *Proceedings del VI Congreso Argentino de Cs. de la Computación, CACiC2000*, pages 453–464, 2000.
- [Gab87] Dov Gabbay. Modal and temporal logic programming. In Antony Galton, editor, *Temporal Logic and their Applications*, pages 197–236. Academic Press, 1987.
- [Pri67a] Arthur Prior. *Past, Present and Future*. Clarendon Press, 1967.
- [Pri67b] Arthur Prior. Stratified metric tense logic. *Theoria* 33, pages 28–38, 1967.
- [RU71] Nicholas Rescher and Alasdair Urquhart. *Temporal Logic*. Springer-Verlag, 1971.

## "Sistemas de Software Distribuido. Aplicaciones"

*De Giusti A.<sup>1</sup>, Pesado P.<sup>1</sup>, Bertone R.<sup>1</sup>, Boracchia M.<sup>1</sup>, Thomas P.<sup>1</sup>, Madoz C.<sup>1</sup>*

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>2</sup>.  
Facultad de Informática. Universidad Nacional de La Plata.*

### Objetivo

Realizar investigación y desarrollo en temas relacionados con Sistemas de Software Distribuido, particularmente los relacionados con los aspectos de Ingeniería de Software que se orientan al desarrollo e implementación de proyectos de sistemas distribuidos, incluyendo especialmente los temas de métricas y calidad de software.

Estas tareas abarcan aspectos propios de la arquitectura, comunicaciones, estructuración de bases de datos, así como especificación, validación y desarrollo de aplicaciones de software en ambientes distribuidos. Asimismo se considera especialmente la investigación de la Ingeniería de Software de los sistemas distribuidos de tiempo real, considerando las extensiones necesarias en las metodologías y herramientas clásicas con el objetivo de considerar las restricciones de tiempo.

### Introducción

Un sistema distribuido consiste en un conjunto de computadoras autónomas conectadas por una red y con soporte de software distribuido. Permite que las computadoras coordinen sus actividades y compartan los recursos de hardware, software y datos, de manera tal que el usuario percibe una única facilidad de cómputo integrada aunque esta pueda estar implementada por varias máquinas en distintas ubicaciones.

El desarrollo de sistemas distribuidos es una necesidad a partir de la utilización de redes de computadoras y de computadores personales de alta performance.

Las ventajas del procesamiento distribuido son:

- Mejora de la disponibilidad: la operación es factible en una configuración reducida cuando algunos nodos están temporalmente no disponibles. No hay un único punto de falla.
- Configuración más flexible: una aplicación puede configurarse de distintas maneras, seleccionando el número apropiado de nodos para una instancia dada.
- Control y administración más localizada: un subsistema distribuido, ejecutando en su propio nodo, puede diseñarse para ser autónomo, de modo que puede ejecutar en relativa independencia de otros subsistemas en otros nodos.
- Expansión incremental del sistema: si existe sobrecarga, el sistema puede expandirse agregando más nodos.
- Costo reducido: con frecuencia una solución distribuida es más barata que una centralizada.
- Balance de carga: en algunas aplicaciones la carga total del sistema puede ser compartida entre varios nodos.
- Mejora en el tiempo de respuesta: los usuarios locales en nodos locales pueden obtener respuestas más rápidas a sus requerimientos.

<sup>1</sup> Docentes-Investigadores LIDI – Facultad de Informática UNLP - E-Mail: [ppesado@lidi.info.unlp.edu.ar](mailto:ppesado@lidi.info.unlp.edu.ar)

<sup>2</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.  
TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

En particular un sistema distribuido de tiempo real debe interactuar con el mundo real, en puntos físicamente distantes y no necesariamente fijos, en períodos de tiempo que vienen determinados por el contexto o las restricciones de la especificación (en muchos casos a partir de una activación asincrónica).

La evolución tecnológica en el tratamiento de señales (locales o remotas) y en los sistemas de comunicaciones ha impulsado enormemente esta área temática, sobre todo en los aspectos de ingeniería de software tales como planificación, desarrollo y verificación de software para Sistemas Distribuidos. Es de fundamental importancia contar con herramientas CASE que asistan a lo largo de todo el ciclo de vida del proyecto.

La necesidad de supervisar el proceso de desarrollo del software torna imprescindible la utilización de métricas. La medición se puede aplicar al proceso del software con el intento de mejorarlo sobre una base continua. También se puede utilizar en el proyecto para ayudar en la estimación, el control de calidad, la evaluación de productividad y el control de proyectos. Finalmente el ingeniero de software puede utilizar la medición para ayudar a evaluar la calidad de productos de trabajos técnicos y para ayudar en la toma de decisiones tácticas a medida que el proyecto evoluciona. En muchos casos es necesario adaptar o ampliar las métricas existentes para poder utilizarlas en sistemas distribuidos, en particular de tiempo real.

Los conceptos de calidad que deben estar presentes en los procesos de desarrollo de software han llevado a crear modelos standarizados que permiten encuadrar a las organizaciones y a los proyectos. En particular existen dos normas ampliamente aceptadas que surgen de dos comunidades diferentes (CMM e ISO 900X).

La mejora continua en los procesos y productos de software es lo que se conoce como calidad total, que en gran parte es conseguida a través de la aplicación y evaluación de las métricas, y la realimentación del proceso productivo con estos resultados.

Desde el punto de vista de las aplicaciones se considerarán en particular las Bases de Datos Distribuidas y aspectos de Tecnología de Software aplicada a educación.

El modelo distribuido de datos hace posible la integración de BD heterogéneas proveyendo una independencia global del administrador de bases de datos (DBMS) respecto del esquema conceptual. Además, es posible implementar una integración tal que reúna varios modelos de datos, representando cada uno de ellos características propias de organizaciones diferentes, asociadas para un trabajo conjunto.

Este modelo de distribución normalmente tiene un soporte multiprocesador tipo MIMD sobre diferentes esquemas de comunicaciones. En tal arquitectura se reproducen los problemas de distribución óptima de datos y procesos, de migración de datos y procesos y de tolerancia a fallas.

En particular interesa estudiar el comportamiento (en tiempo de respuesta y confiabilidad) de grandes bases de datos sobre arquitecturas distribuidas de redes LAN y WAN. Esto significa problemas tales como: Tasa de pérdida de datos; Tiempo máximo necesario para recuperación de información; Complejidad y eficiencia de los algoritmos de recuperación; Tiempo de utilización de recursos del sistema; Incidencia del porcentaje de replicación en el tiempo de respuesta.

En este marco, una línea de investigación actual consiste en desarrollar ambientes de experimentación sobre redes de estaciones de trabajo (NOW) para estudiar grandes Bases de Datos distribuidas que deban ser accedidas en tiempo real (por ello puede requerirse tener un porcentaje de la BD on-line y otro off-line).

La Tecnología de Software aplicada en Educación, es una disciplina en pleno crecimiento, no sólo por el interés y múltiples aplicaciones que ha suscitado en las escuelas, universidades y centros de entrenamiento empresarial, sino porque ha permitido incorporar el tratamiento de la información al proceso educativo.

Esta incorporación de la informática al quehacer educativo involucra a las diferentes modalidades del proceso de enseñanza / aprendizaje - entrenamiento, esto es a distancia, semi presencial o presencial. El desarrollo de ambientes y herramientas con este objetivo (en particular los dedicados WEB based learning) es un tema muy importante en la Ingeniería de Software de Sistemas Distribuidos.

Es por esta razón que los recursos tecnológicos actuales se han constituido en temas de investigación y aplicación fundamentales en los desarrollos multidisciplinarios de Ciencias de la Educación y Ciencia Informática. Por lo tanto, la evolución del mundo educativo en cuanto a métodos y recursos tecnológicos requiere que los usuarios de dicha tecnología (docentes y alumnos) renueven y actualicen sus conocimientos de productos de software educativos y estén preparados para administrar esos nuevos productos. En este contexto la definición y utilización de métricas precisas para la evaluación de calidad de productos de software educativo (en particular en Educación no presencial) es una línea de I/D presente en el proyecto.

#### **Líneas de Estudio/Investigación y Desarrollo:**

- Procesamiento distribuido. Arquitectura, comunicaciones y software. Lenguajes y ambientes para procesamiento distribuido.
- Ingeniería de Software de sistemas distribuidos. Extensiones para el tratamiento de sistemas distribuidos de tiempo real.
- Metodologías de especificación, validación y desarrollo de sistemas de software distribuido. Herramientas CASE orientadas a sistemas de software distribuido.
- Métricas para la evaluación de software. Métricas para la estimación y seguimiento de proyectos. Normas de calidad en software. ISO 900X y CMM.
- Herramientas de integración y mantenimiento de proyectos distribuidos.
- Reingeniería de sistemas complejos que migran por down-sizing a esquemas cliente-servidor distribuidos.
- Bases de Datos distribuidas. Replicación. Consistencia. Migración de datos. Tolerancia a fallas.
- Tecnología de software aplicada en Educación. Educación a distancia. Educación basada en WEB. Métricas de calidad para software educativo.

## Bibliografía

ACM eLearn , ACM Journal of Educational Resources in Computing, ACM Transactions on Computer Systems ,ACM Transactions on Computer-Human Interaction, ACM Transactions on Database Systems ,ACM Transactions on Information Systems, ACM Transactions on Software Engineering and Methodology.

Andrews, "concurrent Programming", Benjamin/Cummings, 1991.

Arthur L. "Improving software quality" , Wiley 1993

Beizer B. "Software testing techniques" Internation Thompson Computer Press 1990

Bell, David; Grimson, Jane, "Distributed Database Systems",. Addison Wesley. 1992

Bobak, A, "Distributed and Multi-Databse Systems", Artech House, 1996

Buretta, M "Data Replication", John Wiley and Sons, 1997

Burleson, D, "Managing Distributed Databases, Building Bridges between Database Island", John Willey 1995.

Communications of the ACM

Coulouris G., Dollimore J., Kindberg T. "Distributed Systems Concepts and Design", Addison Wesley 1994

Date, C.J., "Introducción a los sistemas de Bases deDatos". Addison Wesley 1994.

De Marco."Controlling Software Projects". Yourdon,1998

Frakes, W, Baeza Yates, R "Information Retrieval. Data Structures & Algorithms", Prentice Hall 1992

Ellison K. "Developing real time embedded software" . Wiley 1994.

Hatley D., Pirbhai I., "Strategies for Real-Time System Specification", Dorset House, 1988.

Humphrey W.."Managing the software process". Addison Wesley, 1989.

Huns M, Singh, M Readings in Agents", Morgan Kaufmann Publishers, 1997.

IEEE Computer, IEEE Concurrency, IEEE Realiability and Manteniability Symposium, IEEE Sigact News, IEEE Software Engineering, IEEE Transactions on Parallel and Distributed Processing  
Jajodia S., Kerschberg L. " Advanced transaction models and architectures" Kluwer Academic Pub. 1997

Larson, J., "Database Directions. From relational to distributed, multimedia, and OO database Systems". Prentice Hall. 1995

Leben, M, "Client Server Databases Enterprise Computing", Prentice Hall, 1997.

Loucopoulos, P, Vassilio, K, "System Requirements Engineering" McGraw-Hill Book Company

Marques, P. "Metodología para la elaboración del software educativo en Software Educativo. Guía de uso y Metodología de diseño" , Barcelona 1995.

Moore J. "Software engineering standards" IEEE 2000

Özsu, T, Valduriez, P,"Principels of distributed database systems", Prentice Hall, 1999

Pfleeger S. "Ingeniería de Software. Teoría y Práctica". Addison Wesley 2000.

Pressman, R;"Ingeniería de Software, Un enfoque práctico." Cuarta edición. Mc Graw Hill. 1998

Robinson, W, Volkov, S, "Conflict-Oriented Requirements Restructuring" Departament of Computer Information Systems. Georgia State University.Atlanta. USA.

Seepold, R, Martinez Madrid, N, "Virtual Components Design and Reuse", Kluwer Academic Publishers - 2000.

Shumate K., "Software specification and design for real-time systems", Wiley 1992.

Silbershatz; F., "Fundamentos de las Bases de Datos".. Mc Graw Hill. 1998.

Wolf, W, "Computers As Components: Principles of Embedded Computing System Design", Morgan Kaufmann - 2000.

# Formal Specifications in Component-Based Development

Elsa Estévez    Pablo Fillottrani

Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Av. Alem 1253 – (8000) Bahía Blanca  
Argentina

e-mail: {ece,prf}@cs.uns.edu.ar

## 1 Introduction

Software engineering has entered a new era, the Internet and its associated technologies require a different conceptual framework for building and understanding software solutions. Users ask to develop applications more rapidly, and software engineers need to ensemble systems from preexisting parts. *Components* and *Components-Based Development* (CBD)[16, 15], are the approaches that provide solutions to these arising needs. Components are the way to encapsulate existing functionality, acquire third-party solutions, and build new services to support emerging business processes. Component-based development provides a design paradigm that is well suited to the new requirements, were the traditional *design and build* has been replaced by *select and integrate*. Within this approach, the specification of components plays a crucial role. If we are working on the development of components in order to construct a library for general use, we need to start from a concrete and complete specification of what we are going to construct. If we are assembling our application from pre-existing components, we need a precise specification of the behaviour of the component in order to select it from the library.

The specification is a statement of the requirements of a system. In general, we can view a specification as the statement of an agreement between the producer and a consumer of the service. In particular for software engineering, this agreement holds between the software engineer and the user. Depending on the context, these two roles may present important differences, so the nature of the specification is different. It is obvious we need to specify the system that is our final product, but also all the intermediate elements such as subsystems, components, modules, case tests, etc. Therefore, we can speak about *requirements specification* to represent the agreement between the user and the system developer, *design specification* in terms of the agreement between the system architect

and the implementers, *module specification* or *component specification* as the contract between the programmer using the module and the programmers who develops it. As it is shown, the term *specification* is used in different stages of system development. In all cases a specification at some level states the requirements for the implementation at a lower level, and we can view it as a definition of what the implementation must provide. The relation between the specification and the implementation is often explained in terms of *what* the system must provide and *how* the system will be implemented in order to provide the services. The specification must state *what* a component should do, while the implementer decides *how* to do it.

The specification activity is a critical part of the software production process. Specifications themselves are the result of a complex and creative activity, and they are subject to errors, just as are the products of other activities, like coding. As a result, we can write good specifications, or bad ones. In this sense, most of the qualities required for software products are required for specifications. The first qualities required are that they should be clear, unambiguous, and understandable. These properties are quite obvious, but sometimes specifications are written in natural language and usually hide subtle ambiguities, specially informal specifications. The second major quality required for specifications is consistency, meaning that it should not introduce contradictions. The third prime quality required for specifications is that they should be complete. Because of the difficulties in achieving complete specifications, the use of the incremental principle is essentially important in deriving specifications. That is, one may start with a fairly incomplete specification document and expand it through several steps.

There are many relevant techniques for writing specifications, so we classify them according to different specification styles. Ghezzi, Jazayeri and Mandrioli [2] classify them according to two different orthogonal criteria. Specifications can be stated *formally* or *informally*. Informal specifications are written in a natural language; they can, however, also make use of figures, tables, and other notations to help understanding. They can also be structured in a standardized way. When the notation gets a fully precise syntax and meaning, it becomes a formalism. In such a case, we talk about *formal specifications*. It is also useful to talk of *semiformal specifications*, since, in practice, we sometimes use a notation without insisting on a completely precise semantics.

## 2 Research Topics

On this ground, we restrict our research to the use of formal methods for the specification of software. By formal methods we mean a specification language plus formal reasoning, that includes the use of formalisms such as logic, discrete mathematics, finite state machines, and others. Formal specifications are expressed in languages whose syntax and semantics are formally defined, so they provide hierarchical decomposition and stepwise development. The main goals of using formal methods in requirements specification is to clarify customer's needs and to reveal ambiguity, inconsistency and incompleteness. By using formal specification for requirements, we can make a structural decomposition of the behaviour of the system by specifying the behaviour of each component. Furthermore, there is also the possibility of refinement by demonstrating that lower levels of abstraction

satisfies higher levels. Formal methods enables the verification of software by proving that an implementation satisfies its specification, and likewise it makes easier the validation facilitating testing and debugging. In this context we use *RAISE*, *Rigorous Approach to Industrial Software Engineering*, [3, 4] as the formal method to develop and to specify software. The *RAISE* Method provides a methodology to develop software, a formal specification language *RSL*, and automated tools for proofs and code generation.

Within the formal methods area, algebraic specification is one of the most extensively-developed approaches. The most fundamental assumption underlying algebraic specification is that programs are modelled as many-sorted algebras consisting of a collection of sets of values together with functions over those sets. The overall aim of work on algebraic specification is to provide semantic foundations for the development of software that is *correct* with respect to its requirements specifications. This means that the component developed must exhibit the required input/output behaviour. Algebraic specifications are used to construct software in a stepwise fashion, adding more details in each step of refinement [14, 8, 12].

As new paradigms appeared, different formalisms are used for the theoretical foundations. For example, the object oriented paradigm uses the mathematical concept of coalgebras to model the behaviour of classes [13, 8, 6, 7]. The concept of a class as a black box, that is fully encapsulated and cannot be examined directly, presents a typically co-algebraic view. In dealing with coalgebras, we use destructors to get a result or to change the state of an object. This is more suitable with the concept of encapsulation and information hiding, basis of the object-oriented paradigm. As another example, in component integration we need to abstract implementation details and concentrate only on behaviour aspects. The co-algebraic view is also suitable for modeling the component as a black box.

On our research we describe software components as class expressions in *RAISE*. Each class consists of the state, operations for reading and writing this state and the axioms to relate such operations. On the one hand a class represents a data structure, giving rise to the definition of an abstract data type. This presents a typically algebraic view which allows us to construct and manipulate particular values of the state [14]. On the other hand, a class represents dynamic behaviour: how the component generated from this class can interact with its environment, how its operations invoked externally to produce observable attributes of the state, new states (which attributes we can investigate again) or both. Our studies are based on these two complementary approaches; the algebraic view is more adequate when we need to construct a class from its specification (by refinement) and the co-algebraic view is more suitable for reuse and composition of class expressions. In the latter we require definition of behavioural abstraction which is not supported by refinement. For this purpose we propose to use bisimulations [11, 9]. Bisimulation is a relation between the states of two dynamic systems representing that the two systems cannot be distinguished by interacting with them. It is a central concept in concurrency theory, and it was also considered on the grounds of logic [5], category theory [1], games [10] and co-algebras [8]. We study the application of bisimulation in the framework of *RAISE*, an industrial-strength formal method, in the role of behavioural abstraction for building software from pre-existing components. The goal is to prove how bisimulation ignores what is inessential from the point of view of behaviour, such as differences in operation definitions, data

representations, implementation structures and specification styles. We can thus establish a criteria for comparing components from the behavioural equivalence view.

## References

- [1] A. JOYAL, M. N., AND WINSKEL, G. Bisimulation for Open Maps. *Information and Computation* 127 (1996), 164–185.
- [2] CARLO GHEZZI, MEHDI JAZAYERI, D. M. *Fundamentals of Software Engineering*. Prentice-Hall International, 1991.
- [3] GROUP, T. R. M. *The RAISE Specification Language*. Pr. Hall, 1992.
- [4] GROUP, T. R. M. *The RAISE Development Method*. Prentice Hall, 1995.
- [5] HENNESSY, M., AND MILNER, R. Algebraic Laws for Nondeterminism and Concurrency. *Journal of the ACM* 32 (1985), 137–161.
- [6] JACOBS, B. Coalgebraic Reasoning about Classes in Object-Oriented Languages. *Electronic Notes in Theoretical Computer Science* 11 (1998), 1–12.
- [7] JACOBS, B. Coalgebras in specification and verification for object oriented languages. *Newsletter of the Dutch Association for Theoretical Computer Science* 3 (1999).
- [8] JACOBS, B., AND RUTTEN, J. A tutorial on (co)algebras and (co)induction. *EATCS Bulletin* 62 (1997), 222–259.
- [9] MILNER, R. A Calculus of Communicating Systems. *LNCS 92* (1980).
- [10] NIELSEN, M., AND CLAUSEN, C. Bisimulation, Games and Logic. In *CONCUR94* (1994), vol. 836 of *LNCS*, Springer-Verlag, pp. 385–400.
- [11] PARK, D. Concurrency and automata on infinite sequences. *LNCS 104* (81).
- [12] RAZVAN DIACONESCU, K. F. CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification. *World Scientific, 1998. AMAST Series in Computing, 6* (1998).
- [13] REICHEL, H. An approach to object semantics based on terminal co-algebras. *Mathematical Structures in Computer Science* 5 (1995), 129–152.
- [14] SANNELLA, D., AND TARLECKI, A. Essential Concepts of Algebraic Specification and Program Development. *Formal Aspects of Computing* 9(3) (1997), 229–269.
- [15] SZYPERSKI, C. *Component Software Beyond Object-Oriented Programming*. Addison Wesley Longman Limited, 1998.
- [16] W.BROWN, A. *Large-Scale Component-Based Development*. Prentice Hall International, 2000.

## **SOL: Un lenguaje para el programador**

**Rafael O. Fontao, Claudio Delrieux, Guillermo Kalocai, Gustavo Goñi y Gustavo Ramoscelli**

**Departamento de Ingeniería Eléctrica**

**Universidad Nacional del Sur**

**e-mail: fontao@uns.edu.ar**

### **Resumen**

En este trabajo se propone llevar a la práctica, a la luz de nuevas tecnologías informáticas, un ambiente de concepción de programas cuyos orígenes se toman de un trabajo publicado por el director del proyecto (1978) y desarrollado a través de diferentes implementaciones y aplicaciones de la metodología a lo largo de los últimos años.

La idea central consiste en brindar un lenguaje para el programador, que refleje la metodología de concepción de ideas, más que un lenguaje estándar para programar donde finalmente será escrito el problema a resolver.

En este modelo la estructura de control de un programa se define separadamente del resto de las instrucciones del lenguaje de programación, y es modelada a su vez por el comportamiento de una jerarquía de autómatas finitos.

El enunciado inicial de un problema puede concebirse como un autómata de un solo estado. A cada estado podrá corresponderle hasta dos próximos estados acorde a la evaluación lógica de una condición ( si existe ) al final de su tarea.

A partir de aquí y en forma recursiva se analiza si un estado particular de un autómata puede ser sintetizado directamente por el lenguaje de programación disponible. En tal caso la descomposición se detiene, sino el comportamiento del estado particular se descompone en un nuevo autómata en un nivel inferior. Y así sucesivamente en un número finito de pasos se arribará a que todos los estados del autómata ( jerarquía de autómatas ) pueden ser sintetizados "razonablemente bien" por instrucciones del lenguaje de programación disponible.

La idea de concebir programas mediante este lenguaje ( o metodología ) resulta de interés como medio unificado de programación, ya que la descripción de la solución de un problema puede plantearse con cierto grado

de independencia del lenguaje final a utilizar ( paradigma imperativo ) y por consiguiente puede permitir la reutilización y portabilidad de programas más eficientemente.

## Introducción

La enseñanza clásica de programación se realiza mediante la utilización de un lenguaje práctico disponible ( Ej. Pascal ) a problemas en un contexto académico limitado y aplicando metodologías de programación ( típicamente Programación Estructurada ) de modo tal de hacer que los programas sean eficientes, claros de interpretar, mantener y por supuesto que hagan o que se espera que hagan.

En este sentido, entre la escritura de las especificaciones y el listado del programa en que finalmente será ejecutado para cumplirlas, tradicionalmente no hay más que el relato de buenas intenciones y comentarios adecuados en el texto del programa a modo de documentación, pero de la forma de cómo el programador fue creando o concibiendo la solución de las especificaciones la mayoría de las veces quedan pocos vestigios. Los pasos intermedios en la concepción de un programa ceden su lugar al listado de la solución final y su historia suele desaparecer por completo.

## Un lenguaje para el programador

El lenguaje **SOL** ( por las siglas en Inglés de **Structured Oriented Lenguaje** ) trata de llenar este hueco de historiar la concepción de un programa partiendo de la idea básica de que todo comportamiento secuencial puede modelarse por medio de un autómata finito.

La mera observación de cómo se realiza un trabajo, cualquiera sea su naturaleza, puede concebirse como una secuencia de aplicaciones de herramientas de trabajo, seguida de una herramienta de medida que chequea una condición lógica. Eventualmente la secuencia puede ser vacía y la condición lógica una tautología ( es decir, puede obviarse si siempre es verdadera ). Y a esta secuencia seguida de una condición, la bautizaremos con el nombre de **trabajo elemental**. Nótese que la noción de elemental corresponde al tipo de herramienta usada y no a la tarea en sí. Por ejemplo, si una herramienta es poderosa hacer una tarea con ella puede resultar elemental pero no si se realiza con herramientas más primitivas.

El nexo entre **estado** de un autómata finito y estado de la programación se dá precisamente en la noción de **trabajo elemental**.

## **El Modelo de descripción de un programa como Autómata finito**

Nuestra experiencia docente nos indica que modelar un comportamiento secuencial por medio de autómatas finitos es una noción muy bien aceptada no solo por estudiantes de sistemas digitales sino también por otras disciplinas.

Este hecho está soportado por los trabajos de estudiantes donde autómatas relativamente complejos son diseñados e implementado eficazmente por circuitos digitales.

El comportamiento de un autómata finito ( análogamente máquina secuencial ) se especifica por medio de una tabla en la cual las filas son los estados ( que indican alguna acción a tomar ) y las columnas son los estímulos o entradas al sistema. Cada entrada a esta tabla representa el próximo estado que seguirá el autómata dependiendo del estímulo o entrada. Hay un estado inicial y uno o más estados finales

Para explorar las ventajas de modelar comportamiento secuencial mediante la concepción de autómatas finitos, se propone al siguiente esquema:

Sea L un lenguaje de escritura natural para expresar ideas.

- i) Hay un estado inicial.
- ii) Cada estado se representa por un programa elemental escrito en L. Es decir, una secuencia de aplicaciones de herramientas de trabajo seguida de una aplicación de medida o testeo.
- iii) Hay dos entradas ( columnas ) que denominamos SI y NO ( True and False ) que representan los valores lógicos de la condición al final de cada programa elemental ( por razones de simplicidad y sin perder generalidad, podemos considerar solo una condición lógica ).

El estado actual es el que se está ejecutando y el próximo estado será determinado por el valor lógico de la condición al final del programa elemental.

## Sintaxis de SOL

La definición del lenguaje SOL por producciones BNF es la siguiente:

FSA quiere decir: **Autómata de Estados Finito**.

LPD reemplaza a **Lenguaje Práctico Disponible**

**<program> ::= <control structure> <actions description>**

**<control structure> ::= <state refinement> FIN | <state refinement> <control structure>**

**<state refinement> ::= REF <depth identifier> <FSA definition> END <depth identifier>**

**<depth identifier> ::= null | <identifier>**

**<identifier> ::= <positive integer > | <positive integer> . <identifier>**

**<FSA definition> ::= <state description> | <state description> <FSA definition>**

**<state description> ::= <positive integer> DO <task description> THEN <next state list>**

**<task description> ::= es una descripción escrita del propósito de la tarea elemental**

**<next state list> ::= <next state identifier> | <next state identifier> OR <next state list>**

**<next state identifier> ::= <positive integer> | EXIT <positive integer> | STOP**

**<positive integer> ::= <digit> | <digit> <positive integer>**

**<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9**

**<actions description> ::= DEC <data declarations> < actions list >**

**<data declarations> ::= etapa de declaraciones ( si es que existen ) requeridas por el LPD.**

**<actions list> ::= <action> FIN | <action> <action list>**

**<action> ::= ACT <identifier> <output state> <exit condition>**

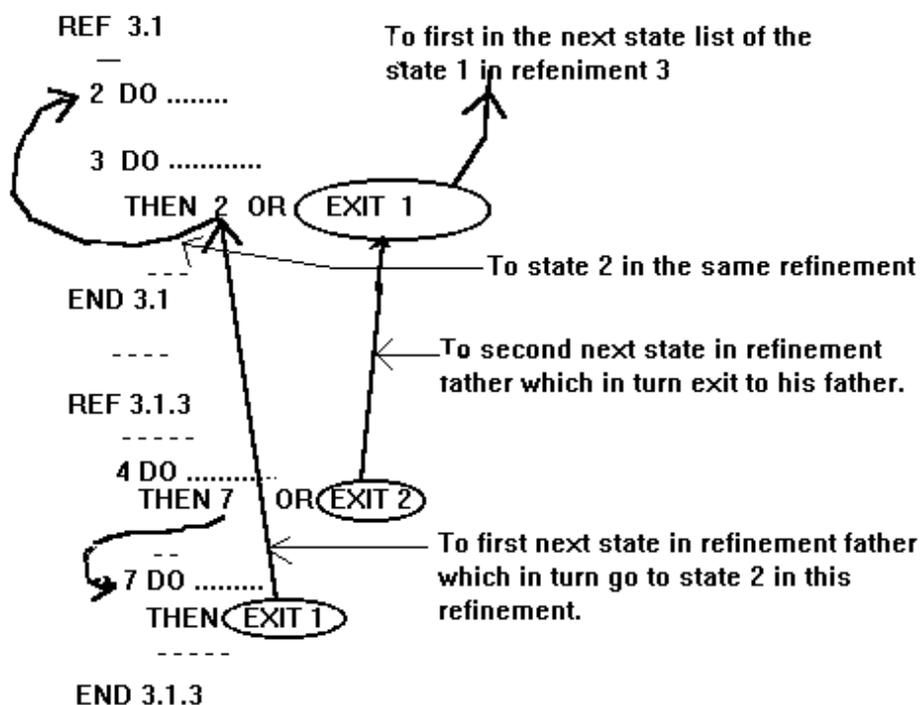
**<output state> ::= sentencias ( en LPD ) que sintetizan la salida del estado**

**<exit condition> ::= NEXT | NEXT ( <logic condition> )**

**<logic condition> ::= expresión lógica permitida en el LPD.**

## Semántica de SOL

Como se establece en sección anterior, la estructura de control se define separadamente del resto de las sentencias que conforman un programa. La estructura de control a su vez, se compone de una sucesión de refinamientos. La definición de una FSA se delimita por las palabras REF y END seguidas de un identificador formado por el identificador que refina, seguido de punto y el entero que identifica al estado que está siendo refinado. Ver la siguiente figura.



## El proceso de refinamientos

Los principios de la programación estructurada establecen que un programa debe concebirse en sucesivos refinamientos: En cada paso o refinamiento un módulo o parte del programa se refina explicitando su comportamiento en mayor detalle con las propias instrucciones del lenguaje. El proceso de refinamiento concluye hasta que prácticamente todo el programa esté escrito lo suficientemente claro en el lenguaje de programación

En forma análoga el proceso de concepción de un programa puede hacerse mediante autómatas finitos de la siguiente manera:

Sea LPD un lenguaje práctico disponible donde finalmente será escrito el programa.

**Paso 0:**

Elija un lenguaje de escritura L para expresar sus ideas y conciba a todo el enunciado del problema como si fuera un FSA de un solo estado y una sola entrada ( SI ).

**Paso 1 a N:**

Mientras que exista algún estado que no pueda ser escrito "razonablemente claro" en el LPD, concíballo expresado en lenguaje L como un nuevo autómata.

Por consiguiente, un estado de un autómata es reemplazado por un nuevo autómata en un nivel inferior. Como regla práctica se aconseja que el nuevo autómata tenga pocos estados. Claro está que al menos deben ser 2 nuevos estados ya que de otro modo no se estaría refinando nada. Solo sería establecer un mero sinónimo o re expresión de lo mismo con otras palabras.

**Paso N+1:**

A esta altura todos los estados podría expresarse claramente en el LPD y en tal caso el paso último es precisamente escribir el código en LPD para todos los estados.

**Características de SOL**

La implementación actual se está llevando en DELPHI y el primer pre compilador será destinado a Pascal por ser este lenguaje el utilizado en los cursos de programación.

La metodología de concebir mediante SOL permite aventurarnos en áreas como la portabilidad y reutilización de programas ya escritos en lenguajes convencionales y traducirlos, al menos parcialmente, en otros lenguajes disponibles.

**Este trabajo de desarrollo e investigación es parte de un PGI de la Universidad Nacional del Sur**

# A Framework for Defining and Checking Constraints in Requirement Gathering and Definition

Rodolfo Gómez and Pablo Fillottrani

Departamento de Ciencias e Ingeniería de la Computación

Universidad Nacional del Sur, Bahía Blanca, Argentina

[rgomez, prf]@cs.uns.edu.ar

## 1 Introduction

Requirements capture user needs about a system [KG99]. They stand for those functional and nonfunctional attributes the system must possess to be considered correct. Requirements also dictate how the system should respond to user interaction. Therefore, they play a very important role in software development. They state what the system should do for the user to be satisfied, and therefore they lead all development stages in the lifecycle. They are used to communicate user needs to all people involved in the development process, the system architecture is built upon them and the testing stage uses them as satisfaction criteria.

*Requirement gathering* is the activity of bringing requirements together, *requirement definition* is the activity of documenting requirements and organizing them into something understandable and meaningful, i.e. a *requirement specification*. Unfortunately these tasks are quite difficult. Users usually cannot precisely what they want from the system. Usually there are more than one user of the system and many conflictive views appear about what the system should do. Frequently, even if the users agree in system functionality, they are not able to see the “big picture” and the specification is incomplete. It is also possible for requirements to be stated by someone who’s not the real user of the system, for example company managers. This may result in requirements that express interactions that do not add real value to users.

Another problem with requirements is that they evolve while users are learning about their needs. Hence, requirements suffer a lot of changes until they can be precisely defined. Also, requirement gathering itself can be a difficult task if users do not have enough time to be properly interviewed, or they are reluctant to cooperate in this process. Finally, requirement specification itself may be cumbersome as it often has no more structure than a simple list. Because this list tries to capture every requirement, including those which add no value to the user real needs, its volume becomes unmanageable by subsequent development stages (e.g. design and testing).

All difficulties just mentioned result in a poor quality requirement specification, with redundant, inconsistent and even incomplete requirements. Recently, UML’s use cases and use diagrams have been adopted as a successful methodology in requirement gathering and definition [KG99]. They have helped to solve the problems we have previously mentioned and are still

useful to drive the entire software development process [IJR99b].

Despite the many advantages use cases offer in requirement definition, we feel this process can be improved by letting requirement redundancy, inconsistency and incompleteness to be automatically controlled as much as possible. The proposed framework should benefit from use-cases structure for discovering patterns expressing redundancy, inconsistency and incompleteness. Also, the framework will be a tool for analysts as it will allow the evolution of requirement definition to be automatically controlled and verified.

## 2 Use cases in Requirement Gathering and Definition

Use cases [IJR99a] are text descriptions of the interactions between some outside actors and the computer system. An actor represents a coherent set of roles that users of use cases play when interacting with these use cases. Typically, an actor represents a role that a human, a hardware device, or even another system plays with a system. Use case diagrams are graphical depictions of the relationships between actors and use cases and between two use cases.

Use cases play an important role in requirement gathering and definition. The interactions that use cases illustrate form the basis of most of the requirements that must be documented. They are effective communication vehicles between analysts and users. Generally, functional requirements can be put into terms of interaction between the actors and the application. Non-functional requirements, such as performance, extensibility and maintainability, can often be stated in terms of use case stereotypes [IJR99a]. Use cases also provides requirement traceability through the lifecycle because they are a building block for system design, units of work, construction iterations, test cases and delivery stages. Finally, use cases discourage premature design because these faults become quite evident when appearing into the use case structure.

Briefly, a use-case driven approach to requirement gathering proceeds as follows [KG99]. Use cases go through a series of refinement iterations until they are complete. Those that model the most valuable functions for the user are identified first. These cases are usually incomplete as they only express brief descriptions of particular interactions. As they are refined, they are added basic course of events, alternative course of events, scenarios, preconditions, postconditions, etc. Last steps of refinement have to do with use case diagrams as a whole, where opportunities for merging use cases with similar functionality, reusing and generalization are looked for.

## 3 Improving requirement gathering and definition

Requirement gathering still relies on the analyst abilities for discovering inconsistencies, redundancy and incompleteness in requirements. Of course, all of these problems have become much more tractable by use cases, but we think there are some ways in which the process can be improved. The goal of our research is to define a framework based on use cases which allow the requirement specification evolution to be automatically controlled and verified.

Use cases become more detailed through refinement iterations. The next iteration completes or corrects the previous one. However, it would be useful for the analyst to define controls to be

carried on while the use case evolves.

Advantages appear when the analyst wishes a given use case diagram semantics to be an invariant with respect to subsequent iterations. These invariants include, for example, actors e.g., in the form of *“this actor cannot perform operation A before operation B has been completed”*, use case relationships e.g., in the form of *“actors can never interact with use case A before they have finished with use case B”*, politics (or business rules as called in [KG99]) e.g., in the form of *“use case diagrams should never contain more than 5 actors and 10 use cases, otherwise they must be decomposed”* and use case structure e.g., in the form of *“all preconditions stated for a use case must be provided as postconditions of other previously-performed use cases”*. Other benefits become clear when analysts postpone some details for future iterations, but they know enough about these details to be documented in the current iteration. For example, the analyst could adorn the use case diagram with a comment expressing that a new actor should be added in the next iteration.

We are currently developing a framework that help analysts to automatically define and maintain the sort of controls they could be interested on during the requirement gathering and definition activities. Roughly, we can compare this framework with a CASE tool in the sense that it will allow the visualization, edition and maintenance of requirement gathering deliverables such as use cases and use case diagrams. Beside that, the focus of our research is concerned with other, paramount features. We think this framework should provide a language which has to be expressive enough to define useful controls while at the same time be capable of automatic verification. For instance, a possible research line may consider executable logics as definition languages and model checkers as verification procedures (see e.g. [BBC<sup>+</sup>99] and [Hol97]).

As an overview of the framework possibilities, consider the following. Firstly, the framework could provide controls over invariants, alerting the analyst if they are not met in new iterations. For example, a previously defined actor could interact with a different use case by adopting a semantics that is inconsistent with that it was originally defined for. Secondly, the framework could remind the analyst about postponed details, such as refining a new actor in the next use case iteration. Finally, it could identify opportunities for reducing redundancy in requirements by finding flows of events which are common to various use cases. Therefore, it would be possible for the analysts to merge the cases or derive a new use case with common functionality.

We are warned about the existent of related work in the area of formal specifications. However, these works seem not to be at the level of abstraction which is present in requirement gathering. For example, RSL, the formal specification language of RAISE (Rigorous Approach to Industrial Software Engineering) [GRO92], [GRO96] is useful for modeling design specifications, i.e. a contract between the system architect and the implementors or for modeling component specifications, i.e. a contract between the user of the program module and the module developer. The major difference between the requirement gathering process our research is concerned about and the process addressed by RSL has to do with inputs. Requirements addressed by use cases come directly from users, so they are much more general, abstract and unstructured than those addressed by RSL, where refinement activities such as analysis and design have transformed these requirements in documents which are more structured, specific and complete. Indeed, we expect our framework to be capable of managing incomplete specifications.

Other work that is worthy to be mentioned is OCL (Object Constraint Language) [WK99]. This language is used to define constraints over both elements of UML models and elements

of the UML metamodel. We still have to investigate if this language can be directly used or otherwise be extended as a mean to specify those controls we are concerned about in requirement gathering. For example, it is possible that preconditions and postconditions for use cases could be specified in OCL to a certain extent, but it is likely OCL does not provide the means for managing common flows of events among use cases.

Our research, then, is currently focused on discovering what kind of controls are useful for the analyst when designing use cases. Then we have find to how these controls are related to use case diagrams. Once these steps are finished, it will be possible to define the expressiveness required for defining and maintaining the controls.

## References

- [BBC<sup>+</sup>99] N. Bjorner, A. Browne, M. Colon, B. Finkbeiner, Z. Manna, B. Sipma, and T. Uribe. Verifying Temporal Properties of Reactive Systems: A STeP Tutorial. *Formal Methods in System Design*, 1999.
- [GRO92] GROUP, T. R. M. *The RAISE Specification Language*. Prentice-Hall, 1992.
- [GRO96] GROUP, T. R. M. *The RAISE Development Method*. Prentice-Hall, 1996.
- [Hol97] Gerard Holzmann. The Model Checker SPIN. *IEEE Transactions on Software Engineering*, 23(5), 1997.
- [IJR99a] Grady Booch Ivar Jacobson and James Rumbaugh. *The Unified Modeling Language User Guide*. ACM Press, Addison-Wesley, 1999.
- [IJR99b] Grady Booch Ivar Jacobson and James Rumbaugh. *The Unified Software Development Process*. ACM Press, Addison-Wesley, 1999.
- [KG99] Daryl Kulak and Eamonn Guiney. *Use Cases, Requirements in Context*. ACM Press, Addison-Wesley, 1999.
- [WK99] Jos Warmer and Anneke Kleppe. *The Object Constraint Language: Precise Modeling with UML*. Addison-Wesley, 1999.

## *Búsquedas en Espacios Métricos*

**Norma Herrera ; Nora Reyes** - {nherrera,nreyes}@unsl.edu.ar  
Departamento de Informática - Universidad Nacional de San Luis

**Ricardo Baeza-Yates ; Gonzalo Navarro** - {rbaeza,gnavarro}@dcc.uchile.cl  
Departamento de Ciencias de la Computación - Universidad de Chile

**Edgar Chávez** - elchavez@fismat.umich.mx  
Escuela de Ciencias Físico-Matemáticas - Universidad Michoacana, México

### 1 Introducción y motivación

La búsqueda es un problema fundamental en Ciencias de la Computación, presente virtualmente en cada aplicación. Las bases de datos tradicionales se construyen basándose en el concepto de búsqueda exacta. Las consultas a la base de datos retornan todos aquellos registros cuyas claves coinciden con la aportada en la búsqueda. Las búsquedas más sofisticadas como búsqueda de rangos sobre claves numéricas, o búsqueda de prefijos sobre claves alfabéticas, se basan en la existencia de un orden lineal sobre las claves de búsqueda.

Actualmente las bases de datos han incluido la capacidad de almacenar nuevos tipos de datos tales como imágenes, sonido, video, etc. Estructurar este tipo de datos en registros, para adecuarlos al concepto tradicional de búsqueda exacta, es sumamente difícil y hasta imposible en algunos casos. Aún cuando pudiera hacerse las consultas a la base serán por objetos similares a uno dado, y no por aquellos exactamente iguales. Este tipo de búsqueda se conoce con el nombre de *búsqueda aproximada o búsqueda por similitud*, y surge en áreas tales como reconocimiento de voz, reconocimiento de imágenes, etc.

La necesidad de una respuesta rápida y adecuada, y un eficiente uso de memoria, hace necesaria la existencia de estructuras de datos especializadas que incluyan estos aspectos.

El planteo general del problema es: dado un conjunto  $\mathcal{S} \subseteq \mathcal{U}$ , recuperar los elementos de  $\mathcal{S}$  que sean similares a uno dado, donde la similitud entre elementos es modelada mediante una función de distancia positiva  $d$ .  $\mathcal{U}$  denota el universo de objetos válidos y  $\mathcal{S}$ , un subconjunto finito de  $\mathcal{U}$ , denota la base de datos en donde buscamos. El par  $(\mathcal{U}, d)$  es llamado *espacio métrico*. La función  $d : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}^+$  cumple con las propiedades propias de una función de distancia:

1.  $d(x, y) = 0 \Leftrightarrow x = y$  (positividad estricta)
2.  $d(x, y) = d(y, x)$  (simetría)
3.  $d(x, y) \leq d(x, z) + d(z, y)$  (desigualdad triangular)

Básicamente, existen tres tipos de búsquedas de interés en espacios métricos:

**Búsqueda por rango:** recuperar todos los elementos de  $\mathcal{S}$  que están a distancia  $r$  de un elemento  $q$  dado.

**Búsqueda del vecino más cercano:** dado  $q$ , recupera el (o los) elemento(s) más cercano(s) a  $q$  en  $\mathcal{S}$ .

**Búsqueda de los  $k$  vecinos más cercanos:** dado  $q$ , recuperar los  $k$  elementos más cercanos a  $q$

En muchas aplicaciones, la evaluación de la función  $d$ , suele ser una operación costosa. Es por esta razón, que se usa como medida de complejidad, la cantidad de evaluaciones de distancias hechas para resolver la búsqueda.

Un caso particular de espacios métricos son los espacios vectoriales  $K$ -dimensionales, con las funciones de distancia  $L_p$  ( $p = 1, 2, \dots, \infty$ ). Para estos casos existen soluciones bien conocidas. Sin embargo, éste no es el caso general.

Las investigaciones en la actualidad tienden al estudio de algoritmos en espacios métricos generales, donde existen varias técnicas conocidas para resolver el problema en un número sublineal de cálculos de distancia, con la condición del preprocesamiento de  $S$ . En general las estructuras sobre las que trabajan dichos algoritmos suponen que los espacios son estáticos.

Nuestro objetivo es estudiar principalmente dos estructuras: *Geometric Near-neighbor Access Tree (GNAT)* y *Spatial Approximation Tree (SAT)*, a fin de optimizar tanto su construcción, como el proceso de búsqueda.

En el caso particular del *SAT*, nos hemos centrado en lograr a partir de él, una estructura totalmente dinámica. Y en el caso del *GNAT*, la atención ha sido puesta en buscar alternativas para su construcción, de forma tal que el árbol resultante tenga un mejor comportamiento en tiempo de búsqueda.

## 2 Árbol de aproximación Espacial (*SAT*)

Para mostrar la idea general de la aproximación espacial se utilizan las *búsquedas del vecino más cercano*.

En este modelo, dado un punto  $q \in \mathcal{U}$  y estando posicionado en algún elemento  $a \in S$  el objetivo es moverse a otro elemento de  $S$  que esté más cerca “espacialmente” de  $q$  que  $a$ . Cuando no es posible realizar más este movimiento, se está posicionado en el elemento más cercano a  $q$  de  $S$ .

Estas aproximaciones son efectuadas sólo vía los “vecinos”. Cada elemento  $a \in S$  tiene un conjunto de vecinos  $N(a)$ . La estructura natural para representar esto es un grafo dirigido. Los nodos son los elementos del conjunto y los elementos vecinos son conectados por un arco. La búsqueda procede sobre tal grafo simplemente posicionándose sobre un nodo arbitrario  $a$  y considerando todos sus vecinos. Si ningún nodo está más cerca de  $q$  que  $a$ , entonces se responde a  $a$  como el vecino más cercano a  $q$ . En otro caso, se selecciona algún vecino  $b$  de  $a$  que está más cerca de  $q$  que  $a$  y se mueve a  $b$ . Para obtener el *SAT* se simplifica la idea general comenzando la búsqueda en un nodo fijo y realmente se obtiene un árbol. El *SAT* está definido recursivamente; la propiedad que cumple la raíz  $a$  (y a su vez cada uno de los siguientes nodos) es que los hijos están más cerca de la raíz que de cualquier otro punto de  $S$ . En otras palabras:  $\forall x \in S, x \in N(a) \Leftrightarrow \forall y \in N(a) - \{x\}, d(x, y) > d(x, a)$ . La construcción del árbol se hace de manera recursiva pudiendo construirlo siguiendo una de dos estrategias posibles: best-fit y first-fit, de acuerdo a si cada elemento  $b$  que no está en  $a \cup N(a)$  se ubicará en el subárbol del vecino más cercano a él de  $N(a)$  o en el primero que sea más cercano a él que  $a$ .

De la definición se observa que se necesitan de antemano todos los elementos de la base de datos para la construcción. Nuestro objetivo es estudiar si es posible que dicha estructura admita la inserción y eliminación de elementos, sin degradar los tiempos de búsqueda.

### 2.1 Inserciones y eliminaciones

Luego de haber analizado distintas formas de realizar inserciones de elementos en el árbol [9], hemos obtenido un nuevo método que permite realizar la construcción incremental de *SAT* a un costo muy inferior a la versión estática y que mantiene el costo de las búsquedas en la mayoría de los casos menor, y en otros levemente mayor que la versión original.

El método obtenido ha surgido como una combinación muy adecuada de las bondades de los dos métodos que demostraron ser mejores entre los previamente analizados (Timestamp e Inserción en la Fringe) y los supera a ambos. Esta nueva técnica construye un árbol con *aridad acotada* y guarda en cada nodo un *timestamp* que marca el tiempo en que se insertó dicho elemento en el árbol.

Al incorporar un nuevo elemento  $x$ , éste se inserta en el punto más apropiado, si es que no se supera la cota de la aridad; en otro caso se continúa la búsqueda del lugar de inserción a partir del vecino más cercano. Sea  $b$  el

punto resultante para la inserción, luego  $x$  se agrega como el vecino *más nuevo* de  $b$  y no se efectúa ningún tipo de reconstrucción.

Ahora claramente, no se fuerza a un elemento a aparecer en la vecindad del más cercano. En tiempo de búsqueda alcanzaremos al elemento insertado porque los procesos de inserción y búsqueda son similares. Las búsquedas se pueden optimizar haciendo uso del *timestamp*.

Las eliminaciones son mucho más complicadas porque los cambios a realizar en la estructura son más costosos, y no son localizados, sino que se pueden propagar al resto de la estructura.

Se han analizado distintas opciones para eliminar elementos, desde esquemas muy simples a otros más sofisticados. Finalmente, el método que resultó más adecuado reinserta en el árbol los vecinos de un nodo eliminado (y los subárboles que cuelgan de ellos). En la reinsertación, se trata de reinsertar subárboles completos para disminuir el número de evaluaciones de distancia realizadas. También se utiliza aquí el *timestamp* para evitar ciertas evaluaciones de distancia ya realizadas cuando se insertó el elemento.

### 3 Geometric Near-neighbor Access Tree

Esta estructura fue presentada por Serge Brin, en el año 1995[2]. El objetivo es que la estructura de datos actúe como un modelo geoméricamente jerárquico de los mismos. Esto se logra construyendo una jerarquía basada en *Diagramas de Voronoi*

La construcción del *GNAT* procede de la siguiente manera: en el primer nivel se seleccionan  $m$  pivotes  $c_1, c_2, \dots, c_m$  de  $\mathcal{S}$ , que se almacenan en el nodo raíz. A cada pivote  $c_i$  se le asocia el conjunto  $\mathcal{S}_{c_i}$  definido como:  $\mathcal{S}_{c_i} = \{x \in \mathcal{S} / d(c_i, x) < d(c_j, x), \forall j = 1 \dots m\}$ . Con cada  $\mathcal{S}_{c_i}$  se construye recursivamente un *GNAT*.

En cada nodo del *GNAT* se almacena además una tabla, a la que llamaremos *rango*, de tamaño  $O(m^2)$  que mantiene información sobre las distancias mínimas y máximas, desde el centro  $c_i$  al conjunto  $\mathcal{S}_{c_j}$ ; esto es:  $rango_{i,j} = [\min_{x \in \mathcal{S}_{c_j}} d(c_i, x), \max_{x \in \mathcal{S}_{c_j}} d(c_i, x)]$ , con  $i, j = 1, \dots, m$ . Esta información, junto con la desigualdad triangular, se usa en el momento de la búsqueda para descartar subárboles. En tiempo de búsqueda, el punto de query  $q$  se compara con algún pivote  $c_i$ , y se descarta cualquier otro pivote  $c_j$  tal que  $d(q, c_i) \pm r$  no interseca  $rango_{i,j}$ .

En la actualidad, nuestro trabajo se ha concentrado en conseguir técnicas de selección de pivotes, que tengan un buen comportamiento cuando se las aplica a un diccionario de palabras como espacio métrico .

#### 3.1 Detección de cluster

Descubrir la estructura subyacente del conjunto de datos es sumamente útil en el diseño de algoritmos de indexación. En nuestro caso particular, saber cómo se agrupan los elementos del espacio métrico nos servirá para identificar puntos, que al ser elegidos como pivotes, producirán una partición razonable del espacio.

Los algoritmos para detección de clusters en espacios métricos, usualmente son de complejidad cuadrática, dado que tiene que considerar todas las posibles relaciones entre pares de objetos de la muestra. En [4] se presenta un algoritmo de análisis de cluster, que tiene una complejidad subcuadrática, medida en términos de cantidad de evaluaciones de la función de distancia  $d$ . Para lograr esta complejidad, el método presentado se basa en la existencia de un índice sobre el espacio métrico.

La idea central es usar búsquedas por rango para la detección de cluster. Para ello se define el grafo de rango  $r$  (*Range r Graph*). En este grafo, se utilizan los elementos del espacio métrico como nodos, y se conecta el nodo  $s_i$  con el nodo  $s_j$ , si  $d(s_i, s_j) \leq r$ . Luego, cada componente conectada de este grafo forma un  $r$  - cluster.

Un aspecto importante, es que los puntos que están alejados simultáneamente de todos los demás (*outliers*) son aislados en este proceso (clusters de cardinalidad 1). Los *outliers* generalmente degradan el comportamiento de las estructuras; por consiguiente, es importante poder detectarlos para tratarlos en forma separada.

Nuestro objetivo es utilizar este algoritmo como paso previo a la creación del *GNAT*. La idea es, primero, aplicar este proceso y obtener los  $r$ -cluster del espacio métrico. Luego, aislar e indexar separadamente los puntos *outliers*. Finalmente, realizar la selección de pivotes entre los restantes cluster. La cantidad y la forma de selección

de pivotes dependerá de la cardinalidad de cada cluster. Si el cluster es pequeño, se elige, como candidato a pivote, un elemento que se encuentre en el centro del cluster (*centroide*); caso contrario se eligen elementos que particionen uniformemente el cluster (*facilities*).

## 4 Trabajo Futuro

En el caso del *SAT* se prevé analizar su comportamiento frente a otro tipo de operaciones no consideradas aún y existentes en Bases de Datos Espaciales. También se estudiará cómo modificarlo, a fin de obtener una buena estructura para memoria secundaria. Claramente en este aspecto puede ayudar el definir la aridad máxima, de manera tal de asegurar que entra todo en una página en memoria secundaria.

En el caso del *GNAT*, se prevé, en función de los resultados que se obtengan con la estrategia planteada, establecer una política eficiente de selección de pivotes y de elección de aridad de cada nodo. El objetivo también es, poder determinar cuál es el  $r$  óptimo para la construcción del *Range  $r$  Graph*, y analizar la forma más conveniente de tratar el conjunto de puntos outliers.

## Referencias

- [1] R. Baeza Yates, N. Herrera, N. Reyes. Búsquedas en espacios métricos: posibles optimizaciones al Geometric Near-neighbor Access Tree In *Actas, III Workshop de Investigadores en Ciencias de la Computación (WICC'01), 2001*
- [2] S. Brin. Near neighbor search in large metric spaces. In *Proc. of the 21st Conference on Very Large Databases (VLDB'95)*, pages 574–584, 1995.
- [3] E. Chávez and G. Navarro. An effective clustering algorithm to index high dimensional metric spaces. In *Proc. 7th South American Symposium on String Processing and Information Retrieval (SPIRE'00)*, pages 75–86. IEEE CS Press, 2000.
- [4] E. Chávez. A Subquadratic Algorithm for Cluster and Outlier Detection in Massive Metric Data. in *Proceedings of SPIRE'2001, IEEE Press, 2001*.
- [5] E. Chávez and G. Navarro. A probabilistic spell for the curse of dimensionality. In *Proc. 3rd Workshop on Algorithm Engineering and Experiments (ALENEX'01)*, pages 147–160, LNCS 2153, 2001.
- [6] E. Chávez, G. Navarro, R. Baeza-Yates, and J. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 2001. To appear.
- [7] N. Herrera, N. Reyes and G. Navarro. Búsquedas en espacios métricos: árbol de aproximación espacial dinámico In *Actas, III Workshop de Investigadores en Ciencias de la Computación (WICC'01), 2001*
- [8] G. Navarro. Searching in metric spaces by spatial approximation. In *VLDB Journal*, 2002. To appear
- [9] G. Navarro and N. Reyes. Dynamic spatial approximation trees. In *Proc. XXI Conference of the Chilean Computer Science Society (SCCC'01), 2001*, pages 213–222. IEEE CS Press.

## *Hacia una experiencia de aprendizaje inicial de Ciencias de la Computación basada en la web*

**Guillermo Feierherd, Beatriz Depetris (1)**

*Departamento Informática – Sede Ushuaia*

*Facultad de Ingeniería – UNPSJB*

*Darwin y Canga – (9410) - Ushuaia – Tierra del Fuego – Argentina*

*TE/FAX: +54 (2901) 44 3513*

*feierherdge@ciudad.com.ar / depetrisb@ciudad.com.ar*

### **Resumen**

La necesidad de satisfacer las crecientes demandas sociales de educación requieren optimizar el uso de los recursos disponibles. Ello implica –particularmente en nuestra región patagónica y en la educación de nivel superior–, emplear todas las combinaciones posibles de técnicas de educación presencial y a distancia.

En este caso particular se tratará de extender los resultados auspiciosos alcanzados mediante experiencias presenciales realizadas con alumnos del último año del nivel Polimodal e ingresantes a la carrera de Informática –originadas inicialmente en una preocupación por los elevados niveles de deserción observados en la misma–, al resto de la población de la Provincia de Tierra del Fuego, incluyendo los residentes en las bases de la Antártida Argentina.

La transformación del curso presencial en un curso a distancia basado en la web, deberá contemplar tanto aspectos técnicos como educacionales y económicos.

EL proyecto está coordinado con la Facultad de Informática de la UNLP<sup>2</sup>.

### **Introducción**

Las demandas educacionales de la sociedad son cada vez mayores y proveer a su satisfacción se ha convertido en un imperativo ético. Imprescindiblemente, este objetivo debe lograrse mediante una optimización de los recursos físicos, económicos y humanos. En todo el mundo, pero particularmente en regiones como nuestra Patagonia, caracterizada por su extensión y su baja densidad de población, dicha optimización implica renunciar a utilizar exclusivamente las modalidades tradicionales de enseñanza. En consecuencia, cómo dice Greville, **“si existe insuficiencia de maestros y de escuelas disponibles deberán encontrarse alternativas distintas de la enseñanza presencial dentro de los linderos de la escuela.”**

Las distintas variantes de educación a distancia –eventualmente combinadas con instancias presenciales–, conforman hoy gran parte de las alternativas requeridas por Greville, y en ellas los gobiernos, las universidades, las empresas y los organismos no gubernamentales depositan sus mayores esperanzas para satisfacer las demandas de educación.

Así, según Prieto Castillo, para millones de estudiantes la oportunidad de aprender a distancia se presenta como una posibilidad –en algunos casos la única–, real, cualquiera sea el nivel de enseñanza. Las posibilidades que ofrece, entre las que se pueden citar la masividad espacial, el menor costo por estudiante, la diversificación de la población escolar, la individualización del aprendizaje, la provisión de cantidad sin desmedro de calidad y la generación de una autodisciplina de estudio indispensable para una educación continua, la convierten en una herramienta poderosa.

En el caso particular de las carreras de informática en la educación superior, un aspecto no menor en la optimización de recursos lo constituye la búsqueda de metodologías y herramientas que permitan disminuir los elevados niveles de deserción. Con ese objetivo, en coordinación con la

---

<sup>1</sup> Profesores de las Facultades de Ingeniería y Ciencias Económicas en la Sede Ushuaia de la Universidad Nacional de la Patagonia San Juan Bosco

<sup>2</sup> Laboratorio de Investigación y Desarrollo en Informática – Facultad de Informática – Universidad Nacional de La Plata

Facultad de Informática de la Universidad Nacional de La Plata, implementamos desde el año 2000 un proyecto para actuar sobre aquellas variables que entendíamos como las principales responsables de esos índices de deserción: el desconocimiento de las características del campo (en el sentido de Bourdieu), y por ende de las competencias y aptitudes para intervenir en el mismo, y distintos problemas metodológico-didácticos en el proceso de enseñanza en las etapas iniciales de las carreras universitarias en informática.

El proyecto implicó la realización de experiencias presenciales en la ciudad de Ushuaia, donde se halla asentada una Sede de la Facultad de Ingeniería de la Universidad Nacional de la Patagonia San Juan Bosco en la que se dicta la carrera de Informática. Los resultados preliminares permiten afirmar que dichos ensayos –tanto los llevados a cabo con los aspirantes a ingresar como con los alumnos de los últimos años del Nivel Polimodal–, pueden considerarse como positivos.

Por otra parte, visitas realizadas a colegios de Tolhuin y Río Grande (las otras dos localidades de la Provincia de Tierra del Fuego) para difundir las características de las carreras que se ofrecen en la Sede Ushuaia y comentar las actividades que se desarrollan en ella, nos permitieron detectar el interés de sus alumnos por participar de experiencias como las mencionadas. Finalmente, existen solicitudes para arbitrar mecanismos que hagan accesibles algunas de las actividades de la Sede local de la Universidad al personal de las bases permanentes en la Antártida Argentina que, en el caso de la base Esperanza, incluye jóvenes en los niveles de EGB y Polimodal. Tanto por razones geográficas como económicas, estos nuevos requerimientos sólo podrán ser satisfechos utilizando estrategias de educación a distancia.

Se plantea entonces la necesidad de extender las experiencias en dos dimensiones: la primera, temporal, permitiendo su realización desde etapas más tempranas del nivel medio; la segunda, espacial, poniéndola al alcance de un mayor número de alumnos del nivel medio y aspirantes a ingresar a estudios superiores de Ciencias de la Computación.

Esto constituye todo un desafío. Para repetir los resultados logrados no debemos olvidar que la educación a distancia no consiste simplemente en reproducir los contenidos de un curso presencial para que puedan ser accedidos a distancia. Implica mucho más, fundamentalmente en este caso en que pretendemos enseñar principios básicos de programación, tarea que ya presenta dificultades en forma presencial.

### **Temas de investigación, desarrollo y experimentación**

- **Analizar y comparar metodologías para la implementación de cursos de computación a distancia**

Se trata de realizar un análisis comparativo de las variantes existentes en educación a distancia, incluyendo sistemas asincrónicos y sincrónicos (en línea) y, fundamentalmente aquellos basados en la web o soportados por ella, eventualmente combinados con instancias presenciales. Este primer análisis considerará exclusivamente los aspectos pedagógicos que puedan favorecer la enseñanza de la programación. El resultado de esta etapa debe arrojar un orden relativo de alternativas posibles.

Cabe un comentario adicional: disponerse a seguir un curso a distancia a través de la web implica adquirir las habilidades necesarias para manejar el sistema informático que permite conectarse a ella. En muchos casos, cuando los contenidos del curso que se pretende seguir están totalmente alejados de estas cuestiones tecnológicas, el tiempo requerido para lograr dichas habilidades (si no se dispone ya de las mismas), puede verse como un tiempo perdido. Esto no es así en el caso que nos ocupa, ya que las destrezas que se adquieran son útiles para desenvolverse en el campo al que se pretende ingresar.

- **Aplicación de limitaciones técnicas y económicas a las alternativas diseñadas en el paso previo.**

Se filtrarán las alternativas diseñadas aplicando criterios de factibilidad técnica y económica.

Se deberán considerar tanto las propias de la institución como la de los alumnos potenciales. Esta última condición implica la necesidad de un relevamiento, no necesariamente exhaustivo, que las determine. Si bien no es posible actuar sobre las posibilidades de hardware y de comunicaciones de que dispone cada alumno, se privilegiará en materia de software el uso de software libre, lo que permitirá su distribución sin costo.

- **Diseño y desarrollo del curso para su implementación a distancia**

El diseño y desarrollo del curso sobre la alternativa elegida deberá contemplar no una simple transcripción de los materiales sino un rediseño que lo adapte al nuevo contexto y que permita aprovechar las ventajas potenciales que éste ofrece.

- **Implementación del curso**

Se llevarán a cabo las distintas instancias del curso.

- **Evaluación de los resultados y retroalimentación**

Se evaluarán los resultados obtenidos y, en función de los mismos se determinará la conveniencia de modificaciones y mejoras.

### **Algunos resultados obtenidos y tareas en curso**

- Durante los años 2001 y 2002 se desarrollaron Talleres de Introducción a la Programación (a nivel del ingreso a la carrera de Informática) y Talleres de Análisis de Aptitudes y Competencias para alumnos de los dos últimos años del Nivel Polimodal residentes en la ciudad de Ushuaia. En todos los casos se utilizó el entorno de programación Visual Da Vinci. Las experiencias realizadas permitieron sugerir modificaciones y extensiones fundadas al ambiente, las que se están desarrollando en coordinación con la UNLP.
- Los Talleres de Introducción a la Programación se prolongaron en actividades de autoaprendizaje y desarrollo de algoritmos (con asistencia de tutores), durante el primer cuatrimestre de cada año, lo que permitió a los alumnos iniciar el Curso de Algorítmica y Programación en mejores condiciones. Se acompañó la investigación experimental con evaluaciones, encuestas a los alumnos y estudios comparativos con alumnos de ciclos anteriores.
- La formación de recursos humanos en el proyecto ha concretado tres Especializaciones en Docencia Universitaria y están en curso cuatro estudios de Magister en temas relacionados.

## Bibliografía Básica

Prieto Castillo D., Gutiérrez Pérez F. *La mediación Pedagógica* Editorial Ciccus – La Crujía, Sexta Edición, Argentina, 1999.

Greville R. *La UNED: Una Evaluación* Editorial UNED, Costa Rica, 1987.

Fainholc A. *La Interactividad en la Educación a Distancia* Editorial Paidós, Argentina, 1999.

Pérez Gómez E. *Volver a pensar la Educación* Editorial Morata, Madrid, 1996

Prieto Castillo D. *La comunicación en la Educación* Editorial Ciccus – La Crujía, Argentina, 1999

Colección de Transactions on Education del IEEE

Colección de Boletines del SIGCSE de ACM

Champredonde R., Palacios A., Ainchil V. *Programming Teaching Based on Thinking Skills* Publicado en First International Congress on Tools for Teaching Logic, Universidad de Salamanca, Junio 2000.

Lucarelli E. *La construcción de la articulación teoría-práctica en las cátedras universitarias: búsqueda y avance* Ponencia en el Instituto de Investigaciones en Ciencias de la Educación, Facultad de Filosofía y Letras, UBA, 1996

Feierherd G., Depetris B., Jerez M. *Tecnología Informática aplicada al aprendizaje inicial de Ciencias de la Computación* Publicado en el III Workshop de Investigadores en Ciencias de la Computación (WICC 2001), San Luis, 2001

Feierherd G., Depetris B., Jerez M. *Una evaluación sobre la incorporación temprana de algorítmica y programación en el ingreso a Informática* Publicado en el VII Congreso Argentino de Ciencias de la Computación, El Calafate (Santa Cruz), 2001

Feierherd G., Depetris B. Proyecto *Evaluación y desarrollo de herramientas multimediales para análisis de competencias y aplicación de una metodología didáctica para mejorar el aprendizaje inicial en Informática* Universidad Nacional de Patagonia San Juan Bosco, 2000

## **“Bases de Datos distribuidas: Agentes Móviles en problemas de Replicación, migración y recuperación de fallas”**

**Ivana Miaton<sup>1</sup>, Patricia Pesado<sup>2</sup>, Rodolfo Bertone<sup>3</sup>, Armando De Giusti<sup>4</sup>**

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>5</sup>.*

*Facultad de Informática. Universidad Nacional de La Plata.*

### **Resumen**

Los avances producidos en los últimos años para el establecimiento de una red de computadoras sin conexión “full time” y aplicaciones portables de la información han dado inicio a una nueva modalidad de trabajo para la Informática en general y las BD en particular. Con la posibilidad de tener agentes móviles, mediante los cuales los usuarios tienen acceso a los datos y a los servicios informáticos sin importar su comportamiento, la localización física o el movimiento, se abre un espacio de investigación de gran interés por sus aplicaciones.

En este marco de investigación, se presentan una propuesta de trabajo para el estudio de recuperación de fallos y actualización de réplicas sobre una BDD con la posibilidad de migrar las réplicas entre las localidades intervinientes, utilizando agentes móviles.

### **Palabras Claves**

Bases de Datos. Bases de Datos Distribuidas. Agentes Móviles.

---

<sup>1</sup> Licenciada en Informática. Becario UNLP. Ayudante Diplomado Dedicación Semi Exclusiva. [imiaton@lidi.info.unlp.edu.ar](mailto:imiaton@lidi.info.unlp.edu.ar)

<sup>2</sup> Licenciada en Informática. Profesor Titular Ded. Exclusiva. [ppesado@lidi.info.unlp.edu.ar](mailto:ppesado@lidi.info.unlp.edu.ar)

<sup>3</sup> Licenciado en Informática. Profesor Adjunto Ded. Exclusiva. [rbertone@lidi.info.unlp.edu.ar](mailto:rbertone@lidi.info.unlp.edu.ar)

<sup>4</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. [degiusti@lidi.info.unlp.edu.ar](mailto:degiusti@lidi.info.unlp.edu.ar)

<sup>5</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina. TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

## Introducción

En la actualidad, con la utilización de redes de computadoras y la necesidad de compartir la información pudiendo tener rápido acceso a la misma, es necesario estudiar mecanismos para la distribución física de los datos, a fin de realizar el proceso de búsqueda y recuperación lo más eficiente posible, en términos de tiempo de respuesta. Para ello es necesario tener en cuenta las características propias de cada problema particular, en función de los datos que manipula y el lugar físico desde donde la información es accedida. Este enfoque hace particularmente atractiva la investigación de la tecnología de Agentes (que pueden tener "inteligencia" en la toma dinámica de decisiones) que permite manejar las decisiones de optimización relacionadas con la migración/replicación y recuperación de datos de la BDD. [DURF 01]

En general, los sistemas de Bases de Datos, tienen mecanismos para facilitar los temas relacionados con la recuperación de información, pero cuando se dispone de un entorno distribuido aparecen una serie de aspectos relacionados con la posición de estos datos dentro de la red, los cuales afectan considerablemente la performance (respuesta y actualización) del sistema. Se han realizado gran cantidad de estudios respecto a la forma en que se puede replicar o fragmentar la información y, en general, no existe un mecanismo genérico que se pueda aplicar a todos los casos. La utilización de Agentes en esta línea de I/D es un enfoque muy actual que tiende a generalizarse en general para todo sistema de software distribuido de alta complejidad, como una evolución de los objetos distribuidos. [JENN 01]

El objetivo de la investigación es estudiar en profundidad los distintos mecanismos de migración, replicación y recuperación de fallas en forma dinámica, analizando su incidencia de acuerdo a la utilización por parte del usuario de la BDD. Por este motivo, se propone desarrollar un entorno que permita simular la distribución de información a lo largo de una red de computadoras conectadas por Internet, variando las tasas de utilización de datos y estudiando el tiempo de respuesta y actualización dentro del problema, con el objetivo final de encontrar la mejor distribución posible de la información. [PAV 01]

## OBJETIVOS

El objetivo general es investigar los problemas de migración, replicación y recuperación de datos en forma dinámica, para Bases de Datos Distribuidas, utilizando la tecnología de Agentes. A partir de los resultados obtenidos con el ambiente desarrollado en investigaciones anteriores, los que fueron presentados en WICC 2001 y CACIC 2001, se profundiza la investigación realizada y se propone desarrollar un ambiente de simulación distribuida, basado en el lenguaje JAVA, que permita estudiar la optimización de distribución de datos (en particular imágenes) a lo largo de una red de computadoras, buscando minimizar el tiempo de respuesta del sistema, tanto a consultas como actualizaciones (considerando condiciones de falla física y lógica).

El estudio inicial estará basado en datos conformados por imágenes, el cual presenta una serie de características especiales:

- Son datos estáticos, se pueden agregar o quitar imágenes a la Base de Datos pero no alterar una preexistente. Esto facilita en cierta medida la manipulación de la actualización, en particular lo relacionado con el tratamiento de errores.
- Conforman una gran Base de Datos, a diferencia de un sistema convencional (tipo gestión) cada imagen conlleva, para su transmisión a lo largo de la red, un gran volumen de bytes. De aquí que sea muy importante encontrar un esquema de distribución que minimice la comunicación.
- Existe un conjunto de datos que se utiliza asiduamente y otro que tiene una utilización parcial, ya sea por lapsos de tiempo (para una época del año) o porque son de actualidad en un momento determinado.
- Existen redes donde el uso de datos compartidos es crítico y donde las distancias físicas son muy importantes (ejemplo: las redes académicas que vinculan Laboratorios de diferentes puntos del mundo investigando un mismo tema). Este tipo de problemas se ajusta en gran medida a la investigación a desarrollar por el requerimiento de migración/replicación y recuperación dinámica de datos (un caso típico son los Laboratorios dedicados a temas tales como los estudios genómicos).

**TEMAS DE INVESTIGACION**

- Bases de Datos Distribuidas. Transacciones Distribuidas. Protocolos.
- Teoría de Agentes. Agentes inteligentes. Agentes móviles. Aplicación a BDD
- Sistemas Distribuidos. Lenguajes y ambientes para Sistemas Distribuidos. JAVA.
- Fragmentación de información. Aplicabilidad al caso de estudio. Localidad y migración dinámica de datos.
- Replicación de información. Alternativas: replicación estática Vs. Dinámica.
- Errores en BDD. Recuperación de fallas.
- Nivel de replicación de datos: réplicas parciales o totales.
- Aplicación de la teoría de Agentes a los problemas de migración, replicación y recuperación de fallas en BDD.
- Generación de casos de prueba. Trazas. Métricas y mediciones de resultados.
- Definición de patrones de interés, que permitirán evaluar cada uno de los resultados obtenidos y sacar conclusiones.

**BIBLIOGRAFIA BASICA**

- [AGRA 95] *Coding Based Replication Schemes for Distributed Systems*. Ganga Agrawal, Pankaj Jalote. IEEE transactions on parallel and distributed systems. Vol 6 No. 3 March 1995
- [BELL 92] *Distributed Database Systems*, Bell, David; Grimson, Jane. Addison Wesley. 1992
- [BERN 84] *An algorithm for concurrency Control and Recovery in Replicated Distributed Databases* Bernstein, Goodman. ACM Trans. Database Systems, vol 9 no. 4, pp. 596-615, Dec 1984.
- [BERS 92] *Client Server Architecture*. Berson, Alex. Mc Graw Hill Series. 1992
- [BERT 99] *Ambiente de experimentación para Bases de Datos Distribuidas*. Bertone, Rodolfo; De Giusti, Armando; Ardenghi, Jorge. Anales WICC. Mayo 1999. San Juan Argentina.
- [BHAS 92] *The architecture of a heterogeneous distributed database management system: the distributed access view integrated database (DAVID)*. Bharat Bhasker; Csaba J. Egyhazy; Konstantinos P. Triantis. CSC '92. Proceedings of the 1992 ACM Computer Science 20th annual conference on Communications, pages 173-179
- [BIGU 01] *Constructing Intelligent Agents Using Java*. Joseph Bigus, Jennifer Bigus. Editorial Wiley.
- [BURL 94] *Managin Distributed Databases. Building Bridges between Database Island*. Burleson, Donal. 1994
- [DATE 94] *Introducción a los sistemas de Bases de Datos*. Date, C.J. Addison Wesley 1994.
- [DIPA 99] *Un ambiente experimental para evaluación de Bases de Datos Distribuidas*. Di Paolo, Mónica; Bertone, Rodolfo; De Giusti, Armando. Paper presentado en la ICIE 99. Fac. Ingeniería. UBA. Buenos Aires. Argentina
- [DURF 01] *Scaling Up Agent Coordination Strategies*. Edmund M. Durfee. Computer IEEE, Julio 2001. Páginas 39-46.
- [GARG 99] *Performance and Reliability Evaluation of Passive Replication Schemes in Application Level Fault Tolerance*. Sachin Garg. Yennun Huang. Chandra Kintala. IEEE 1999.
- [GEIS 94] *PVM: Paralell Virtual Machine. A user guide and tutorial for networked parallel computer*. Al Geist, Adam Beguelin, Jack Dongarra, Weicheng Jiang, Robert Manchek, Vaidy Sunderam, The MIT Press, 1994.
- [GRAY 96] *The Dangers of Replication and a Solution*. Jim Gray, Pat Helland, Patrick O'Neil, Dennis Shasha. SIGMOD 96. Montreal Canada, ACM 1996.
- [GUY 90] *Name Transparency in Very Large Scale Distributed File Systems*. Richard Guy, Thomas Page, John Heidemann. Second IEEE Workshop on Experimental Distributed Systems. October 1990. Pag 20-25
- [HOLL 99] *The Performance of Database Replication with Group Multicast* Joanne Holliday, Divyakant Agrawal, Amr El Abbadi. IEEE 1999.
- [HOLL 99] *Database Replication: If you must be :Lazy be Consistent*. Joanne Holliday, Divyakant Agrawal, Amr El Abbadi. IEEE 1999.
- [JENN 01] *An Agent-based approach for Building Complex Software systems*. Nicholas Jennings. communications of the ACM. Abril 2001. volumen 44, Nro 4, páginas 35-41.
- [KEMM] *A siute of Database Replication Protocols based on Group Communicaton Primitives* Bettina Kemme, Gustavo Alonso. IEEE
- [LARS 95] *Database Directions. From relational to distributed, multimedia, and OO database Systems*. Larson, James. Prentice Hall. 1995
- [LIN] *An Efficient Quorum-Based Scheme for Managing Replicated Data in Distributed Systems*. Ching-min Lin, Ge-Ming Chiu, Cheng-Hong Cho. IEEE
- [MIAT 98] *Experiencias en el análisis de fallas en BDD*. Miaton, Ivana; Ruscuni, Sebastián; Bertone, Rodolfo; De Giusti, Armando. Anales CACIC 98. Neuquén Argentina.
- [MORS94] *Practical Parallel Computing*, Stephen Morse, AP Profesional 1994.
- [ÖZSU 91] *Principles of Distributed Database Systems*. Özsu, M. Tamer; Valduriez, Patric. Prentice Hall 1991.
- [PAGN] *Improving Replication Protocols through Priorities* Henning Pagnia, Oliver Theel. IEEE
- [PAV 01] *Agentes ed Software*. Juàn Pavòn Mestras. Universidad Complutense Madrid. <http://gradia.fdi.ucm.es/jpavon/agentes/am.pdf>
- [PDNE] *Transaction Reorderin in Replicated Dataases*. Fernando Pedone, Rachid Guerraoui, André Schiper. IEEE.

- [RUSC 00] *Estudio de recuperación de errores en BDD*. Sebastián Rusconi. Tesina de Grado. Fac. de Informática. Diciembre 2000.
- [SCHU 94] *The Database Factory. Active database for enterprise computing*. Schur, Stephen. 1994.
- [SHEI] *Replication of Multimedia Data Using Master-Slave Architecture*. Sohail Sheikh, Raj Ganesan.
- [SHET 90] *Federated database systems for managing distributed, heterogeneous, and autonomous databases*. Amit P. Sheth; James A. Larson. ACM Computing Surveys. Vol. 22, No. 3 (Sept. 1990), Pages 183-236
- [SILB 98] *Fundamentos de las Bases de Datos*. Silbershatz; Folk. Mc Graw Hill. 1998.
- [SUP] *A replica Control Method for Improving Availability for Read-only Transactions*. Chang Sup Park, Myoung Ho Kim, Yoon Joon Lee.
- [THEL 99] *Distributed Caching and Replication*. Oliver Theel. Markus Pizka. Proceedings of the 32<sup>nd</sup> Hawaii International Conference on System Sciences. 1999
- [THOM 90] *Heterogeneous distributed database systems for production use*. Thomas, Charles; Glenn R. Thompson; Chin-Wan Chung; Edward Barkmeyer; Fred Carter; Marjorie Templeton; Stephen Fox; Berl Hartman. ACM Computing Surveys. Vol. 22, No. 3 (Sept. 1990), Pages 237-266
- [UMAR 93] *Distributed Computing and Client Server Systems*. Umar, Amjad. Prentice Hall. 1993.
- [WEB1] [www.seas.gwu.edu/faculty/shmuel/cs267/textbook/tpcp.html](http://www.seas.gwu.edu/faculty/shmuel/cs267/textbook/tpcp.html)
- [WEB2] [www.sei.cmu.edu/str/descriptions/dtpc\\_body.html](http://www.sei.cmu.edu/str/descriptions/dtpc_body.html)
- [WEB3] [www.datanetbbs.com.br/dclobato/textos/bddcs/parte2.html](http://www.datanetbbs.com.br/dclobato/textos/bddcs/parte2.html)
- [WEB4] [www.itlibrary.com/library/1575211971/ch45.htm](http://www.itlibrary.com/library/1575211971/ch45.htm)
- [WEB5] [www.itlibrary.com/library/1575211971/ch26.htm](http://www.itlibrary.com/library/1575211971/ch26.htm)
- [WEB6] [java.sun.com/docs/books/tutorial/networking/sockets/definition.html](http://java.sun.com/docs/books/tutorial/networking/sockets/definition.html)
- [WU] *A new Quorum Based Replica Control Protocol* Yu/ting Wu, Yao-Jen Cahg, Shyan Ming Yuan, Her Kung Chang. IEEE

## Project Line: Formally Specifying the Domain of Balanced Scorecard

G. Montejano, R. Uzal, D. Riesco, A. Dasso, A. Funes

*Universidad Nacional de San Luis*

*Ejército de los Andes 950 – (5700) San Luis – Argentina*

*Tel: +54-2652-424027 ext. 251 Fax: +54-2652-430059*

*{gmontel\ruzal\driesco\arisdas\lafunes}@unsl.edu.ar*

### Abstract

*In today's world, in both the public or private organizations, there exists the problem that the strategies, that are the only sustainable forms that the organizations have to create value, are changing, but the tools to measure them are not. On the other hand, in the current economy, the intangible assets are most important sources of competitive advantages, and it's necessary to have tools that describe assets based on knowledge and strategies for value creation. Without these tools, the organizations have difficulties to manage things that they are not able to describe or measure. Also, there exists a scheme to support the organizational management, called Balanced Scorecard, that enables organizations to clarify their vision and strategy, and translate them into action.*

*There are not too much integrated tools that carry from strategic mapping to Balanced Scorecard. We believe that if this scheme has an automated solid base to help the management of the projects and the organization, the managers will have the possibility to make decisions with bigger speed and accuracy, as well as to observe the deviations of the reality regarding the desirable thing with more anticipation.*

*Here, we present a first proposal for formally construct tools to support the Balanced Scorecard in the organizational environment. In order to construct high quality software, it is important to formally outline the specification of the domain and the requirements. We do this in RAISE Specification Language, and its RAISE method, because of its intended use on real developments in the software industries. This language and this method are formal, but it allows intermediate specifications not formal –rigorous–and it is sufficiently flexible to conveniently work with distinct degrees of formality without the strict techniques required by the formal methods.*

**Keywords:** Software Engineering, Formal Methods, Management, CAM.

### 1. Introduction

The capacity to execute a strategy is more important than the quality of the strategy itself. The application of the strategy is the most important factor on the valuation of an enterprise and its management [1-3]. A problem is that the strategies, that are the only sustainable forms that the organizations have to create value, are changing, but the tools to measure them are not. More recent studies have

estimated that by the end of the 20th century, the countable value of the tangible assets represented only from 10% to 15% of market value of the enterprises. It is clear that to create value opportunities, we are going from material asset management to management of strategies based on knowledge that develop the immaterial assets of the organization: relations with clients, innovative products and services, efficient operating processes of high quality, information technology and databases, and also employees capabilities, skills and motivations.

In the current economy, however, where intangible assets are most important sources of competitive advantages, it's necessary to have tools that describe assets based on knowledge and strategies for value creation. Without these tools, the enterprises have difficulties to manage things that they are not able to describe or measure [4]. The organizations need a type of management system to be explicitly designed to manage the strategy, not the tactic [5].

The strategy application requires, therefore, that all employees, as well as all business units and support units, be aligned and linked to the strategy. And with the quick changes in technology, competition and regulations, the formulation and application of the strategy should be a continuous and participative process. The current organizations need a language that allows them to communicate the strategy, as well as processes and systems that help them to implement it and to obtain information or feedback on it. The Balanced Scorecard became a tool to manage the strategy.

### 2. Review of Balanced Scorecard

A approach to strategic management was developed in the early 1990's by Drs. Robert Kaplan (Harvard Business School) and David Norton (Balanced Scorecard Collaborative) [6-8]. They named this system the 'balanced scorecard'. Which provides a prescription as to what companies should measure in order to 'balance' the financial perspective.

The Balanced Scorecard is a *management system* (not only a measurement system) that enables organizations to clarify their vision and strategy and translate them into action. It provides feedback around both the internal business processes and external outcomes in order to continuously improve strategic performance and results [9].

The following four figures show the central point of its definition.

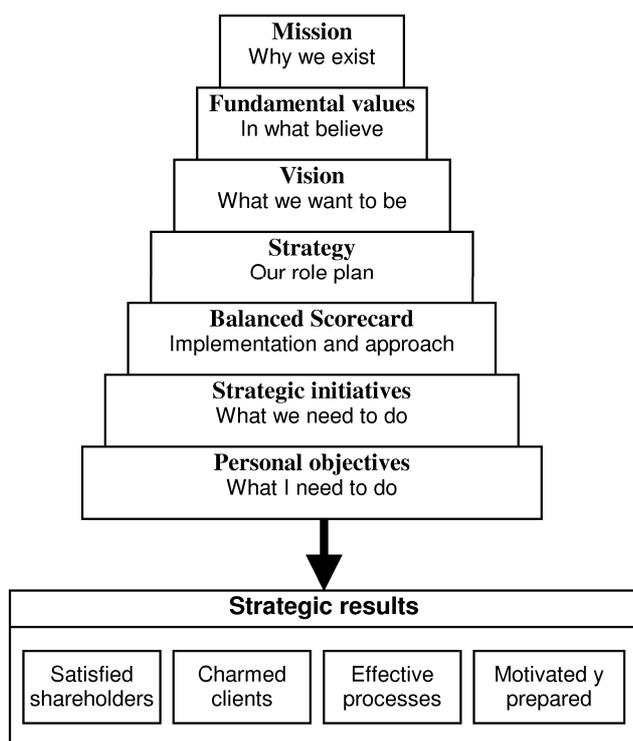


Figure 1. Translation a mission to searched results.

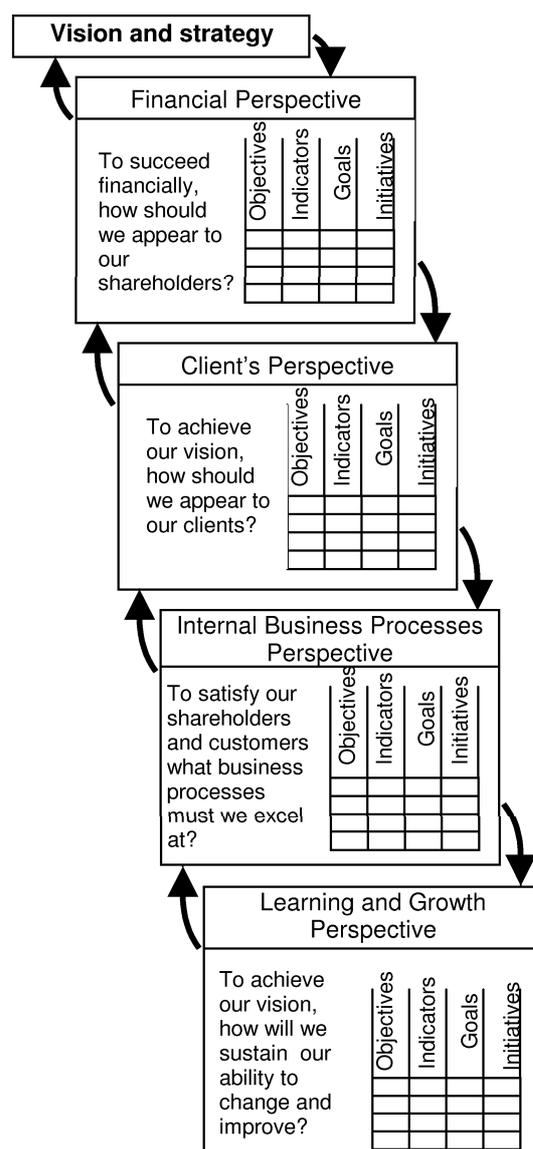


Figure 2. Definition of the cause-effect relationships of the strategy.

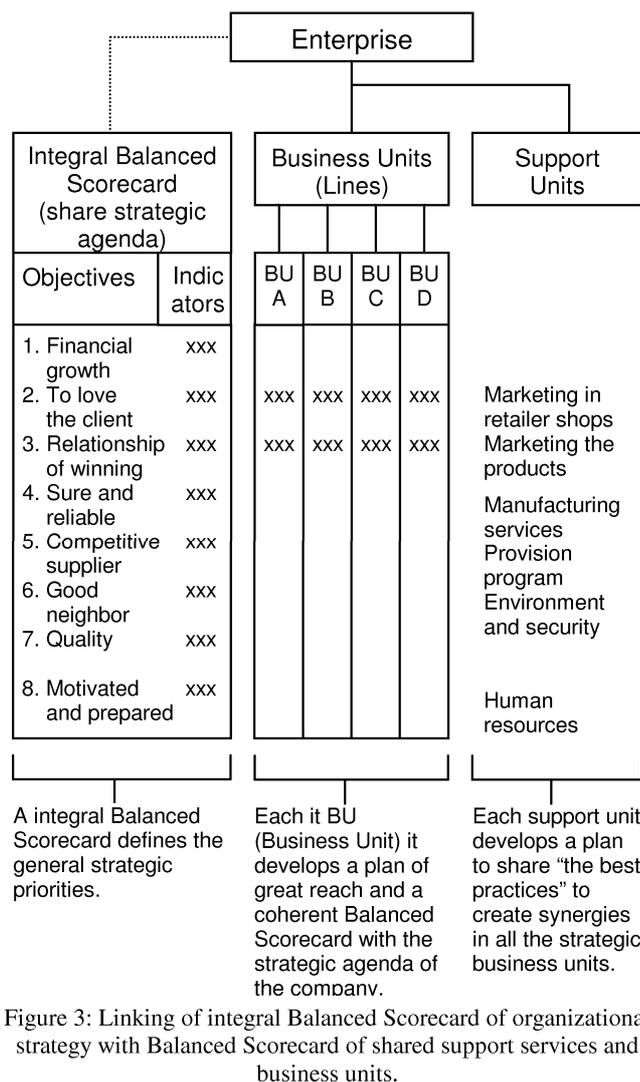


Figure 3: Linking of integral Balanced Scorecard of organizational strategy with Balanced Scorecard of shared support services and business units.

Financial perspective	Client perspective	Internal processes perspective	Learning and growth perspective
To optimize the distribution of capital (value for shareholders)	To promote the crossed sale (to share bank accounts)	To optimize shared processes: -Infrastructure -Acquisitions -Bookcases space	Share better practices To develop key skill
To balance the growth with the risk (rate of the investment adjusted to the risk)	To be centered in the client (satisfaction and fidelity of client)	Scale economies: -Distribution -Manufacture  To integrate of chain of value (Cost by unit) (Quote of market)	

Figure 4. Practical example of the spectrum of organizational synergies.

### 3. Formalization of the Domain in RAISE Specification Language

We first present the stakeholders specification, as integral part of the Balanced Scorecard management system. Then, we present additional components needed to form the complete specification for integral and strategic Balanced Scorecard concepts. Some ideas have been taken from the technical reports published by Dines Bjørner [10-12]. The syntax and the semantic are based on the RAISE specification language [13] and the specifications are presented in five different sections for clarity of the readers.

RAISE is an acronym for “Rigorous Approach to Industrial Software Engineering”. RAISE takes the word “industrial” seriously. The method is intended for use on real development, not just toy examples. The method is based on a number of principles: separate development, step-wise development, invent and verify, and rigor [14].

In this first work, we are centered on the specification of Balanced Scorecard domain. We try to distinguish between the domain analysis and the product analysis. Here we begin with the first domain specification as shown.

**Specification 1:** The *stakeholders* (interested parts) can be formally defined as shown below:

The BSC of the interested parts (stakeholders) identifies to main constituent parts of the business –shareholders, clients and employees– and also some others like the suppliers and the community.

```

scheme STAKEHOLDERS =
  class
    type
      Stakeholder,
      Main_Stakeholder = {| s:Stakeholder •
        is_Main_Stakeholder(s) |},
      Secondary_Stakeholder = {| s:Stakeholder
        • is_Secondary_Stakeholder(s) |},
      Client ={|s:Stakeholder • is_Client(s)|},
      Employee = {| s:Stakeholder •
        is_Employee(s) |},
      Shareholder = {| s:Stakeholder •
        is_Shareholder(s) |},
      Supplier = {| s:Stakeholder •
        is_Supplier(s) |},
      Community = {| s:Stakeholder •
        is_Community(s) |}
    value
      is_Client : Stakeholder -> Bool,
      is_Employee : Stakeholder -> Bool,
      is_Shareholder : Stakeholder -> Bool,
      is_Supplier : Stakeholder -> Bool,
      is_Community : Stakeholder -> Bool,
      is_Main_Stakeholder: Stakeholder -> Bool
      is_Main_Stakeholder(s) ≡
        is_Client(s) ∨ is_Employee(s) ∨
        is_Shareholder(s),
      is_Secondary_Stakeholder :

```

```

      Stakeholder -> Bool
      is_Secondary_Stakeholder(s) ≡
        is_Supplier(s) ∨ is_Community(s)
    axiom
      ∀ s : Stakeholder •
        is_Shareholder(s) ⇒
          ~is_Client(s) ∧ ~is_Supplier(s),
      ∀ s : Stakeholder •
        is_Client(s) ⇒ ~is_Shareholder(s),
      ∀ s : Stakeholder •
        is_Supplier(s) ⇒ ~is_Shareholder(s)
    end

```

**Specification 2:** The strategic structure, i.e. the *translation from the mission to searched results* (pointing to reach Total Quality Management) to construct the strategy can formally be defined as follows:

```

STAKEHOLDERS
scheme STRATEGIC_STRUCTURE =
extend STAKEHOLDERS with
  class
    type
      Mission, Fundamental_Values, Vision,
      Strategy, Integral_BSC, Goal, Time,
      Strategic_Initiatives,
      Personal_Objectives,
      Actual_Indicator = Indicator,
      Management_Indicator = Indicator,
      Final_Indicator = Indicator,
      Result_Indicator = Indicator,
      is_Actual_Indicator,
      is_Management_Indicator,
      is_Final_Indicator, is_Result_Indicator,
      Strategic_Results, Value = Real,
      Deviation = Real,
      Indicator = {| (g, v, t, ld, rd) :
        (Goal×Value×Time×Deviation) •
        is_Indicator(g, v, t, ld, rd) |},
      Total_Quality_Management ==
        Total_Quality_Management_1 |
        Total_Quality_Management_2,
      Total_Quality_Management_1 =
        {| (g, ai, mi) : (Goal ×
          Actual_Indicator×Management_Indicator)•
          is_Total_Quality_Management(g, ai, mi)|},
      Total_Quality_Management_2 =
        {| (g, fi, ri) : (Goal ×
          Final_Indicator × Result_Indicator) •
          is_Total_Quality_Management(g, fi, ri)|}
    value
      M_FV : Mission  $\xrightarrow{m}$  Fundamental_Values,
      FV_V : Fundamental_Values  $\xrightarrow{m}$  Vision,
      V_S : Vision  $\xrightarrow{m}$  Strategy,
      S_IBSC : Strategy  $\xrightarrow{m}$  Integral_BSC,
      IBSC_SI : Integral_BSC  $\xrightarrow{m}$ 
        Strategic_Initiatives,
      SI_PO : Strategic_Initiatives  $\xrightarrow{m}$ 
        Personal_Objectives,
      PO_G : Personal_Objectives × Employee  $\xrightarrow{m}$ 
        Goal,
      PO_Indicators_SR : Personal_Objectives ×
        Management_Indicator × Result_Indicator

```

```

→ Strategic_Results,
is_Actual_Indicator,
is_Management_Indicator,
is_Final_Indicator,
is_Result_Indicator : Goal × Value ×
Time × Deviation × Deviation → Bool,
is_Indicator : Goal × Value × Time ×
Deviation × Deviation → Bool
is_Indicator(g, v, t, ld, rd) ≡
is_Management_Indicator(g, v, t, ld, rd) ∨
is_Result_Indicator(g, v, t, ld, rd) ∨
is_Actual_Indicator(g, v, t, ld, rd) ∨
is_Final_Indicator(g, v, t, ld, rd),
is_Total_Quality_Management :
Goal × Indicator × Indicator → Bool,
precede : Time × Time → Bool
axiom
∀ d : Deviation • d ≥ -100.0 ∧ d ≤ 100.0,
∀ (g1, v1, t1, left_dev_allow1,
right_dev_allow1) : Result_Indicator,
(g2, v2, t2, left_dev_allow2,
right_dev_allow2) : Result_Indicator •
g1 = g2 ⇒ v1 = v2 ∧ t1 = t2 ∧
left_dev_allow1 = left_dev_allow2 ∧
right_dev_allow1 = right_dev_allow2,
∀ g1 : Goal, (g2, vactual, tactual,
left_dev_actual, right_dev_actual) :
Actual_Indicator,
(g3, vdesirable, tdesirable,
left_dev_allow, right_dev_allow) :
Management_Indicator •
g1 = g2 ∧ g2 = g3 ∧
(vactual - left_dev_actual) ≤
(vdesirable - left_dev_allow) ∧
(vactual + right_dev_actual) ≤
(vdesirable + right_dev_allow) ∧
precede(tactual, tdesirable) ⇒
is_Total_Quality_Management(g1,
(g2, vactual, tactual,
left_dev_actual, right_dev_actual),
(g3, vdesirable, tdesirable,
left_dev_allow, right_dev_allow)),
∀ g1 : Goal,
(g2, vfinal, tfinal, left_dev_final,
right_dev_final) : Final_Indicator,
(g3, vdesirable, tdesirable,
left_dev_allow, right_dev_allow) :
Result_Indicator •
g1 = g2 ∧ g2 = g3 ∧
(vfinal - left_dev_final) ≤
(vdesirable - left_dev_allow) ∧
(vfinal + right_dev_final) ≤
(vdesirable + right_dev_allow) ∧
precede(tfinal, tdesirable) ⇒
is_Total_Quality_Management(g1,
(g2, vfinal, tfinal, left_dev_final,
right_dev_final),
(g3, vdesirable, tdesirable,
left_dev_allow, right_dev_allow))
end

```

With the last expression, we are specifying the

is\_Total\_Quality\_Management predicate, which evaluates to true if all system Actual\_Indicator are valid into Management\_Indicator, and all system Final\_Indicator are valid into Result\_Indicator at end of time.

The Balanced Scorecard of the interested parts (stakeholders) provides Balanced Scorecard of the key indicators of activity. Where the second factor of Result\_Indicator, which is a Real, determines “the desirable value” to obtain as result on the measure of its Goal. Similarly, the second factor of Management\_Indicator determines “the actual value” that it measure of its Goal.

Examples of values for Personal\_Objectives type are by:

- a Client as Main\_Stakeholder is: “An excellent place to buy”
- a Employee as Main\_Stakeholder is: “An excellent place to work”
- a Shareholder as Main\_Stakeholder is: “An excellent place to invest”

**Specification 3:** We formalize the *translation from the strategy to operative terms* (from strategic approach to tactic approach) as expressed below:

```

STRATEGIC_STRUCTURE
scheme STRATEGY_TO_TACTICS =
extend STRATEGIC_STRUCTURE with
class
value
Describe_Strategy :
Strategy × Main_Stakeholder  $\xrightarrow{m}$ 
Strategic_Initiatives,
Execute_Strategy :
Strategic_Initiatives  $\xrightarrow{m}$ 
Personal_Objectives × Indicator
end

```

**Specification 4:** The four Balanced Scorecard *perspectives* can be specified as follows:

```

STRATEGY_TO_TACTICS
scheme PERSPECTIVES =
extend STRATEGY_TO_TACTICS with
class
type
Financial_Perspective, Financial_Terms,
Profits, Clients_Perspective, Clients,
Valor_Proposition, Reengineering,
Products_and_Services_Innovation,
Clients_Management_Processes,
Internal_Processes_Perspective,
Internal_Processes, Technology_List,
Learning_and_Growth_Perspective,
Employees, Skills, Capabilities
value
FP : Strategy × Financial_Terms  $\xrightarrow{m}$ 
Financial_Perspective × Profits,
CP : Strategy × Clients  $\xrightarrow{m}$ 
Clients_Perspective × Valor_Proposition
× Products_and_Services_Innovation
× Clients_Management_Processes,

```

```

IPP : Strategy × Internal_Processes  $\xrightarrow{m}$ 
      Internal_Processes_Perspective ×
      Technology_List × Reengineering,
LaGP : Strategy × Employees  $\xrightarrow{m}$ 
      Learning_and_Growth_Perspective ×
      Skills × Capabilities

```

end

Where Strategy, Clients and Employees types are defined as above.

**Specification 5:** Finally, the *complete Balance Scorecard management system concept* can be formally defined as showed below:

```

PERSPECTIVES
scheme BSC =
extend PERSPECTIVES with
class
  type
    BSC
  value
    BSC : Financial_Perspective ×
           Clients_Perspective ×
           Internal_Processes_Perspective ×
           Learning_and_Growth_Perspective ×
           Strategic_Initiatives ×
           /* Strategy × Main_Stakeholder */
           Secondary_Stakeholder ×
           Strategic_Results
           /* Goal, Employee,
              Personal_Objectives,
              Management_Indicator,
              Result_Indicator */
            $\xrightarrow{m}$  Bool
axiom
   $\forall g : \text{Goal}, i1, i2 : \text{Indicator} \bullet$ 
    is_Total_Quality_Management (g, i1, i2)
end

```

There is an overload of definition on Strategic\_Integral\_BSC because it should be the same concept and content, viewed from three different angles:

- from strategic BSC of stakeholders (each stakeholder has their own BSC, plus the internal processes),
- from the four perspectives, that is the strategic map plus desirable results,
- from business units, i.e. the internal synergy that contribute each of strategic business units.

#### 4. Conclusions

The scheme to support the organizational management, called Balanced Scorecard, is useful in the organizations, but there are not too much integrated tools that carry from strategic mapping to Balanced Scorecard. We have proposed a first proposal for the construction of tools to support the Balanced Scorecard in the organizational environment. Then, in this first step we formally model the domain of Balanced Scorecard.

For constructing high quality software, it is important to formally outline the specification of the domain and the requirements. We do this in RAISE Specification Language, and its RAISE method, to develop industrial software. This language and this method is not formal, but it is rigorous and it is sufficiently flexible to conveniently work without the techniques required by the formal methods.

In this first approach, we give the specifications of the domain, that they will serve for to outline a general scheme to start to define tools, environments and interfaces to obtain a general framework for supporting the Balanced Scorecard management system.

It is desirable, in the future, to develop an automated solid base to take the management of the projects and of the organization in general. Thus, the managers will have the possibility to make decisions with bigger speed and accuracy, as well as to observe the deviations of the reality regarding the desirable thing with more anticipation.

#### References

- [1] Robert Kaplan, David Norton, "Cómo utilizar el Cuadro de Mando Integral: Para implementar y gestionar su estrategia", Ed. Gestión 2000, 2001.
- [2] Nils-Göran Olve, Jan Roy, Magnus Wetter, "Implantando y Gestionando el Cuadro de Mando Integral (Performance Drivers)", Ed. Gestión 2000, 2000.
- [3] Robert Kaplan, David Norton, "The Strategy Focused Organization", Harvard Business School Publishing Corporation, 2001.
- [4] Uzal, R. et al, "A General Architecture for Decision Support Systems Based on Kaplan & Norton Concept of Balanced Scorecard" *Proceedings of the International Conference on Computer and Information Science*, Orlando - Florida - USA, October 2001.
- [5] John Rockart, "Chief Executives define their own data needs", *Harvard Business Review*, March-April 1979.
- [6] Paul Arveson, "What is the Balanced Scorecard?", The Balanced Scorecard Inst., <http://www.balancedscorecard.org>, '98
- [7] Workforce Performance, "Using a Balanced Scorecard Approach to Measure Performance", *Newsletter Reprint*, <http://www.opm.gov/perform/articles/095.htm>, April 1997.
- [8] David P. Norton & Robert S. Kaplan, *Balanced Scorecard Report*, Harvard Business School Publishing, 1999 - 2001.
- [9] Roberto Uzal et al, "Using Balanced Scorecard to Implement an Executive Information System at State Government Level", *Proceedings of the 30 Argentine Conference on Computer Science and Operational Research (JAIIO) Second Argentine Symposium on Software Engineering (ASSE 2001)*, 10 al 14 de Setiembre de 2001 Buenos Aires - Argentina.
- [10] Dines Bjørner, "PIMaC<sup>2</sup>S: A Project Planning, Information, Monitoring, Control & Communication System", *Technical report*, Technical University of Denmark, <http://www.it.dtu.dk/~db/s99>, April 1999.
- [11] Dines Bjørner, "P<sup>2</sup>IMCaCS: A Project Planning, Information, Monitoring, Control & Communication System", *Technical report*, Technical University of Denmark, <http://www.it.dtu.dk/~db/s99>, April 1999.
- [12] Dines Bjørner, "Work Flow Systems", *Technical report*, Technical University of Denmark, 1989.
- [13] The RAISE Language Group, "The RAISE Specification Language", The BCS Practitioner Series, PrenticeHall, 1992.
- [14] The RAISE Method Group, "The RAISE Development Method", The Practitioner Series, PrenticeHall, 1995.

**Spectra in Taxonomic Evidence in Databases III.  
Application in Celestial Bodies. Asteroids Families.**

**Autores:**

*Angel Luis Plastino*  
*PROTEM Laboratory*  
*Department of Physical Sciences*  
*Faculty of Sciences*  
*University of La Plata*  
*C.C. 727 or (115 # 48/49)*  
*(1900) La Plata*  
*Buenos Aires*  
*Argentina*  
*Phone: (54 221) 483-9061 (ext. 247)*  
*E-MAIL: [Plastino@venus.fisica.unlp.edu.ar](mailto:Plastino@venus.fisica.unlp.edu.ar)*

*Rosa Beatriz Orellana*  
*Mechanics Laboratory*  
*Celestial Mechanics Department*  
*Faculty of Astronomical and Geophysical Sciences*  
*University of La Plata*  
*Paseo del Bosque*  
*(1900) La Plata*  
*Buenos Aires*  
*Argentina*  
*Phone: (54 221) 421-7308*  
*E-MAIL: [rorellan@fcaglp.fcaglp.unlp.edu.ar](mailto:rorellan@fcaglp.fcaglp.unlp.edu.ar)*

*Gregorio Perichinsky*  
*Operating System and Database Laboratory*  
*Computer Science Department - South Wing - School of Engineering*  
*University of Buenos Aires [gperi@mara.fi.uba.ar](mailto:gperi@mara.fi.uba.ar)*  
*Paseo Colón N° 850 - (1063) Buenos Aires - Argentina*  
*Phone: (54 11) 4343-0891 (int. 140/145)*  
*FAX: (54 11) 4331-0129*

**Abstract.**

Numerical Taxonomy aims to group in clusters, using so-called structure analysis of operational taxonomic units (OTUs or taxons or taxa) through numerical methods.

These clusters constitute families. Structural analysis, based on their phenotypic characteristics, exhibits the relationships, in terms of degrees of similarity, between two or more OTUs.

Entities formed by dynamic domains of attributes, change according to taxonomical requirements: Classification of objects to form families or clusters.

Taxonomic objects are here represented by application of the semantics of the Dynamic Relational Database Model.

Families of OTUs are obtained employing as tools i) the Euclidean distance and ii) nearest neighbor techniques. Thus taxonomic evidence is gathered so as to quantify the similarity for each pair of OTUs (pair-group method) obtained from the basic data matrix.

The main contribution of the present work is to introduce the concept of spectrum of the OTUs, based in the states of their characters. The concept of families' spectra emerges, if the superposition

principle is applied to the spectra of the OTUs, and the groups are delimited through the maximum of the Bienaymé-Tchebycheff relation, that determines Invariants (centroid, variance and radius).

Applying the integrated, independent domain technique dynamically to compute the Matrix of Similarity, and, by recourse to an iterative algorithm, families or clusters are obtained.

A new taxonomic criterion is thereby formulated.

An astronomic application is worked out. The result is a new criterion for the classification of asteroids in the hyperspace of orbital proper elements (the well-known Families of Hirayama).

Using an updated database of asteroids we ascertain the robustness of the method. Thus, a new approach to Computational Taxonomy is presented, that has been already employed with reference to Data Mining.

The Informatics (Data Mining and Computational Taxonomy), is always the original objective of our researches.

## **1. Introduction.**

Classification is an abstraction technique used to collect objects with common properties.

The following hypothesis: 1) each object belongs to one (and only one) class and 2) for each class at least one object belongs to the classification, allow us to delimit the domain of objects.

The association of concepts in systematic way by recourse to numerical variables has been the source of a great variety of numerical classification techniques, that have their origin in Numerical Taxonomy.

The search of classification concepts that facilitate a robust classification structure (not modifiable by the addition of new information and not altered by the incorporation of new entities) constitutes an important endeavor. In such a line, this work develops tools based on Information Theory and, as a result, a new classification technique is found.

We discuss first principles and methods and then, as an application, we investigate the application of our ideas to the classification of asteroids (an application to celestial bodies), and the classification of plants in biological taxonomy, and the classification of objects of other disciplines.

A correspondence with ideas pertaining to the field of the Dynamic Databases is also established.

Taxonomic objects are here represented by the application of the semantics of the Dynamic Relational Database Model: Classification of objects to form clusters or families [34][35][36][37].

## **2. Conclusions.**

The integrated and dynamical treatment of the domains permits an accessible normalization process, a rapid assignation of attribute - domain - values, the implementation of a model of data base Dynamic and their utilization in Numerical Taxonomy.

Our contribution here is of both a theoretical and an empirical nature. The agglomeration of objects forming classes produced by following the steps of the method (ALGORITHM) presented in this communication allows one to obtain clusters and domains with normalized values. Additionally, the density and range (in terms of the radius of the set) can be visualized as INVARIANT CHARACTERISTICS of the OTUs.

From a taxonomic viewpoint, we introduce the new and original concepts of characteristic spectra of OTUs and families' characteristic spectra.

The states of the characters of the attributes of the OTUs are considered in the spectra using principles of both superposition and interference. For the same purpose we use as well as the density and range concepts. An alternative point of view is that of regarding the radius of the clusters as an invariant characteristic [22] [23] [27] [42].

The methodology introduced here has been applied, as an illustrative example, to the important astronomical problem of asteroids' classification. We are able to reproduce the celebrated families

of Hirayama by straightforward application of our algorithm, avoiding the intuitive and ad-hoc considerations of previous researchers [2] [3] [4] [31] [46].

Measures that allow to determine the stability of a family by its attributes' sensibility have also been studied.

The Families proposed by Hirayama have been validated by the numerical method described above. The spectra have been used in a rather original way so as to confirm that the results obtained with our numerical methodology are congruent (consistent).

Our main contribution lies the use of dynamically independent domains to compute a CHARACTERISTIC SPECTRUM for each OTU (Object) from the Resemblance Matrix. We have computed this matrix by recourse to an iterative algorithm that allows us to obtain the clusters or families.

In the taxonomic space this method of clustering delimits taxonomic groups that can be visualized as FAMILIES' CHARACTERISTIC SPECTRA.

One should be encouraged by the present results to evaluate an Invariant relative to the Centroid :: Density :: Radius cluster for the incorporation of new members to the clusters, and also to design structure using other methods that employ our technology.

### References.

- [1]Abramson,N., "Information Theory and Coding". McGraw Hill. Paraninfo. Madrid. 1966.
- [2]Arnold,J.R., "Asteroids Families and Jet Streams". The Astronomical Journal. 74: pp 1235-1242. 1969.
- [3]Carusi,A., Massaro,E. "On Asteroids Classifications in Families". Astronomy and Astrophysics. Supplements 34, p 81. 1978.
- [4]Carusi,A., Valsecchi,G.B. "On Asteroids Classifications in Families". Astronomy and Astrophysics. pp 327-335. 1982.
- [5]Batini, C., Ceri, S., Navathe, S.B. "Conceptual Databases Design" Addison Wesley. 1998.
- [6]Codd E. F. "Relational Completeness of Data Base Sublanguages". Database Systems, Courant Computer Science Symposia Series 6, Englewood Cliffs, New Jersey, Prentice-Hall. 1972.
- [7]Codd E. F. "Extending the Data Base Relational Model to Capture More Meaning" ACM TODS 4, 4 pp 397-434. 1979.
- [8]Codd E. F. "Relational Data Base. A Practical Foundation for Productivity" CACM 25, 2. 1982.
- [9]Codd E. F. "How Relational is your Database Management System ?". Computer World. 1985.
- [10]Codd E. F. "The Relational Model for Database Management: Version 2". Addison Wesley. 1990.
- [11]Cramer, Harald. "Mathematics Methods in Statistics". Aguilar Edition. Madrid. Spanish. 1958.
- [12]Crisci, J.V. , Lopez Armengol, M.F. "Introduction to Theory and Practice of the Numerical Taxonomy" , A.S.O. Regional Program of Science and Technology for Development. Washington D.C. Spanish. 1983.
- [13]Date,C.J. "An Introduction to Datase Systems Vol. I". 6<sup>th</sup> Ed. Addison Wesley. 1995.
- [14]Date,C.J. "Relational Database: Selected Writings". Addison Wesley. 1986.
- [15]Date,C.J. "Relational Database: Further Misconceptions #1". Info DB, spring, 1986.
- [16]Date,C.J. "A SQL Standard". Addison Wesley. 1987.
- [17] Date,C.J. "Where SQL Falls Short". Datamation pp 84-86. 1987.
- [18]Date,C.J. "An Introduction to Datase Systems Vol. I". 5<sup>a</sup> Ed. Addison Wesley. 1998.
- [19]Date,C.J. "Date on Databases" On proceeding of the Codd & Date Relational Database Symposium". Madrid. 1992.
- [20]de Miguel, A., Piattini, M. "Concepts and Design of Databases." Addison Wesley.1994. Spanish.
- [21]Elmasri,R., Navathe,S. "Fundamentals of Database Systems".The Benjamin/Cummings Publishing Company and Addison Wesley. 1997.

- [22] Feynman, R.P., Leighton, R.B. & Sands, M. "Lectures on physics, Mainly Mechanics, Radiation and Heat". pp. 25-2 ff, 28-6 ff, 29-1 ff, 37-4. 1971.
- [23] Frank, N.H. "Introduction to Mechanics and Heat". Science Service. Washington. Editorial Atlante. Spanish. 1949.
- [24] Freeman, J.A., Skapura, D.M. "Neural Networks. Algorithms, applications and techniques of programming". Addison Wesley. Iberoamericana. Spanish. 1991.
- [25] Gennari, J.H. "A Survey of Clustering Methods" (b). Technical Report 89-38. Department of Computer Science and Informatics. University of California., Irvine, CA 92717. 1989.
- [26] Hamming, R.W. "Coding and information theory". Englewood Cliffs, NJ: Prentice Hall. 1980.
- [27] Hetcht, E. and Zajac, A., "Optic". Fondo Educativo Interamericano. pp. 5-11-206-207-293-297-459-534. Spanish 1977.
- [28] Hirayama, K. "Groups of Asteroids Probably Common Origin". Proceeding of Physics-Mathematics Society. Japan II:9. pp 354-351. 1918.
- [29] Hirayama, K. "Groups of Asteroids Probably Common Origin". The Astronomical Journal: 31, pp 185-188. 1918.
- [30] Hirayama, K. "Present State of the Families of Asteroids". Proceeding of Physics-Mathematics Society. Japan II:9. pp 482-485. 1933.
- [31] Knêzević, Z., Milani, A. "Asteroids Proper Elements from an Analytical Second Order Theory". Astronomy and Astrophysics. pp 1073. 1990.
- [32] Kohonen, T. "Self - Organization and Associative Memory". Berlin: Springer - Verlag. 1989.
- [33] Perichinsky, G. "Multiple states of multiple state automata to key fast validation". 11th. International Symposium Computer at University. Catvat. Zagreb. Yugoslavia. 1989.a.
- [34] Perichinsky, G., Servetto, A., Crocco, E. "Relational Data Bases Structured on Dynamic Domains of Attributes". 18th Sessions. Operative Research and Informatic's Argentine Society. SADIO. 1989.b.
- [35] Perichinsky, G., Servetto, A. "Dynamically Integrated Independent Domains on Data Bases". 19th Sessions Operations Research and Informatic's Argentine Society. SADIO. 1990.
- [36] Perichinsky, G. et Al. "Data Base Model Manager Structured on Independent Domains". Faculty of Science. National University of La Plata. Spanish. 1992.
- [37] Perichinsky, G., Feldgen, M., Clúa, O. "Conceptual Contrast of Dynamic Data Bases with the Relational Model" in Proceedings International Association of Science and Technology for Development. 14<sup>th</sup> Applied Informatics Conference. Innsbruck, Austria. 1996.
- [38] Perichinsky, G., Feldgen, M., Clúa, O. "Dynamic Data Bases and Taxonomy" in Proceedings International Association of Science and Technology for Development. 15<sup>th</sup> Applied Informatics Conference. Innsbruck. Austria. 1997.
- [39] Perichinsky, G., Jimenez Rey, E.; Grossi, M. "Domain Standardization of Operational Taxonomic Units (OTUs) on Dynamic Data Bases" in Proceedings International Association of Science and Technology for Development. 16<sup>th</sup> Applied Informatics Conference. Garmisch-Partenkirchen. Germany. 1998.
- [40] Perichinsky, G., Jimenez Rey, E.; Grossi, M. "Spectra of Objects of Taxonomic Evidence on the Dynamic Data Bases" in Proceedings International Association of Science and Technology for Development. 16<sup>th</sup> Applied Informatics Conference. Garmisch-Partenkirchen. Germany. 1998.
- [41] Perichinsky, G., Jimenez Rey, E.; Grossi, M. "Application of Dynamic Data Bases in Astronomic Taxonomy" in Proceedings International Association of Science and Technology for Development. 17<sup>th</sup> Applied Informatics Conference. Innsbruck. Austria. 1999.
- [42] Sawyer, R.A. "Experimental Spectroscopy". Dover Publication. New York. 1963.
- [43] SEI "Rationale for SQL Ada module Description language (SAMeDL)" Ver. 2.0 CMU/SEI-92-TR-16, oct 1992.
- [44] Sokal, R.R., Sneath, P.H.A. "Numerical Taxonomy". W.H. Freeman and Company. 1973.
- [45] Wiederhold, G. "Data Base Design". McGraw-Hill Book Company. 1983. Edition 1985.

- [46]Williams,J.G. "Asteroids Family Identifications and Proper Elements". Jet Propulsion Laboratory. Palomar-Leiden minor planets. Asteroids Edition. University of Arizona Press. p 1034. 1989.
- [47]Fenton, N.E., Pflieger, Sh.L. "Software Metrics". PWS Publishing Company. 1997.
- [48]Knezevic,Z., Milani,A. Farinella,P., Froehle,Ch., Froehle, Cl, "Asteroids Family Identifications and Proper Elements". Icarus, 93, 316. 1991.
- [49]Kitchenham,B., Pickard,L., Pflieger, S.L. "Case studies for method and tool evaluation". IEEE Software, 12(4) pp 52-62. 1995.
- [50]Milani,A. "Asteroids Family Identifications and Proper Elements".Celestial Mechanics. 57,59. 1993.
- [51]Williams,J.G., "Asteroids Family Identifications and Proper Elements".Icarus. 96,251. 1992a.
- [52]Williams,J.G., "Asteroids Family Identifications and Proper Elements".Icarus. Sbm. 1992b.
- [53]Zappala, V, Cellino,A., Farinella,P., Knêzevíc,Z. "Asteroid Families. I. Identification by Hierarchical Clustering and Reliability Assessment " The Astronomical Journal, 100, 2030. 1990.
- [54]Zappala, V, Cellino,A., Farinella,P., Milani,A., "Asteroid Families. II. Extension to Unnumbered Multiopposition Asteroids" The Astronomical Journal, 107, 772. 1994.
- [55]Perichinsky, G., Orellana, R., Plastino, A.L., Jimenez Rey, E. y Grossi, M.D. "Spectra of Taxonomic Evidence in Databases." Proceedings of XVIII International Conference on Applied Informatics. (Paper 307-7-1 ).Innsbruck. Austria. 2000.

## Integración de los meta-modelos de Workflow y UML

Daniel Riesco  
driesco@unsl.edu.ar  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950  
5700 – San Luis – Argentina  
Tel: + 54 (0) 2652 – 424027 ext. 251  
Fax: + 54 (0) 2652 – 430059

Edgardo Acosta, Marcelo Ariel Uva, Adela Grando  
eacosta@dc.exa.unrc.edu.ar, uva@dc.exa.unrc.edu.ar  
Universidad Nacional de Río Cuarto, Argentina

### Resumen

La identificación y el modelado de los procesos de negocio son de gran importancia en el desarrollo de cualquier industria. Las tecnologías Workflow, que comenzaron a tomar empuje a mediados de la década pasada, apuntan a la automatización total o parcial del proceso de negocio. Actualmente existe un estándar, propuesto por la WfMC (Workflow Management Coalition) para el modelado y especificación de procesos: la gramática de WPD (Workflow Process Definition Language). Dentro del estándar de UML (Unified Modeling Language), propuesto por la OMG (Object Management Group), se utilizan los Diagramas de Actividades para el modelado de procesos pero, aunque cubren aspectos fundamentales, existen detalles que estos diagramas no alcanzan a modelar. Esta línea de investigación propone una extensión al meta-modelo de grafo de actividades de UML para equiparar su poder expresivo con el de WPD y permitir cumplir el estándar de Workflow.

### Introducción

La identificación y el modelado de la cadena de actividades necesarias para generar un producto o ejecutar un servicio es de gran importancia en el desarrollo de cualquier industria. Un conocimiento claro y ordenado del proceso de negocio facilita su optimización y adaptación. Esta característica, que provee a la empresa dueña del proceso de una gran capacidad de reacción, resulta aún más apreciable si el proceso está automatizado [1,2].

Las tecnologías de administración de flujos de trabajo (Workflow Management - WFM) han tenido un crecimiento notable en una gran variedad de industrias [4]. Día a día son más los productos que ingresan al mercado aprovechando la riqueza de las tecnologías WFM para el modelado y ejecución de procesos de negocios.

La característica principal que presentan los sistemas Workflow es la automatización, en parte o total, de los procesos de negocios logrando una interacción entre los recursos humanos y aquellos basados en máquinas [6].

Ante la gran diversidad de productos Workflow surge la necesidad de un lenguaje común que posibilite la interoperabilidad entre dichos productos.

Con el fin de establecer una estandarización para la automatización de procesos de negocio, un grupo de compañías se reúne para formar la WfMC. El estándar propuesto por la WfMC incluye un meta-modelo de los procesos Workflow y un lenguaje de especificación (WPD) basado en ese meta modelo [5].

La notación de diagramas de actividades impone cierto nivel de abstracción en el modelado de procesos organizacionales [3]. Algunos detalles no pueden ser modelados y quedan espacios en blanco

donde pueden surgir ambigüedades y/o inconsistencias. Es por ello que una especificación de proceso basada en diagramas de actividades no puede ser interpretada y ejecutada automáticamente. Esta limitación es superada por la notación del meta modelo Workflow. Una especificación de proceso escrita en WPDL no sólo puede ser interpretada y ejecutada automáticamente sino que, por tratarse de una notación estándar, puede ser intercambiada por distintas herramientas sin perder ni alterar su semántica.

En esta línea hace referencia a dos alternativas para el modelado de procesos de negocios y se propone la extensión de una de ellas para equipararla con la otra. La primera alternativa utiliza como herramienta los diagramas de actividades definidos en el meta-modelo de UML y la segunda hace uso de los elementos presentes en el meta-modelo Workflow.

La propuesta de este trabajo de investigación es desarrollar una extensión al meta-modelo de grafo de actividades de UML para aumentar su poder expresivo y para que cumpla con el estándar de Workflow.

## 1. UML

El Lenguaje de Modelado Unificado UML, es un lenguaje estándar para el modelado de software, el cual permite la visualización, especificación, construcción y documentación de los elementos intervinientes en el modelado de un sistema [8, 10, 11].

También da a los desarrolladores la posibilidad de visualizar los resultados de su trabajo en esquemas estandarizados. UML proporciona elementos, relaciones y diagramas. Dentro de esta última categoría podemos encontrar nueve tipos de diagramas diferentes: de Casos de uso, de Clases, de Colaboración, de Objetos, de Estados, de Actividades, de Componentes y de Despliegue.

### 1.1 Diagrama de Actividades

Los diagramas de actividades definidos por la especificación de UML pueden ser utilizados para dos objetivos diferentes, uno de ellos modelar los aspectos dinámicos de los sistemas, es decir modelar los procesos computacionales. El segundo uso para los diagramas de actividades se aplica al modelado organizacional, para la ingeniería de procesos de negocio y modelado de flujos de trabajo.

Al modelar un flujo de trabajo se pone especial interés en las actividades, tal y como son percibidas por los actores que colaboran con el sistema. A menudo en el entorno de los sistemas con gran cantidad de software existen flujos de trabajo utilizados para visualizar, especificar, construir y documentar procesos de negocio que implican al sistema que se está desarrollando. En este uso de los diagramas de actividades es particularmente importante el modelado del flujo de objetos.

El contexto donde reside un sistema incluye actores que interactúan con él. Es posible encontrar sistemas automáticos que trabajen en el contexto de procesos de negocio de más alto nivel, como es el caso de software de empresas para misiones críticas. Estos procesos de negocio son del tipo de workflow ya que muestran el flujo de trabajo y de objetos a través del negocio. Los diagramas de actividades pueden utilizarse para modelar los procesos de negocio en los que colaboran éstos sistemas automáticos y humanos.

Un diagrama de actividades es básicamente una proyección de los elementos de un *grafo de actividades*, un caso especial de máquina de estados en la cual la mayoría de los estados son estados de actividad y en la cual todas o casi todas las transiciones se disparan al terminar la acción en el estado origen. Los diagramas de actividades aplican todas las características de las máquinas de estado, lo que significa que el proceso modelado por un diagrama de actividades puede contener estados simples y compuestos, bifurcaciones, divisiones y uniones.

La OMG (Object Management Group) define mediante un diagrama de clases la sintaxis abstracta para los diagramas de actividades.

## 2. Workflow

La administración del flujo de trabajo (Work Flow Management) es una tecnología que está evolucionando rápidamente y está siendo explotada en una gran variedad de industrias. Su principal característica es la automatización de procesos que involucran combinación de actividades humanas y basadas en máquinas, particularmente aquellas en las que hay interacción con aplicaciones y herramientas IT (Information Technology).

Sin embargo no había aún estándares definidos para permitir a diferentes productos WfMC trabajar conjuntamente, lo que dio como resultado procesos automatizados e incompatibles entre sí. Por ello es que surge la WfMC (Workflow Management Coalition), establecida para identificar las áreas de funciones comunes y desarrollar especificaciones apropiadas para la implementación de productos workflow. Se pretende que esas especificaciones permitan la interoperabilidad entre productos workflow heterogéneos y mejoren la integración de aplicaciones workflow con otros servicios IT tales como correo electrónico y administración de documentos. De esa forma se mejorarían las oportunidades para el uso efectivo de tecnología workflow dentro del mercado IT, para beneficio de vendedores y usuarios de ambas tecnologías.

### 2.1. Modelo de Referencia Workflow

Existen en la actualidad gran cantidad de productos workflow pero a pesar de ello es posible encontrar un modelo genérico el cual engloba las características comunes a todos ellos. A partir de ese modelo genérico de workflow se ha desarrollado el modelo de referencia de workflow [12], identificando las interfaces dentro de las estructuras que permiten a los productos interoperar en una variedad de niveles. Para lograr interoperabilidad entre productos workflow es necesario un conjunto de interfaces estandarizadas y de formatos de intercambios de datos entre esos componentes.

Para esta línea de trabajo solo interesa la Interfaz 1 [13], vinculada con la definición de procesos. Es el meta-modelo definido en esta interfaz sobre el cual se debe trabajar en esta línea de investigación, y para ello definir el meta-modelo siguiendo la forma que lo hace la OMG, para luego utilizar dichos conceptos e incorporarlos como una extensión del meta-modelo del grafo de actividad que está definido en el estándar UML de la OMG.

## 3. Conclusiones

En esta línea de investigación propone extender el meta-modelo de grafos de actividades de UML para que cumpla con el estándar de workflow.

El objetivo básico de la extensión es permitir el modelado de procesos haciendo uso de los conceptos de diagramas de actividades de UML más algunos otros propios de workflow que se agregan. Esto posibilita que los procesos modelados por medio de la extensión verifiquen el estándar definido por la WfMC logrando interoperabilidad y portabilidad con otros procesos definidos por medio de ese estándar.

Al agregarse y especializarse clases del meta-modelo de grafos de actividad de UML además de lograr el cumplimiento del estándar workflow se logra adaptar UML a las nuevas tecnologías Workflow y se amplía su poder expresivo en cuanto a modelado de procesos de negocio.

Esta extensión permite que los diagramas de actividad de UML incorporen: la posibilidad de ser simulados, la capacidad de definir/decidir en tiempo de ejecución el participante ejecutor, la opción de asignarles un recurso individual, lo cual no están contemplados en el meta-modelo UML.

La extensión no sólo agrega elementos sino que también debe impone restricciones para excluir situaciones permitidas en los grafos de actividades pero no en workflow. Estas restricciones deben seguir con el estándar OMG por lo cual deben ser definidas usando OCL.

## References

- [1] Davenport, T.H. y Young, J.E., "The New Industrial Engineering: Information Technology and Business Process Redesign", Sloan Management Review, 1990.
- [2] Champy, J. "Reengineering Management", Harper Business, 1995
- [3] Jacobson, I, Booch, G., Rumbaugh, J, "The Unified Software Development Process", Addison-Wesley, 1999.
- [4] Embedded & Autonomous Workflow: A WfMC White Paper, [www.wfmc.org](http://www.wfmc.org)
- [5] R. Allen. "The Workflow Handbook 2001". Published in association with the Workflow Management Coalition (WfMC) Oct. 2000.
- [6] Workflow: An Introduction. Rob Allen, Open Image Systems Inc., United Kingdom Chair, WfMC External Relations Committee.
- [7] Hammer, M. and Champy, J. "Reengineering the Corporation: A Manifesto for Business Revolution", Harper Collins Publishing, Inc., 1993
- [8] Booch, G., Rumbaugh, J, Jacobson, I, "The Unified Modeling Language User Guide", Addison-Wesley, 1999.
- [9] Jacobson, I. "Objectifying Business Process Reengineering", Addison Wesley, 1996
- [10] OMG, "Unified Modeling Language Specification", [www.omg.org](http://www.omg.org).
- [11] James Rumbaugh, Ivar Jacobson, and Grady Booch, "The Unified Modeling Language Reference Manual", Addison Wesley, 1999.
- [12] David Hollingsworth, The Workflow Management Coalition Specification Workflow Management Coalition. The Workflow Reference Model. Document Number TC00-1003. Document Status - Issue 1.1. 1995, [www.wfmc.org](http://www.wfmc.org).
- [13] Work Group 1, Workflow Management Coalition Interface 1: Process Definition Interchange Process Model, Document Number WfMC TC-1016-P - Version 1.1 (Official release), 1999, [www.wfmc.org](http://www.wfmc.org).

## Extensiones al Meta-modelo UML Desarrollando Nuevos Esterotipos

Daniel Riesco  
[driesco@unsl.edu.ar](mailto:driesco@unsl.edu.ar)  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950  
5700 – San Luis – Argentina  
Tel: + 54 (0) 2652 – 424027 ext. 251  
Fax: + 54 (0) 2652 – 430059

Alicia Grumelli, Alcides Macció, Paola Martellotto  
[agrumelli@exa.unrc.edu.ar](mailto:agrumelli@exa.unrc.edu.ar), [pmartellotto@exa.unrc.edu.ar](mailto:pmartellotto@exa.unrc.edu.ar)  
Departamento de Computación, Universidad Nacional de Río Cuarto, Argentina

### Resumen

---

UML es un lenguaje universal de modelado de aplicaciones orientadas a objetos que cubre un amplio rango de dominios.

Sin embargo, ningún paradigma es suficiente para expresar claramente todas las vistas de los dominios posibles en el mundo. Por ello, se han hecho muchas extensiones a UML que reflejan dominios específicos.

En el estándar de la OMG (Object Management Group) se proveen tres mecanismos para extender el meta-modelo UML: valores etiquetados, restricciones y estereotipos.

Esta línea de trabajo se propone complementar uno de los mecanismos de extensión, los estereotipos, con otro, las restricciones, ampliando además la posibilidad de expresar semántica adicional a los mismos. Se profundiza en el análisis de tres estereotipados existentes y se propone trabajar en una nueva extensión, *los estereotipos evolutivos*, los cuales permiten extender el meta-modelo a través del establecimiento de una *interacción* entre el meta-modelo, sus desarrolladores y las herramientas CASE sobre UML. Los estereotipos evolutivos permiten integrar no solo la posibilidad de especificar restricciones OCL al Meta-modelo sino también definir ampliaciones semánticas al mismo. Estos estereotipos serán utilizados en una herramienta que permite definir dinámicamente nuevos estereotipos a partir del meta-modelo UML.

---

### 1. Introducción

UML es un lenguaje de modelado estándar aplicable a diferentes tipos de sistemas, dominios, métodos y procesos [OMG02] [BOO00] [JAC00]. Es un lenguaje muy expresivo que cubre las diferentes vistas necesarias para desarrollar sistemas. Sin embargo, ningún paradigma es suficiente para expresar claramente todos los matices de todos los modelos en todos los dominios. Por lo tanto, UML ha sido diseñado para ser abierto-cerrado, haciendo posible extender el lenguaje de forma controlada.

Los estereotipos han sido ampliamente utilizados para distintos dominios particulares. En Hypermedia Design Model [BAR01] se define diferentes estereotipos UML para instanciar el modelo de aplicaciones web. En otro tipo de dominio totalmente diferente, como lo es el diseño de sistemas de tiempo real orientados a objetos, su aplicación es realmente interesante. En [MIG97] se especifican

diferentes estereotipos para sistemas de tiempo real, aunque su especificación simplemente no se hace utilizando el lenguaje natural. En [MCL98] se presentan extensiones a UML que permiten solucionar las deficiencias del mismo en áreas críticas para el desarrollo de sistemas de tiempo real, tales como tiempo (timing), concurrencia e interfaces hardware/software. Una integración más interesante, aunque no lo define explícitamente como estereotipos, es la dada en [HO00] donde asocia elementos de su propuesta de framework con elementos del metamodelo de UML asociándole restricciones OCL.

El GDPL es un lenguaje para capturar la estructura y comportamiento de protocolos de comunicación. Este lenguaje está basado en UML. En [PAR00] se muestra el encadenamiento y los distintos compiladores necesarios para usar una herramienta comercial como el Rational Rose [ROS00]. Se define y se utiliza diferentes estereotipos cuya semántica es especificada externamente a la herramienta Rational Rose a través de los distintos compiladores (R2D2, TPDL2SDL, SDT).

## 2. Estereotipos evolutivos

Los estereotipos evolutivos son incorporados en las herramientas de modelado para que los desarrolladores de modelos puedan modificar el meta-modelo UML, incorporando nuevos elementos con su correspondiente semántica.

Estos estereotipos no necesitan la codificación a mano de un módulo para chequear el modelo ni de un módulo para generar código, las herramientas que soportan estos tipos de estereotipos los codifican automáticamente. Además, proveen mecanismos de simulación para modelar los aspectos dinámicos de los modelos creados. Además, no solamente extienden la semántica del meta-modelo, sino también permiten cambiar la semántica de los elementos que ya están definidos en el meta-modelo, a través de la auto modificación del mismo código que los define. De esta manera el ambiente de la herramienta cambia dinámicamente su apariencia y funcionalidad para permitir a los usuarios utilizar en los diagramas las extensiones realizadas en el meta-modelo.

Los estereotipos evolutivos surgen como una propuesta alternativa para extender el meta-modelo UML, proveen la capacidad de interactuar con el desarrollador para incorporar nuevos elementos a dicho meta-modelo. A continuación se presenta la clasificación de estereotipos propuesta por Schleicher y Westfechtel [SCH01].

### 2.1. Clasificación de los estereotipos

Schleicher y Westfechtel distinguen las siguientes clases de estereotipos:

- **Estereotipos decorativos:** son usados para adaptar la notación de UML a un dominio específico, carecen de toda clase de soporte en el meta-modelo.
- **Estereotipos descriptivos:** introducen nuevos elementos que no cambian la semántica de UML. Sirven para expresar elementos del dominio subyacente, es decir, permiten al usuario crear modelos UML usando elementos del meta-modelo del dominio específico.
- **Estereotipos restrictivos:** son nuevos elementos semánticos agregados a UML, no cambian la base de UML, solamente la extienden. Van más allá de los estereotipos descriptivos asociando restricciones que son definidas declarativamente en vez de utilizar código.

Sin entrar en detalles en cada uno de ellos, describimos los estereotipos evolutivos.

### 3.2 Estereotipos evolutivos

Los estereotipos evolutivos son incorporados en las herramientas de modelado para que los desarrolladores puedan modificar el meta-modelo UML, incorporando nuevos elementos con su

correspondiente semántica. De esta manera el ambiente de la herramienta cambia dinámicamente su apariencia y funcionalidad para permitir a los usuarios utilizar en los diagramas las extensiones realizadas en el meta-modelo.

Estos estereotipos permiten el acceso a la definición del meta-modelo para realizar las extensiones (sintáctica y semánticamente). Como ejemplo, el lenguaje de programación Delphi provee sentencias tales como “Class”, que permiten el desarrollo del *Meta-modelo*. Con ellas es posible definir el objeto *ModelElement* del Meta-modelo de UML de la siguiente manera:

```

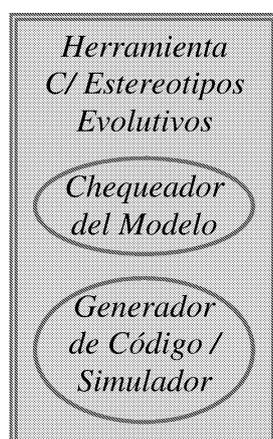
type
  ModelElement = class (Element)
    Name: String;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

```

De esta manera es posible implementar todas las clases y relaciones que componen el meta-modelo para UML.

La incorporación de estereotipos determinará la creación de nuevas clases en el meta-modelo. Los estereotipos incorporados no son únicamente decorativos sino que introducen nueva semántica al meta-modelo. Para ello se requiere de un mecanismo formal para la definición de la semántica; así, se utilizan reglas OCL y declaraciones de atributos y métodos Delphi.

Gráficamente, una herramienta que soporta estereotipos evolutivos tiene la siguiente estructura.



Como se puede observar, tanto el módulo de chequeo del modelo como el módulo de generación de código están incorporados en la herramienta, lo cual facilita la tarea del desarrollador a la hora de extender el meta-modelo UML; a diferencia de otras propuestas, como la de [PAR00], donde se deben implementar compiladores separados para definir la semántica de los estereotipos incorporados.

Además, el módulo de generación de código incorpora la posibilidad de modelar los aspectos dinámicos de los diagramas.

### 3. Conclusiones

Si bien UML es un lenguaje universal para modelar aplicaciones en un amplio rango de dominios, muchas veces no es suficiente para expresar ciertos posibles dominios del mundo. Por ello

UML es un lenguaje abierto que permite extender su semántica y su sintaxis para modelar ese tipo de aplicaciones específicas.

Esta línea de investigación está centrada en desarrollar teóricamente y prácticamente, a través de una herramienta CASE, a los estereotipos evolutivos. Estos son un tipo de extensiones al meta-modelo que se integran naturalmente a UML. Incorpora el módulo de chequeo del modelo para asegurar la consistencia del modelo con respecto al meta-modelo y el módulo de generación de código/simulación, que permite generar código automáticamente a partir de la definición del nuevo estereotipo y mostrar los aspectos dinámicos del modelo a través del simulador.

Como se ha visto, en otros trabajos se utilizan los estereotipos como extensiones a UML, pero sólo los estereotipos restrictivos, que a diferencia de los evolutivos, no permiten asociar una semántica a los elementos incorporados. Una ventaja de los estereotipos evolutivos es que permiten especificar la semántica de los elementos nuevos y generar código automáticamente, manteniendo la consistencia con el modelo UML.

La interacción con el usuario es otra gran ventaja, los estereotipos evolutivos son incorporados en las herramientas de modelado, a través de la interacción entre los usuarios y la herramienta, para que éstos puedan modificar el meta-modelo UML, incorporando nuevos elementos con su correspondiente semántica. Así, el ambiente de la herramienta cambia dinámicamente su apariencia y funcionalidad para permitir a los usuarios utilizar en los diagramas los estereotipos previamente definidos. Las extensiones evolutivas al meta-modelo de UML en una herramienta integrada, facilitan la interacción entre los modelos, los desarrolladores de modelos y las herramientas de desarrollo.

#### 4. Referencias

- [OMG01] OMG: *OMG Unified Modeling Language Specification*. [www.omg.org](http://www.omg.org). 2000.
- [BOO00] Grady Booch, James Rumbaugh, Ivar Jacobson: *El Lenguaje Unificado de Modelado*. S.A., Madrid, 2000.
- [JAC00] Ivar Jacobson, Grady Booch, James Rumbaugh: *El Proceso Unificado de Desarrollo de Software*. S.A., Madrid, 2000.
- [MCL98] Michael J. McLaughlin, Alan Moore: *Real Time Extensions to UML. Timing, concurrency, and hardware interfaces*. Diciembre de 1998.
- [BAR01] Luciano Baresi, Franca Garzotto, Paolo Paolini, Extending UML for Modeling Web Applications, Proceedings of the 34th Hawaii International Conference on System Sciences, IEEE Press, 2001.
- [MIG97] Miguel de Miguel, Alejandro Alonso, Juan de la Puente, Object-Oriented design of Real time Systems with Stereotypes, Proceedings of the 9th Euromicro Workshop on Real Time Systems, IEEE Press, 1997.
- [HO00] Wai-Ming Ho, Francois Pennaneac'h, Noel Plouzeau, UMLAUT: A Framework for Weaving UML-Based Aspect-Oriented Designs, Proceedings of the Technology of Object-Oriented Languages and Systems (TOOLS 33), IEEE Press, 2000.
- [PAR00] Juha Parssinen, Niklas von Knorring, Jukka Heinonen, Tellabs Oy, Markku Turunen, "UML for Protocol Engineering - Extensions and Experiences", Proceedings of the Technology of Object Oriented Languages and Systems (TOOLS 33), IEEE Press, 2000.
- [ROS00] Rational inc. Rational Rose 2000. [www.rational.com](http://www.rational.com)
- [SCH01] Schleicher y Westfechtel: *Beyond Stereotyping: Metamodeling Approaches for the UML*. 2001. Página 4. 34<sup>th</sup> Hawaii international conference on system Sciences.

## Modelling of Processes and Formal Methods

D. Riesco, G. Montejano, R. Uzal

Universidad Nacional de San Luis  
Departamento de Informática  
Ejército de los Andes 950  
5700 San Luis - Argentina  
driesco@unsl.edu.ar

### Abstract

We proposed here a technique, which can be employed within the methodology known as process reengineering. This technique was applied in a government environment, which included an Information System and Geographical Information System, developed with financial support from The World Bank.

One model used in process reengineering is the process model diagram. It allows finding the tasks, to be completed in each area of the organisation. To understand the domain is crucial to be able to specify each one of these tasks.

We show here how to use modelling of processes to find the tasks and to formalise their description using RAISE formal method. In this way, using a model of process as input, an engineer employs a systematic technique to create –as a starting point– the main functions (tasks) of the domain using the RAISE formal method.

**Keywords:** Reengineering of Processes, Reverse Engineering, Formal Method, RAISE Language

### 1. Introduction

Business Process Reengineering (BPR) is a method to radically redesign the way in which an organisation performs its tasks [2,3]. BPR is not a method for system development, but a method, which helps in developing a new system, from an old one, which changes the way to do business.

There are two tendencies in the proposed approaches for Process Reengineering [8]. The first of them has been called “white page”. It proposes to ignore the past as a conditioning element and emphasises that nowadays the outstanding quality of a manager is to forget (probably what he has “learnt” will serve for nothing).

Another tendency is the one that includes “Reverse Engineering”, that is, the study of the previous situation to the Process re-planning.

BPR leads to the following changes in:

- Process, the way that the people do the business.
- The software, upgrades of a new system, and how it supports the Process’ Tasks.
- Interface, radical changes in the applied technology.
- Hardware, upgrades to client/server architecture.
- Data, transition to a Relational/Object Oriented Data Base.
- People, change the way the people work, which is affected by the reengineering of processes.

In the implementation of a new system, there are different degrees of changes [7], which show the consequences of processes reengineering:

- Reaffirm: explicit decision to change nothing.
- Repack: change of the user interface (i.e.: move from character interface to GUI).
- Rehost: change of hardware platform (i.e.: migration from centralised environment with mainframe to client/server environment).

- Re-architecture: change of technology kernel without explicitly affecting the way to do the business.
- Reengineering: change the way to do business including all points above.

Therefore, applying process reengineering implies a big change, which affects the way to do business. One of the basic tools applied in reengineering of processes is modelling of process (described in section 2).

The main advantage of modelling of processes is to be able to visualise the way the organisation works, showing the information flow and its different transformations, among different organisation areas.

One of the disadvantages is that the method is informal. Hence, the proposal, presented here, is to combine it with RAISE and obtain all the advantages of a formal method. Section 3 gives a brief description of RAISE formal method, and the proposed technique.

## 2. Modelling of Processes

Modelling of processes is one of the tools used in process reengineering. This tool allows visualising the processes and flows as well as the organisation areas affected by the processes. It is used conceptually to show what actually happen with the organisation (domain engineering), as well as to be able to visualise what will happen to the new managerial restructuring.

A diagram is created, which represents the areas, departments or sections of the company, showing the processes, data flows and data storages, which belong to each section (or what will happen in the new reorganisation). The tool allows decomposing the diagram of a high level process into other sub processes. The diagram gives a visual representation of the new business activities or those that already exist.

Some key concepts are defined below:

- A Process may be defined as a set of activities that are interlaced or chained. The Process transforms a stimulus born in an external entity to the organisation, into an answer to that external entity. The Process rules the relation of the boundary studied with its environment. In principle, each Process must satisfy one of the components of the demand of the external medium (stimulus generated out of the studied area) that arrive to the organisation area of study.
- Task or Activity is every action conveniently related that constitutes a Process. When a Process is studied, its separation in Tasks constitutes an essential step.

The enchainment of Tasks transforms a stimulus of the environment in the answer to it. It is important to do the analysis of the value added by each Task to the Process it is part of. The same Task may form part of different Processes. Each Task and may be associated to a job description.

- Separation of Tasks or Activities in Steps. In case of very complex Processes, the simple separation in Tasks is not sufficient to permit an adequate study; each Task should be separated in Steps. A Step can not be separated.
- Formalisation of Steps. Steps are specified using RAISE. They determine the subtasks, which have to be clearly specified to understand the domain.

## 3. RAISE Formal Method

RAISE [4] is an acronym for “Rigorous Approach to Industrial Software Engineering”. RAISE takes seriously the word “industrial”. The method is intended for use on real development, not just toy examples.

The method is based on a number of principles: separate development, step-wise development, invent and verify, and rigour.

The main point, used in this work, is the concept of separate development. If we want to develop

systems of any size, then we must be able to decompose their description into components and compose the system from the components. Each component can be specified and developed by different engineers. The problem is that an engineer writes a function that others want to use (reuse). It is simple to check the name of the function, which parameters it has and what its result type is. But it is not so easy to be exact about the semantics of the function.

To specify formally the semantic of a function RAISE is used. RAISE's scheme construct is used to specify each component. RAISE uses a language called RSL (RAISE Specification Language) [5]. In RSL, a scheme (module) is basically a named collection of declarations. A module is a class expression. A class expression represents a class (essentially a set) of models. An object is a named model of a class of models represented by some class expression (the instance of a class).

In a class, we can specify types: build-in types, abstract type (sorts), and compound types ( $\times$  for Cartesian Product,  $-$ set for Set,  $*$  for List,  $\mapsto$  for Maps, Records, Variants).

One of the main aspect of a RAISE specification is the definition of functions ( $\text{type\_expr} \rightarrow \text{type\_expr}$ ). The specification can be written in a variety of styles: abstract property oriented style (algebraic), model oriented style and concrete algorithm oriented style as an extreme of the spectrum. The properties can be specified as axioms in the class.

#### 4. Formal Specification of the steps specified in the Modelling of Processes.

The technique proposes to write a formal specification in RSL for each step of each task specified in each Process Model.

For that the following points should be kept in mind:

- A Step is specified by a RSL function.
- The functionalities specified in a step have to be absorbed in a RSL scheme.
- As a heuristic, there will be a scheme for each process model data store. This scheme will contain all functions pertaining to the step that access this data store.
- The properties needed to execute the step are specified with function preconditions.
- It is possible to specify other auxiliary functions within the scheme. This helps to understand the step formalisation.
- If it is needed to use functions of other classes (reuse), the scheme will contain a declaration of this class (object declaration).

Above is presented a model applied to the Reengineering of Processes of a State (Province) Land Register environment, where a GIS (Geographical Information System) maintain information about all parcels (lots) of the state.

The step "Unification of Parcels" is very complex to describe unambiguously in natural language; therefore we specify it using RSL formal language. Unification means to create a new parcel with the following properties:

- The unified parcel area is the sum of the area of the parcels, which are unified.
- The building area in the new parcel is the sum of the building area of each unified parcel.
- The adjacent parcels are the same that the adjacent parcels of each unified parcel. The parcels located to the north of the new parcel correspond to the parcels located to north of each unified parcel. The same will occur with the parcels located to the south, east and west.
- Any other properties are also specified using RSL formal language.

Since RSL is used in the steps specification, the RSLTC (RAISE Specification Language Type Checking) tool is employed to type check it. The use of this tool has the advantage of formally checking a specification that has been applied in one model of reengineering of processes.

Here, we present the formal specification of Step “Unification of Parcels”:

**scheme** PARCEL =

**class**

**type**

Pid, /\* Parcel Identification \*/  
 P, /\* Parcel (Abstract Type)\*/  
 PState, /\* Parcels of State/Province \*/  
 PP = P × P,  
 AdjacentNS = PP-**set**, /\* North-South Adjac.Parcel\*/  
 AdjacentWE = PP-**set**, /\* West-East Adjac. Parcel \*/  
 Adjacents = AdjacentNS × AdjacentWE

**value** /\* Step Specification \*/

unification : P-**set** × Pid × PState → PState, /\* All adjacent parcels of state \*/  
 adjacents : PState → Adjacents,  
 parcels : PState → P-**set**, /\* All parcels of the state\*/  
 adjacentsNS : P × Adjacents → AdjacentNS /\* All adjacents North-South of the parcel \*/  
 adjacentsNS(p, (adjacentNS, adjacentWE)) ≡  
 {(p, s) | s : P • (p, s) ∈ adjacentNS} ∪  
 {(n, p) | n : P • (n, p) ∈ adjacentNS},  
 adjacentsWE : P × Adjacents → AdjacentWE /\* All adjacent West-East of the parcel \*/  
 adjacentsWE(p, (adjacentNS, adjacentWE)) ≡  
 {(p, e) | e : P • (p, e) ∈ adjacentWE} ∪  
 {(w, p) | w : P • (w, p) ∈ adjacentWE},  
 sum : (P → Real) × P-**set** → Real /\* A parcel set is applied a generic function (formal parameter) and sum this result \*/  
 sum(f, ps) **as** result  
**post** ps = {} ⇒ result = 0.0 ∧  
 ps ≠ {} ⇒ **let** p : P • p ∈ ps **in**  
 result = sum(f, ps \ {p}) + f(p)  
**end**,  
 id : P → Pid, /\* Functions about Parcel atributes \*/  
 groundArea : P → Real,  
 buildingArea : P → Real

**axiom**

∀ ps : P-**set**, pid : Pid, pState : PState •  
 unification(ps, pid, pState) **as** pStateR  
**post** ∃p' : P • id(p') = pid ∧  
 groundArea(p') = sum(groundArea, ps) ∧  
 buildingArea(p') = sum(buildingArea, ps) ∧  
 /\* Adjacent Parcels locate to N-S of new parcel\*/  
 adjacentsNS(p', adjacents(pState)) =  
 {(n, p) | n : P, p : P • p ∈ ps ∧  
 (n, p) ∈ adjacentsNS(p, adjacents(pState)) ∧  
 n ∉ ps} ∪  
 {(p, s) | s : P, p : P • p ∈ ps ∧  
 (p, s) ∈ adjacentsNS(p, adjacents(pState)) ∧  
 s ∉ ps} ∧  
 /\*Adjacent Parcels locate to W-E of new parcel\*/  
 adjacentsWE(p', adjacents(pState)) =  
 {(w, p) | w : P, p : P • p ∈ ps ∧  
 (w, p) ∈ adjacentsWE(p, adjacents(pState)) ∧ w ∉ ps} ∪  
 {(p, e) | e : P, p : P • p ∈ ps ∧  
 (p, e) ∈ adjacentsWE(p, adjacents(pState)) ∧  
 e ∉ ps} ∧  
 /\* Rest of the parcels keep same adjacents \*/

$$\exists p : P \cdot p \neq p' \wedge$$

$$p \in \text{parcels}(pState) \wedge p \notin ps \Rightarrow$$

$$\text{adjacentsNS}(p, \text{adjacents}(pState)) =$$

$$\text{adjacentsNS}(p, \text{adjacents}(pState))$$

end

## 5. Conclusions

The technique presented here has all the advantages of the use of formal methods in the first step of the reengineering of processes. We apply the modelling of processes and specify each task of it using a formal specification, in this case using the RAISE formal method.

The technique was applied in a government environment using the Modelling of Processes of Oracle Designer 2000, and the RSLTC (RAISE Specification Language Type Checking) to specify each task described with the Oracle tool.

This technique allows the engineering group, which will do the forward engineering, to have a clear, unambiguous specification of each task done by the organisation. In this way the problem of vagueness inherent in an informal description is avoided which helps to construct a more reliable system. It also facilitates the rigorous specification and analysis of complex software systems.

The RAISE specification can be a contract between the developers and the users, as well as between the developers, in charge of specifying the model of the processes, and the developers, in charge of the forward engineering.

## References

- [1] Jacobson, I. and others "Object Oriented Software Engineering. A use Case Driven Approach", Reading MA Addison Wesley, 1992
- [2] Jacobson, I. "Objectifying Business Process Reengineering", Addison Wesley, 1996
- [3] Jacobson, I. and Lindström F., "Re – engineering of old systems to an object – oriented architecture", OOSPLA'91, pp. 340 – 350
- [4] The RAISE Method Group, "The RAISE Development Method", Prentice Hall, 1995.
- [5] The RAISE Language Group, "The RAISE Specification Language", Prentice Hall, 1992.
- [6] Hammer, M. and Champy, J. "Reengineering the Corporation: A Manifesto for Business Revolution", Harper Collins Publishing, Inc., 1993
- [7] Champy, J. "Reengineering Management", HarperBusiness, 1995
- [8] Manganelli, R. and Klein, M. "The Reengineering Handbook", AMACON, 1994
- [9] Lincoln, T., "Managing Information System for Profit", John Wiley, 1990
- [10] Martin, J. "Information Engineering", PrenticeHall, 90
- [11] McCabe T. J. "A Software Complexity Measure", IEEE Trans. Software Engineering, vol 2, no. 6, Dec76
- [12] Arthur, Lowell Jay, "Rapid Evolutionary Development", Wiley, 1992
- [13] Rockart, John F., "Chief executives define their own data needs", Harvard Business Review, March – April, 79

# A Systematic Approach to Generate Test Cases based on Combinations of Information

Marisa A. Sánchez<sup>1</sup> and Miguel A. Felder<sup>2</sup>

<sup>1</sup> Universidad Nacional del Sur, Bahía Blanca

E-mail: [mas@cs.uns.edu.ar](mailto:mas@cs.uns.edu.ar)

<sup>2</sup> Pragma Consultores, Buenos Aires

Url: <http://www.pragma.com.ar>

E-mail: [mfelder@pragma.com.ar](mailto:mfelder@pragma.com.ar)

## 1 Introduction

Software based systems incrementally provide critical services to users. Mobile telephone systems, for example, are used in circumstances in which the malfunctioning may have disastrous consequences. During the last years, software has been incorporated in devices used in daily life, such as audio and television. The diversity of systems in which software is incorporated is increasing. Thus, the software development process has to consider a variety of specification techniques and models, incorporating also techniques from engineering sciences. In particular, the validation and verification processes have to be adapted to these new developments. For example, the testing based solely on the software specification is incomplete. First, there is an implicit objective to verify that the program works correctly (as in the testing model of the 1957–1978, [GH88]). Myers [Mye83] says that with the aim of demonstrating that a program does not fail, we can unconsciously select data that has a low probability of exposing faults. On the other hand, if the objective is to demonstrate that a program has faults, test data will have a higher probability of revealing them. In specification-based testing we select data for which the desired behavior for the system is defined. As stated by Boris Beizer [Bei95] testing should include both clean and dirty tests. Dirty tests are designed to “break” the software; clean tests are designed to demonstrate that software executes correctly. Specifications only provide clean tests.

It is clear that a system has more possible behaviors than those deduced from the specification. The problem is that, given the diversity of information that we have to consider to understand a system, it is not obvious how to define those behaviors. In this work, we propose a systematic approach generate test cases based on combinations of information:

- information based on the software specification;
- information about the behavior of other system components, such as, memory resources, network availability, deadlock of resources;
- information about different operative conditions.

## 2 Research direction

The main problems that we encounter when we try to deduce information outside the specification are the following:

- (a) For the case of specification-based testing, the number of possible behaviors is bounded by what is described in the specification. If we also consider information outside the specification, the number of possible behaviors is infinite.
- (b) We have to deal with specifications provided in different languages, with different levels of granularity and abstraction, and that they consider different views of the system.

In Sections 2.1 and 2.2 we discuss these points.

### 2.1 Selection of “dirty” behaviors

To address the first point, we propose to characterize possible behaviors and to give a priority according to some criteria. We use Fault Tree Analysis to determine how an undesirable state (failure state) can occur in the system [us881]. This analysis is a widely used technique in industrial developments, and allows to describe how individual component failures or subsystems can combine to effect the system behavior.

A fault tree consists of the undesired top state linked to more basic events by logic gates. Once the tree is constructed, it can be written as a Boolean expression and simplified to show the specific

combinations of identified basic events sufficient to cause the undesired top state. The sets of basic events that will cause the root event are regarded as Minimal Cut Sets.

## 2.2 Integration of different sources of information

Concerning the second point mentioned in the introduction of this section, we have to integrate Fault Tree Analysis results with statecharts. We assume we have a specification of the desired behavior for the system using statecharts [Har87]. This formalism is widely used within the software engineering community, and has been adopted by the Unified Modeling Notation (UML) [DH89,HLN<sup>+</sup>90,BRJ98].

The results of the Fault Tree Analysis are related to the system specified behavior to determine how we can reproduce these scenarios. In our work we propose to interpret each Minimal Cut Set as a Duration Calculus formula [ZHR91]. We defined some conversion rules of a formula to a statechart. These rules are applied to the syntactic categories of Duration Calculus formulas. By combining both sources of information within a common semantic framework, we can systematically build a testing model. The testing model provides a representation of the way the system behavior can be compromised by failures or abnormal conditions or interactions.

In particular, if this conditions refer to a peak activity that exceed system limitations, we are doing stress testing. Stress testing evaluates the behavior of systems that are pushed beyond their specified operational limits [Ngu01]. Stress testing requires an extensive planning effort for the definition of workload, and this involves the analysis of different components that effect system behavior (*e.g.* memory resources, network bandwidth, software failures, database deadlocks, operational profiles). However, this analysis is usually performed ad hoc. We propose to use Fault Tree Analysis helps to define workload scenarios. The results of this analysis are composed with the specification statecharts, and we obtain a model that describes how a given workload can be reproduced.

## 2.3 Reduction of the testing model

The testing model is specified using statecharts. Although its semantics is very intuitive, the inherent complexity of many of today's

applications may lead to large and complex statecharts. To address this problem, we propose to reduce statecharts using slicing techniques.

Program slicing is a technique for decomposing programs by analyzing their data flow and control flow. The traditional definition of slicing is concerned with slicing programs written in imperative programming languages [Wei84]. Therefore, it is assumed that programs contain variables and statements, and slices consist solely of statements. Sloane *et al.* extended the concept of slicing to a generalized marking of a program's abstract tree [SH96]. This generalization allows slicing based on criteria other than the use of a variable at a given statement. We also base our approach to slicing on a marking of the abstract syntax tree, and for this purpose we define a formal grammar to describe correct syntax for statecharts. A slicing criterion of a statechart is defined by a state. The criterion determines a projection on the sequences of the statechart that throws out all states and transitions that do not contribute to reach the state of interest.

## 2.4 Tool support

The testing process can become very tedious, if not unpractical because the amount of information needed to describe a test case is large in most real problems. So it becomes necessary to support any testing approach with tools, otherwise it may be useless for the practitioner. We direct our efforts towards developing an approach that requires as little human intervention as possible. Thus, the resulting approach allows a level of automation that can significantly enhance the productivity of the testing process. Much of the ongoing work is directed at developing tool support.

## References

- [Bei95] Boris Beizer. *Black-Box Testing. Techniques for Functional Testing of Software and Systems*. John Wiley Sons, Inc., 1995.
- [BRJ98] G. Booch, J. Rumbaugh, and I. Jacobson. *The Unified Modeling Language. User Guide*. Object Technology Series. Addison Wesley Longman, Reading, MA, USA, 1998.
- [DH89] D. Drusinsky and D. Harel. Using Statecharts for Hardware description and Synthesis. *IEEE Transactions on Computer-Aided Design*, 8:798–807, 1989.

- [GH88] David Gelperin and Bill Hetzel. The Growth of Software Testing. *Communications of the ACM*, 31(6):687–695, June 1988.
- [Har87] D. Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8:231–274, 1987.
- [HLN<sup>+</sup>90] D. Harel, H. Lachover, A. Naamad, A. Pnueli, M. Politi, R. Sherman, A. Shtultrauring, and M. Trakhtenbrot. Statemate: A Working Environment for the Development of Complex Reactive Systems. *IEEE Transactions on Software Engineering*, 16, 1990.
- [Mye83] G. Myers. *El Arte de Probar el Software*. Librería El Ateneo Editorial, Buenos Aires, 1983. Spanish translation.
- [Ngu01] Hung Q. Nguyen. *Testing Applications on the Web*. John Wiley and Sons, Inc., 2001.
- [SH96] A. M. Sloane and J. Holdsworth. Beyond traditional program slicing. In *Proceedings of the International Symposium on Software Testing and Analysis*, pages 180–186, 1996.
- [us881] Fault Tree Handbook. Nureg-0492, U.S. Nuclear Regulatory Commission, Washington, D.C., Jan. 1981.
- [Wei84] Mark Weiser. Program Slicing. *IEEE Transactions on Software Engineering*, 10(4), July 1984.
- [ZHR91] Chaochen Zhou, C.A.R. Hoare, and Anders P. Ravn. A Calculus of Durations. *Information Proc. Letters*, 40(5):269–276, Dec. 1991.

**INTEGRATED ENVIRONMENT OF SYSTEMS AUTOMATED ENGINEERING.***Autores*

*Arturo Carlos SERVETTO, Ramón GARCÍA MARTÍNEZ, Gregorio PERICHINSKY*  
*aserve/rgm/gperi@mara.fi.uba.ar*  
*Tel. (011) 4342-9184 / 4343-0891, Int. 142*  
*Laboratorio de Bases de Datos y Sistemas Operativos*  
*Facultad de Ingeniería*  
*Universidad de Buenos Aires*

**KEYWORDS:** static models, dynamic models, integrated environment, conceptual pattern.

**ABSTRACT.**

The Standardization of a development process for systems of medium complexity, entirely supported by an integrated development environment, could mean a double contribution to the industry of systems development. On the one hand it would contribute to the reduction of costs by shortening times of development and facilitating the maintenance of the systems. On the other hand, by using conceptual models of a high level of abstraction, final users would be able to visualize and follow the progress of systems development and to commit themselves in the process actively, guaranteeing their success.

The purpose of this series of papers is to identify, design, develop and integrate the components of an integrated environment for a system automated development, starting from high-level-abstraction formal specifications. It is intended to achieve a generation of systems starting from only two models: the static or data structure model, and the dynamic or functional model. The former is based on an adaptation of the conceptual pattern of entities and relationships, and the latter on the formal specification of operations in objects relational algebra and on the finite automaton theory. The maintenance of the systems generated by the tool would be made by operating directly on the static and dynamic models, with no need for either re-coding or making reverse engineering.

The strength of the proposal is based on integration of the Software Engineering and the analysis and metrics of quality. Storage protocols of data definitions and storage protocols of interface definitions corresponding to a graph, through the state transitions and their gradation so that the user could visualize the development of the life cycle[6][32][33][47].

Any artifact or document can be measured and estimated if it is produced during the software life cycle.

This study is an approximation to a process of activities supported by an effective help tool. Abstraction and visualization capacities are provided to the artifacts with the system design and analysis (CASE). Such artifacts under controlled conditions trace the software life cycle taking information of the structure (framework) of the application; with Quality, Reliability and Metrics.

**PROCESS ORIENTED.**

Similarly to what happened with the imperative programming and the structured systems development, at the beginning the Object Oriented (O-O) paradigm was applied exclusively to the programming. The following phase, more or less in the 80s, was characterized by the interest explosion in the interfaces of the user. The most evident effect is reflected in the current abundance of libraries of classes for the development of interfaces[24][36][38].

The current phase of the history of the O-O is characterized by the emphasis which has been moved from programming to analysis and design, and because there is conscience of the problem of the open systems and of the need for standards. There exists an important trend toward the incorporation of methods guided to objects, in the systems of database management as well as in the structured methods of existing management and the CASE tools that give support to systems of database and structured methods[2]. The proposal of this work consists in the specification of a process and behavior model of systems for a CASE tool that, based on the analysis by scenarios and in the classification of objects in a system in objects of application and of interface, synthesizes the

functionality and the behavior of a system in a finite states machine by scenery [1][9][11][17] [18][25] [26][34][35][37][39][45][46][53] .

The states in the model represent objects of interface and the transitions represent messages of activation to other objects of interface or well messages to classes of application that give as a result a set of objects that is associated with an object interface. At the same time each object interface, associated with an empty set, unitary or vast of objects of application, it can have some of these active objects or in area (inclusive null or undefined object).

Storage protocols of data definitions and the definitions grammar corresponding to a graph can be simple (defined attribute, aggregate entity, aggregate attribute, identifier and hierarchy) or composites (attributes, entities and relations). Another protocol are the storage protocols of interface definitions and the definitions grammar corresponding to a graph can be simple (report, attribute value, computed value, conjunction term and code) or composites (intefaces, forms, transitions, relations between forms and a disjunction term).

An adjustment is used for the formal specification of the messages to classes of application of relational algebra, whose operations determine instance subsets of classes of application or joint of instances of temporary classes, created as a result of consulting. The model results apt for the automation of the office-supply systems development or from traditional management and reduces the behavioral specification and functionality of a system to an only graph by all scenarios, synthesizing the scenarios or cases of use, the interaction among objects (sequence graphs and of collaboration) and the state sequences, proposed by the UML (Unified Modeling Language) of Booch, Rumbaugh and Jacobson.

## REFERENCES

- [1] Anfossi, D; Servetto, A., 1997, Review of Object Oriented CASE tools: Comparisons, Evaluation and Conclusion; proceeding of the III ICIE (International Conference on Informatics Engineering). University of Buenos Aires; pp. 449-463. Spanish.
- [2] Bandinelli, S. 1995. Modelling and Improving an Industrial Software Process. IEEE Transaction on Software Engineering. Vol.21, N°5, Pág. 440-454.
- [3] Bass, L. , Clements, P. and Kazman, R. "Software Architecture in Practice". Addison - Wesley. 1998.
- [4] Bell, D. and Morrey, I. "Software Engineering, Second Edition".Prentice Hall.1992.
- [5] Böehm 1981 B. Boehm, Software Engineering Economics, Prentice Hall, 1981.
- [6] Böehm, B. 1988. A Spiral Model for Software Development an Enhancement. IEEE Computer. Mayo, Pág. 61-72.
- [7] Brown, A. W. 1989, Database Support for Software Engineering. New York: Wiley.
- [8] Brown, A. W. & McDermid, J. A., 1991, On Integration and Reuse in a Software Development Environment, F. Long & M. Tedd (Editors), Ellis Horwood, Mar 7.
- [9] Chappell, C., Downes, V. & Tully, T., 1989, Real-Time CASE: The Integration Battle, Ovum.
- [10] Constantine, L. and Lockwood, L. "Software for use". A Practical Guide to the Models and Methods of Usage - Centred Design. Addison - Wesley. 1999.
- [11] Cortese, A., 1989, DEC Challenges IBM CASE Strategy, Computerworld, 23, 41 (Oct),
- [12] Carlyle, R., 1990, Is Your Data Ready for the Repository?, Datamation, 36, 1 (Jan) 43-47.
- [13] Davis, A. y Sitram, P. 1994. A Concurrent Process Model for Software Development. Software Engineering Notes. Vol.19, N°2, Pág. 38-51.
- [14] Fagan M.E., "Advances in Software Inspections," IEEE Transactions on Software Engineering, Vol. 12, No. 7, July, 1986, pp. 744-751.
- [15] Feiler P.H. and Humphrey W.S., Software Process Development and Enactment: Concepts and Definitions, CMU/SEI-92-TR-4, ADA258465, March 1992.
- [16] Fenton, N. and Pfleeger, S. Software Metrics. PWS Publishing Company. 1997.
- [17] Feuche, M. Atherton, K, 1988, DEC to Boost CASE Standard, MIS Week, 9, 19 (May),33.
- [18] Forte, G., 1989, In Search of the Integrated CASE Environment, C/A/S/E Outlook 89, 2.
- [19] Fowler P. and Rifkin S., Software Engineering Process Group Guide, Software Engineering Institute, CMU/SEI-90-TR-24, ADA235784, September, 1990.

- [20] Freedman D.P. and Weinberg G.M., Handbook of Walkthroughs, Inspections, and Technical Reviews, Third Edition, Dorset House, New York, NY, 1990.
- [21] Ghezzi, C. , Jazayeri, M. , Mandrioli, D. "Fundamentals of Software Engineering". Prentice Hall. 1991.
- [22] Gibson, S., 1989a, CASE Buyers Await Repository, Computerworld, 2, 20 (May), 2.
- [23] Gibson, S., 1989b, Some Win, Some Lose When Repository Debuts, Computerworld, 23, 11 (Mar), 141.
- [24] Graham, I. 1994, Métodos Orientados a Objetos. Addison-Wesley / Díaz de Santos,
- [25] Grudin, J., 1989, The CASE Against User Interface Consistency, Communications of the ACM, 32, 10 (Oct), 1164-74.
- [26] Hecht, A. & Harris, M. 1995, A CASE Standard Interchange Format: Proposed Extension to EDIF 2.0, Cadre Technologies Inc.
- [27] Humphrey W.S., "Characterizing the Software Process," IEEE Software, Vol. 5, No. 2, March, 1988, pp. 73-79.
- [28] Humphrey W.S., Managing the Software Process, Addison-Wesley, Reading, MA, 1989.
- [29] IEEE-STD, 1990, Computer Society Task Force on Professional Computing Tools, A standard Reference Model for Computing System Tool Interconnections.
- [30] IEEE-STD-610 ANSI/IEEE Std 610.12-1990, "IEEE Standard Glossary of Software Engineering Terminology," February 1991.
- [31] IEEE-STD 1074-1991 Customer Request IEEE Standard for developing Software Life Cycles Process.
- [32] Juristo, N. y Pazos, J. 1993. Towards a Joint Life Cycle for Software and Knowledge Engineering.
- [33] Kitson D.H. and Masters S., An Analysis of SEI Software Process Assessment Results: 1987-1991, CMU/SEI-92-TR-24, July 1992.
- [34] Margolis, N., 1988, CASE Fights to Beat 'All Talk, No Action' Image, Computerworld, 22, 52 (Dec), 45-48.
- [35] Martin, J., 1990, Integrated CASE Tools a Must for High-Speed Development, PC Week, 6, 3 (Jan), 78.
- [36] Martin, J.; Odell, J.J. 1994, Object-Oriented Analysis and Design, Prentice Hall Inc.
- [37] Myers, E., 1989, CASE Standards Connecting: Efforts Worldwide Attempting Coordination, Software Magazine, 9, 12 (Oct), 23-27.
- [38] Nierstrasz, S. 1992. Component-Oriented Software Development. Communications ACM Vol.35, N°9, Pág. 160-165.
- [39] Philips, B., 1989a, A CASE foe Working Together, The Electronic System Design Magazine, 19, 12 , 55-58.
- [40] Paulk, M.C. Curtis, B. Chrissis, M.B., et al, Capability Maturity Model for Software, CMU/SEI-91-TR-24, ADA240603, August 1991.
- [41] Paulk, M.C. Curtis, B. Chrissis, M.B. and Weber, Charles V. Capability Maturity Model for Software, Version 1.1, CMU/SEI-93-TR-24, February 1993.
- [42] Paulk, M.C. Weber, C.V. Garcia, S. Chrissis, M.B. and Bush, M. Key Practices of the Capability Maturity Model, Version 1.1 (a), CMU/SEI-93-TR-25, February 1993/(Hilburn Thomas 1995).
- [43] Weber, C.V. Paulk, M.C. Wise, C.J. and Withey, J.V. Key Practices of the Capability Maturity Model, CMU/SEI-91-TR-25, ADA240604, August 1991.
- [44] Pfleeger, S. "Software Engineering: Theory and Practice". Prentice Hall. 1998.
- [45] Phillips, B., 1989b, Software and CASE, Electronic Design, 37, 1 (Jan), 64-72.
- [46] Poston, R. M., 1989, Proposed Standard Eases Tool Interconnection, IEEE Software, 6, 6 (Nov), 69-71.
- [47] Raccoon, P. 1994. The Chaos Model and the Chaos Life Cycle. Software Engineering Notes. Vol.20,N°1, pp 55-66
- [48] Sommerville, I. "Software Engineering". Addison - Wesley. 1996.
- [49] Thomas, I., 1989, PCTE Interfaces: Supporting Tools in Siftware Engineering Environments, IEEE Software, 6, 6 (Nov), 15-23.
- [50] Thomas, I. & Nejme, B. A., 1992, Definitions of Tool Integration for Environments, IEEE Software,9,2, 29-35.

- [51] Wasserman. A. I., 1988, Integration and Standardization Drive CASE Advancements, Computer Design, 27, 22.
- [52] Wasserman, A., 1990, Tool Integration in Software Engineering Environments,.: Proceedings of the international Workshop on Environments, F. Long (Editor), Springer-Verlag , 137-149.
- [53] Yourdon, E., 1990, DEC's CASE Environment, America Programmer, 3, 1 (Jan) 4-14.
- [54] Perichinsky, G., Servetto, A.C.et al. 2001. Integrated Environment of Systems Automated Engineering 2. Product Oriented. Proceedings of the XIX International Conference on Applied Informatics. ( pp 706-713 ).Innsbruck. Austria.
- [55] Perichinsky, G., Servetto, A.C.et al. 2002. Integrated Environment of Systems Automated Engineering 3. Proceedings of the XX International Conference on Applied Informatics. ( accepted ).Innsbruck. Austria.

## Proceso y Métodos de Ingeniería Liviana de Sistemas

### *Autores*

*Arturo Carlos SERVETTO, Ramón GARCÍA MARTÍNEZ, Gregorio PERICHINSKY*

*aserve/rgm/gperi@mara.fi.uba.ar*

*Tel. (011) 4342-9184 / 4343-0891, Int. 142*

*Laboratorio de Bases de Datos y Sistemas Operativos*

*Facultad de Ingeniería*

*Universidad de Buenos Aires*

### **Introducción**

Las metodologías de desarrollo de software surgieron a raíz de la necesidad de controlar y documentar proyectos cada vez más complejos, impulsadas principalmente por instituciones económicamente importantes y con requisitos de seguridad y fiabilidad en sus sistemas sumamente estrictos, como el Departamento de Defensa de los Estados Unidos.

Pero la evolución de estas metodologías, que sugieren o imponen a menudo varias actividades paralelas para cada fase del ciclo de vida de los sistemas y que asimismo pueden requerir más de un modelo para cada actividad, determinó que el costo y el esfuerzo requeridos en producir y mantener los documentos relativos al proceso de desarrollo crecieran considerablemente, a tal punto que hoy en día sólo instituciones muy grandes o que desarrollan sistemas muy complejos las adoptan y cumplen formalmente [1].

Las metodologías de desarrollo tradicionales modelan independientemente tres aspectos o dimensiones de los sistemas informáticos: estructura de la información, funciones y comportamiento. Cada aspecto requiere distintos modelos y los documentos resultantes pueden ser múltiples en los últimos dos aspectos, y en general son bastante costosos de mantener durante la evolución de los sistemas [2, 4, 7, 8, 11, 13].

En el desarrollo de sistemas pequeños o de mediana complejidad, la adopción de una metodología formal se percibe como un obstáculo más que como una ventaja, puesto que las ventajas sólo se hacen notorias a largo plazo, y los objetivos primarios de cualquier institución son producir software en el menor tiempo posible y a bajo costo para resultar competitiva.

Paralelamente, el empleo de herramientas de automatización del desarrollo de sistemas aún resulta bastante limitado debido a la complejidad, diversificación y falta de estandarización de las metodologías, que impide que una única herramienta pueda cubrir procesos completos de desarrollo o soportar al mismo tiempo metodologías distintas; así es que, para un desarrollo asistido en toda su extensión se debe contar con varias herramientas que a menudo no son compatibles entre sí. Además, sólo para empresas importantes o de desarrollo de sistemas de alta complejidad resulta económicamente viable emplear este tipo de herramientas para cubrir íntegramente el proceso de desarrollo [1].

Actualmente el mayor esfuerzo de la industria de sistemas informáticos está dirigido a la búsqueda de una metodología estándar así como a la estandarización de notaciones y representaciones de la información a los efectos de lograr la integración de herramientas de desarrollo [14, 15, 16].

La consolidación de nuevas tecnologías como las basadas en objetos, con su rico bagaje de modelos de interfaces y sus facilidades para la reusabilidad, hace propicio el desarrollo de herramientas de automatización integral del desarrollo de sistemas mediante un proceso de ingeniería liviana; es

decir, minimizando la cantidad de modelos e integrándolos en una única herramienta en la que se puedan mantener sin mayores costos [6].

### **Proyecto**

El proyecto que motiva esta presentación tiene como objetivo el desarrollo de una herramienta CASE de automatización integral que produzca sistemas, esto es, el código y la documentación completos, a partir de especificaciones gráficas de alto nivel de abstracción. Para ello se definió un proceso de ingeniería liviana basado en la *prototipación evolutiva*, con ciclos conformados por la especificación o evolución de requerimientos, la especificación o evolución de diseño y la generación o regeneración del sistema [10].

Para la *especificación de requerimientos* se emplea un modelo gráfico que sintetiza funcionalidad y comportamiento en base a la teoría de autómatas finitos [9]. Se concibe a todo sistema como a un autómata cuyos estados se asimilan a interfaces, y sus transiciones a funciones. Cada interfaz es entonces un escenario operativo o de interacción con el sistema, y las opciones a partir de ella casos de uso del escenario [5]. Se puede asociar uno o más grupos de usuarios como actores tanto a escenarios como a casos de uso, de manera que si un grupo se asocia a un escenario será actor en todos los casos de uso, y si se asocia a algún caso de uso en particular quedará excluido como actor en los restantes casos de uso; esto permite definir grupos de usuario jerárquicos, con acceso restringido a funciones específicas.

La *especificación de diseño* se efectúa a partir del diagrama de requerimientos, asociando:

- A cada escenario, el diseño detallado de una ventana como especificación de formulario o reporte
- A cada caso de uso, el diseño detallado de un diagrama conceptual de entidades y relaciones comprometidas y un guión en álgebra relacional de entidades conceptuales

La generación o regeneración del sistema es automática, y consiste en crear o evolucionar los esquemas de datos a partir de diagramas nuevos o cambios en diagramas preexistentes, y en recrear el código a partir de los guiones.

Lo original del proyecto es la definición del proceso de ingeniería liviana, la aplicación de la teoría de autómatas finitos para sintetizar especificación de requerimientos y comportamiento de sistemas, y la definición del álgebra relacional de entidades conceptuales y su extensión para la especificación de guiones funcionales.

Se completó el diseño de la herramienta y ya se ha comenzado su desarrollo. Los metadatos asociados a los diagramas de especificación se modelaron separando la información de definición de la información de representación, y se prevé como futuro proyecto agregar a la herramienta el manejo de repositorios con control de versiones.

Para la modelación conceptual de entidades y relaciones se adapta una extensión del modelo propuesta por los autores Batini, Ceri y Navathe [3, 12] de gran riqueza expresiva, y que admite la definición de atributos compuestos, atributos polivalentes, cardinalidades mínima y máxima de entidades en relaciones y jerarquías de generalización con cobertura.

Si bien en la herramienta se induce a la definición de un diagrama de entidades y relaciones por caso de uso, se prevé el reuso, coordinación e integración panóptica de diagramas, así como la denotación contextual de entidades, relaciones y atributos.

Para la definición de esquemas de datos a partir de diagramas conceptuales se define una heurística de transformación de modelos conceptuales a lógicos (relacionales) basada en reglas sobre el diagrama conceptual que pueden cambiar según el estado de la base de datos.

También se definen métricas de calidad, tanto para las estructuras de datos como para las funciones, para validar y orientar el diseño, y métricas de complejidad para la valorización de los sistemas que se produzcan.

Se cree que esta herramienta importa una contribución significativa para la comunidad informática dedicada al desarrollo de sistemas de pequeña y mediana complejidad, dado que implica la adopción de una metodología simple y precisa que favorece la participación de los usuarios finales y mantiene a todo desarrollo permanentemente documentado.

La participación y el compromiso de los usuarios finales en desarrollos basados en esta herramienta se presumen garantizados debido a que los modelos empleados para las especificaciones son de un alto nivel de abstracción y comprensibles para personas no especializadas; además, la combinación del modelo de casos de uso, que permite verificar la completitud y rastrear el cumplimiento de requerimientos, con la posibilidad de la prototipación temprana, asequible con la generación de sistemas a partir de la especificación del diseño de interfaces, optimiza las relaciones contractuales facilitando la aprobación de fases.

### **Referencias**

- [1] Anfossi, D. y Servetto, A. "Relevamiento de Herramientas CASE Orientadas a Objetos: Comparación, Evaluación y Conclusiones". Anales del III ICIE (Congreso Internacional de Ingeniería en Informática) de la Facultad de Ingeniería de la Universidad de Buenos Aires; pág. 449-463.
- [2] Bass, L. , Clements, P. and Kazman, R. "Software Architecture in Practice". Addison - Wesley. 1998.
- [3] Batini, C., Ceri, S., Navathe, S. "Diseño Conceptual de Bases de Datos: Un enfoque de Entidades-Interrelaciones". Addison-Wesley Iberoamericana, 1991; ISBN 0-201-60120-6.
- [4] Bell, D. and Morrey, I. "Software Engineering, Second Edition". Prentice Hall. 1992.
- [5] Booch, Grady; Rumbaugh, James; Jacobson, Ivar. "The Unified Modeling Language - User Guide". Addison-Wesley, 1999.
- [6] Brown, A. W. & McDermid, J. A., 1991, On Integration and Reuse in a Software Development Environment, Software Engineering Environments '91, F. Long & M. Tedd (Editors), Ellis Horwood, Mar 7.
- [7] Constantine, L. and Lockwood, L. "Software for use". A Practical Guide to the Models and Methods of Usage - Centred Design. Addison - Wesley. 1999.
- [8] Ghezzi, C. , Jazayeri, M. , Mandrioli, D. "Fundamentals of Software Engineering". Prentice Hall. 1991.
- [9] Grimaldi, Ralph P. "Matemáticas discreta y combinatoria. Introducción y aplicaciones". Addison-Wesley Iberoamericana, 1989.
- [10] IEEE STD 1074-1991 Customer Request IEEE Standard fo developing Software Life Cycles Process, Identify Ideas or Needs, Preliminary Statement of Need, Feasibility Studies, Statement of need.
- [11] Pfleeger, S. "Software Engineering: Theory and Practice". Prentice Hall. 1998.
- [12] Servetto, A. "Mecanismos de Abstracción en la Modelación Conceptual de Datos y su Aplicación en una Ampliación del Modelo de Entidades y Relaciones" 3ras Jornadas de Informática e Investigación Operativa de la Universidad de la República de Uruguay (12 al 14 de diciembre de 1996).
- [13] Sommerville, I. "Software Engineering". Addison - Wesley. 1996.

- [14] Thomas, I. & Nejme, B. A., 1992, Definitions of Tool Integration for Environments, IEEE Software, 9, 2 (Mar), 29-35.
- [15] Wasserman, A., 1990, Tool Integration in Software Engineering Environments, Software Engineering Environments: Proceedings of the international Workshop on Environments, F. Long (Editor), Springer-Verlag , 137-149.
- [16] Wasserman. A. I., 1988, "Integration and Standarization Drive CASE Advancements, Computer Design", 27, 22 (Dec), 86.

## **Estudio de Técnicas apropiadas para modelar aplicaciones de Hidroinformática en el contexto de los Sistemas de Información Ambiental –**

**Urciuolo Adriana, Iturraspe Rodolfo, Sandoval Sandra, Parson Ariel**

Universidad Nacional de la Patagonia San Juan Bosco – Sede Ushuaia, Darwin y Canga, (9410) Ushuaia. TE/FAX: 430892  
e-mail: [urciuolo@tdfuego.com](mailto:urciuolo@tdfuego.com), [iturraspe@tdfuego.com](mailto:iturraspe@tdfuego.com), [sandrasandoval@ciudad.com.ar](mailto:sandrasandoval@ciudad.com.ar), [a-parson@impsat1.com.ar](mailto:a-parson@impsat1.com.ar)

### **Resumen**

El objetivo general de la línea de investigación es el estudio de técnicas apropiadas para el desarrollo de sistemas de hidroinformática, que permitan manejar la complejidad, representar en forma adecuada la información y simular los procesos relacionados con el ambiente natural, en el contexto de un Sistema de Información ambiental.

En forma específica durante la primera etapa del proyecto, se pretende definir técnicas adecuadas para modelar los sistemas de hidroinformática, mediante la aplicación de un enfoque combinado a un campo particular del dominio: los modelos matemáticos lluvia-escorrentía. Se proponen microarquitecturas de diseño OO (en base a la aplicación de patrones de diseño) aptas para el desarrollo de modelos hidrológicos en el contexto de un Sistema de Información Ambiental y soluciones evolutivas para la optimización de los parámetros del modelo. El proyecto se desarrolla en la Fac. de Ingeniería de la UNPSJB. Asesores externos del proyecto: Dra. Silvia Gordillo (UNLP), PhD. Zbigniew Michalewicz.

### **Introducción**

Los Sistemas de Información ambiental se relacionan con el manejo de los datos correspondientes a los distintos componentes interactuantes del ambiente: el suelo, el agua, el aire y las especies existentes. El presente proyecto de investigación en desarrollo, focaliza su atención en el tratamiento de la información correspondiente a uno de dichos componentes: el agua.

La gran demanda de información ambiental y de herramientas apropiadas para su organización y análisis, así como la necesidad de que las organizaciones de manejo ambiental cuenten con sistemas de información eficientes, motiva actualmente en forma creciente el interés por esta área de investigación. La complejidad de los sistemas relacionados al ambiente hídrico, generó el desarrollo de un gran número y diversidad de modelos computacionales (Bavovic, 1996), que a través de años de evolución dieron surgimiento a un nuevo paradigma que tiene sus bases en la hidráulica computacional: la hidroinformática (Abbot, 1991), el cual se *ocupa del desarrollo y aplicación de modelos matemáticos y tecnología avanzada de información a problemas de hidrología, hidráulica e ingeniería ambiental. Provee sistemas de soporte de decisión basados en computadoras para ser utilizados por ingenieros, autoridades del agua y organismos gubernamentales de la gestión hídrica y ambiental* (IHE Delf, 1996). Los problemas más comunes que manejan estos sistemas son: modelado de ríos, cuencas, agua subterránea, transporte de sedimentos, transporte de contaminantes, calidad de aguas, predicción de crecidas, riesgos hidrológicos, cambio climático, manejo y remediación de suelos y aguas subterráneas contaminadas, conservación de calidad de aguas, etc. El tratamiento de estos problemas, normalmente se realiza en las instituciones, mediante el uso de *modelos matemáticos* que simulan distintas situaciones, asumiendo que los resultados de los mismos, una vez ajustados, corresponden al comportamiento del sistema hidrológico real.

Dentro de este dominio, los modelos computacionales aplicables al campo de la hidrología superficial conocidos como de *lluvia-escorrentía*, sufrieron un proceso de cambios y evolución, determinado por los avances en la computación, a los cuales se fueron adaptando; se distinguen cinco generaciones de modelos. Si bien en la última se han incorporado componentes de inteligencia artificial encapsulando conocimiento de expertos y de otras metodologías para sistemas soporte de toma de decisión, se

observan aún problemas en el software, derivados de la falta de utilización sistemática de técnicas de ingeniería de software apropiadas para el dominio. Pueden citarse:

*1. Falta de metodologías adecuadas para el manejo de la complejidad.*

La complejidad en el manejo de la información y el modelado de ambientes naturales está presente por muchas razones, sin embargo podríamos citar como fundamentales las siguiente causas:

- Gran cantidad de datos a procesar. Las técnicas clásicas de almacenamiento y manejo de datos no son eficientes en la mayoría de los casos.
- Escala temporal: los fenómenos ambientales cambian a través del tiempo.
- Necesidad de representación espacial de la información.
- Los objetos ambientales presentan estructuras complejas, generalmente se componen de subobjetos.
- Los procesos hidrológicos involucran información de tipo continua y modelos no lineales
- Necesidad de considerar la influencia humana sobre los ecosistemas.
- Manejo de la incertidumbre.
- Consideración de restricciones legales, normativas existentes, etc.
- Requerimiento del estudio de complejas conexiones lógicas entre datos, dada la diversidad de áreas de estudio interrelacionadas.

*2. Falta de integración de modelos a sistemas de información hídrica y ambiental.*

La mayoría de modelos tradicionales existentes en el mercado, se ejecuta en forma aislada de los sistemas de información hídrica y ambiental de las organizaciones, sin poder utilizar la información disponible en los mismos. En la actualidad, algunos modelos se han comenzado a integrar a SIG (Sistemas de Información Geográfica) para beneficiarse de sus prestaciones de almacenamiento y análisis de datos, pero dicha integración se consigue a través de complejos programas de interface para adecuar las estructuras de datos de modelos y SIG y no a través de una arquitectura común de diseño. Lo mismo sucede en cuanto a la integración con otros modelos de hidrología (por ejemplo de calidad de aguas, transporte de sedimentos, etc.) o de otras áreas, que permitirían agregar funcionalidad a la existente o modificar los métodos de cálculo.

## **Temas de investigación y desarrollo**

De lo expuesto, surge la necesidad de analizar metodologías apropiadas para manejar la complejidad y de caminar hacia una infraestructura común integradora de desarrollo del software para el dominio, que permita la ampliación de la frontera de utilización a nuevos problemas y usuarios y brinde flexibilidad para la actualización y/o extensión de los modelos existentes en las organizaciones.

Las actuales investigaciones en hidroinformática incluyen el estudio de técnicas apropiadas para el manejo de la complejidad y de la incertidumbre.

Se observa que en el software correspondiente a modelos de quinta generación, ha comenzado a utilizarse el enfoque OO, pero básicamente a nivel de los componentes de control del modelo. Abbot postula que la OO debería jugar un rol importante en esta generación, en la modelación de los tipos de datos (Abbot, 1991). No obstante lo propuesto, si bien se encuentran en nuestros días aplicaciones que utilizan este enfoque, sólo se observa la definición de algunas jerarquías de herencia y relaciones básicas, sin el objetivo de obtener diseño reusable.

Dentro de este proyecto de investigación, se propone el desarrollo de microarquitecturas de diseño OO para este tipo de sistemas, basadas en la utilización de patrones, que permitan solucionar en parte, los

problemas planteados: representación adecuada de la información continua, espacio-temporal, flexibilidad en la elección del modelo a utilizar por los componentes de un sistema hidrológico real, modelos integrados a Sistemas de información.

El planteo del proyecto implica una concepción inversa a la tradicional en el modelado de recursos hídricos, donde el énfasis fue siempre puesto en los procesos físicos siendo representados y la forma en la cual obtener los parámetros de un ambiente particular jugaba un rol menor. El énfasis se pone en primer lugar en la descripción del ambiente de un sistema hidrológico real y luego en la definición de los modelos de procesos que puedan utilizar los datos disponibles. Se utilizan metodologías apropiadas para desacoplar la funcionalidad, de modo que distintos modelos independientemente del objetivo de la simulación tengan acceso a los datos observados y almacenados en sistemas de información.

Por otra parte, dada la necesidad de manejar la incertidumbre, se hace necesaria la optimización de parámetros de los modelos para que se ajusten al sistema real que representan.

Existe en la actualidad un incremento cada vez mayor de la utilización de técnicas de computación evolutiva aplicadas a los sistemas de información ambiental (Goldberg, 2000), como herramienta de optimización de funciones y de toma de decisiones; es intención de este proyecto, probar asimismo su adaptabilidad a procesos característicos de las aplicaciones de hidrología superficial, como la optimización de parámetros, Se propone para ello el estudio de diferentes técnicas evolutivas, en particular de algoritmos genéticos a los efectos de determinar la conveniencia de su uso para este fin.

### *Etapas del Proyecto*

#### *1. Análisis del Dominio*

Se realizaron los siguientes estudios: Conceptos básicos del paradigma de hidroinformática, Modelos hidrológicos e hidráulicos existentes, Análisis de aplicaciones de software existentes y de documentación relacionadas con los Sistemas de Información Ambiental y con el dominio de estudio en particular, Análisis y comparación de metodologías utilizadas en el dominio.

*1.1. Requerimientos candidatos:* Se definieron los siguientes requerimientos candidatos para el software del dominio de los sistemas de hidroinformática:

- Proveer una forma adecuada y flexible de modelar los procesos del mundo natural.
- Adaptarse tanto a los objetivos de simulación numérica, resolviendo las ecuaciones correspondientes, como al manejo de la gran cantidad y complejidad de la información disponible.
- Proveer medios para intercambio de datos o integración entre los sistemas de simulación hidráulicos o hidrológicos con herramientas de análisis de impactos ambientales y económicos.
- Facilitar su integración con SIG (Sistemas de información geográfica) a los efectos de manejar el ingreso y almacenamiento de datos, así como la presentación de resultados mediante una interface adecuada.
- Utilizar herramientas adecuadas de simulación y optimización.
- Proveer facilidades para su extensión y modificación, así como para la inclusión de rutinas de simulación existentes

#### *1.2. Definición del contexto*

Se realizó el análisis del contexto de las aplicaciones correspondientes a Sistemas de información hídrica y a Modelos lluvia/escorrentía, considerando diferentes niveles de diferente abstracción.

Se definieron los siguientes tipos de objetos, para cada uno de los niveles expuestos:

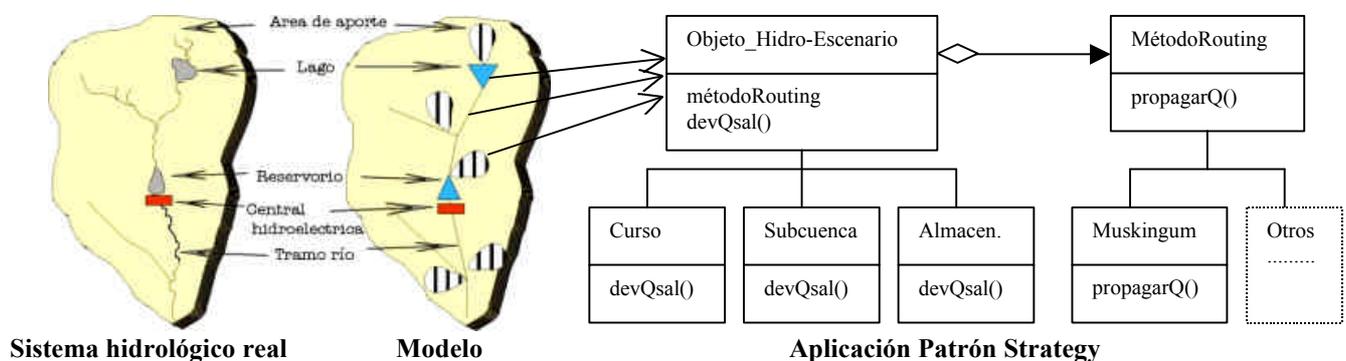
- Nivel de Información hídrico/ambiental: Objeto-Hidrológico. Este nivel está compuesto por los objetos del mundo real que interesa almacenar en un “inventario hídrico” (cuena, lago, planta, etc.) Las entidades observadas se clasificaron en hidrológicas, climáticas, tecnológicas y ambientales.
- Nivel de representación espacial: GeoObjeto-Hidrológico: en este nivel, se adiciona comportamiento geográfico a aquellos objetos identificados en el nivel anterior, que deban incluir su representación espacio/temporal. Los objetos de este nivel pueden ser ubicados en un mapa de acuerdo a diferentes representaciones (raster, vector, etc.)
- Nivel de escenarios de simulación: Objeto-Hidrológico-Escenario. En este nivel los objetos tienen comportamiento hidrológico, es decir pueden simular diferentes procesos de la hidrología superficial, mediante distintos métodos.

Se analizaron los procesos naturales involucrados y los métodos de simulación correspondientes a diferentes escenarios de modelos lluvia-escorrentía.

## 2. Diseño

Se definieron microarquitecturas de diseño, en base a patrones OO, apropiadas para:

- Composición flexible del sistema hidrológico real a representar. Se utiliza el patrón de diseño Composite (Gamma, 1997), que permite representar jerarquías todo-parte.
- Representación espacial/temporal de los objetos de un sistema hidrológico real. Se utiliza el “Modelo para aplicaciones geográficas”, Gordillo, 1998. Este modelo propone utilizar el patrón Decorator (Gamma, 1997), el cual permite adicionarle características espaciales. Además resuelve la definición de la Locación del objeto (información referente a la posición y al tiempo) y considera la geometría y sistema de referencia utilizado para el objeto.
- Adición de “comportamiento hidrológico” a los objetos del sistema de información seleccionados. Se utiliza el patrón de diseño Decorator, (Gamma, 1997) a los fines de adicionar comportamiento específico a los objetos del sistema de información.
- Representación de procesos correspondientes a modelos conceptuales ESMA y modelos distribuidos basados en el proceso físico. Se utiliza el patrón de diseño Strategy, que define una familia de algoritmos, encapsula cada uno y los hace intercambiables
- Integración de diferentes escenarios de simulación al nivel del Sistema de Información hídrica. Se utilizan los patrones Strategy y Mediator.



Esta etapa del Proyecto aún no ha sido finalizada

### 3. *Diseño de Procesos utilizando algoritmos genéticos (Próxima etapa)*

En esta etapa del proyecto, se realizará el estudio de distintas técnicas de computación evolutiva y se utilizarán algoritmos genéticos para la optimización de parámetros en modelos lluvia-escorrentía.

#### **Conclusiones y trabajos futuros**

Los resultados obtenidos hasta el presente permiten establecer que mediante el uso de técnicas de modelación apropiadas, pueden ser resueltos gran parte de los problemas mencionados en relación a los sistemas de hidroinformática. En particular, el uso de patrones de diseño OO permite obtener: una forma conveniente de modelar información de tipo continua y su representación espacio/temporal, flexibilidad en la selección del método y/o modelo hidrológico a utilizar por parte de los distintos componentes de un sistema de información, composición flexible del sistema hidrológico real a modelar, facilidad para extender las aplicaciones a diferentes tipos de modelos hidrológicos y ambientales y modelos desarrollados en un contexto de sistemas de información ambiental. Durante la próxima etapa del proyecto, se pretende demostrar que los algoritmos genéticos constituyen una valiosa herramienta para el desarrollo de estos sistemas.

En futuros trabajos se espera poder investigar sobre técnicas que permitan la incorporación a estos sistemas de restricciones legales, normativas vigentes y otras reglas; se considera que en este aspecto, sería conveniente analizar la utilización de Inteligencia Artificial en Hidroinformática.

#### **Bibliografía**

- ◆ Abbot M.B. Hydroinformatics: Information Technology and the aquatic environment. Avebury Technical, Adlershot, UK, 1991
- ◆ Alfredsen. An object oriented framework for application development and integration in hydroinformatics. Dr. Eng. Thesis, Norwegian University of Science and Technology, 1998
- ◆ Bavovic, V. Emergence, Evolution, Intelligence: Hydroinformatics. IHE-Delft Balkema, Rotterdam Netherlands, 1996
- ◆ Beven, K., Rainfall-Runoff Modelling. Wiley, 2000
- ◆ Hydroinformatics 98, Copenhagen, Denmark Editors: Babovic, Larsen. Proceedings: A.A. Balkema, Rotterdam, 1998. Hydroinformatics 00, Iowa, USA. Electronic Proceedings : Delft, 2000
- ◆ Booch G., Object-Oriented Analysis and Design With Applications Addison-Wesley Publications, 2<sup>nd</sup> Edition, 1994
- ◆ Booch G., Jacobson I., Rumbaugh J. The Unified Software Development Process. Addison-Wesley Publications, 1999.
- ◆ Gamma, E. H. Design Patterns. Elements of Reusable OO Software. Addison-Wesley, 1997
- ◆ Gordillo, S. Tesis de Magister en Ingeniería de Software, UNLP. "Modelización de campos continuos en Sistemas de Información Geográfica", 1998
- ◆ Günther, O. Environmental Information Systems. Springer-Verlag, Berlín, Germany, 1998
- ◆ Laurini, R. Thompson, D. Fundamentals of Spatial Information Systems. Academic Press, 1999
- ◆ Babovic, V. Evolutionary Algorithms as a Theme in Water Resources Scientific presentations AIO meeting 93, Delft University of Technology, pp. 21-36, 1993
- ◆ Goldberg, D.E. A Hydroinformatician's Approach to Computational Innovation and the Design of Genetic Algorithms. Publication Illinois Genetic Algorithms Laboratory, 2000
- ◆ Michalewicz, Z. How to Solve It. Springer-Verlag, Berlín Germany, 2000

## EL CICLO DE VIDA DE LOS SISTEMAS: LOS CONCEPTOS Y LOS ALGORITMOS DE EVOLUCIÓN SIMPLE Y CICLICA

Yuriy D. Zubenko (1,2), Fernando J. Lage (2), Zulma Cataldi (2)

1. Universidad Argentina John F. Kennedy;

2. Facultad de Ingeniería, UBA. Universidad de Buenos Aires.

Paseo Colón 850. 4° Piso. 1063 – Ciudad de Buenos Aires. [informat@mara.fi.uba.ar](mailto:informat@mara.fi.uba.ar)

### RESUMEN

*Se presenta el estado actual de las investigaciones que se están llevando a cabo que estudian el análisis de los sistemas desde los fundamentos teóricos–matemáticos que los sustentan. Estos se pueden presentar en forma de conceptos, algoritmos y sus aplicaciones. El análisis de sistemas tiene como objeto de investigación los propios sistemas, con sus herramientas teóricas y metodológicas, a partir de la definición del sistema como una colección de propiedades del objeto real que realiza una función.*

*Para el estudio acerca de la vida de los sistemas se define la evolución simple o la vida de un sistema con los estadios de aparición y síntesis; de funcionamiento; y de degradación o descomposición. La evolución cíclica está definida como consecuencia de las evoluciones simples con una función recurrente. Para cada estadio se presenta el concepto teórico, el algoritmo abstracto y los ejemplos. Los resultados permiten suponer, que el análisis de sistemas permite resolver problemas prácticos en sistemas de diferentes tipos y destinos.*

### INTRODUCCIÓN

Se toma la siguiente definición de sistema: "Un sistema es una colección de propiedades del objeto real, que realiza una función" y se lo designa con la letra **S**. Las nociones y las relaciones de los sistemas se presentan en forma de la Teoría General de Sistemas como la subteoría de la teoría **L** del cálculo proposicional [7,9,21].

En correspondencia con la **Hipótesis I**: "Cada sistema es finito en espacio y en tiempo" se definen los estadios de evolución simple de los sistemas y la evolución cíclica del sistema como una secuencia de evoluciones simples de sistemas con la misma función.

### DESARROLLO

**El análisis de sistemas estáticos [21].** En los sistemas estáticos se investigan las nociones tales como objetos, partes, características, etc. y, las relaciones entre ellas, los componentes y las estructuras. Luego se construye el modelo abstracto de acuerdo a la Teoría General de Sistemas, después los modelos derivados en álgebra lógica, lógica probable, las teorías interdisciplinarias, siendo posible el uso del espacio topológico [10], en las teorías aplicadas, hasta obtener el modelo con sus características y relaciones. Paralelamente, se crean los modelos gráficos y se aplican las herramientas metodológicas. Con la nueva definición de sistema se reúnen las propiedades  $s_i$  y se recibe la función **F** del sistema como una nueva propiedad. El espacio topológico **T<sub>p</sub>** contiene todas las propiedades posibles, incluyendo la nueva propiedad **F<sub>i</sub>**. Por eso se destaca esa nueva propiedad **F<sub>i</sub>** en el espacio **T<sub>p</sub>**. La creación del sistema significa una transformación de las propiedades:  $S_i \rightarrow F_i$ .

#### **Ejemplo 1: El Análisis estático de los sistemas de computación en gestión [22].**

A estos sistemas se los llama sistemas de gestión automatizados (SGA) que se denomina **S**. Por definición, el SGA es la colección de las propiedades informáticas del hardware, software y del personal siendo el objeto real (**N**), que realiza la función de gestión (**F**). Las propiedades que forman al sistema son informáticas (**s<sub>1</sub>**). La base (**B**) son los técnicos o hardware (**H**), los programas o software (**W**), la organización o el personal (**P**), la calificación (**C**), las leyes, reglas y normas (**R**). El medio exterior (**V**) es el objeto de gestión (**V1**) y las partes adyacentes del proyecto de SGA son las condiciones exteriores, locales, la alimentación eléctrica, etc. denominada (**V2**). La frontera **G1** son los límites físicos del objeto, por ejemplo, la entrada y la salida informática del SGA. La frontera **G2** es la relación entre las propiedades informáticas (**s<sub>1</sub>**) del sistema y otras propiedades del objeto real o la base **B**. La estructura de las relaciones exteriores del SGA define la función:

$Y=F(x)$ , donde  $x$  y  $Y$  son las relaciones iniciales y finales del sistema (la entrada y la salida). La estructura interior del SGA es la estructura de sus propiedades informáticas realizados por la base  $B$ . Los elementos de la base  $B$  son: los elementos discretos electrónicos, eléctricos, magnéticos de la computadora; los elementos biológicos del personal; los elementos funcionales de la técnica de gestión analógica. En el caso de elementos discretos de la base  $B$  los procedimientos informáticos en SGA son las colecciones de operaciones booleanas, donde la adición 1 es la señal de entrada, la suma es la señal de salida y la adición 2 es el estado del elemento.

En el SGA existen dos tipos de elementos de análisis: en la estructura funcional el elemento es una tarea; en la estructura de los procedimientos informáticos el elemento es un bit de información. La estructura interior lineal del SGA puede ser representada por código binario. La estructura lineal es muy voluminosa y por ello no es efectiva, son mejores estructuras flexibles: con interconexiones entre los elementos no lineales y distribuidas. En la estructura lineal los componentes del vector de salida dependen solamente de los componentes del vector de entrada, y en las flexibles no. La estructura interior de SGA tiene los componentes siguientes: la base de datos, el sistema de clasificación y de codificación, las formas de documentos, la información masiva, las señales de entrada y de datos, las señales de salida y los documentos, la estructura de los procedimientos de datos, el sistema de representación de la información, la interface, etc.

### **El algoritmo abstracto de investigación en los sistemas.**

La investigación en sistemas posee ocho estadios básicos que se describen debajo. La **Entrada** del algoritmo es **el concepto** teórico y la **información inicial y la salida** del algoritmo es la definición del **problema**.

Los estadios básicos son:

**Estadio 1.** Primero es necesario identificar el sistema (**S**). Para eso se define el objeto real (**N**), el medio exterior (**V**), la frontera **G1** entre ellos. En el objeto **N** se identifican las propiedades (**s<sub>N</sub>**), entre esas propiedades se identifican las propiedades del sistema (**s<sub>i</sub>**) y otras propiedades (**s<sub>B</sub>**). Luego se definen el sistema **S** con propiedades **s<sub>i</sub>** y la función **F**. Se identifica la base **B** con propiedades **s<sub>B</sub>**. Se define la frontera **G2** como las relaciones entre las propiedades **s<sub>i</sub>** y las propiedades **s<sub>B</sub>**. Para lo anterior se usan la Teoría General de Sistemas, identificando al sistema **S** en la clasificación de los sistemas.

**Estadio 2.** Se define la función **F** como el equilibrio entre las relaciones exteriores del sistema (**SS, SB, SV, VS, BS**) y la organización de las propiedades particulares del sistema. Se define la estructura funcional (**|F|**) y la estructura de las propiedades particulares del sistema (**|s<sub>i</sub>|**) y la correlación entre esas dos estructuras del sistema. Se usa la definición de la función.

**Estadio 3.** Se identifica la estructura de la base **|B|**, sus partes y las relaciones entre ellas. Se definen las relaciones entre la base **B** y el sistema **S** (**BS, SB**), para lo que se usa la Teoría General de Sistemas y la metodología propia.

**Estadio 4.** Se limita el medio exterior del sistema, se identifican sus partes importantes para el sistema y su estructura **|V|**. Se definen las relaciones entre el medio exterior **V** y el sistema **S** (**SV, VS**), y entre el medio exterior **V** y la base **B** (**BV, VB**).

**Estadio 5.** En quinto lugar, se crea **el modelo abstracto** del sistema en la Teoría General de Sistemas, usando los axiomas de la Teoría L y de la TGS.

**Estadio 6.1.** Se crea **el modelo derivado** (más detallado y aplicado) del sistema, usando las teorías interdisciplinarias, espacios y teorías aplicadas.

**Estadio 6.2.** Se crea paralelamente al anterior **el modelo empírico** (detallado y aplicado) del sistema, usando la metodología propia de los sistemas (principios, nociones, métodos, opiniones de expertos, estadística).

**Estadio 7.** En séptimo lugar, se crea **el modelo con las características** del sistema, usando la ciencia aplicada correspondiente.

**Estadio 8.** Se identifica el **estadio de la vida** del sistema (aparición y síntesis; funcionamiento; degradación y/o descomposición) o **evolución cíclica**. Se identifica el **tipo de problema** y las tareas que tienen que resolver, usando la clasificación de los problemas y tareas.

### El concepto y algoritmo abstracto de aparición y síntesis de los sistemas.

Crear un sistema significa organizar algunas propiedades  $s_i$  de un objeto real en forma de una función  $F$ , definida como:  $F(s_i, s_{ij}, s_{ijk}) = \langle SS, SB, SV, BS, VS \rangle$ , con condiciones de integridad, estabilidad y estacionariedad de las propiedades y de las relaciones, donde:  $s_{ij}$  son las características de las propiedades,  $s_{ijk}$  son los valores de las características. La función del sistema está definida por la relación del sistema, su base y su medio exterior. Se pueden representar  $S$ ,  $B$  y  $V$  mediante sus propiedades, características y valores:  $S \sim \langle s_i, s_{ij}, s_{ijk} \rangle_S$ ,  $B \sim \langle s_i, s_{ij}, s_{ijk} \rangle_B$ ,  $V \sim \langle s_i, s_{ij}, s_{ijk} \rangle_V$  y la aparición de la función se puede representar como consecuencia en álgebra lógica [6,12]:  $\langle s_i, s_{ij}, s_{ijk} \rangle_S \wedge \langle s_i, s_{ij}, s_{ijk} \rangle_B \wedge \langle s_i, s_{ij}, s_{ijk} \rangle_V \rightarrow F$ . La  $F$  se la puede representar como una función con argumentos:  $F = F(S, B, V)$ , y la existencia de  $F$  se puede expresar por la función del álgebra lógica:

$f_F(x_{ij})$ ,  $x_{ij} = 0$  ( si  $s_{ij}$  existe ),  $= 1$  ( si  $s_{ij}$  está ausente). La función  $f_F$  puede ser representada por:  $f_F \sim \wedge x_{ij}$ .

La probabilidad de creación de la función mediante la relación de las características independientes puede ser presentada como la multiplicación de las probabilidades de los valores, por lo que es posible la combinación de las probabilidades condicionales e incondicionales.

Según la **Hipótesis 2 del análisis de sistemas: Las posibilidades de la materia para organizarse y desorganizarse tiene un carácter casual [16,17]**. Esto significa que la aparición y la síntesis del sistema tienen carácter casual. El sistema aparece si  $f_F = 1$  con alguna probabilidad  $P_F$ , pero en el caso (a) el sistema se aparece como materia no organizada (por ejemplo todos los sistemas naturales no orgánicos), en caso (b) el sistema aparece parcialmente como materia organizada en forma de otros sistemas (por ejemplo todos los sistemas naturales orgánicos), en caso (c) el sistema aparece como materia organizada en forma de otros sistemas (por ejemplo todos los sistemas artificiales, porque la función  $F$  tiene un carácter probable y por eso se lo puede estudiar con la ayuda de la teoría de las funciones casuales [19]. En los sistemas la función casual es la forma concreta de la estructura interior  $F(s_i, s_{ij}, s_{ijk})$  de las propiedades  $s_i$  para realizar una estructura de las relaciones exteriores  $\langle SS, SB, SV, BS, VS \rangle$ . La función casual se caracteriza por su esperanza matemática  $M[F]$ , dispersión  $D[F]$  y otras características clásicas. Pero a diferencia entre las ciencias y los sistemas el algoritmo de aparición y síntesis de los sistemas tiene un carácter abstracto. La síntesis del sistema se realiza en dos formas: conceptual y formal. Esas dos formas se usan: una después de la otra, una paralelamente con la otra o en alguna combinación. La síntesis de los sistemas comienza con el análisis del medio exterior y con la definición de sus partes principales. Se definen la naturaleza de la base, sus partes principales y su estructura. En el sistema se definen la estructura funcional y la estructura de las propiedades. Se definen las relaciones exteriores del sistema, su estructura interior y sintetizan el sistema en forma de correspondencia entre relaciones exteriores y la estructura interior en forma iterativa. Se pueden ver en la bibliografía citada los ejemplos 2 y 3 que se citan debajo.

**Ejemplo 2: La síntesis de una empresa privada de producción [4,13,21].**

**Ejemplo 3: La síntesis de una planta de energía atómica [11,15,21].**

### El concepto y algoritmo abstracto de funcionamiento de los sistemas.

Los sistemas dinámicos tienen dependencia con el tiempo en ambas partes de su función:

$$F(s_i(t), s_{ij}(t), s_{ijk}(t)) = \langle SS(t), SB(t), SV(t), BS(t), VS(t) \rangle,$$

donde la parte derecha está definida por la configuración de las relaciones exteriores del sistema y la parte izquierda está definida por las posibilidades del sistema de realizar su función. Estas últimas posibilidades del sistema tienen tres niveles cualitativos diferentes: el nivel 1, el más simple, por cambio de los valores  $s_{ijk}(t)$  de las características; el nivel 2, más complejo, por cambio de las

características  $s_{ij}(t)$  de las propiedades; y el nivel 3, el más complejo, por cambio de las propiedades  $s_i(t)$  del sistema.

**El funcionamiento** del sistema significa un cambio de los valores (nivel 1); **adaptación** significa el cambio de las características (nivel 2); **evolución** significa el cambio de las propiedades (nivel 3). Se puede representar la función dinámica  $F(t)$  como una consecuencia de valores (expresiones) de esa función en algunos momentos del tiempo en el intervalo  $\Delta T$ , es decir en forma estática:  $F(t) \sim F(t_1) \wedge F(t_2) \wedge F(t_3) \wedge \dots$ , donde  $t_1, t_2, t_3 \in \Delta T$ .

La función del sistema tiene particularidades de integridad, estacionariedad y estabilidad de relaciones y estabilidad de las propiedades. Entonces en el sistema que funciona en forma real, la función recrea al mismo en algún intervalo  $\Delta T$  de existencia del sistema:

$F(s_i(t_1), s_{ij}(t_1), s_{ijk}(t_1)) \rightarrow F(s_i(t_2), s_{ij}(t_2), s_{ijk}(t_2))$ , donde  $t_2 > t_1$  y  $t_1, t_2 \in \Delta T$ .

La función del sistema tiene carácter casual, por eso su estructura interior (parte izquierda) puede ser presentada como una alternativa entre las estructuras posibles para la misma configuración de las relaciones exteriores (parte derecha). Entonces la estructura interior más probable puede ser presentada como esperanza matemática  $M[F]$  y su límites como la dispersión  $D[F]$ . Es posible presentar el funcionamiento de los sistemas naturales no orgánicos (el sistema solar, el átomo) como reproducción de las relaciones interiores del sistema. El funcionamiento de los sistemas naturales orgánicos se puede presentar como adaptación de las relaciones interiores a las relaciones exteriores (por ejemplo, la homeostasis del organismo). Los ejemplos 4 y 5 citados debajo ilustran el funcionamiento de los sistemas.

**Ejemplo 4. El análisis de funcionamiento de los sistemas de gestión automáticos [20].**

**Ejemplo 5. El análisis de funcionamiento de una empresa estatal [1,3,4,21].**

### **El concepto y algoritmo abstracto de degradación y descomposición de los sistemas.**

La existencia del sistema significa la realización de su función con condiciones, que significa algunas limitaciones para todas las características y relaciones:

$s_i = \text{const}$  (propiedades del sistema son permanentes por definición);

$s_{ij}^* \leq s_{ij} \leq s_{ij}^{**}$  (las características son limitadas por debajo (\*) y por arriba (\*\*)); etc.

Por la hipótesis II, la causa común del final de la existencia del sistema es su función y las condiciones de funcionamiento. Entonces, la violación del equilibrio de la función o de sus condiciones inicializa la degradación y la descomposición del sistema. Como inicio del algoritmo de análisis de degradación y descomposición del sistema se usa el problema con un sistema, definido en el estadio de investigaciones del sistema. Se verifican los límites de todos los argumentos de la función y después se verifica la estabilidad de la función en forma entera. Por ejemplo, la estabilidad de los sistemas de gestión automáticos está estudiada por la teoría de estabilidad con las herramientas matemáticas correspondientes. La estabilidad de los sistemas de gestión automática está definida por su estructura funcional interior, o por su movimiento libre. El movimiento libre del sistema lineal está descrito por una ecuación diferencial homogénea. La parte dependiente del  $y(t)$  depende de la acción exterior  $x(t)$  y no influye a la estabilidad del sistema. En este caso se citan a continuación los ejemplos 6, 7 y 8 y la bibliografía de referencia.

**Ejemplo 6. El análisis del sistema natural inorgánico (sistema solar) [21].**

**Ejemplo 7. El análisis del sistema natural orgánico (la biósfera) [8,18,21].**

**Ejemplo 8. El análisis del sistema artificial (de una central de energía atómica) [11].**

### **El concepto y algoritmo abstracto de la evolución cíclica de los sistemas.**

La evolución cíclica es la prolongación de la evolución simple en la dirección del cambio (aumento o disminución) del nivel de estabilidad y estacionariedad de sistema. La función del sistema nace por interconexión del sistema, su base y su medio exterior y sus posibilidades están definidas por la estructura interior, por eso el nivel de estacionariedad y estabilidad del sistema depende de sus relaciones exteriores y de la estructura interior. La función es más estable si se ha construido más

variable la estructura interior o si han tomado en consideración mayor cantidad de propiedades, características y valores.

La evolución cíclica orientada al objetivo de los sistemas orgánicos y artificiales está definida por las condiciones existentes en la Tierra y en el Sol. Para los sistemas inorgánicos, como el sistema solar, tales condiciones están ausentes o no tienen significado o no se conocen. Para los sistemas inorgánicos, tal como un átomo las condiciones existentes definen su evolución cíclica, como repetición de las evoluciones simples. La forma común de evolución cíclica de sistema es una consecuencia de sus estados estables e inestables [5]. Se puede consultar la bibliografía citada debajo a fin de ver los ejemplos 9 y 10 que se describen.

**Ejemplo 9. La interpretación en sistemas de la teoría de evolución de Darwin [2,14].**

**Ejemplo 10. El pronóstico de los sistemas de computación [22].**

## CONCLUSIONES

El análisis de la bibliografía, de las investigaciones teóricas y de las aplicaciones prácticas muestran que el análisis de los sistemas, con el concepto de sistemas como una colección de propiedades con una sola función, es aplicable y efectivo en la investigación de los sistemas de todo tipo y en todos los estadios de su vida. Es posible investigar los sistemas en empresas (tecnológicos, económicos, financieros, de gestión), los sistemas informáticos, macroeconómicos, ecológicos, etc. Cada ciencia tiene sus propios sistemas y el análisis de sistemas permite estudiarlos en forma unificada. De este modo se pueden comparar los sistemas de diferentes tipos y orígenes y usar los resultados de los estudios realizados en una ciencia para aplicarlos en otra.

## TRABAJOS FUTUROS

De acuerdo a lo antedicho, se piensa investigar diferentes aspectos que hacen a los sistemas. Se tomará el caso de una organización con problemas de índole tecnológico o informática, comenzando por la representación del problema que se desea resolver, la dependencia del mismo con los factores internos y externos al propio sistema, pasando luego a la simulación del mismo.

## REFERENCIAS BIBLIOGRAFICAS

1. Análisis de la actividad financiera comercial de empresas. Moscú, "Finansi i Statistika", 1988.
2. Darwin Charles. El origen de las especies. Moscú, "Selhozizdat", 1952.
3. Demchenkov V.S., Mileta V.I. Análisis de sistemas en la actividad de la empresa. Moscú, "Finansi i Statistika", 1990.
4. Economía del mercado. Moscú, "Somintek", 1992.
5. Evin I.A., Yablonskii A.I. Modelo de evolución y la teoría de catástrofes. En la Revista de la Academia de Ciencias de URSS "Investigaciones en sistemas", Moscú, "Nauka", 1982.
6. Gindikina S.G. Álgebra lógica en ejercicios. Moscú, "Nauka", 1972.
7. Ivs G., Niusom K.V. Lógica matemática y filosofía de la matemática. Moscú, "Znanie", 1968.
8. Kairukshtis L. Una alternativa de la evolución regional con balance ecológico. En la Revista de la Academia de Ciencias de URSS "Investigaciones en sistemas", Moscú, "Nauka", 1988.
9. Klini S.K. Lógica matemática. Moscú, "Mir", 1973.
10. Kolmogorov A.N., Fomin S.V. Elementos de la teoría de funciones y del análisis funcional. Moscú, "Nauka", 1981.
11. Korolev V.V. Sistemas de gestión y defensa de las plantas atómicas. Moscú, "Energoatomizdat", 1986.
12. Latotin L.A., otros. Lógica matemática. Minsk, "Visheishaya Shkola", 1991.
13. Lipsits I.V. Plan de negocio como el fundamento de la suerte. Moscú, "Máshinostroenie", 1993.
14. Malinovskii A.A. La lógica del Darwinismo en sistemas. En la Revista de la Ac. Ciencias de URSS "Priroda", 10, 1983.
15. Popirin L.S. Modelos matemáticos y optimización de las plantas atómicas. Moscú, "Nauka", 1984.
16. Rapoport A. Aspectos matemáticos del análisis abstracto de los sistemas. En Revista "Investigaciones por la teoría general de sistemas", Moscú, "Progress", 1969.
17. Sachkov Y.V. Probabilidad en la evolución de las investigaciones en sistemas y estructuras. En la Revista de la Academia de Ciencias de URSS "Investigaciones en sistemas", Moscú, "Nauka", 1969.
18. Starostin B.A. El concepto de sistemas, los parámetros y complejidad de los objetos biológicos. En la Revista de la Academia de Ciencias de URSS "Investigaciones en sistemas", Moscú, "Nauka", 1974.
19. Ventsel E.S. Teoría de probabilidades. Moscú, "Nauka", 1969.
20. Voronov A.A. Teoría de gestión automática. Moscú, "Vischaya Shkola", 1986.
21. Zubenko Y. D. Análisis de sistemas. Universidad Estatal Técnica de Donetsk, Ucrania. 1995.
22. Zubenko Y.D., Iliin A.A. Optimización de resolución de las tareas en una empresa. Moscú, 1977.

## Controles Semánticos en el Modelamiento Orientado a Objetos

Susana Kahnert – Pablo Fillottrani  
Depto. Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Av. Alem 1253 – Bahía Blanca, Argentina  
e-mail: {sak,prf}@cs.uns.edu.ar

### 1.- Introducción

En un sentido general, un modelo es "una abstracción de algo con el propósito de entenderlo antes de construirlo" [2]. Los modelos omiten detalles no esenciales, facilitando así el análisis de las propiedades de entidades demasiado complejas para ser entendidas directamente. Así como desde tiempos remotos las obras de ingeniería, arte o artesanía se construyen a partir de modelos, también para las distintas etapas del desarrollo de todo software resulta el Modelamiento una actividad crucial.

Es posible construir distintos modelos de un mismo sistema, cada uno enfatizando determinados aspectos que abstraen al sistema desde un punto de vista particular. Como cada uno de estos modelos enfoca la atención sobre determinados detalles mientras simplifica otros que no resulten importantes desde esa perspectiva, un sistema complejo podrá ser representado por una serie de pequeños modelos complementarios que facilitarán su entendimiento. En el proceso de desarrollo del software, un mismo modelo puede ser expresado a niveles diferentes de abstracción, según sea su propósito mostrar los requisitos y conocimiento del dominio, ayudar en el diseño del sistema, comunicar las decisiones de diseño, organizar un sistema complejo, documentar el sistema final, etc. Por consiguiente, la elección del modelo influirá profundamente en el enfoque del problema y en el aspecto de su solución.

Para que el modelo sea útil, es necesario formularlo en un lenguaje que pueda ser compartido por todas las personas involucradas en el desarrollo del sistema. En sistemas de cierta complejidad, la diversidad de personas intervinientes -Jefes de Proyecto, Analistas, Diseñadores, Programadores, Usuarios-, cada uno aportando su enfoque particular del problema, evidencia la necesidad de contar con un lenguaje formalmente definido. Una propuesta en tal sentido es UML (Lenguaje Unificado de Modelamiento) [1,3]. UML es un lenguaje gráfico para visualizar, especificar, construir y documentar los elementos de un sistema de software, es decir para definir los modelos de ese sistema. En 1995 Booch, Jacobson y Rumbaugh combinaron sus propios lenguajes intentando conservar sus ventajas particulares, y crearon la primera versión de UML. En enero de 1997 UML se propuso como un estándar ante el Object Management Group (OMG). La propuesta fue modificada, y finalmente aprobada en noviembre del mismo año, como versión 1.1. Después de varias revisiones por el OMG, surgió la versión 1.3 en 1999.

El desarrollo del Software requiere de un modelo que va evolucionando hasta llegar al sistema implementado. UML es un lenguaje para especificar ese modelo en sus distintas etapas. La notación provista por UML para expresar un modelo incluye elementos gráficos y de texto. Adosada a estos elementos, hay una interpretación semántica que intenta capturar el significado del modelo. Las herramientas de soporte para el desarrollo de sistemas orientados a objetos en UML no sólo deberán proveer de un editor gráfico para la notación, sino también ayudar a asegurar la coherencia de sus aspectos semánticos. Esta última no es una tarea sencilla porque el significado podrá cambiar de acuerdo al propósito del modelo en un momento específico. Por ejemplo, a veces puede ser deseable verificar la consistencia de las relaciones entre las clases, pero después podría ser más importante asegurar la consistencia de tipos en cada invocación a métodos

En el grupo de Investigación y Desarrollo de Ingeniería de Software se está actualmente trabajando en una herramienta llamada HEMOO (Herramienta para Edición de Modelos Orientados a Objetos), para soportar el desarrollo de modelos orientados a objetos, usando UML. Su objetivo es asistir al usuario en el manejo de los aspectos semánticos y notacionales del modelo. Este trabajo se enfoca especialmente en el control de los aspectos semánticos.

## 2. Semántica en Modelamiento orientado a Objetos

El Metamodelo UML define el lenguaje para especificar modelos, es decir, provee la descripción formal de los elementos que conformarán el modelo, tales como clases, atributos, operaciones [4]. Dado que UML es un lenguaje de especificación, proporciona un vocabulario y un juego de reglas para combinarlo. Las unidades del vocabulario se dividen en tres categorías: los *elementos* que son abstracciones de las cosas que constituyen al modelo, las *relaciones* que conectan los diferentes elementos del modelo, y los *diagramas* que reúnen un conjunto de elementos relacionados. En particular, un diagrama es una representación gráfica de un conjunto de elementos, normalmente mostrado como un grafo conexo donde los nodos son los elementos y los arcos las relaciones. UML proporciona nueve tipos diferentes de diagramas. Algunos muestran los aspectos estructurales, presentando a los componentes estáticos del modelo, e indican características que se mantienen durante el ciclo de vida de un sistema. Entre ellos están los Diagramas de Clases, de Objetos, de Componentes y de Distribución. Otros presentan el comportamiento del sistema, definiendo cómo interactúan los objetos a lo largo del tiempo; son los Diagramas de Casos de Uso, de Secuencia, de Actividad, de Estados y de Colaboración.

Para construir estos diagramas, los diferentes elementos no se unen de una manera arbitraria, sino a través de reglas definidas en el Metamodelo. Además, dado que el mismo elemento puede aparecer en distintos diagramas, cada aparición del elemento en un diagrama debe ser consistente con las otras. Para contemplar estas situaciones, UML proporciona el concepto de *modelo bien formado*. Se dice que un modelo es bien formado cuando está correctamente construido usando las reglas del lenguaje, y satisface el significado semántico previsto. En otros términos, un modelo bien formado debe ser sintácticamente y semánticamente correcto. Para expresarlo formalmente, se definen los *controles semánticos*.

Los chequeos semánticos pueden ser caracterizados de acuerdo a diversos aspectos, tal como se ve en la Figura 1. Esta clasificación adquiere importancia teniendo en cuenta que la herramienta deberá poner a disposición del usuario la posibilidad de efectuar chequeos sobre un ámbito determinado –diagramas, relaciones o clases-, o también pudiendo decidir si desea verificar el cumplimiento de los chequeos que afectan a determinados elementos modelo. Este trabajo se enfoca en este último aspecto, es decir a la clasificación de los controles de acuerdo a su generalidad.

critérios	categorías
<b>Generalidad</b>	<ul style="list-style-type: none"> <li>◆ Reglas               <ul style="list-style-type: none"> <li>• Predefinidas</li> <li>• Específicas del modelo</li> </ul> </li> <li>◆ Restricciones</li> </ul>
<b>Ámbito</b>	<ul style="list-style-type: none"> <li>◆ diagramas</li> <li>◆ relaciones</li> <li>◆ elementos</li> </ul>

Figura 1 Clasificación de los controles semánticos

En este aspecto, las condiciones que restringen el valor de todos los elementos respectivos en el modelo se denominan *reglas*, mientras que las que afectan a ítems específicos se llaman *restricciones*.

Las reglas son condiciones semánticas generales que el modelo en su conjunto debe satisfacer para ser considerado bien formado. Es posible identificar dos clases de reglas: las *predefinidas* y las *específicas del modelo*. Las reglas predefinidas son las que introduce UML para especificar cada uno de sus elementos estándares y, en general están definidas como un conjunto de invariantes de una instancia de la metaclassa. Están descritas en una combinación de lenguaje natural (semántica) y lenguaje formal (reglas de buena formación).[4]

### Ejemplo:

Una de las reglas de buena formación que define UML para la metaclassa `Class` establece que toda `Class` que sea concreta debe tener todas sus operaciones implementadas. Para la representación de esta regla, se requiere el atributo `isAbstract`, que pertenece a la clase `GeneralizableElement`, ancestro de `Class`. Figura 2.

En la representación en Prolog de la metaclass `Class` de UML, debe evidenciarse que `Class` descende de `Classifier`, cuyos ancestros son `Namespace` y `GeneralizableElement`, que, a su vez descienden de `ModelElement`, que es heredero de `Element`

Las reglas específicas del modelo permiten establecer la semántica de un modelo en particular. Por ejemplo, el responsable de un determinado desarrollo podrá especificar que cada diagrama tendrá como máximo una dada cantidad de clases, o que no se admitirán clases con más de una cantidad tope de operaciones, o que todos los atributos deben ser privados.

Las restricciones son condiciones semánticas referidas a elementos específicos en el modelo. Es posible definir las tanto en vistas estructurales como en vistas dinámicas. En el ámbito de los Diagramas de Clase, existen algunas restricciones que los objetos deben cumplir durante todo su ciclo de vida, y otras que cumplen solamente cuando interactúan con otros objetos. Las primeras se llaman *Invariantes de Clase*. Las segundas se definen sobre las operaciones que permiten la comunicación, y son las denominadas *precondiciones* y *postcondiciones*. Las precondiciones deben ser verdaderas en el momento de invocar una operación, y la responsabilidad de que esto se cumpla recae en quien la invoca. Las postcondiciones deben ser verdaderas al momento de finalizar la invocación a una operación, recayendo la responsabilidad de su satisfacción en el código que implementa esa operación.. Ambas deben ser consistentes con el conjunto de todas las demás restricciones definidas sobre la clase.

Las reglas pueden ser escogidas por el usuario para restringir en general las características de un modelo bien formado, y además para definir en particular la semántica de un modelo específico. El problema es que UML no establece, de manera clara, cómo agregar reglas específicas a un modelo existente. Con respecto a las restricciones, UML permite al usuario especificarlas en cualquier lenguaje, formal o informal. Precisamente, la importancia de herramientas como HEMOO reside en ofrecer al usuario la posibilidad de adaptar el concepto de modelo bien formado según el contexto y el momento particular en el proceso de desarrollo de software, ayudando en la identificación de posibles problemas en su definición.

El prototipo de HEMOO en el que se está trabajando está compuesto por tres subsistemas: Editor Gráfico, Editor de Propiedades, y Control Semántico. Ambos editores generan las representaciones internas para ser procesadas por el subsistema del Control Semántico.

El subsistema del Editor Gráfico tiene la responsabilidad de proveer al usuario las herramientas necesarias para definir un modelo en UML. Incluye facilidades para usar no sólo los elementos visuales, sino también los de texto, tales como restricciones, pseudocódigo y comentarios. Su implementación es en lenguaje Java. Este subsistema genera una representación interna de los elementos en un modelo que usará el subsistema del Control Semántico.

Para que la consistencia de las restricciones, y también su compatibilidad con las reglas semánticas sea verificada por HEMOO, la herramienta proporciona una sintaxis precisa para describirlas. Este lenguaje se llama el HEMOO Lenguaje de Restricciones (HEMOOLR), y tiene asociado una semántica formal, de manera que pueda ser controlada su consistencia..

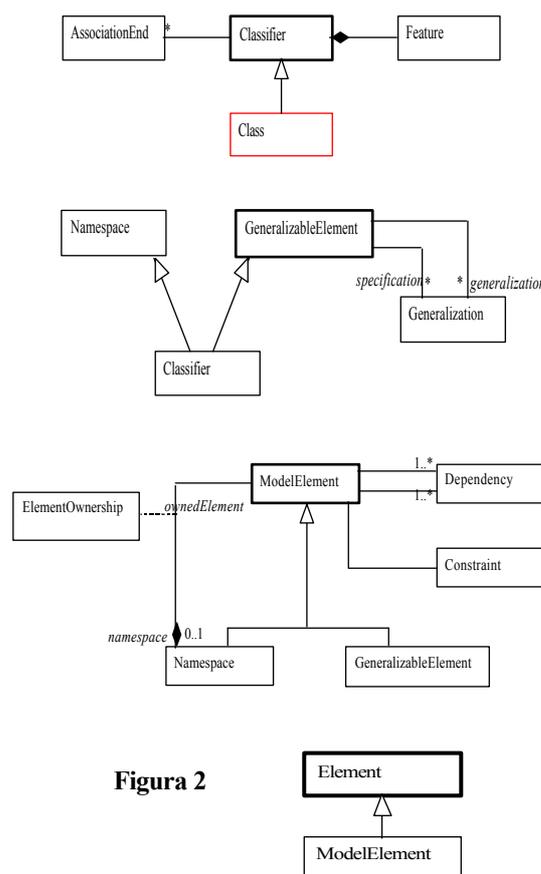


Figura 2

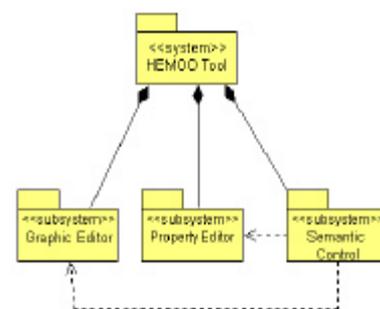


Figura 3

El subsistema del Editor de Propiedades es responsable de proporcionar las herramientas necesarias para introducir en el sistema las nuevas reglas definidas por el usuario. Esto no sólo incluye el nombre, y los posibles argumentos generales (tal como una clase o una relación), sino también su definición semántica. Para introducir la definición semántica de una regla, el subsistema permite al usuario especificar todas las propiedades existentes, predefinidas y definidas por el usuario, combinadas con elementos de programación lógica tales como conjunción, disyunción, negación por falla y predicados de segundo orden. La definición de propiedades así generada se usará en el Subsistema del Control Semántico.

El subsistema del Control Semántico es responsable de inducir al usuario a especificar las propiedades que quiere sean verificadas, e informar los resultados. Usa la representación del modelo y las propiedades generada por los otros dos subsistemas.

En una primera etapa, en esta línea de investigación, la herramienta estará enfocado a la Vista Estructural del sistema, y en particular a los Diagramas de Clase, que contiene clases, interfaces, paquetes, y las relaciones entre ellos. Una vez completada esta etapa, se planea extender el alcance a todos los Diagramas involucrados en la Vista Estructural.

### 3. Controles Semánticos Estructurales

La implementación de los controles semánticos se hace en lenguaje Prolog. El hecho de que este lenguaje provea una semántica bien definida para sus programas permite establecer una correspondencia directa con la semántica del modelo en desarrollo. Además, la existencia de intérpretes eficientes facilita el desarrollo de las herramientas automáticas necesarias para el control de tanto las reglas predefinidas como de las específicas del modelo. Otra ventaja es que se pueden aprovechar otro tipo de herramientas existentes como los generadores de intérpretes para gramáticas formales o la programación con restricciones (CLP).

La especificación del metamodelo UML incluye constructores abstractos –no instanciables-, cuya finalidad es la de definir estructuras compartidas y dar organización al modelo, y constructores concretos que son instanciables y reflejan los constructores del modelo. Entre los primeros se pueden mencionar los constructores `GeneralizableElement`, `ModelElement`, `Classifier`, mientras que entre los últimos se encuentran los constructores `Class`, `Attribute`, `Association`. Esta especificación, para cada uno de los constructores incluye tres aspectos:

- Sintaxis abstracta, escrita mediante notación gráfica y lenguaje natural,
- Reglas de buena formación: escritas en lenguaje formal (OCL) y lenguaje natural.
- Semántica: escrita en lenguaje natural – solo definida para los constructores concretos-.

La representación en Prolog consiste en una base de datos con los hechos que representan a cada uno de los constructores del metamodelo y un conjunto de reglas con la información necesaria para contemplar cada uno de los aspectos recién mencionados, y controlar su cumplimiento mediante el subsistema de Control Semántico.

Con respecto a las Restricciones, solo será posible controlar su correcta definición. Se entiende por correcta definición que: las expresiones booleanas sean sintácticamente válidas, que exista concordancia de tipos, que la visibilidad de los servicios involucrados sea la adecuada, que sean relaciones entre funciones, funciones y atributos o entre atributos. En el caso de que alguno de estos controles no pueda efectuarse, por no haberse completado aún la definición de la clase, deberá generar las *warnings* correspondientes.

El control de la correcta definición tanto de las Reglas específicas del modelo, como de las Restricciones, la efectúa el Editor de Propiedades, en el momento en que éstas son ingresadas.

Una vez asegurada su correcta sintaxis, el control de la consistencia de las restricciones y las reglas, se realiza traduciendo a Prolog las reglas específicas del modelo que estén bien definidas. En este punto, para controlar la inexistencia de conflictos entre reglas y/o restricciones, y dada la generalidad de lo que queremos

chequear, se presentarán tipos básicos primitivos sobre los que será posible aplicar restricciones, y se usarán herramientas (lenguajes CLP) que permiten controlar que estas restricciones son compatibles.

Dado que el modelo evoluciona, y no siempre es deseable controlar todas las reglas simultáneamente, la herramienta provee la posibilidad de que, si bien el modelo, una vez terminado, deberá cumplir todas las reglas, el usuario dispondrá de la facilidad de seleccionar cuáles reglas chequear en cada una de las etapas del desarrollo.

Actualmente se está desarrollando un prototipo de la herramienta con la funcionalidad descrita en este trabajo. La posibilidad de extensiones futuras es amplia. Aunque los Diagramas de Clase son importantes, pueden agregarse otros tipos de diagramas para la definición de un modelo. En otra línea de investigación se está trabajando actualmente en los Diagramas de Casos de Uso. En particular, como próximo paso planeamos extender el alcance de la herramienta al conjunto de todos los diagramas estructurales de UML.

### Referencias

[1] *The Unified Modeling Language User Guide*, Grady Booch, James Rumbaugh, Ivar Jacobson. Addison Wesley, 1999.

[2] *Object-Oriented Modeling and Design*, James Rumbaugh, Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen. Prentice Hall, 1991.

[3] *The Unified Modeling Language Reference Manual*, James Rumbaugh, Ivar Jacobson, Grady Booch. Addison Wesley, 1999.

[4] *OMG Unified Modeling Language Specification (draft)*. Object Management Group, Inc. Versión 1.3a1, January 1999.

[5] *Applying Logic Programming Techniques in a Tool for Object-Oriented Modeling in UML*, Pablo Fillottrani, Elsa Estévez, Susana Kahnert. Proceedings Software Engineering & Knowledge Engineering, June 2001.

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

# WICC 2002

*16 y 17 de mayo de 2002*

## Procesamiento Concurrente, Paralelo y Distribuido



# Construcción de un sistema GNU para plataforma SPARC de 64 bits

Hugo J. Curti <hcurti@exa.unicen.edu.ar>

INCA/INTIA - Facultad de Ciencias Exactas

Universidad Nacional del Centro de la Provincia de Buenos Aires

Campus Universitario - Pje. Arroyo Seco s/n (7000) Tandil - (02293)432466

## Resumen

El presente artículo tiene como objetivo describir la experiencia obtenida en la construcción de un sistema GNU completo sobre plataforma SPARC. Dicho sistema incluye todas las utilidades necesarias para formar una estación de trabajo, una estación de desarrollo o un servidor. Incluye un kernel del sistema operativo LINUX con sus módulos, las librerías más importantes, todos los comandos básicos de UNIX, el ambiente de desarrollo completo, el sistema XFREE86 completo y el ambiente de escritorio GNOME.

## 1 Introducción

Un sistema GNU está formado por un conjunto de programas aportados por diferentes desarrolladores en todo el mundo. Estos programas tienen en común que se encuentran protegidos por la licencia GNU. Las características fundamentales de esta licencia son las siguientes:

- Protege los derechos de autor de los programas.
- Garantiza la distribución del código fuente de los programas.
- Permite el libre uso y distribución de los programas.

Estas condiciones de licencia facilitaron la orientación de las herramientas de la llamada *cadena*

*de desarrollo* para diferentes plataformas y arquitecturas. A su vez, todo programa preparado para construirse con esta cadena de desarrollo puede construirse con poco o ningún esfuerzo extra en todas las plataformas hacia las cuales dicha cadena haya sido orientada. Esta operación de orientación de un programa a diferentes plataformas se denomina *portado*.

El kernel del sistema operativo, como cualquier otro programa, también puede ser portado. Sin embargo, las partes del mismo que interactúan directamente con el hardware, denominadas *drivers*, requieren código específico para cada plataforma, lo cual requiere un esfuerzo de desarrollo considerable. Algo similar ocurre con el servidor de terminal gráfica, que requiere código específico para cada controladora de video.

Como corolario de lo expuesto, se puede decir que para portar un sistema GNU a una plataforma es necesario que exista el código necesario para la orientación tanto en la cadena de desarrollo como en el kernel del sistema operativo. Si se desea portar la interfaz gráfica, se requiere además que el driver para la controladora de video esté presente en el servidor de terminal gráfica.

En la sección 2 se describen los componentes principales de un sistema GNU y su configuración. En la sección 3 se revisan los aspectos relevantes a este trabajo de la arquitectura SPARC y sus modalidades. En la sección 4 se describe la distribución ROCK LINUX, elegida como base para este trabajo. En la sección 5 se muestra el orden utilizado para desarrollar el trabajo. Por úl-

timo, en la sección 6 se presentan las conclusiones del trabajo.

## 2 Configuración de los programas GNU

Debido a la diversidad de arquitecturas, de plataformas y de sistemas que existen en plaza, para que un código pueda construirse en cualquiera de ellos se requiere una adaptación de dicho código a cada sistema particular. Este proceso de adaptación fue automatizado en la mayoría de los programas GNU por una herramienta denominada *autoconf*. A su vez, existe otra herramienta que automatiza el proceso de construcción de código denominada *make*.

### 2.1 La herramienta autoconf

La herramienta *autoconf* es utilizada por los desarrolladores para detectar y resolver problemas comunes de portabilidad en sus programas, a la vez que permite generar un script denominado *configure*.

El código de cada programa se distribuye con su propia versión de *configure*, con lo cual no es necesaria la presencia de la herramienta *autoconf* para realizar la configuración.

Para construir un programa GNU normalmente se debe ejecutar primero el script *configure* que generará los archivos de partida para la herramienta *make*. *Configure* puede recibir parámetros que permiten modificar su comportamiento o ajustar manualmente parámetros que no son correctamente detectados en forma automática.

### 2.2 La herramienta make

La herramienta *make* es un manejador de dependencias que permite automatizar la construcción y la instalación de un programa. *Make* decide, mediante sus archivos de configuración, la herramienta que se debe invocar para generar cada uno de los objetivos, que normalmente corresponden a los archivos binarios finales de los programas y sus archivos intermedios.

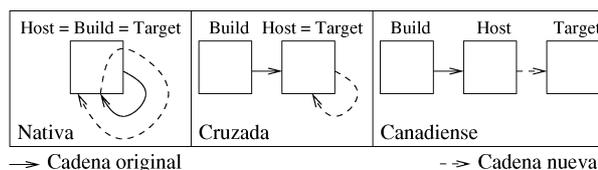


Figura 1: Modalidades de construcción.

### 2.3 La cadena de desarrollo de GNU

La cadena de desarrollo es la columna vertebral de todo el sistema [Frea]. Está formada por el conjunto de herramientas que permite crear código ejecutable a partir del código fuente de un programa: las utilidades binarias, el compilador, un conjunto de librerías estándar y un depurador.

### 2.4 Modalidades de construcción de la cadena de desarrollo

Existen tres parámetros que permiten definir la modalidad de construcción de la cadena de desarrollo [Freb]. Estos parámetros se pueden especificar en el script *configure* de las utilidades binarias y del compilador. Las librerías standard y el depurador se construyen utilizando los dos primeros elementos una vez que estén creados.

- El sistema de construcción (*build*) es aquel en el que se va a hacer la construcción de la cadena de desarrollo. Este sistema debe poseer ya una cadena de construcción correctamente configurada e instalada para crear programas en el sistema anfitrión.
- El sistema anfitrión (*host*) es aquel en el cual funcionará la nueva cadena de desarrollo.
- El sistema de destino (*target*) es aquel para el cual se generará código ejecutable.

Se denomina *construcción nativa* (*native build*) a la configuración en la cual los parámetros *build*, *host* y *target* son iguales. Se denomina *construcción cruzada* (*cross build*) a la configuración en la cual *build* y *host* son iguales y *target* es diferente y *construcción canadiense* (*canadian build*) a la configuración en la cual *build* y *host* son diferentes entre sí independientemente del valor de *target* [How]. La figura 1 muestra gráficamente las tres configuraciones.

## 3 La arquitectura *sparc*

### 3.1 Descripción

Para este trabajo se utilizó una estación de trabajo *Sun Ultra Sparc 10* con un procesador *ultrasparc II* que puede trabajar en un modo nativo con un ancho de palabra de 64 bits y en un modo de compatibilidad con procesadores anteriores de la misma familia cuyo ancho de palabra es de 32 bits. Además utiliza para el almacenamiento en memoria de los enteros el formato denominado *big endian*, en el cual se almacenan primero los bytes de mayor peso.

### 3.2 La cadena de desarrollo para *sparc* y *sparc64*

Los dos modos de trabajo del procesador conforman dos arquitecturas diferentes que requieren cadenas de desarrollo diferentes. La arquitectura del modo de 64 bits se denomina *sparc64*, mientras que del modo de 32 bits se denomina *sparc*. Al momento de la realización de este trabajo la cadena de desarrollo GNU para *sparc* se encontraba en un estado de madurez elevado, con lo que se podía construir perfectamente la cadena completa. No ocurría lo mismo con la versión para *sparc64*, para la cual las utilidades binarias funcionaban, el compilador requería ajustes y las librerías estándar aún no estaban desarrolladas.

### 3.3 El kernel de Linux para *sparc* y *sparc64*

Para el kernel de Linux tanto la versión *sparc* como la versión *sparc64* estaban desarrolladas al momento de la realización de este trabajo. Sin embargo no es posible utilizar un kernel para *sparc* en un procesador *ultrasparc II* debido a que este procesador debe ser inicializado con código de 64 bits [Roc02].

La versión 2.95 del compilador (última versión considerada estable al momento del desarrollo de este trabajo) no estaba preparada para construir el kernel en la arquitectura *sparc64*. Por lo tanto no se podía utilizar la vía convencional para construir el kernel. Existían sin embargo dos formas

de hacerlo:

- Utilizar una versión especial del compilador denominada *egcs64*, que contiene una serie de *parches* que permiten construir el kernel, aunque no sirve para construir aplicaciones.
- Utilizar la versión 3.0.4 del compilador (considerada inestable para uso general), que fue anunciada mientras este trabajo estaba en progreso y que sirve tanto para construir el kernel como para construir aplicaciones.

## 4 La distribución ROCK LINUX

Se eligió para este trabajo una distribución denominada ROCK LINUX [SSC00, W<sup>+</sup>]. Dicha distribución presenta las siguientes ventajas que facilitaron este trabajo [Wol99]:

- Se distribuyen códigos fuentes.
- El proceso de descarga, construcción e instalación se encuentra gobernado exclusivamente por *scripts* de shell que pueden ser seguidos y modificados con facilidad.
- Existe un grupo de trabajo activo que tiene como fin portar la distribución a la plataforma *sparc*.
- Posee una política de manejo de *parches*<sup>1</sup> mínima y eficiente.

### 4.1 Descarga

Los *scripts* y los archivos de configuración que conforman ROCK LINUX se distribuyen en forma separada de los paquetes. Una vez instalados los *scripts* se puede ejecutar *Download*, que buscará cada paquete por la Internet y lo descargará desde el sitio de su desarrollador y mediante

<sup>1</sup>Un *parche* (*patch*) es un archivo que codifica diferencias entre dos árboles de código fuente. Cuando se hace necesario modificar el código fuente original se genera este archivo parche que puede luego ser aplicado al código original para obtener el código modificado. Además puede ser transferido en forma semiautomática a nuevas versiones de código original.

una herramienta de espejado de web. Cualquier cambio o error que pueda producirse en los sitios puede reflejarse en forma sencilla en los archivos de configuración de la distribución y reintentar el *Download* [Pri02, WB<sup>+</sup>].

Los paquetes se dividen en dos conjuntos: el conjunto *base* formado por todos los paquetes considerados esenciales y el conjunto *extensiones*, conformado por los paquetes de menos relevancia. Para este trabajo se utilizó el conjunto *base* exclusivamente.

## 4.2 Construcción

La construcción es completamente automática mientras no haya errores y se realiza en cinco pasos [Pri02]:

- En el primer paso se construye un árbol de directorios básico de UNIX, se construye y se instala la cadena de desarrollo completa en dicho árbol y se configura el kernel.
- En el segundo paso se realiza un *chroot*<sup>2</sup> al árbol recién creado, se reconstruye la cadena de desarrollo, se construye el kernel, el shell y los comandos básicos de UNIX.
- En el tercer paso se reconstruye todo lo anterior y luego se construye el grueso de los paquetes.
- En el cuarto paso se construye la documentación que suele depender de los paquetes construidos en el paso anterior.
- En el quinto paso se reconstruyen todos los paquetes una vez más y se crean las versiones binarias para distribuir.

### 4.2.1 Errores de compilación

Cuando se producen errores de compilación todo el *script* de construcción se detiene. Entonces se debe revisar y corregir el error en forma manual. Luego se debe generar el *patch* y colocarlo junto

<sup>2</sup>La llamada *chroot* permite cambiar el directorio raíz de un proceso. Al hacer un cambio de raíz sobre el proceso de construcción, la misma se realizará sobre el mismo árbol donde residirá una vez instalada.

con los archivos de configuración de la distribución. El proceso de construcción se reanuda en el punto donde se detuvo.

### 4.2.2 Imagen ISO

Una vez construidos todos los paquetes de la distribución, siempre mediante *scripts*, se puede generar una imagen ISO que se puede grabar en un CD. El CD puede ser utilizado para iniciar un kernel que cargará en un disco emulado en memoria un sistema mínimo y permitirá iniciar la instalación.

Para este trabajo no fue posible crear una imagen ISO de inicio porque la herramienta no funcionaba correctamente para *sparc* en ese momento [Roc02].

## 4.3 Instalación y configuración

Para instalar se deben crear las particiones y los sistemas de archivos en forma manual. Luego se puede iniciar el *script* de instalación donde se pueden elegir los paquetes a instalar. Los paquetes generarán el mismo árbol de directorios donde fueron construidos. Por último se podrán generar en forma semiautomática o manual las configuraciones de los paquetes principales, quedando de esta forma el sistema instalado.

## 5 Desarrollo del trabajo

Debido al estado de inmadurez de la cadena de desarrollo en la arquitectura *sparc64* y a la necesaria disposición de un kernel *sparc64* se optó por una configuración mixta, donde el kernel es *sparc64* pero las librerías y todos los programas de usuario son *sparc*.

Al comenzar el trabajo se disponía de la estación de trabajo *Sun Ultrasparc 10* ya descrita con un sistema MANDRAKE LINUX antiguo y sin mantenimiento. El trabajo se desarrolló en los siguientes pasos:

- Se realizó una construcción nativa de la última versión de la cadena completa de desarrollo para *sparc*, utilizando la versión estable del compilador (gcc 2.95).

- Utilizando la cadena nueva se realizó una construcción cruzada de las utilidades binarias y el compilador (versión inestable gcc 3.0.4) hacia *sparc64*. Este paso requirió ajustes en el proceso de construcción del compilador debido a errores en la configuración de la herramienta *make*.
- Utilizando la cadena cruzada se construyó un kernel de Linux *sparc64* con los *drivers* necesarios para manejar la estación de trabajo, reemplazando con éste al viejo kernel del sistema MANDRAKE. La versión del kernel compilado correspondía al de la distribución.
- Utilizando la cadena de desarrollo para *sparc* se construyó la distribución ROCK LINUX completa. Los errores de compilación que fueron presentándose durante la construcción se debieron a problemas de portabilidad, por lo cual se generaron *parches* necesarios para que algunos paquetes puedan funcionar correctamente en *sparc*. Con la excepción de aquellos paquetes cuyos autores indicaban explícitamente que *no funcionaban en sparc*, toda la distribución se pudo compilar exitosamente.
- Ante la imposibilidad de crear la imagen ISO, se preparó una partición nueva en el disco y se corrió el *script* de instalación sobre ella.
- Se reemplazó manualmente el kernel de la distribución (*sparc*) por el kernel en uso y se instaló el cargador manualmente, porque la distribución solamente contemplaba el cargador para las arquitecturas x86.
- Se reinició el nuevo sistema. Se eliminó por completo el viejo sistema y se ordenaron las particiones.
- Se configuraron las aplicaciones y los servidores.
- Por último se instaló el servidor de terminal gráfica, que en un principio no se construía correctamente. Requirió la búsqueda de una serie de *parches* y la adaptación de los mismos a la distribución en uso.

## 6 Conclusiones

Como resultado de este trabajo se obtuvo un sistema GNU completo funcionando sobre una plataforma *sparc*, basado exclusivamente en software de libre distribución y con una interface que permite que un usuario de un sistema similar funcionando en una arquitectura x86 pueda utilizar la plataforma *sparc* con un costo de migración prácticamente nulo.

Los *parches* obtenidos a partir de este trabajo fueron enviados a [Roc02] para su inclusión en la distribución oficial.

## Referencias

- [Frea] **Free Software Foundation.** “GCC installation instructions.” <http://gcc.gnu.org/install>.
- [Freb] **Free Software Foundation.** “GCC manual.” <http://gcc.gnu.org/onlinedocs/gcc>.
- [How] **S. Howard.** “CrossGCC FAQ.”
- [Pri02] **P. Prins.** “ROCK linux guide.” <http://www.rocklinux.org>, 2002.
- [Roc02] “ROCK ports mailing list.” [rockports@rocklinux.org](mailto:rockports@rocklinux.org), 2002.
- [SSC00] **SSC.** “Rock linux.” *Linux Journal*, página 12, Oct 2000.
- [W<sup>+</sup>] **C. Wolf et al.** “ROCK linux distribution.” <http://www.rocklinux.org>.
- [WB<sup>+</sup>] **C. Wolf, D. Bakke et al.** “ROCK linux FAQ.” <http://www.rocklinux.org>.
- [Wol99] **C. Wolf.** “Rock linux philosophy.” <http://e-zine.nluug.nz/cid=1>, Nov 1999.

## "Modelos de predicción de performance en sistemas paralelos"

Laura De Giusti <sup>1</sup>, Marcelo Naiouf <sup>2</sup>, Armando De Giusti<sup>3</sup>

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>4</sup>.  
Facultad de Informática. Universidad Nacional de La Plata.*

### Resumen

Esta línea de investigación (dentro del proyecto de Procesamiento Concurrente, Paralelo y Distribuido del LIDI) se coordina con la Universidad Autónoma de Barcelona estudiando los modelos de predicción de performance en sistemas paralelos y analizando algunas modificaciones a los mismos según el tipo de arquitectura y paradigma de paralelismo utilizado.

En el estudio se analiza la respuesta de los modelos para "clases" de arquitecturas reconocidas (Memoria compartida, Memoria Distribuida, Memoria compartida Distribuida) con diferentes esquemas de comunicación: punto a punto, bus, múltiple bus. Asimismo se considera en cada caso la adaptación de diferentes paradigmas de programación paralela al modelo de arquitectura utilizado.

El aporte de la línea de investigación tiende a la especificación de un modelo unificador para la predicción de performance, en particular para procesos asincrónicos bajo el paradigma cliente-servidor y sobre una arquitectura tipo cluster de workstations.

### Palabras Claves

Paralelismo - Concurrencia - Modelos de Computación Paralela – Sincronización – Mensajes asincrónicos – Predicción de performance

---

<sup>1</sup> Licenciada en Informática. Becario UNLP. Jefe de Trab. Práct. SD Idgiusti@lidi.info.unlp.edu.ar

<sup>2</sup> Profesor Adjunto Dedicación Exclusiva. mnaiouf@lidi.info.unlp.edu.ar

<sup>3</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. degiusti@lidi.info.unlp.edu.ar

<sup>4</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.

TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

## Introducción

El *paralelismo*, asociado con *multiprocesamiento* en hardware y *conurrencia* en software requiere especificar e implementar procesos explotando la concurrencia implícita o explícita en el problema a resolver, y *al mismo tiempo* optimizar la adaptación del algoritmo desarrollado a la arquitectura física de soporte.

Los paradigmas de expresión de algoritmos paralelos se asocian entonces en forma directa con la arquitectura de procesamiento que soporta la ejecución, dando lugar a la noción de *sistema de procesamiento paralelo*, que incluye el software propio de la aplicación y el modelo de hardware y comunicaciones elegido.

Cuatro paradigmas básicos relacionados con los sistemas de procesamiento paralelo son los modelos *master-worker*, *pipeline*, *single processor multiple data* y *divide and conquer*. Cada uno de estos paradigmas (con sus variantes particulares) se adapta mejor a *clases* de problemas en programación paralela.

Avanzando en el concepto anterior, existen *patrones* de solución a problemas de procesamiento paralelo utilizando cada uno de estos cuatro paradigmas básicos, y dentro de cada uno de ellos hay un *modelo de prestación* o *performance* que nos puede indicar los límites alcanzables en cuanto a *speedup*, eficiencia y rendimiento de la solución paralela, en función de los parámetros propios de la aplicación y la arquitectura física de soporte.

Un área importante de investigación es la **predicción de performance** alcanzable por un *sistema paralelo* (normalmente asociado a uno de los patrones mencionado anteriormente). La especificación, transformación, optimización y evaluación de algoritmos distribuidos y paralelos, según las predicciones realizadas marca una posibilidad de optimización de soluciones en función de distintos modelos de arquitectura y el estudio de métricas de complejidad y eficiencia.

En el cómputo secuencial, el modelo RAM ha sido prácticamente aceptado como un estándar. En el procesamiento distribuido/paralelo no hay un modelo teórico unificador. Precisamente el objetivo de esta línea de investigación es profundizar el estudio de los modelos actualmente utilizados (PRAM, LOGP, BSP, CCM, BSPRAM, OBBSP) especialmente considerando arquitecturas MIMD débilmente acopladas, con procesadores homogéneos. En particular interesan los procesos asincrónicos bajo el paradigma cliente-servidor, sobre una arquitectura tipo cluster de workstations.

La *performance* obtenida en un sistema paralelo está dada por una compleja relación en la que intervienen una gran cantidad de factores (el tamaño del problema, la arquitectura de soporte, la distribución de procesos en procesadores, la existencia o no de un algoritmo de balanceo de carga, etc). Por otra parte, existen varias métricas para evaluar sistemas paralelos, tales como el tiempo de ejecución paralelo; el *speedup*; la eficiencia; la escalabilidad o la *isoeficiencia*, métricas que pueden tener mayor o menor importancia según la aplicación particular del sistema paralelo.

En este marco, interesa *predecir y evaluar performance* de sistemas paralelos, utilizando modelos de propósito general y medir la calidad de las estimaciones sobre diferentes arquitecturas reales.

## Temas de Investigación

- Sistemas Paralelos. Paradigmas de Programación paralela. Estudio de patrones para los paradigmas master-worker, pipeline, single processor multiple data y divide and conquer.
- Modelos de predicción de performance clásicos: PRAM, LOGP, BSP, CCM, OBSP, BSPRAM
- Análisis del ajuste de los modelos a “clases” de sistemas paralelos. Desarrollos experimentales.
- Métricas de performance en sistemas paralelos.
- Dependencia de los modelos respecto del balance de carga en los procesadores y la homogeneidad / heterogeneidad de los mismos.
- Análisis de la dependencia de los resultados obtenidos en función de cambios en la arquitectura de soporte. En particular pasar de la red homogénea con memoria distribuida y comunicación vía bus a un esquema multiprocesador con memoria compartida distribuida.
- Análisis de la escalabilidad de las soluciones para los diferentes patrones.
- Análisis de las extensiones necesarias para tener un modelo de predicción de performance para procesos asíncronos bajo el paradigma cliente-servidor.
- Aplicaciones paralelas de tratamiento de imágenes: Transformadas aplicables en la codificación de imágenes con pérdida, Clustering, Reconocimiento de patrones, Análisis de similitud de imágenes.

## Equipamiento de experimentación

En el LIDI se dispone de un cluster de PCs con 16 equipos homogéneos. Por otra parte se puede utilizar la computadora Clementina que tiene 40 procesadores con memoria compartida distribuida.

Naturalmente también se puede experimentar con arquitecturas seudo-paralelas tal como redes heterogéneas con un soporte de comunicaciones tipo PVM o MPI.

**BIBLIOGRAFIA BASICA**

- [And91]** Gregory R. Andrews. "Concurrent Programing" The Benjamin/Cummings Publishing Company. 1991.
- [And99]** Gregory R. Andrews. "Foundations on Multithread and Distributed Programing" Addison Wesley. 1999.
- [Bax94]** Gregory A. Baxes "Digital Image Processing. Principles and Applications" John Wiley & Sons Inc. 1994.
- [Bri95]** Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- [Bub97]** Bubak, Funika, Moscinski, "Performance Analysis of Parallel Applications under Message Passing Environments", [www.icsr.agh.edu.pl/publications/html/perf\\_full/](http://www.icsr.agh.edu.pl/publications/html/perf_full/), 1997.
- [Bust88]** Bustard, Elder, Welsh, "Concurrent Program Structures", Prentice Hall, 1988.
- [Cof92] M. Coffin, "Parallel programming- A new approach", Prentice Hall, Englewood Cliffs, 1992.
- [Cha88]** Chandhi K. M., Misra J., "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- [Esp 99]** Espinosa A., Parcerisa F., Margalef T., Luque E. "Relating the Execution Behaviour with the Structure of the Application" (Euro PVM/MPI) Lecture Notes in Computer Science, vol. 1697, 91-98. Springer, Alemania, 1999.
- [Esp 00a]** Espinosa A., Margalef T., Luque E. "Automatic Performance Analysis of Master/Worker PVM Applications with Kpi" (Euro PVM/MPI) Lectures Notes in Computer Science, vol. 1908, 47-55. Springer . Alemania, 2000.
- [Esp 00b]** Espinosa A., Margalef T., Luque E. "Integrating Automatic Techniques in a Performance Analysis Session" (Euro-Par 2000) Lectures Notes in Computer Science, vol. 1900, 173-177. Springer, Alemania, 2000.
- [Gon92]** Rafael C. Gonzáles, Richard E. Woods, "Digital Image Processing", Addison-Wesley Publishing Comp., 1992.
- [Heer91]** D. W. Heermann, A. N. Burkitt, "Parallel Algorithms in Computational Science", Springer-Verlag, 1991.
- [Hoar85]** C. A. R. Hoare, "Communicating Sequential Processes", Pentice-Hall, 1985.
- [Hus91]** Zahid Hussain "Digital Image Processing" Ellis Horwood Limited. 1991.
- [Hwa93]** Hwang K., "Advanced Computer Architecture: Paralelism, Scalability, Programability", McGraw-Hill, 1993.
- [IEEE]** Colección de Transactions on Parallel and Distributed Processing.
- [Kum94]** Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Desing and Analysis of Algorithms", Benjamin/Cummings, 1994.
- [Laws92]** H. Lawson, "Parallel processing in industrial real time applications", Prentice Hall 1992.
- [Lei92] Leighton F. T., "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 1992.
- [Luq95]** Luque E, Ripoll A, Cortès A, Margalef T, "A Distributed Diffusion Method for Dynamic Load Balancing on Parallel Computers", Proceedings of the EUROMICRO Workshop on Parallel and Distributed Processing, IEEE Computer Society, Jan. 1995.
- [Mor94]** Morse F., "Practical Parallel Computing", AP Professional, 1994.
- [Pri01]** Printista Marcela, "Modelos de Predicción en Computación Paralela", Tesis de Magister en Ciencias de la Computación, 2001.
- [Sim97]** Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.
- [Tin98]** Tinetti F., De Giusti A., "Procesamiento Paralelo. Conceptos de Arquitectura y Algoritmos", Editorial Exacta, 1998.
- [Zom96]** Zomaya A. (ed), "Parallel Computing. Paradigms and Applications", International Thomson Computer Press, 1996.

## *Procesamiento Paralelo y Distribuido en tratamiento masivo de datos*

**Armando De Giusti<sup>1</sup>, Marcelo Naiouf<sup>2</sup>, Cecilia Sanz<sup>3</sup>,  
Fernando Tinetti<sup>4</sup>, Laura De Giusti<sup>5</sup>, Rodolfo Bertone<sup>6</sup>**

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>7</sup>.  
Facultad de Informática. Universidad Nacional de La Plata.*

### **Palabras Clave**

Procesamiento Paralelo y Distribuido. Programas Paralelos. Análisis de performance. Balance de carga. Escalabilidad. Métricas del paralelismo. Aplicaciones de tratamiento masivo de datos.

### **Resumen**

Investigar el procesamiento paralelo y distribuido aplicado a tratamiento masivo de datos.

Los temas fundamentales propuestos en el proyecto se refieren a la especificación, transformación, optimización y verificación de algoritmos concurrentes ejecutables en sistemas paralelos/distribuidos, la optimización de clases de soluciones en función de modelos de arquitectura multiprocesador y las métricas de complejidad y eficiencia relacionadas con el procesamiento paralelo.

Se trabaja experimentalmente son cuatro modelos de arquitectura multiprocesador, disponibles/accesibles en el LIDI.

Interesa especialmente la aplicación de estas investigaciones al tratamiento masivo de datos de imágenes (genéricas, médicas y de cultivos), de datos numéricos (cómputo científico) y de bases de datos distribuidas.

### **Introducción**

El procesamiento ha evolucionado hacia el paralelismo prácticamente desde el inicio mismo de las computadoras digitales. En la actualidad resulta innegable la importancia y el creciente interés en el procesamiento distribuido y paralelo dentro del espectro de la Ciencia de la Computación.

Existen diversas razones para esta realidad, entre las que se pueden citar:

- El crecimiento de la potencia de cómputo, dado en la evolución de la tecnología de los componentes y en las arquitecturas de procesamiento (supercomputadoras, hipercubos de procesadores homogéneos, grandes redes de procesadores no-homogéneos, procesadores de imágenes, de audio, etc.)
- La transformación y creación de algoritmos que explotan al máximo la concurrencia implícita en el problema a resolver, de modo de distribuir el procesamiento minimizando el tiempo total de respuesta. Naturalmente esta transformación también debe adaptarse a la arquitectura física de soporte.
- La capacidad del cómputo distribuido/paralelo de reducir el tiempo de procesamiento en problemas de cálculo intensivo o de grandes volúmenes de datos.
- La necesidad de tratar con sistemas de tiempo real distribuidos, donde los requerimientos en relación al tiempo de respuesta son críticos.

---

<sup>1</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. [degiusti@lidi.info.unlp.edu.ar](mailto:degiusti@lidi.info.unlp.edu.ar)

<sup>2</sup> Profesor Titular. [mnaiouf@lidi.info.unlp.edu.ar](mailto:mnaiouf@lidi.info.unlp.edu.ar)

<sup>3</sup> Profesor Adjunto. [csanz@lidi.info.unlp.edu.ar](mailto:csanz@lidi.info.unlp.edu.ar)

<sup>4</sup> Profesor Adjunto. [fernando@lidi.info.unlp.edu.ar](mailto:fernando@lidi.info.unlp.edu.ar)

<sup>5</sup> Jefe de Trabajos Prácticos. [ldgiusti@lidi.info.unlp.edu.ar](mailto:ldgiusti@lidi.info.unlp.edu.ar)

<sup>6</sup> Profesor Adjunto. [pbertone@lidi.info.unlp.edu.ar](mailto:pbertone@lidi.info.unlp.edu.ar)

<sup>7</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.

TE/Fax +(54) (221) 422-7707. <http://lidi.info.unlp.edu.ar>

- El límite físico alcanzado por las computadoras secuenciales, que en algunos casos torna inaceptable el tiempo para resolver determinados problemas, y hace que la solución paralela sea la única factible.
- La existencia de sistemas en los que no es tan importante la velocidad de cómputo sino la necesidad de resolver problemas en más de una ubicación física a la vez, capacidad que puede asociarse rápidamente a una configuración paralela
- Las posibilidades que el paradigma paralelo ofrece en términos de investigación de técnicas para el análisis, diseño y evaluación de algoritmos.

Es fundamental referirse a un algoritmo paralelo mencionando el modelo de computación paralela para el que se lo diseñó. En el cómputo secuencial, el modelo RAM ha sido prácticamente aceptado como estándar. En el procesamiento distribuido/paralelo no existe un modelo teórico unificador. Esto se debe a que los diferentes modelos paralelos (PRAM para memoria compartida, BSP para memoria distribuida, LogP para redes de procesadores asincrónicos, etc), enfatizan determinados aspectos a costa de otros. Una razón más sutil es que no es probable una máquina paralela universal; para cada aplicación existe una máquina óptima. Por otro lado, existen distintas alternativas de implementación sobre sistemas de cómputo paralelo con hardware de procesamiento homogéneo o heterogéneo y con acoplamiento fuerte o débil de sus componentes.

Por esto, lo correcto es hacer referencia a *sistemas paralelos* como la combinación de algoritmo y arquitectura. La performance obtenida en un sistema paralelo está dada por una compleja relación en la que intervienen una gran cantidad de factores (el tamaño del problema, la arquitectura de soporte, la distribución de procesos en procesadores, la existencia o no de un algoritmo de balanceo de carga, etc).

Por otra parte, existe un gran número de métricas para evaluar sistemas paralelos (tiempo efectivo, speed-up, eficiencia, producto procesador-tiempo, escalabilidad, isoeficiencia).

En este marco, interesa evaluar la performance de distintas clases de aplicaciones sobre diferentes arquitecturas reales para obtener resultados concretos. El énfasis está en adecuar los resultados teóricos ideales en las métricas a la realidad de la implementación de determinada clase de problema sobre distintas plataformas. Este interés se basa en que muchos sistemas paralelos no alcanzan su capacidad teórica, y las causas de esta degradación son muchas y no siempre fáciles de determinar.

El análisis sobre plataformas disponibles permite estudiar el impacto que tienen algunos de estos factores sobre las implementaciones, y adecuar las métricas clásicas a las mismas. En particular puede estudiarse la influencia de las estrategias de distribución de procesos y datos, y la carga (estática o dinámica) asignada a cada procesador sobre el speedup, la eficiencia y la escalabilidad. Más allá de las características teóricas/algorítmicas de las aplicaciones, el rendimiento real y por lo tanto el tiempo necesario para resolver los problemas siempre o en la mayoría de los casos dependerá de la arquitectura de procesamiento elegida para la implementación.

Muchas disciplinas entre las que se encuentran el reconocimiento de patrones en tiempo real, el tratamiento y transmisión de video en tiempo real y la visión por computadora, requieren un importante esfuerzo en la investigación de algoritmos paralelos aplicables en áreas tales como robótica, industria manufacturera, ingeniería forestal y medicina. Interesa investigar la optimización de algoritmos paralelos para el reconocimiento de patrones de similitud, enfocando particularmente aplicaciones orientadas a grandes bases de datos de imágenes. En particular interesa el problema de *análisis de similitud de imágenes* basado en el estudio de la transformada Haar-Wavelet multidimensional que permite obtener una "firma digital" (codificación) de la imagen en estudio (que sea invariante a las traslaciones, rotaciones y escalado) y buscar otras imágenes que tengan una codificación similar dentro de una Base de Datos. La complejidad del procesamiento con esta técnica requiere dividir la imagen en regiones

y a su vez procesar la transformada wavelet dentro de cada región mediante una ventana deslizante de tamaño variable, lo que obliga a estudiar la implementación de algoritmos paralelos que permitan tener tiempos de respuesta razonables, más aún si el análisis de similitud se requiere que sea realizado en tiempo real. El objetivo es desarrollar una arquitectura de software paralela que permita hacer análisis de similitud de imágenes (por ejemplo fotográficas o de video) en tiempo real.

Otras áreas de interés incluyen el análisis de imágenes hiperspectrales de cultivos, los cálculos sobre matrices, y los problemas de Reconocimiento de patrones complejos.

### **Temas de Investigación y desarrollo**

- Optimización del empleo de recursos de procesamiento.
  - ✓ Balance de carga, tanto en procesamiento como en comunicaciones.
  - ✓ Distribución óptima de procesos y procesadores.
  - ✓ Migración de datos y/o procesos en forma dinámica.
  - ✓ Ajuste dinámico de código paralelo.
- Paralelización de aplicaciones.
  - ✓ Transformación de algoritmos.
  - ✓ Adaptación de algoritmos a modelos de arquitectura.
  - ✓ Especificación y verificación de sistemas paralelos de tiempo real
- Métricas de paralelismo.
  - ✓ Análisis de complejidad de los algoritmos secuenciales y paralelos.
  - ✓ Análisis de rendimiento de las soluciones paralelas.
  - ✓ Caracterización de paradigmas de procesamiento paralelo y modelos de performance asociados.
- Arquitecturas de procesamiento paralelo.
  - ✓ Rendimiento en función de la capacidad potencial de procesamiento de las arquitecturas de cómputo paralelo (tal como supercomputadoras paralelas y redes o clusters de estaciones de trabajo)
  - ✓ Reconfiguración dinámica del hardware y/o del software para adaptarse a la aplicación y/o proponer tolerancia a fallas
  - ✓ Diseño de hardware paralelo orientado a clases de aplicaciones.
- Aplicaciones específicas
  - ✓ Análisis de similitud de imágenes
  - ✓ Análisis de imágenes hiperspectrales a áreas agrícolas
  - ✓ Aplicaciones médicas de Redes neuronales paralelas
  - ✓ Computo científico sobre arquitecturas distribuidas
  - ✓ Análisis de rendimiento y replicación en Bases de Datos distribuidas

### **Equipamiento de experimentación**

Los modelos de arquitectura experimental propuestos (y disponibles para el proyecto) son cuatro: arquitecturas multiprocesador homogénea fuertemente acopladas tipo cubo de transputer, arquitecturas multiprocesador distribuida (homogénea y heterogénea) débilmente acoplada tipo NOW, multiprocesadores dedicados a tratamiento de imágenes basados en DSPs y arquitecturas multiprocesador de memoria compartida distribuída tipo SGI Origin 2000 (Clementina).

## Bibliografía Básica

- ACM Siggraph (Conferencias realizadas anualmente desde 1995)
- Akl S, "Parallel Computation. Models and Methods", Prentice-Hall, Inc., 1997.
- Baxes G.. "Digital Image Processing". Wiley, 1994
- Bell, David; Grimson, Jane, "Distributed Database Systems", Addison Wesley. 1992
- Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- Date, C.J., "Introducción a los sistemas de Bases de Datos". Addison Wesley 1994.
- Glassner A., "Principles of Digital Image Synthesis", Morgan Kaufmann, 1995.
- Gonzalez and Woods, "Digital Image Processing", Addison-Wesley, 1992
- Gupta A., Kumar V., "Performance properties of large scale parallel systems", Journal of Parallel and Distributed Computing, November 1993.
- Hwang K., "Advanced Computer Architecture: Paralelism, Scalability, Programability", McGraw, 1993.
- IEEE Transactions on Image Processing (colección de revistas 1990-2002)
- IEEE Transactions on Parallel and Distributed Processing (colección de revistas 1990-2002)
- Jain Anil, "Fundamentals of Digital Image Processing", Prentice Hall, 1989
- Jensen. "Introductory Digital Image Processing. A remote sensing perspective" 2da edición. Prentice Hall. 1996
- Kohonen T., "Self-Organizing Maps", Springer-Verlag, 1997
- Kosko B., "Neural Networks and Fuzzy Systems", Prentice Hall, 1992
- Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Design and Analysis of Algorithms", Benjamin/Cummings, 1994.
- Larson, J., "Database Directions. From relational to distributed, multimedia, and OO database Systems". Prentice Hall. 1995
- Lawson H., "Parallel processing in industrial real time applications", Prentice Hall 1992.
- Leighton F. T., "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 1992.
- Maravall Gomez-Allende, "Reconocimiento de Formas y Visión Artificial", Addison-Wesley Iberoamericana, 1994.
- Miller R., Stout Q. F., "Algorithmic Techniques for Networks of Processors", CRC Handbook of Algorithms and Theory of Computation, M. J. Atallah, ed, 1998.
- Morse F., "Practical Parallel Computing", AP Professional, 1994.
- Peddle. "An Empirical comparison of evidential reasoning, linear discriminant analysis, and maximum likelihood algorithms for alpine land cover classification". Canadian Journal of Remote Sensing. Vol.19. Nro.1- 1993
- Peddle. "Knowledge Formulation for Supervised Evidential Classification". Photogrammetric Engineering & Remote Sensing. Vol. 61- Nro. 4 – 1995
- Silbershatz, F., "Fundamentos de las Bases de Datos".. Mc Graw Hill. 1998.
- Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.
- Thomas M Lillesand, Ralph W Kiefer. "Remote Sensing and Image Interpretation" 3<sup>rd</sup> Edition. John Wiley. 1994
- Tinetti F., De Giusti A., "Procesamiento Paralelo. Conceptos de Arquitectura y Algoritmos", Editorial Exacta, 1998.
- Zomaya A., "Parallel Computing. Paradigms and Applications", Int. Thomson Computer Press, 1996.

# Una solución estática al problema de Buffer Overflow

Javier Echaiz\*      Rafael B. García      Jorge R. Ardenghi

Laboratorio de Investigacion de Sistemas Distribuidos (LISiDi)  
Departamento de Ciencias e Ingenier a de la Computacion  
Universidad Nacional del Sur  
Bah a Blanca - Buenos Aires - Argentina  
e-mail: {je,rbg,jra}@cs.uns.edu.ar

## Resumen

El Buffer Overflow es desde hace mas de una decada el mayor de los problemas que amenaza la seguridad en sistemas. La clave del exito para tratar de mitigar sus efectos parece encontrarse en el analisis estatico. Es nuestra intencion desarrollar un sistema estatico (que extiende a LCLint) de deteccion de posibles problemas de esta naturaleza analizando el codigo fuente de los programas de aplicacion. Una de las mayores ventajas de un sistema de este tipo es que se pueden evitar problemas de seguridad antes de que los programas se encuentren en funcionamiento.

**Palabras Clave:** buffer overflow, seguridad en sistemas, sistemas operativos.

## 1 Introduccion

Durante el año 2000 dos estudios [1, 2] indicaron que la mayoría de los ataques no provienen de problemas de seguridad recientemente descubiertos sino de viejos y conocidos problemas. El CERT y la NSA señalaron que los ataques por explotación de buffer overflow son hoy en día el mayor problema de seguridad [3] y que seguirán siendo un problema importante durante los próximos 20 años [4].

Los programas escritos en lenguaje C son particularmente susceptibles a ataques por buffer overflow, pues en las consideraciones de diseño de dicho lenguaje se priorizó espacio y performance sobre seguridad. Como consecuencia de este diseño, C permite manipular punteros sin ningún tipo de chequeo de límites, característica peligrosa por naturaleza debido a que muchas librerías estándar de C incluyen muchas funciones que se tornan “inseguras” si no son utilizadas cuidadosamente. La importancia del problema reside en el hecho de que la mayor parte de los programas críticos de seguridad están escritos en este lenguaje.

Una posible solución al problema es emplear mecanismos en tiempo de ejecución para mitigar los riesgos asociados al persistente buffer overflow. Sin embargo, estas técnicas no están siendo suficientemente empleadas como veremos en la próxima sección. Nuestra propuesta es la de detectar posibles vulnerabilidades en buffers mediante el análisis del código fuente de la aplicación.

---

\*Becario de la Comisión de Investigaciones Científicas de la Provincia de Buenos Aires, Argentina.

Este trabajo involucra una serie de ideas provenientes de diversas áreas de la computación, incluyendo análisis de programas, teoría, compiladores y seguridad en sistemas. Esta nueva herramienta tendrá en cuenta los siguientes tres principios: (1) explotar comentarios semánticos agregados al código fuente para controlar propiedades interprocedurales; (2) emplear técnicas estáticas que logren buena performance y escalabilidad; (3) emplear heurísticas sobre *loops*, de forma tal que sea posible analizar loops presentes en programas típicos.

## 2 Técnicas Dinámicas de Defensa y Ataque

El ataque más simple y conocido de buffer overflow es el *stack smashing* [5]. Consiste en sobrescribir un buffer en el stack con el objetivo de reemplazar la dirección de retorno. Cuando la función ejecuta el `return`, en lugar de saltar a la dirección de retorno original, el control pasará a la dirección que fue colocada en el stack por el atacante. Esto concede al atacante la capacidad de ejecutar código arbitrario. Como mencionamos en la sección previa, C brinda acceso directo a memoria y aritmética de punteros sin ningún tipo de control de límites. Más grave aún, C incluye funciones inseguras en su librería estándar, como `gets`, que permiten copiar una cantidad ilimitada de información introducida por el usuario en un buffer de tamaño fijo sin controlar los límites de dicho buffer. Muchas veces estas funciones reciben buffers pasados como parámetro mediante el stack, facilitando la tarea del atacante, quien únicamente debe ingresar una entrada de longitud mayor a la del buffer y codificar un programa binario en dicha entrada. El famoso Internet Worm de 1988 [7] aprovechaba esta vulnerabilidad presente en el `fingerd`. Existe otro tipo de ataque de buffer overflow, donde se emplea el heap en lugar del stack. Este ataque es más sofisticado que el anterior, pues la mayoría de los sistemas operativos no permiten saltar a direcciones cargadas desde el heap o hacia código almacenado en el heap.

Existen varias implementaciones que atacan el problema en tiempo de ejecución. StackGuard [8] es un compilador que genera binarios que incluyen código diseñado para prevenir ataques del tipo stack smashing. Este compilador ubica un valor especial a continuación de la dirección de retorno y chequea antes de saltar que ésta no haya sido modificada. Baratloo, Singh y Tsai describen dos métodos que actúan en tiempo de ejecución: uno reemplaza las funciones inseguras de la librería estándar de C por versiones seguras; el otro modifica los ejecutables con la intención de chequear la dirección de destino alojada en el stack antes de saltar.

Otra técnica dinámica es la denominada *software fault isolation* (SFI), mediante la cual se insertan instrucciones antes de las operaciones de memoria para prevenir el acceso a regiones fuera de rango [10]. Esta técnica ofrece escasa protección contra ataques típicos de buffer overflow dado que no evita que un programa escriba en la dirección del stack donde se almacena la dirección de retorno. Existen versiones de SFI que agregan controles más extensivos alrededor de las instrucciones potencialmente peligrosas para restringir de forma más general el comportamiento de los programas y así lograr una mayor protección. Algunos ejemplos de sistemas que utilizan esta técnica son Janus [11], Naccio [12] y Generic Software Wrappers [13].

El sistema operativo puede prevenir algunos tipos de ataque por buffer overflow haciendo que el código y los datos se encuentren en segmentos de memoria separados, donde el segmento de código es *read only* y no se pueden ejecutar instrucciones almacenadas en el segmento de datos. Esto no elimina totalmente el problema, pues el atacante todavía puede sobrescribir una dirección almacenada en el stack y hacer que el programa salte hacia cualquier dirección del segmento de código. Peor aún, si el programa utiliza librerías compartidas, el atacante puede saltar a una dirección (en el segmento de código) que puede ser utilizada maliciosamente (e.g.,

una llamada a `system`). Es claro entonces que esta técnica no solo no soluciona el problema real sino que también evita la ejecución de código automodificable legítimo.

### 3 Prototipo de una Solución Estática

Sin considerar la disponibilidad de las técnicas vistas que actúan en tiempo de ejecución, es evidente que no son actualmente una solución viable al problema de buffer overflow. Probablemente esto se debe a la falta de conciencia acerca de la severidad del problema y a la falta de soluciones prácticas. Por otra parte es cierto que en algunos ambientes de trabajo, las soluciones en tiempo de ejecución son inaceptables. Por ejemplo StackGuard reporta un overhead en la performance del 40% [6]. El otro problema real que presentan este tipo de soluciones es que convierten un ataque de buffer overflow en uno de *denial of service*, pues generalmente no pueden hacer más que abortar la ejecución del programa frente a un intento de explotación de buffer overflow.

El control estático soluciona estos inconvenientes detectando posibles vulnerabilidades antes de que el programa se ejecute. Detectar vulnerabilidades de buffer overflow analizando código fuente es evidentemente un problema no decidible<sup>1</sup>. La intención de éstos métodos no es la de tratar de verificar que un programa no tenga vulnerabilidades debido a buffer overflow, sino la de detectar la mayoría de los posibles problemas de este tipo antes de que el programa se encuentre en funcionamiento. Podemos tratar de conseguir entonces buenos resultados a pesar de que esta técnica sea incompleta y no sensata. Esto significa que nuestra herramienta ocasionalmente generará falsas alarmas (falsos positivos) y fallará esporádicamente en detectar algunos problemas reales (falsos negativos). A pesar de esto, nuestro objetivo es desarrollar una herramienta capaz de producir resultados útiles sobre programas reales utilizando un esfuerzo razonable, dejando a cargo del programador el chequeo “a mano” de un número reducido de posibles vulnerabilidades.

Nuestra herramienta estática ampliará las capacidades de LCLint [9], una herramienta de chequeo estático que emplea anotaciones en el código fuente. LCLint es capaz de detectar entre otras, violaciones a ocultamiento de información, referencias a punteros nulos, problemas en la instanciación de parámetros y modificación inconsistente de variables globales.

El usuario agregará comentarios semánticos (de aquí en adelante llamados *anotaciones*) al código fuente tanto de la aplicación como de las librerías estándar. Las anotaciones describirán las intenciones y asunciones del programador, las cuales pasarán desapercibidas por el compilador de C pero serán reconocidas como entidades sintácticas por nuestra herramienta. Utilizaremos entonces la siguiente notación (a grandes rasgos) para especificar una anotación dentro del código fuente:

$$\text{anotación} := * \$ \text{requerimiento} \$ *$$

Podemos por ejemplo utilizar `* $ notnull $ *` para indicar que el valor de una variable debe ser distinto de NULL.

No es difícil imaginar que el *requerimiento* debe ser suficientemente expresivo como para poder realizar una detección efectiva de vulnerabilidades de buffer overflow. LCLint contempla un número pequeño de casos posibles, y será necesario por ejemplo saber cuanta memoria fue asignada a un determinado buffer. Por lo tanto deberemos extender LCLint para que soporte un

---

<sup>1</sup>Trivialmente podemos reducir el problema de la detención al problema de detección de buffer overflow agregando instrucciones que causen un buffer overflow antes de todas las instrucciones `halt`.

lenguaje de anotaciones más expresivo mientras mantenemos la simplicidad de sus comentarios semánticos.

El programador podrá explicitar el estado de una función mediante precondiciones y poscondiciones utilizando cláusulas<sup>2</sup>. Podemos utilizar dichas cláusulas para describir las asunciones acerca de los buffers que se pasan como parámetros a funciones y controlar que se cumplan las restricciones cuando la función retorna. Nuestra herramienta utilizará (inicialmente) cinco tipos de asunciones y restricciones: *minSet*, *maxSet*, *minRead*, *maxRead* y *nullterminated*.

Cuando *minSet* y *maxSet* se encuentran presentes en una cláusula como requerimiento, indicarán los límites iniciales y finales (respectivamente) que puede soportar un buffer de forma segura (e.g., del lado izquierdo de una asignación (*lvalue*)). Por ejemplo, consideremos una función que recibe dos parámetros: un arreglo *a* y un entero *i* y que tiene la precondición *maxSet(a) >=i*. El análisis asume que al principio del cuerpo de la función *a[i]* puede ser utilizado como *lvalue*. Si *a[i+1]* fuese usado antes de cualquier modificación a *a* o a *i*, LCLint generaría un *warning* ya que las precondiciones de la función no son suficientes para garantizar que *a[i+1]* puede utilizarse de forma segura como *lvalue*. Los arreglos en C comienzan en 0 por lo tanto la declaración `char buffer[MAXSIZE]` genera las restricciones *maxSet(buffer) =MAXSIZE-1* y *minSet(buffer) = 0*.

Análogamente, las restricciones *minRead* y *maxRead* aplicadas a un buffer indican los límites de lectura seguros, mínimo y máximo respectivamente. Obviamente *maxRead(buffer)* siempre será menor o igual que *maxSet(buffer)*. En aquellos casos donde existan algunos elementos de un buffer todavía no inicializados *maxRead(buffer)* puede ser menor que *maxSet(buffer)*.

Cuando LCLint encuentra una llamada a función, controla que las precondiciones implicadas por la cláusula de requerimientos sean satisfechas antes de realizar dicha llamada. Para el ejemplo de *maxSet(a) >=i* devolvería un *warning* si no puede determinar con certeza si el arreglo pasado en *a* puede alojar al menos tantos elementos como el valor pasado en *i*.

Si tenemos *nullterminated* en una anotación, podremos monitorear que el buffer de almacenamiento termine con el caracter NULL. Muchas funciones de la librería estándar de C requieren la presencia del terminador nulo en los *strings* que reciben como parámetros, si este requerimiento no se satisface pueden aparecer vulnerabilidades por explotación de buffer overflow.

Muchos problemas de buffer overflow aparecen en funciones de librería, como por ejemplo `strcpy`. Parte de nuestra tarea será entonces agregar anotaciones a librerías estándar para así poder detectar inconvenientes incluso antes de agregar anotaciones en el programa de aplicación.

### 3.1 Implementación y Trabajos Futuros

Nuestra herramienta será implementada en C bajo Solaris combinando técnicas tradicionales de construcción de compiladores con generación y resolución de restricciones. Los programas serán analizados a nivel de función y los análisis interprocedurales se efectuarán mediante la información contenida en las anotaciones. En el caso de que se obtengan resultados promisorios implementaremos una versión de esta herramienta que haga uso de un sistema subyacente de DSM<sup>3</sup> con la intención de lograr una mayor performance, especialmente en el procesamiento de código fuente de aplicaciones de gran tamaño (decenas o centenas de miles de líneas de código).

<sup>2</sup>LCL, predecesor de LCLint, incluía una notación general de pre y poscondiciones mediante cláusulas.

<sup>3</sup>Nuestro grupo de investigación se encuentra desarrollando un sistema de DSM que cuenta con un modelo de consistencia scope y un protocolo de coherencia basado en home.

## 4 Conclusiones

Los mecanismos estáticos de detección de problemas de buffer overflow desarrollados previamente indican que éste es un buen curso de acción frente al persistente problema de buffer overflow que atenta contra la seguridad en sistemas. Es claro que las técnicas estáticas no son ni sensatas ni completas (no es una solución óptima) pero somos optimistas con respecto a los posibles resultados que obtendremos mediante la aplicación de la herramienta descrita brevemente en este trabajo. Este sistema será capaz de evitar muchos de los problemas de seguridad antes de que los programas se encuentren en funcionamiento, evitando así que el usuario tenga que aplicar un *patch* a un programa luego de que el atacante aprovechó una vulnerabilidad.

El principal objetivo de esta herramienta es mejorar la seguridad de los sistemas en general y el problema de buffer overflow en particular, pero además, agregar anotaciones al código fuente redundará en un mejor entendimiento del funcionamiento de los programas y en una mejor capacidad para su mantenimiento.

## Referencias

- [1] C. Cowan, C. Pu. Buffer Overflows: Attacks and Defenses for the Vulnerability of the Decade. *DARPA Information Survivability Conference*, enero 2000.
- [2] D. Wagner, J. Foster. A First Step Towards Automated Detection of Buffer Overrun Vulnerabilities. *Network and Distributed Security Symposium*, febrero 2000.
- [3] R. Pethia. Bugs in Programs. *SIGSOF Symposium on the Foundations of Software Engineering*, noviembre 2000.
- [4] B. Snow. Future of Security. *IEEE Security and Privacy*, mayo 1999.
- [5] Aleph One. Smashing the Stack for Fun and Profit. *BugTraq archives*, 1996.
- [6] C. Cowan, S. Beattie. Protecting Systems from Stack Smashing Attacks with StackGuard. *Linux Expo.*, mayo 1999.
- [7] E. Spaford. The Internet Worm Program: An Analysis. *Purdue Tech Report 832*, 1988.
- [8] C. Cowan, D. Maier. Automatic Detection and Prevention of Buffer-Overflow Attacks. *7<sup>th</sup> USENIX Security Symposium*, enero 1998.
- [9] D. Evans, J. Guttag. LCLint: A Tool for Using Specifications to Check Code. *SIGSOF Symposium on the Foundations of Software Engineering*, diciembre 1994.
- [10] R. Wahbe, S. Lucco. Efficient Software-Based Fault Isolation. *14<sup>th</sup> ACM Symposium on Operating Systems Principles*, 1993.
- [11] I. Goldberg, D. Wagner. A Secure Environment for Untrusted Helper Applications: Confining the Wily Hacker. *6<sup>th</sup> USENIX Security Symposium*, julio 1996.
- [12] D. Evans, A. Twyman. Flexible Policy-Directed Code Safety. *IEEE Symposium on Security and Privacy*, mayo 1999.
- [13] T. Fraser, L. Badger. Hardening COTS Software with Generic Software Wrappers. *IEEE Symposium on Security and Privacy*, mayo 1999.

# Un Modelo de Programación Paralelo para Memoria Distribuida

Jacqueline Fernandez, Mónica Fuentes, Fabiana Piccoli, Marcela Printista \*

Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950  
5700 - San Luis  
Argentina

e-mail: {gfuentes, jmfer, mpiccoli, mprinti}@unsl.edu.ar

## 1 Introduction

En los primeros estados de la computación paralela una de las soluciones más comunes para introducir paralelismo fue extender un lenguaje secuencial con algún conjunto de *constructores* comúnmente denominados *forall*. Aunque la mayoría tiene la misma sintaxis, ellos difieren en su semántica e implementación. Entre las implementaciones más populares están las propuestas por High Performance Fortran (HPF)[7] y OpenMP [9]. En este trabajo se propone una extensión a la iteración *forall* para el lenguaje C. La iteración *forall* define un modelo de programación, el cual se caracteriza por:

- Permitir y explotar varios niveles de paralelismo: *Paralelismo anidado*[2], [3].
- Expandir el modelo de pasaje de mensajes.
- Posibilitar la asociación de un modelo de complejidad.

En consecuencia, este modelo brinda, por un lado la posibilidad de anidar sentencias *forall* y generalizar el modelo de programación de pasaje de mensajes (cada *forall* crea un comunicador, hablando en términos de MPI[8]); por el otro, posibilita el análisis y la predicción de la performance en base al modelo de complejidad asociado.

## 2 Modelo: Sintaxis y Semántica

Como se mencionó, el modelo de programación que se propone extiende el paradigma secuencial imperativo con la inclusión de nuevas sentencias: Iteraciones Paralelas. A continuación se muestra una versión simplificada de su sintaxis:

**forall(i,first,last,f(i),res<sub>i</sub>,size<sub>i</sub>)**

La variable *i* representa la variable de iteración, la cual toma valores enteros entre *first* y *last*. Cada iteración sobre *i* es independiente de las demás y se realiza en paralelo. Las variables *res<sub>i</sub>* y *size<sub>i</sub>* representan, respectivamente, el resultado de la *i*-th iteración y su correspondiente tamaño.

---

\*Grupo subvencionado por la UNSL y ANPCYT (Agencia Nacional para la Promoción de la Ciencia y Tecnología)

Para explicar la semántica se necesita definir una máquina paralela. Supongamos una arquitectura con infinitos procesadores, cada uno con su memoria privada y unidos a los otros a través de una red de interconexión. Los procesadores son organizados en grupos, al comienzo de la computación todos pertenecen al mismo grupo. Los procesadores de un mismo grupo tienen el mismo estado de memoria, y se asume que tienen los mismos datos de entrada y ejecutan el mismo programa. Cada procesador es una máquina *RAM*[1] y se diferencia de las demás en el estado de su registro interno, inaccesible para el programador, el cual contiene su nombre, *NAME*. Cuando un procesador encuentra una iteración *forall* decide, según su *NAME*, el valor de *i* y en consecuencia a que subgrupo pertenece y cual es su nuevo *NAME*. Los procesadores de cada subgrupo ejecutan la correspondiente  $f(i)$ . Las siguientes expresiones muestran como se realiza la división de los procesadores, el mapping de estos a las distintas tareas y el intercambio de los resultados.

Cantidad de Iteraciones a realizar:  $M = last - first + 1$

Iteración a hacer:  $i = first + NAME \% M$

for ( $j = 1; j < M; j++$ )

$partner[j] = \Phi + (NAME + j) \% M$

donde  $\Phi$  es:  $\Phi = M \times (NAME / M)$

El nuevo valor de *NAME* es calculado por:  $NAME = NAME / M$

Al finalizar el *forall*, cada procesador recupera su propio valor de *NAME*. Cada vez que una iteración *forall* es ejecutada, la memoria de los procesadores del grupo contiene exactamente los mismos valores. En tal punto la memoria es dividida en dos partes: aquella que será modificada y aquella que no tendrá ningún cambio dentro de la iteración. Las variables que pertenecen al segundo conjunto estarán disponibles en el ámbito del *forall* para lectura. Las otras serán particionadas entre los distintos grupos. En todo momento de la ejecución, los procesadores pertenecientes a un mismo grupo deben tener una visión consistente de la memoria y parecer un mismo proceso. Para lograr esto es necesario proveer de un mecanismo de intercambio de información entre los procesadores socios dentro del *forall*.

La semántica impone dos restricciones:

1. Los resultados  $res_i$  y  $res_j$  correspondientes a la ejecución de las iteraciones independientes  $i$  y  $j$ , cumplen con:

$$res_i \cap res_j = \emptyset \quad \forall i, j$$

2. Para todo  $T_i$  (denominamos  $T_i$  a la ejecución del cuerpo de la  $i$ -th iteración  $f(i)$ ), el espacio de memoria para sus resultados debe ser ubicado previamente a la ejecución de  $T_i$ . Esto indica que no es posible usar estructuras dinámicas de memoria, como listas y árboles, en el resultado de cada iteración.

La primer condición es mandatoria, durante la ejecución de cada  $T_i$ , los resultados  $res_i$  no pueden compartir direcciones de memoria.

### 3 Mapping y scheduling

Desafortunadamente una máquina infinita, tal como la descrita aquí, es una máquina ideal. Las máquinas reales poseen un número limitado de procesadores. Cada procesador tiene un registro interno, *NUMPROCESSORS*, donde se almacena el número de procesadores disponibles en su grupo corriente.

Tres diferentes situaciones deben ser consideradas en la ejecución de un *forall* con  $M = last - first + 1$  iteraciones:

- $NUMPROCESSORS = 1$ ;
- $M \geq NUMPROCESSORS$ ;
- $NUMPROCESSORS > M$ ;

El primer caso es trivial. Un único procesador ejecuta todas las iteraciones secuencialmente y no existe la oportunidad de explotar el paralelismo intrínseco.

El segundo caso ha sido ampliamente estudiado como paralelismo chato. Su principal problema es el balance de carga. Para solucionarlo varias políticas de scheduling fueron propuestas [9]. Muchas políticas de asignación son posibles, la asignación puede ser por: *bloques*, *bloques-cíclica*, *guiada* o *dinámica*.

El tercer caso trae consigo problemas adicionales como es el balance de la carga. Su solución implica que los subgrupos en que se dividieron los procesadores no serán del mismo tamaño. La solución al balance de la carga se encuentra en la modificación de la política de distribución de procesadores vista anteriormente. Solucionar el balance de la carga trae consigo otro problema: determinar la relación de sociedad entre los procesadores de distintos grupos. La relación de sociedad genera un *polytope* donde cada procesador, en cada grupo, tiene la propiedad de tener uno y sólo un socio, al cual le envía sus resultados.

Actualmente, existe una implementación del modelo, la cual utiliza la librería de pasaje de mensajes sobre *puBSP*[4]. Considerando el modelo y como un test exhaustivo, se están desarrollando varias aplicaciones. Entre ellas se encuentran: la *Multiplicación de Matrices*[10], *La Transformada rápida de Fourier*[6] y el problema de los *N-cuerpos*[5].

## 4 Conclusiones

El modelo de programación aquí expuesto, permite expresar paralelismo a través de las sentencias *forall*. Admite además, varias extensiones como son las cláusulas de reducción.

El modelo propuesto extiende el modelo de pasaje de mensajes estandar, a la vez que se caracteriza por ofrecer:

- Facilidad de Programación.
- Portabilidad a otras plataformas.
- Predictibilidad de la performance al tener asociado un modelo de predicción confiable.

Todas estas características lo hacen un modelo muy interesante para la investigación y el desarrollo de otros modelos orientados a él.

## Referencias

- [1] Aho, A. V. Hopcroft J. E. and Ullman J. D.: The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Massachusetts, (1974).
- [2] Ayguade E., Martorell X., Labarta J., Gonzalez M. and Navarro N. Exploiting Multiple Levels of Parallelism in OpenMP: A Case Study Proc. of the 1999 International Conference on Parallel Processing, Aizu (Japan), September 1999.
- [3] Blikberg R., Sørøvik T.. Nested parallelism: Allocation of processors to tasks and OpenMP implementation. Proceedings of The Second European Workshop on OpenMP (EWOMP 2000). Edinburgh, Scotland, UK. 2000

- [4] Bonorden O., Huppelshauer N., Juurlink B., Rieping I. PUB library, Release 6.0 - User guide and function reference. University of Paderbon, Germany. (1998)
- [5] Barnes J., Hut P. A hierarchical  $O(N \log N)$  force calculation algorithm. nature. Vol 324.P. 446,1986.
- [6] Cooley, J. W. and Tukey, J. W.: An algorithm for the machine calculation of complex Fourier series, Mathematics of Computation, **19**, 90, (1965) 297–301.
- [7] High Performance Fortran Forum: High Performance Fortran Language Specification. Version 2.0 <http://dacnet.rice.edu/Depts/CRPC/HPFF/versions/hpf2/hpf-v20/index.html> (1997)
- [8] MPI Forum: MPI-2: Extensions to the Message-Passing Interface, <http://www.mpi-forum.org/docs/mpi-20.ps.Z> (1997).
- [9] OpenMP Architecture Review Board: OpenMP Specifications: FORTRAN 2.0. <http://www.openmp.org/specs/mp-documents/fspec20.ps> (2000).
- [10] Quinn M. Parallel Computing. Theory and Practice. Second Edition. McGraw-Hill, Inc.

# Consistencia de Memoria en Sistemas DVSM

Rafael B. García      Javier Echaiz\*      Jorge R. Ardenghi

Laboratorio de Investigación de Sistemas Distribuidos (LISiDi)  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur  
Bahía Blanca - Buenos Aires - Argentina  
e-mail: {je,rbg,jra}@cs.uns.edu.ar

**Palabras clave:** Shared Memory, SM. Distributed Shared Memory, DSM. Distributed Virtual Shared Memory, DVSM. Uniform Memory Access, UMA. Non-Uniform Memory Access, NUMA. Massively Parallel Processing, MPP. Secuential Consistency, SC. Scope Consistency ScC.

## Introducción a los DVSM

Las arquitecturas paralelo, obtenidas combinando varios procesadores, pueden suministrar potencia de cómputo órdenes de magnitud superior a las arquitecturas de procesador único.

Un aspecto clave que diferencia a estas arquitecturas es el mecanismo de soporte para la comunicación. En las arquitecturas de pasaje de mensaje cada procesador tendrá su propia memoria local solo accesible por el mismo, requiriendo que los procesos se comuniquen a través de mensajes explícitos. Contrariamente, multiprocesadores con un espacio único de memoria, sistemas SM, al permitir que toda la memoria pueda accederse desde los distintos procesadores, posibilita una comunicación con operaciones de lectura y escritura en memoria.

Los sistemas SM, referidos usualmente como rigidamente acoplados, proporcionan una memoria global la cual será accesible por igual por todos los procesadores. Esto permite compartir datos a través de un mecanismo uniforme de lectura/escritura sobre la memoria común, UMA, aunando facilidad de programación y portabilidad. Como contrapartida estos sistemas estarán expuestos a un incremento en la contención y latencia de los accesos a memoria, afectando performance y escalabilidad.

Si bien con sistemas eficientes de caché e interconexiones con un gran ancho de banda se atenúan estas limitaciones al punto de posibilitar arquitecturas con varias decenas de procesadores, ejemplo Starfire Ultra 10000 con 24 a 64 procesadores, este esquema resulta inviable para combinar un número muy superior de procesadores.

Se tiene así la necesidad de distribuir la memoria, reteniendo el concepto de un espacio único de direcciones, dando lugar a los sistemas DSM. Una porción de la memoria total compartida se asigna a cada procesador, dando lugar a dos tipos de acceso a memoria: local y remoto. Mejora la escalabilidad pero incorpora un mecanismo no uniforme para la memoria por la disparidad entre los tiempos de acceso locales y remotos, en lo que se conoce como sistemas NUMA. Su limitación es la asignación estática de los espacios de direccionado.

A su vez los sistemas distribuidos, llamados normalmente multicomputador, corresponden a nodos de procesamiento con su memoria independiente, vinculados por alguna red de interconexión. Estos sistemas han demostrado ser escalables y ciertamente posibilitan sistemas de enorme potencia de cómputo, ejemplo los sistemas MPP.

Tradicionalmente, la comunicación entre procesos en un sistema distribuido se basa en el modelo de pasaje de datos. Sistemas con pasaje de mensajes o sistemas que soportan llamadas a procedimientos remotos, RPC, adhieren a este modelo. En sí extiende el mecanismo de comunicación del sistema: abstracciones como pórticos o mailbox junto con primitivas como Send y Receive son usadas para la comunicación entre procesos, pudiendo ocultar esa funcionalidad a través del lenguaje como con el mecanismo RPC. En cualquier caso los procesos pasan información por valor, contrariamente a esto, sistemas con un espacio de dirección compartido podrán hacerlo por referencia.

Resulta de interés retener en estos sistemas debilmente acoplados el paradigma de memoria compartida, dando lugar a los sistemas con memoria virtual distribuida compartida, DVSM.

La principal ventaja de un DVSM sobre el convencional de pasaje de datos es la abstracción mas simple y entendible que posibilita al programador. Permite una transición mas natural de las aplicaciones secuenciales a las distribuidas. En principio, cómputos paralelo y distribuido escritos para SM podrán ser ejecutados en DVSM sin cambios. El sistema de memoria compartida oculta el mecanismo de comunicación remota de los procesos permitiendo que estructuras complejas puedan ser pasadas por referencia, simplificando substancialmente la programación y facilitando la migración de procesos. Más aún, el dato en un DVSM podrá persistir mas allá del tiempo de vida del proceso que accede la memoria compartida.

El modelo de pasaje de mensajes fuerza al programador a ser consciente todo el tiempo del movimiento de datos entre procesos, por el manejo explícito de las primitivas de comunicación y los pórticos o canales. Además dado que el pasaje de datos es entre múltiples espacios de dirección, resulta difícil pasar estructuras complejas de datos. Las mismas deberán ser marshaled y unmarshaled por la aplicación. (Marshaling se refiere a la linearización y empaquetado de una estructura de datos en un mensaje).

Los sistemas DVSM comparten metas con una variedad de sistemas como por ejemplo memoria caché de los procesadores de un sistema SM, memoria local en los sistemas NUMA, caché distribuida de los file systems y base de datos distribuidas entre otros. Particularmente, todos ellos intentan minimizar el tiempo de acceso a datos que son potencialmente compartidos y que serán mantenidos consistentes. Aunque similares a distancia, tendrán cuestiones de detalle e implementación que varían significativamente por la diferencia en el costo y uso de sus parámetros.

## Clasificación de los sistemas DVSM

La gran atención que ha merecido éste tipo de arquitecturas y el importante número de trabajos de investigación y desarrollo relacionado con el tema lo podemos entender por dos cuestiones convergentes:

1. El gran desarrollo de los microprocesadores, tanto a nivel de su arquitectura como de la tecnología de circuitos integrados de gran escala VLSI
2. El desarrollo de las redes de interconexión del tipo local y de sistemas, que proporcionan significativos anchos de banda en la comunicación

Lo anterior hace potencialmente competitiva la combinación de cientos o miles de estos micro-procesadores potentes y de reducido costo por su producción de gran escala.

Convengamos que un DVSM posibilita el paradigma de programación de memoria compartida en sistemas que poseen memoria físicamente distribuida, mas específicamente sistemas sin un espacio de direcciones único. Esta abstracción se logra con acciones específicas para acceder desde los diversos nodos a los datos del espacio virtual global compartido. Hay tres cuestiones fundamentales para la performance que se refieren a las acciones para traer el dato al sitio donde es requerido y a su vez mantenerlo consistente con respecto a los otros procesadores, las que dan a su vez espacio para clasificarlos según

- Cómo se realiza el acceso **Algoritmo**
- En dónde el acceso es implementado **Nivel de implementación**
- Cúal es la semántica de consistente **Modelos de consistencia**

## Algoritmo de soporte de la memoria compartida

El primer tópico da lugar a diferentes algoritmos de memoria compartida según la existencia o no de múltiples copias de un dato y de sus permisos de acceso pudiendo aplicarse a la distribución de datos la estrategia de replicación o migración.

- SRSW single reader/single writer
  1. Sin migración
  2. Con migración
- MRSW multiple reader/single writer
- MRMW multiple reader/multiple writer

La replicación no está permitida en SRSW mientras que podrá darse en los algoritmos MRSW y MRMW. Resultan fundamentales en el desempeño del algoritmo las características particulares de la aplicación. Si bien los algoritmos predominantes son MRSW, los MRMW aunque más complejos de implementar, posibilitan un superior nivel de concurrencia que en ciertas implementaciones resultará decisivo.

El aspecto negativo de la performance en los algoritmos que implementan memoria distribuida es que son sensitivos al comportamiento de las aplicaciones. No se tendrá un único algoritmo que funcione aceptablemente en un amplio rango de aplicaciones. Esto sugiere un esfuerzo para adoptar el algoritmo apropiado dada una aplicación. Más aún, podría incrementarse la performance significativamente con sintonía fina sobre cómo la aplicación realiza el acceso a memoria o sintonía del propio algoritmo para el particular comportamiento de la aplicación, aunque eliminando la ventaja de transparencia en los accesos a memoria compartida.

## Nivel de la Implementación

Como se realiza el acceso es una de las decisiones mas importantes, dada su incidencia tanto en la programación como en la performance y costo. Se tendrá:

- Software
  - Librerías de run time
  - Sistema Operativo

- 2. Fuera del kernel
  - Compilador
    - Hardware
    - Combinación Hardware/Software

Las implementaciones por software son normalmente librerías de run time que serán linkeadas con la aplicación que usa dato compartido, ejemplo Trademark, o esta abstracción podrá realizarse a nivel de lenguaje de programación, dado que el compilador puede detectar accesos compartidos e insertar llamadas a rutinas de sincronización y coherencia. Otro tipo de solución se alcanza incorporando un mecanismo DSM en el sistema operativo distribuido, dentro o fuera del kernel. A menudo se combinan elementos de distintas alternativas, por ejemplo IVY que es una runtime library que incluye modificaciones al SO.

Las implementaciones de hardware buscan superar las limitaciones propias de las alternativas de software, como velocidad y granularidad de los accesos (típicamente de una página) entre otras. Así, podemos mencionar MERLIN que a través de una memoria reflectiva mapeada dinámicamente permite al usuario compartir datos a relativa alta velocidad y con baja latencia, de manera automática y concurrente con el cómputo. Otro ejemplo de este nivel de implementación es el KSR1 que plantea, inspirado en lo que fue en su momento el advenimiento de la memoria virtual, algo análogo para las arquitecturas MPP, en el sentido que el usuario se desentienda del movimiento entre nodos tanto de los datos como del código, en lo que sería entre dos niveles para el ejemplo de memoria virtual.

Referido a la alternativa híbrida encontramos que soluciones de hardware, como COMA, se complementan con software como en Simple-COMA. En general se podrán hacer los casos infrecuentes por software para minimizar complejidad, o alternativamente a las soluciones de software incorporarles hardware para acelerar las operaciones frecuentes.

## Modelos de consistencia

Dado que un sistema de memoria compartida permite que múltiples procesadores lean y escriban simultáneamente la misma locación de memoria, los programadores requieren un modelo conceptual para la semántica de la operación de memoria que le permita usarla correctamente. Esto se conoce como modelo de consistencia de memoria o modelo de memoria. Este modelo deberá ser intuitivo y fácil de usar. Normalmente optimizaciones a nivel de arquitectura y de compilador que posibilitan eficiencia en un espacio de direcciones único complicarán el comportamiento de memoria llevando a que diferentes procesadores tengan distinta visión de la memoria compartida. El problema en diseñar un sistema de memoria compartida se sintetiza en presentar al programador una visión del sistema de memoria que sea simple de usar y aun así posibilite las optimizaciones necesarias para un soporte eficiente del espacio de memoria compartido. Un tercer factor de importancia resulta ser la portabilidad.

Muchos de los modelos propuestos fallan en ofrecer una semántica razonable de programación, o bien sacrifican performance por la programabilidad.

En el estudio y evaluación de los diversos modelos tres cuestiones deberán ser tenidas en cuenta: primero cómo el modelo se presenta al programador y cómo este impacta tanto a la programación como a la portabilidad, luego las restricciones que el modelo impone en el sistema y las técnicas para una implementación eficiente, y por último la performance del modelo y complejidad requerida en la implementación para alcanzar dicha performance.

Los programadores prefieren razonar con un modelo simple e intuitivo como es el SC, luego el

información sobre las operaciones de memoria en una ejecución secuencialmente consistente. Esta le permite al sistema determinar optimizaciones en el ordenamiento que pueden ser explotadas sin violar la consistencia secuencial. Mas aún, la misma información podrá ser usada para portar el programa a distintas implementaciones, de manera automática y eficiente.

## Lineamientos del Trabajo

Estamos trabajando en un sistema híbrido que, si bien es por software aprovechando los mecanismos de memoria virtual, introduce hardware para la actualización automática. Comparte con Brazos el trabajar con el modelo ScC pero se diferencian en que este último es por software unicamente y no trabaja con el concepto de home para las páginas. Las ventajas de la implementación en estudio son reducción significativa de la estructura de datos y de procesamiento, con perspectiva de mejora en performance y escalabilidad.

Referido a performance se reduce el trabajo a realizar dentro de las zonas críticas, muy conveniente por el efecto serializador en situaciones de gran contención, además de desaparecer efectos secundarios como cache pollution. En cuanto a escalabilidad tiene la ventaja de que la estructura de datos escalaría con la complejidad del problema y no con el tamaño del cluster, como ocurre típicamente en este tipo de sistemas. Además, el protocolo en cuestión manejaría satisfactoriamente aplicaciones con granularidad fina, por el prefetch implícito a través de la actualización automática.

Se está trabajando en su posible simulación y además en la posibilidad de manejar home dinamicamente para hacerlo más eficiente, sopesando la eventual sobrecarga inducida.

## Referencias

- [1] David Mosberger: Memory Consistency Models. TR 93/11 Department of Computer Science, The University of Arizona
- [2] Pete Keleher, Alan L. Cox, and Willy Zwaenepoel. Lazy Release Consistency for Software Distributed Shared Memory, Department of Computer Science, Rice University
- [3] Liviu Iftode, Cezary Dubnicki, Edward W. Felten and Kai Li. Improving Release Consistent Shared Virtual Memory using Automatic Update, 2nd IEEE Symposium on High-Performance Computer Architecture, February 1996
- [4] Liviu Iftode, Jswinder Pal Singh and Kai Li. Scope Consistency: a Bridge between Release Consistency and Entry Consistency, 8vo Annual ACM SPAA96
- [5] Evan Speight and John K. Bennett. Brazos: A Third Generation DSM System, Proceedings of the First USENIX Windows NT Workshop, August, 1997
- [6] Mathias Blumrich, Kai Li, Richard Alpert, Cezary Dubnicki, Edward Felten, and Jonathan Sandberg. Virtual Memory Mapped Network Interface for the SHRIMP Multicomputer, Proceedings of the 21st International Symposium on Computer Architecture, April 1994
- [7] Creve Maples and Larry Wittie. MERLIN: A superglue for Multicomputer Systems, COMCON'90

## Perception and Enterprise Communication Networks to Improve the Requirements Elicitation Process

Juan M. Luzuriaga, Rodolfo Martínez, and Alejandra Cechich  
*Departamento de Informática y Estadística, Universidad Nacional del Comahue,  
Buenos Aires 1400, Neuquén, Argentina*  
*phone: (54) 299 - 4490312, fax: (54) 299 4490313*  
*Email: jluzuria, rodom, acechich@uncoma.edu.ar*

**Abstract.** The requirements elicitation phase of software development projects is characterised by the intensity and importance of communication activities. Requirements elicitation is a traditional exploratory phase in which context is analysed and an abstraction is performed as a consequence. However, exploratory processes are characterised by a deep interaction with environmental factors. In this paper, we present a process for managing communication during the requirements elicitation phase. Our process would help get well-defined requirements by using knowledge inside organisations and a classification of requirements perception based on how environment is controlled by stakeholders. Our approach can be used to guide the elicitation process as well as to validate the requirements.

### 1. Introduction

In spite of the highly technical and formal nature of software, creating it remains a very people-intensive activity. It is well known that users should be involved in a requirements elicitation process, however, there are few advances in this particular area, since the process still depends on the skills and experience of the system analyst who is eliciting the requirements. Unfortunately, deficient requirements are the single biggest cause of software project failure [1]. Stakeholder dialog is a pillar of the requirements elicitation process. Typically, the analyst converses with the customers about their needs. In turn, the analyst may raise questions about the requirements, which lead to further conversations. Like many dialogs, requirements elicitation can be difficult to manage [2]. It is widely recognised that communication problems are a major factor of software projects. The organisational domains into which such software is introduced are often too intricate to be fully understood. Hence requirements descriptions are necessarily uncertain [3].

From our empirical studies on auditing quality management standards in enterprises, which are trying to adapt or implement a quality management system, we have built a set of heuristics and strategies for improving the requirements elicitation process. Principles, such as *transparent communication* [4], have been used to detect how to gather information and how to define requirements in such a way that feedback and user's support are always encouraged.

We know that most RE activities are exploratory. RE engineers acquire and discover requirements and knowledge about the problem domain by conducting an information search. Exploratory activities are characterised by a deep interaction between stakeholders and the environment, which is always abstracted and interpreted. Environment affects stakeholders before and after receiving behaviour as a response of a stimulus. Environment conditions, behaviour itself, and consequences are interrelated in such a way that a stimulus detected in the environment acquires certain control on answers. In this case, stimulus does not trigger an answer as a reflex behaviour. Simply, the occurrence of an answer is more probable, and even more, it can occur combined with other conditions that affect its probability. It is common that an answer reinforced in a given situation occurs in a similar situation due to a generalisation process. However, if the answer is reinforced

only when a particular property is present, it acquires an exclusive control by a process called *discrimination* [5]. Our strategy also present a classification of stakeholders' perception of requirements based on controlling environmental behaviour, which can be used for detecting incomplete and conflicting requirements.

## 2 A Strategy for Improving the Elicitation Process

Knowledge is extracted from information transported by a signal (oral language, gesture language, etc.) through a channel. In spite of different types of organisations exist, the following enterprise communication networks, which are the support of all information, can be found:

- *Hierarchical Network*: a formal organisation that is represented as a hierarchy connecting managers to subordinates – a diagram known as flow-chart. In general, this network is not effective to allow information flows.
- *Informational Network*: information related to organisation's functions. This view describes relationships among the information used by the enterprise. It includes all information forms and notes how their placement and distribution support users and applications.
- *Expert Network*: information needed to accomplish a task. It includes how a particular task should be done as well as how some problems should be solved. The main pitfall of this kind of network is that, sometimes, specialised information exists, but experts don't become aware of the methods needed to get it. To deal with this situation, a new information network can be produced as a consequence. Expert networks can be composed of manuals, specific procedures, and of course, experts.

Enterprise communication networks constitute the formal and informal structure on which enterprise communication flows. Strictly speaking, an enterprise has its own formal communication channels that are called *communicating flows* or *vectors* used to transmit messages throughout the organisation [4]. Regarding to the hierarchical and informational networks, we should first analyse communicating flows inside the organisation. They can be (1) *upward and downward*: these flows allow that communication through the hierarchical network occurs, and they are used as a command-reporting channel; (2) *horizontal*: these flows allow that communication among employees on the same hierarchical level occurs, and they are used for accomplishing routine teamwork; and (3) *transversal*: these flows allow that communication among employees in different areas that are not connected through the hierarchical network occurs.

In our empirical studies on several enterprises in different realms (Clinics, Newspapers, China producers, Hydroelectric enterprises) a sample of tasks that involves more than one area of each organisation was used to detect communicating flows or vectors, and to determine their predominant direction – upward/downward, horizontal, or transversal. After that, a strategy was defined to conduct the requirements elicitation process [6][7]. As a first heuristic, we found that *when predominant vectors are horizontal and upward/downward, starting elicitation through organisation's areas is more convenient. On the other hand, when transversal vectors are significant, starting elicitation through organisation's processes is easier.*

Depending on our behaviour and the executive director's decisions, our arrival is always a cause of uncertainty in an organisation. Uncertainty generates entropy, in such a way that the greater uncertainty our presence generates the greater is the entropy. As a consequence, a considerable

amount of gathered information is useless because users are unwilling to share their domain knowledge. Therefore, as a second heuristic, *we should try to minimise entropy by using a strategy that considers every kind of communicating flow inside the organisation.*

Our strategy is based on three pillars: reducing uncertainty generated by changes, detecting enterprise communication networks to know where requirements should be elicited, and detecting the communication flows or vectors that flow inside the networks to know how to proceed in the elicitation process. This strategy can be used along with many other techniques. For example, contextual-inquiry technique uses five models in which, after an appropriate number of site visits, the requirements team interprets the data and compiles a user profile that represents the common traits of all the users observed and interviewed [8]. Analysing requirements could perfectly be done by “guiding” the interviewing process using enterprise networks.

Two kinds of products are generated when our strategy is applied: (1) a diagram of each enterprise network written in UML notation for deployment diagrams [9], and (2) bi-dimensional tables representing the communicating flows and also comments and “fears” of our users.

The following strategy’s activities are usually applied without a specific order. It was introduced here as a way of clarifying the concepts.

1. *Detect Enterprise Communication Networks*: Selecting appropriate people is a fundamental activity to achieve a successful elicitation process. Specifically, the way of approaching a selection depends on the communication network (hierarchical, informational, or expert) that we want to model. Knowledge depends on previous exposition to reinforced contingency conditions, but we also acquire a special kind of knowledge when we are capable of proclaim rules or laws. That is, a person can “know” how to use an instrument because he or she has read the instructions, even though he or she has never used it. Knowledge that allows people to describe contingencies is very different from knowledge identified with the behaviour produced by contingencies. Decomposing contextual perceptions into its constituting parts can help define more precise and traceable software requirements specifications. To do that, some activities should be taken into account. First of all, every element of a contextual perception (CP) should be clearly stated and documented. Several techniques for documenting data, processes, and constraints exist, so we omit the details here for brevity. According to our approach, elements of a CP are analysed from different perspectives (provided by stakeholders) and they are also influenced by the role played by stakeholders. So, we have more than one view of the same reality and we need to conciliate these different views. We propose using a contextual perception matrix as a documentation of all perceptions made on the same reality (one matrix’s row). Every element of the matrix is a reference to the documents that collect all the information (data, processes, and constraints) supplied by a particular stakeholder. Of course, the information is strongly influenced by the role played by the stakeholder. It should be validated taking into account the actual source of the information (an expert, a naïve user, etc.). Perception of different realities can be documented in the same way, and hence new rows are added to the matrix [10].
2. *Use Communication Flows*: As we previously mentioned, we should analyse every area of an organisation to detect how messages flow through the networks in order to accomplish different works. Identifying the direction of the vectors is the main goal when we are detecting them. Focusing on vectors underlines the importance of a multi-directional communication. Therefore, a horizontal vector among peers could facilitate the interchange of information among areas, increase internal cohesion, facilitate management processes, and encourage communication. An

upward vector could help improve the social dialog in the enterprise stimulating consensus and supporting a participative system. A downward vector ensures that every member of a process or sector knows and understands the goals and objectives of the organisation. Interpersonal communication is also improved. Finally, a transversal vector could help define a common language among all stakeholders and select activities according to the organisation's principles.

3. *Manage Uncertainty*: Researchers have developed requirements negotiation models, however little attention is given to the socio-psychological aspects of the process [11] even though, achieving a high quality communication reduces uncertainty [12]. High quality communication starts establishing "harmony" among stakeholders, where harmony is defined as "compatibility" or "affinity". People who get harmony among each other are comfortable working together, they are open mind and they are honest in their communications. Of course, this is a desirable situation when we are eliciting requirements. The challenge is to keep a harmonic state between stakeholders during enough time. Users' doubts concerning the elicitation process itself and/or its consequences are part of the information we should obtain. This knowledge would be useful to get commitment and trust on people collaborating to define requirements.

### Conclusions and Future Work

We have presented a strategy for eliciting requirements based on detecting and defining the elements that compose traditional enterprise communication networks. These networks are present in every sort of organisation and constitute a basement on which requirements analysis decisions can be built. As a consequence, uncertainty is reduced and selecting experts is more appropriate. However, in spite of the strategy has been applied in several cases, its impact on the elicitation process has only been informally recorded. In the next stage of our work, we are extending the strategy to include some features to be measured as well as a measurement procedure, in order to provide an objective evaluation. Several on-going experiments based on the GQM (Goal/Quality/Metric) approach [13] will quantify the impact of our strategy on traditional software development projects.

### References

- [1] Hofmann H. and Lehner F., 2001. Requirements Engineering as a Success Factor in Software Projects. In *IEEE Software*, Vol. 18, No. 4, 58-66. IEEE Press.
- [2] Robinson W., and Pawlowski S., 1999. Managing Requirements Inconsistency with Development Goal Monitors. In *IEEE Transactions on Software Engineering*, Vol. 25, No. 6, 816-835. IEEE Press.
- [3] Al-Rawas A., and Easterbrooks S., 1996. Communication Problems in Requirements Engineering. A Field Study. In *Proceedings of the First Westminster Conference on Professional Awareness in Software Engineering*, London.
- [4] Ciamberlani L. And Steimberg L., 1999. *Comunicación para la Transparencia*, Granica Ed.
- [5] Skinner, 1974. *Sobre el Conductismo*. Ediciones ORBIS, S.A.
- [6] Luzuriaga J., and Martínez R., 2001. Recopilación de Información en un Proceso de Auditoria de Normas de Calidad, *Reporte Técnico 010*, Universidad Nacional del Comahue, Argentina.
- [7] Luzuriaga J., Martínez R., and Cechich A., Managing Enterprise Communication Networks to Improve the Requirements Elicitation Process, *Proceedings of the International Conference on Enterprise Information Systems*, ICEIS 2002, April 3-6 Ciudad Real, Spain.
- [8] Beyer and Holtzblatt, 1998. Contextual Design: Defining Customer-Centred Systems, Morgan Kaufmann.
- [9] Rumbaugh J., Jacobson I., and Booch G., 1999. *The Unified Modeling Language Reference Manual*, Addison-Wesley.

- [10] Cechich A., Luzuriaga J., Martínez R., Classifying Requirements Perceptions to Guide Exploratory Processes in Requirements Elicitation, *Proceedings of IDEAS 2002 (Workshop Iberoamericano en Ingeniería de Requisitos y Ambientes de Software)*, April 23-26 La Habana, Cuba.
- [11] Herlea D. et al, 2000. Using Different Communication Media in Requirements Negotiation, *IEEE Software Magazine*, Vol. 17(3), pp. 28-35
- [12] Harrington H., 1993. *Mejoramiento de los Procesos de la Empresa*, Mc Graw-Hill.
- [13] van Solingen R., *The Goal/Question/Metric Method: a practical guide to quality improvement of software development*, Mc-Graw Hill, 1999.

## *Escalabilidad y Balance de carga en Sistemas Paralelos*

Marcelo Naiouf<sup>1</sup>, Armando De Giusti<sup>2</sup>

*LIDI - Laboratorio de Investigación y Desarrollo en Informática<sup>3</sup>.  
Facultad de Informática. Universidad Nacional de La Plata.*

### **Palabras Clave**

Procesamiento Paralelo. Programas Paralelos. Análisis de performance. Balance de carga. Escalabilidad. Métricas del paralelismo.

### **Resumen**

Investigar el balance de carga y la escalabilidad en procesamiento paralelo.

Los temas fundamentales se refieren a los algoritmos de balance de carga propiamente dichos y a su aplicación en la resolución paralela de problemas, poniendo especial énfasis en los atributos de performance de estas combinaciones. En particular, interesan medidas tales como el speedup, la eficiencia, el costo y la escalabilidad de las soluciones.

Se trabaja experimentalmente con diferentes modelos de arquitectura multiprocesador, disponibles o accesibles en el LIDI.

Interesa la aplicación de las investigaciones al tratamiento de imágenes, de datos numéricos en cómputo científico y de bases de datos distribuidas.

### **Introducción**

El procesamiento computacional evolucionó hacia el paralelismo prácticamente desde el inicio mismo de las máquinas digitales. Actualmente no puede negarse la importancia y el creciente interés en el procesamiento paralelo dentro del espectro de la Ciencia de la Computación.

Esto se debe a un gran número de razones. Entre ellas se puede hacer referencia al crecimiento de la potencia de cálculo dada por la evolución de la tecnología, la transformación y creación de algoritmos que explotan la concurrencia implícita de los problemas para obtener mejores tiempos de respuesta, la necesidad de tratar sistemas de tiempo real distribuidos con requerimientos críticos en tiempo, el límite físico alcanzado por las máquinas secuenciales que convierte a la solución paralela en la única factible, las posibilidades que ofrece el paradigma paralelo en términos de investigación de técnicas para el análisis, diseño y evaluación de algoritmos.

En términos generales, las computadoras paralelas hacen posible resolver problemas de creciente complejidad y obtener resultados con mayor velocidad. Pero, en algunos casos el costo del paralelismo puede ser muy alto en términos de esfuerzo de programación. En teoría, el paralelismo es simple: aplicar múltiples CPUs a un único problema. Para el científico computacional resuelve algunas de las restricciones impuestas por las computadoras de una sola CPU. Además de ofrecer soluciones más rápidas, las aplicaciones que fueron paralelizadas (convertidas en programas paralelos) pueden resolver problemas más grandes y complejos cuyos datos de entrada o resultados intermedios exceden la capacidad de memoria de una CPU, las simulaciones pueden ser corridas con mayor resolución, los fenómenos físicos pueden ser modelados de manera más realista.

---

<sup>1</sup> Profesor Titular. [mnaiouf@lidi.info.unlp.edu.ar](mailto:mnaiouf@lidi.info.unlp.edu.ar)

<sup>2</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva. [degiusti@lidi.info.unlp.edu.ar](mailto:degiusti@lidi.info.unlp.edu.ar)

<sup>3</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.  
TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

En la práctica, el paralelismo tiene un alto precio. La programación paralela involucra una curva creciente de aprendizaje y es de esfuerzo intensivo: el programador debe pensar sobre la aplicación de maneras novedosas y puede terminar debiendo reescribir todo el código serial. Las técnicas para “debugging” y “tuning” de performance de programas seriales no se extienden fácilmente al mundo paralelo.

Un aspecto de fundamental importancia es referirse a un algoritmo paralelo mencionando el modelo de computación para el que se lo diseñó. A diferencia del cómputo secuencial, en el procesamiento paralelo no hay un modelo teórico unificador, ya que diferentes modelos enfatizan determinados aspectos sin obtener una generalización. Por otro lado, no es probable una máquina paralela universal, pues para cada aplicación existe una máquina óptima, y distintas alternativas de implementación sobre sistemas de cómputo paralelo con hardware homogéneo o heterogéneo y con acoplamiento fuerte o débil de sus componentes. Por esto, lo correcto es hacer referencia a *sistemas paralelos* como la combinación de algoritmo y arquitectura.

*Para mejorar la utilización de los procesadores y el tiempo de respuesta, las computaciones paralelas requieren que los procesos sean distribuidos en procesadores de manera que la carga computacional tienda a ser equitativa (“balanceada”) en el tiempo. Este es uno de los temas fundamentales dentro del cómputo paralelo.* La performance obtenida en un sistema paralelo está dada por una compleja relación en la que intervienen una gran cantidad de factores: el tamaño del problema, la arquitectura de soporte, la distribución de procesos en procesadores, la existencia o no de un algoritmo de balanceo de carga, etc.

Existe una variedad de técnicas de balanceo de carga. La evolución ha marcado la necesidad de que las mismas sean diseñadas para ser escalables, portables, y relativamente fáciles de usar. En general interesa que sean de rápida convergencia hacia un estado balanceado, manteniendo la localidad de comunicación, minimizando las transferencias de carga, y significando un bajo overhead en la aplicación total.

En el *balance de carga estático*, esta distribución de carga de trabajo se hace en la fase de inicialización. Hubo numerosos estudios usando teoría de grafos, programación entera, teoría de colas, etc, así como enfoques heurísticos. Al realizarse sólo una vez, es útil únicamente en problemas con carga más bien estática durante la ejecución, y definitivamente no es adecuada para computaciones con una carga cambiante dinámicamente.

Para mantener la carga balanceada durante la ejecución de problemas de carga variable es necesario realizar balanceo de carga en varias etapas durante el tiempo de corrida (*balance de carga dinámico*).

La ejecución del procedimiento de balanceo dinámico requiere algún medio de mantener una visión global del sistema y algún mecanismo de negociación para migraciones de procesos entre procesadores cercanos. Cada estrategia tiene que resolver los temas de cuándo invocar un balanceo, quién toma las decisiones de balanceo y de acuerdo a qué información, y cómo manejar las migraciones de procesos entre procesadores. Combinando distintas respuestas se tiene un gran espacio de diseños posibles de métodos de BCD para multicomputadores.

Existe un gran número de métricas para evaluar sistemas paralelos. Entre ellas se encuentran el tiempo efectivo, speed-up, eficiencia, producto procesador-tiempo, overhead paralelo, grado de concurrencia, escalabilidad, isoeficiencia, isospeed, etc. En el caso del balanceo de carga, existen medidas tales como la performance normalizada y el tiempo de estabilización.

En esta investigación es de interés la evaluación de performance de distintas clases de aplicaciones sobre diferentes arquitecturas reales, de modo de adecuar los resultados teóricos ideales de las métricas a la realidad. Esto se basa en que muchos sistemas paralelos no alcanzan su capacidad teórica, y las causas de esta degradación son muchas y no siempre fáciles de determinar. El análisis sobre plataformas disponibles permite estudiar el impacto que tienen algunos de estos factores sobre las implementaciones, y adecuar las métricas clásicas a las mismas.

En particular se estudia la influencia de las estrategias de distribución de procesos y datos, y la carga (estática o dinámica) asignada a cada procesador sobre el speedup, la eficiencia y la escalabilidad. Más allá de las características teóricas/algorítmicas de las aplicaciones, el rendimiento real y por lo tanto el tiempo necesario para resolver los problemas siempre o en la mayoría de los casos dependerá de la arquitectura de procesamiento elegida para la implementación.

### **Temas de Investigación y desarrollo**

- Paralelización de aplicaciones. Esto incluye la transformación de algoritmos secuenciales en paralelos y su adecuación a diferentes modelos de arquitectura.
- Estudio de las diferentes métricas de evaluación de performance de sistemas paralelos. Abarca el análisis de la complejidad de los algoritmos, el rendimiento de las soluciones paralelas y la caracterización de los modelos de performance asociados.
- Balance de carga en sistemas paralelos. Involucra el balanceo de cargas de procesamiento y comunicaciones, la distribución de procesos en procesadores, la migración de datos y procesos, y la aplicación de diferentes técnicas a la resolución de problemas reales.

### **Equipamiento de experimentación**

Los modelos de multiprocesadores propuestos y disponibles para el trabajo experimental son: arquitectura multiprocesador homogénea fuertemente acopladas tipo cubo de transputer, arquitectura multiprocesador distribuida (homogénea y heterogénea) débilmente acoplada tipo NOW, y arquitectura multiprocesador de memoria compartida distribuida tipo SGI Origin 2000 (Clementina).

## Bibliografía Básica

- Akl S, "Parallel Computation. Models and Methods", Prentice-Hall, Inc., 1997.
- Andrews G. "Foundations on Multithread and Distributed Programming" Addison Wesley. 1999.
- Bell, David; Grimson, Jane, "Distributed Database Systems", Addison Wesley. 1992
- Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- Bubak, Funika, Moscinski, "Performance Analysis of Parallel Applications under Message Passing Environments", [www.icsr.agh.edu.pl/publications/html/perf\\_full/](http://www.icsr.agh.edu.pl/publications/html/perf_full/), 1997.
- Coffin M. "Parallel programming- A new approach", Prentice Hall, Englewood Cliffs, 1992.
- Chandi K. M., Misra J., "Parallel Program Design. A Foundation", Addison Wesley, 1988.
- Espinosa A., Parcerisa F., Margalef T., Luque E. "Relating the Execution Behaviour with the Structure of the Application" (Euro PVM/MPI) Lecture Notes in Computer Science, vol. 1697, 91-98. Springer, Alemania, 1999.
- Espinosa A., Margalef T., Luque E. "Automatic Performance Analysis of Master/Worker PVM Applications with Kpi" (Euro PVM/MPI) Lectures Notes in Computer Science, vol. 1908, 47-55. Springer . Alemania, 2000.
- Espinosa A., Margalef T., Luque E. "Integrating Automatic Techniques in a Performance Analysis Session" (Euro-Par 2000) Lectures Notes in Computer Science, vol. 1900, 173-177. Springer, Alemania, 2000.
- Gonzalez and Woods, "Digital Image Processing", Addison-Wesley, 1992
- Gupta A., Kumar V., "Performance properties of large scale parallel systems", Journal of Parallel and Distributed Computing, November 1993.
- Hwang K., "Advanced Computer Architecture: Parallelism, Scalability, Programability", McGraw, 1993.
- Heermann, D. Burkitt A., "Parallel Algorithms in Computational Science", Springer-Verlag, 1991.
- IEEE Transactions on Parallel and Distributed Processing (colección de revistas 1990-2001)
- Kosko B., "Neural Networks and Fuzzy Systems", Prentice Hall, 1992
- Kumar V., Grama A., Gupta A., Karypis G., "Introduction to Parallel Computing. Design and Analysis of Algorithms", Benjamin/Cummings, 1994.
- Lawson H., "Parallel processing in industrial real time applications", Prentice Hall 1992.
- Leighton F. T., "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes", Morgan Kaufmann Publishers, 1992.
- Luque E, Ripoll A, Cortès A, Margalef T, "A Distributed Diffusion Method for Dynamic Load Balancing on Parallel Computers", Proceedings of the EUROMICRO Workshop on Parallel and Distributed Processing, IEEE Computer Society, Jan. 1995.
- Miller R., Stout Q. F., "Algorithmic Techniques for Networks of Processors", CRC Handbook of Algorithms and Theory of Computation, M. J. Atallah, ed, 1998.
- Morse F., "Practical Parallel Computing", AP Professional, 1994.
- Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.
- Printista Marcela, "Modelos de Predicción en Computación Paralela", Tesis de Magister en Ciencias de la Computación, UN Sur 2001.
- Sima D, Fountain T, Kacsuk P, "Advanced Computer Architectures. A Design Space Approach", Addison Wesley Longman Limited, 1997.
- Tinetti F., De Giusti A., "Procesamiento Paralelo. Conceptos de Arquitectura y Algoritmos", Editorial Exacta, 1998.
- Watts J., Taylor S., "A Practical Approach to Dynamic Load Balancing", IEEE Transactions on Parallel and Distributed Systems, Vol. 9, No. 3, March 1998, pp. 235-248
- Xu C., Lau F., "Iterative Dynamic Load Balancing in Multicomputers"
- Zomaya A., "Parallel Computing. Paradigms and Applications", Int. Thomson Computer Press, 1996.

## Distributed and Parallel Processing for the Embgrid Project

Sanz, Cecilia<sup>2</sup>, Tinetti Fernando<sup>3</sup>, Russo Claudia<sup>4</sup>, Denham Monica<sup>5</sup>, De Giusti, Armando<sup>1</sup>, Grau Oscar<sup>6</sup>

{degiusti, csanz, fernando, crusso, mdenham}@lidi.info.unlp.edu.ar  
grau@biol.unlp.edu.ar

*LIDI. Laboratorio de Investigación y Desarrollo en Informática<sup>7</sup>.  
Facultad de Informática. UNLP.  
50 y 115. 1er Piso. La Plata*

### Summary

The goal of EMBgrid is to overcome relevant infrastructure needs in Bioinformatics development and services in order to address future needs for the provision of useful Bioinformatics solutions to the Biosciences community. The proposed approach will make use of an extended Grid architecture, built on top of existing EU-DATAGRID services and deployed over a wide number of European nodes, to enable delivery of advanced tools and solutions to the growing demands of Biosciences. We contemplate creation of a pan-european Grid resource, related to other EU Grid initiatives and devoted mainly to Bioinformatics. This initiative will raise Europe's competitiveness in this and related fields. Successful deployment and dissemination will be facilitated by EMBnet long track in delivering tools, services and training to the European scientific community.

There is a cooperative argentinian group involved in this project, and has the responsibility of analyze the specification, transformation, optimization and verification of concurrent algorithms executable in distributed/parallel systems; they also have to study the optimization of solutions types in function of multiprocessor architecture models and complex and efficiency metrics related to the parallel processing.

In Biocomputing, the need of distributing advanced tools and solutions has arisen for the growing data generation which takes place in a hyper-exponential manner, and for the solution of complex problems.

The specific goal proposed for this project is to analyze some Emboss programs, which are used in Biocomputing area (in particular, those related with alignment, and pattern matching), in order to transform them into parallel and distributed algorithms.

---

<sup>1</sup> Investigador Principal CONICET. Profesor Titular Ded. Exclusiva.

<sup>2</sup> Profesor Dedicación Exclusiva - Facultad de Informática. UNLP.

<sup>3</sup> Profesor Dedicación Semi Exclusiva - Facultad de Informática. UNLP.

<sup>4</sup> Profesor Dedicación Exclusiva - Facultad de Informática. UNLP.

<sup>5</sup> Ayudante Alumno - Becaria LIDI - Facultad de Informática. UNLP.

<sup>6</sup> Profesor Dedicación Exclusiva - Director IBBM - Facultad de Ciencias Exactas. UNLP.

<sup>7</sup> LIDI - Facultad de Informática. UNLP - Calle 50 y 115 1er Piso, (1900) La Plata, Argentina.

TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

## Research Topics

### Processing Resources Use Optimization

Obtaining the maximum efficiency of a multiprocessor architecture represents an essential objective in parallel processing. The research in techniques that lead to upgrade the performance of a given architecture, or to adjust dynamically the processing resource assignation, is the axis of this line of work. As examples, we can quote specific techniques of:

- Charge Balance: processing and/or communications, depending on the application and the parallelization. All deviation as regards the charge equilibrium leads to the loss of the resource use and, therefore, to the loss of the general output.
- Process and processors distribution: the processes distribution in the architecture is specifically related to the processing charge balance and, therefore, it must be analyzed and optimized [1].
- Migration of data and/or processes: by analyzing the cost-benefit relation of keeping the data static distribution and processing of all the application, or adapt it dynamically on the basis of the parallel execution.

### Parallelization of Applications

The parallelization of applications with large computing requirements are objects to be studied for their computational resolution with parallel processing [2][3]. The classical techniques of :

- a) Algorithm transformations,
- b) Adaptation of algorithms to architecture models, and
- c) Specification and verification of real time parallel systems,

are under process of research, looking for their improvement (at least, by areas of massive data processing problems), trying to turn them into applicable systematic methodologies with the minor quantity of potential adaptations to different problems within the area of the given problem.

### Parallelism Metrics

In general, they are dealt with from several points of view, such as [4]:

- a) Complexity analysis of sequential and parallel algorithms. Considering this analysis as a basis, a type of solution can be adopted rationally.
- b) Output analysis, with which the quality or the cost-benefit relation to the parallel solutions proposed for the solution of the specific problem can be assessed.
- c) Characterization of parallel applications and architectures in order to clearly identify both the applications and architectures, and to attain a supported proposal of the proposed solution type.

### Parallel Processing Architectures [5]

In data massive processing, the following topics are relevant:

- a) The cost-benefit relation and the output in function of the processing potential capacity of parallel computing architectures (such as parallel supercomputers and workstations nets or clusters)
- b) Dynamic reconfiguration of hardware and/or software in order to adapt to the application and/or to propose failure tolerance.

Parallel hardware design oriented to application classes.

## Experimental work [6][7]

- Analysis of Emboss sequential programs

It is necessary to evaluate which programs need to be improve in order to obtain a better performance for them in cases of massive data treatment. The algorithm specification and the parallel programming paradigm should be studied in order to optimize time response.

- Parallel system evaluation

On of the analysis to make is evaluate the costs of the parallel solutions. There are different alternatives for the parallel systems implementation with homogeneous or heterogeneous hardware and with weakly or strongly coupled components. These alternatives give various possibilities that can be traduced in costs (including communication costs), which must be evaluated taken into account the relation cost/performance. The costs are not only related with hardware but with the development of algorithms and/or the algorithms adjustment for the processing architecture.

The performance of a parallel system is given by a complex relation among different factors (size of the problem, support architecture, process distribution among processors, balance charge algorithm existence, etc.). There are numerous metrics to evaluate parallel systems, such as, speed up, efficient, etc.

- Developed parallel solutions

Developed parallel/solutions for Emboss programs related with alignment and pattern matching on a cluster of homogeneous PCs, studying absolute performance and speed up.

Study of the best distribution (analyzing the local replication grade) for large data bases related with this project, in order to optimize their accessibility. To achieve this goal, a simulation environment for data base development (implemented by LIDI's group) will be used.

The experimental architecture models proposed (and available for the project) are four: homogeneous multiprocessor architectures highly coupled, transputer cube type; distributed multiprocessor architectures (homogeneous or heterogeneous) weakly coupled NOW type; multiprocessors dedicated to DSPs-based image treatment; and multiprocessor architectures of shared memory SGI Origin 2000 type (Clementina).

## Conclusions and Future work

An important research area is the specification, transformation, optimization and evaluation of distributed and parallel algorithms. The optimization of solutions in function of different architecture models and the study of complexity and efficiency metrics are of special interest. These encompass the development of parallel processes, the transformation of sequential algorithms in parallel (exploiting the implicit or explicit concurrency in the problem to be solved) and the performance assessment metrics on different supporting platforms (both of hardware and software).

There exist several areas in which parallel and distributed programming is required - as it can be observed in the project "Development of Grid Environment for Biological Applications", in which is involved this research line.

The group has experience in the area of parallel and distributed processing applied to complex problems where large volumes of data are involved. Actually, emboss programs are being analyzed and studied, in order to obtain parallel solutions.

## Bibliography

- [1] Bell, David; Grimson, Jane, "Distributed Database Systems",. Addison Wesley. 1992
- [2] Brinch Hansen, P., "Studies in computational science: Parallel Programming Paradigms", Prentice-Hall, Inc., 1995.
- [3] Gupta A., Kumar V., "Performance properties of large scale parallel systems", Journal of Parallel and Distributed Computing, November 1993.
- [4] Hwang K., "Advanced Computer Architecture: Paralelism, Scalability, Programability", McGraw, 1993.
- [5] IEEE Transactions on Parallel and Distributed Processing (colección de revistas 1990-2002)
- [6] Zomaya A., "Parallel Computing. Paradigms and Applications", Int. Thomson Computer Press, 1996.
- [7] <http://www.embnet.org/>

## **Evaluación del Rendimiento y Tiempo de Transferencia de la Información en Redes de Datos Mediante Modelos Analíticos y Técnicas de Simulación.**

Oscar Roque Segre<sup>1</sup> Rubén Jorge Fusario<sup>2</sup> Cristóbal Santa María<sup>3</sup>  
 Facultad de Informática, Ciencias de la Comunicación y Técnicas Especiales  
 Universidad de Morón  
 TE: 5627-2000- Int. 273 Fax 5627-4598

### **Resumen**

Este primer trabajo ha consistido en la evaluación de posibles modelos analíticos y de técnicas de simulación a los efectos de ser aplicados a las Redes de Datos con el objetivo de evaluar su comportamiento y su rendimiento.

En este trabajo han sido simulados, utilizando distintas leyes estadísticas, los siguientes procesos: generación de tráfico, tamaño de la trama, tiempo de demora, número de WS conectadas a la red y su carga variable y la generación de congestión, entre otros procesos los que pueden ser observados y analizados en el desarrollo del trabajo.

Simulado el funcionamiento de una Red de Datos se puede también simular sobre esta un protocolo de comunicaciones seleccionando, en este trabajo, el IEEE 802.3 el que fue elegido por contener aspectos como el acceso aleatorio y contencioso al MFE.

Para ello se ha parametrizado el modelo a los efectos de posibilitar su implementación como así también realizar evaluaciones sobre la variación (sensibilidad) de las variables relacionadas específicamente con el tráfico de la red.

El trabajo plantea la metodología de la simulación genérica de las Redes de Datos y coteja los resultados teóricos con los resultados obtenidos sobre redes reales en funcionamiento –mediciones efectuadas con el instrumental adecuado- a los efectos de comprobar la validez de la metodología adoptada para la simulación tanto de la Red como la del protocolo de comunicaciones que fue seleccionado.

### **1.- HIPÓTESIS DEL MODELO**

El método a utilizar en este análisis, es el denominado algoritmo exponencial binario el que, pese a su complejidad, aporta las mejores soluciones para la simulación del funcionamiento de una red.

Otros procedimientos de análisis ( como el desarrollado por Metcalf y Boggs [27] en 1986 ) suponen que se tiene una probabilidad constante para la transmisión de una trama en una ranura habilitada a ese propósito. En esta hipótesis adoptada, se supone que las estaciones de trabajo (WS) conectadas a la red se encuentran: transmitiendo, en estado de contienda o en reposo.

La probabilidad P para que otra estación intente adquirir el MFE durante ese mismo tiempo está dado

$$P = kp(1 - p)^{k-1}$$

Siguiendo el mismo proceso deductivo, podríamos afirmar que, la inversa del tiempo total que insume transmitir una trama Tt, es el número de tramas que se transmitirán en la unidad de tiempo al podemos designar como:

$$Vt \text{ (tramas/ seg.)} = 1/Tt = 1/( Tm + Tp + R2 \tau )$$

$$Vt \text{ ( tramas/seg )} = 1/( k Tt ) = \lambda$$

Esta forma de tratamiento es incompleta, ya que resulta evidente que Tt =f(k).

### **2.-EL PLANTEO TEÓRICO DEL MODELO**

El parámetro  $\lambda$  representa la esperanza matemática de una distribución de Poisson cuya variable aleatoria modela el número de tramas emitidas en la unidad de tiempo.

Esta distribución responde a siguiente ecuación:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad \text{donde } x \text{ es entero.}$$

<sup>1</sup> osegre@unimoron.edu.ar

<sup>2</sup> jrfusario@ara.mil.ar

<sup>3</sup> smaria@sion.com.ar

La elección de una variable poissoniana para modelar este fenómeno se apoya primero en la evidencia experimental (ajuste de por medio) y por otra parte en la bibliografía disponible de carácter técnico (Ref. Schwartz [10], Tanenbaum [3] y otros).

### **3.- LOS PARÁMETROS UTILIZADOS**

Además de  $\lambda$ , promedio de tramas emitidas en la unidad de tiempo y  $k$  número de estaciones conectadas a la red, utiliza dos parámetros adicionales. Estos son:

Tr: tiempo de ranura y N: número de tramas totales transmitidas en un intervalo T.

### **4.- PLANTEO TEÓRICO DE LA SIMULACIÓN**

La variable de decisión de este modelo de simulación es el rendimiento entendiendo por tal a la proporción existente entre la suma del total de los tiempos de transmisión calculados cada uno en función de la longitud de la trama y la suma de los tiempos reales de transmisión por efecto de las colisiones y sus posteriores resoluciones según lo explicado en el punto anterior. Entonces:.

- Rendimiento =  $T_o/T_t$

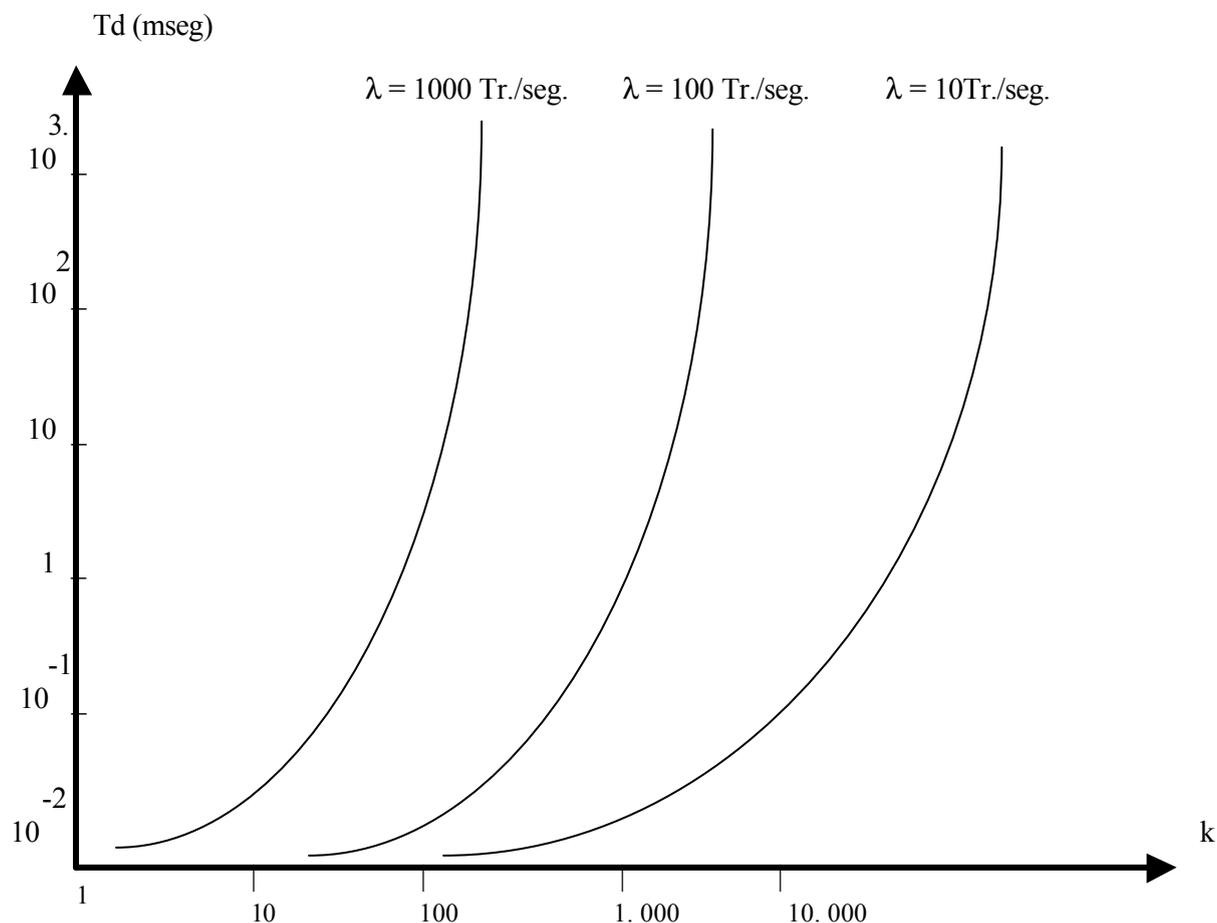
La técnica de simulación elegida para el cálculo del rendimiento es la de Montecarlo Aproximada.

## **CONCLUSIONES**

Analizados y comparados los valores obtenidos a partir de la simulación de la red con topología bus y la simulación del protocolo de comunicaciones IEEE 802.3 CSMA/CD, con los obtenidos de una red real - la que se está operativa y funcionando - mediante el instrumental adecuado descrito en el informe final, se obtienen las siguientes conclusiones:

1. La simulación de la red y del protocolo de comunicaciones fue programada en C++, como se puede ver en el programa que se adjunta dentro del informe. No se originaron inconvenientes como no sea el propio de la elaboración del programa de simulación. La bibliografía que se tomó como base para el desarrollo del proyecto, si bien es cierto que relata que las redes de datos son posibles de ser simuladas, no se han encontrado en ella antecedentes sobre los procedimientos utilizados sobre todo en cuales han sido las leyes estadísticas utilizadas para la simulación del tráfico, cantidad de tramas emitidas por segundo, tamaño de las tramas, etc. Esto hizo necesario que dichos valores fueran obtenidos previamente por experimentación y ajuste a partir de los datos medidos sobre la red real observada y de la cual se obtuvieron los valores que se adjuntan.
2. El programa calcula el parámetro  $\lambda$  (tramas/seg.) que se pueden transmitir, a partir de una configuración de la red dada por la cantidad  $k$  de WS conectadas y que emiten tráfico. A los efectos de que el tiempo de procesamiento no sea demasiado grande se tomó el número  $k$  de WS reducido. Dicho parámetro  $\lambda$  es la esperanza matemática de la variable aleatoria - cantidad de tramas emitidas por segundo - que se supone en este caso distribuida según ley de Poisson.
3. La determinación de este parámetro  $\lambda$  es la que mayor dificultades ofreció y abre una posibilidad de continuación en otro proyecto de investigación o tesis de maestría, para que pueda ser ajustado con mayor precisión ya que la cantidad de tramas por segundo que una red de datos - trabajando con cualquier protocolo - pueda emitir es un valor estadístico que depende de los elementos constitutivos de la red siendo los más importantes: cantidad de WS conectadas, WS que pretenden ocupar el MFE que es variable en el tiempo, tamaño de la trama, tiempo de ranura, tiempo de colisión, etc.
4. Con la precisión con la que se determine la ley de probabilidad y su valor medio  $\lambda$ , será el grado de aproximación con que la red simulada se ajuste a una red real. Este es el objetivo primario de la investigación ya que sobre la simulación de la red se puede simular cualquier protocolo estandarizado o que pueda ser desarrollado a partir de condiciones previamente establecidas de congestión, enrutamiento, direccionamiento, tamaño de trama, etc.
5. En la simulación se ha preferido calcular el parámetro  $\lambda$  a través del mismo programa lo cual supone fijar que en la red existe un determinado tráfico ó número de tramas emitidas por las WS por segundo, y no precisar cuales han sido las WS que emiten esas tramas o generan ese tráfico. Los valores obtenidos a partir de la simulación justifican este procedimiento aunque, como fue establecido en el punto 4 de estas conclusiones, habría que ajustar con mayor precisión este valor de  $\lambda$  considerando todos los parámetros de los cuales depende.

6. El uso en el protocolo IEEE 802.3 del algoritmo exponencial binario está justificado pues se pudo probar que se ajusta al número probable de las colisiones que se producen en la transmisión de la información, cuanto mayor es el tráfico generado por las WS y cuanto mayor es el número de estas WS conectadas a la red.[ ver gráfico de  $T_d = f(k, \lambda)$  siguiente]



Tamaño de la trama : Variable dentro de límites del protocolo

$k \Rightarrow$  Número de WS conectadas a la red

$T_d \Rightarrow$  Tiempo de demora en milisegundos

$\lambda \Rightarrow$  Tráfico en la red en tramas/ segundo

## **BIBLIOGRAFÍA**

- [1].-J.F. Kurose and H.T.Mouftah "Computer Aided Modeling, Analysis and Design of Communications Networks". IEEE Journal en selected Areas in Communications , Vol.6, N°1, January 1988, pp 130-145.
- [2].-R Jain "The Art of computer System Performance Analysis: Techniques Experimental Design, Measurements, Simulation and Modeling" . Wiley, NY,1991.
- [3].-A Tenenbaum "Computers Networks" 3ra. Edición, Prentice Hall, NJ, 1996.
- [4].-A Keshenbaum "Telecommunications Networks Design Algorithms" G Hill NY 1993.
- [5].-J Tindle, Brewis and Ryan "Advanced Simulation and Optimisation of the Telecommunications Networks" BT Technology Journal, Vol. 14, N°2 , April1996.
- [6].- T N Saadawi, Ammar and Hakeen "Fundamentals of Telecommunications Networks" Wiley ,NY, 1997.
- [7].-T G Robertazzi "Computing Networks and Systems: Queueing Theory and Performance Evaluation" Springer, Vielag, NY, 1994.

- [8].-Bikel and Doksum "Mathematical Statistics" Holden Day, SF, 1997.-
- [9].-Biedway and Salameh "From LAN's to GAN'S" Telecommunications, Vol. 26, N° 7 July 1992, pp 23-27.
- [10].-M Schwartz "Redes de Telecomunicaciones: Protocolos, modelos y análisis" Addison, MX, 1987.
- [11].-Papoulis "Probability, Random Variables and Stochastic Process" Mc Graw Hill,
- [12].-Azarang y García "Simulación y Análisis de Modelos Estocásticos" Mc Grow Hill,
- [13].-Gordon "Simulación de Sistemas" Diana
- [14].-F Halsall "Comunicación de Datos, Redes de Computadoras y Sistemas Abiertos" 4ª Edición, Addison, Iberoamericana, 1998.
- [15].-U Black "Data Link Protocols" Prentice Hall, Englewood Cliffs, 1993.
- [16].-A Huurdeman "Guide to Telecommunications Systems" Artech House, Norw. 1997.
- [17].-J Martin, J Leben and K Chapman "Local Area Networks: Architectures and Implementations" 2ª Edición, Prentice Hall, Englewood Cliffs, 1994.
- [18].-J Hsu "Computers Networks: Architectures, Protocols and Communications Software" Artech House, Norwood, 1997.
- [19].-I Garcés, D Franco and E Luque "Analytical Modelling of the Network Traffic Performance" Proceedings of 7º International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems 99(MASCOTS' 99) IEEE Computer Society, pp 190-196.
- [20].-J Meggers and M Wallbaum "Application Level Error Recovery Using Active Networks Nodes" Proceedings of 5º IEEE Symposium on Computers Communications 2000 ( ISCC 2000) pp 540-545.
- [21].- D Betsekas and R Gallager "Data Networks" Prentice Hall, Engl. Cliffs, NY, 1992.
- [22].-W Lelang, M Taqqu, W Willinger and D Wilson "On the Self-Similar Nature of Ethernet Traffic" IEEE/ACM Transactions on Networking, vol2, N° 1, pp 1-14, Feb. 1994.
- [23].-R Metcalf and D Boggs "Ethernet: Distributed Packet Switching for Local Computers" Communications of the ACM, Vol.19, July 1986, pp 395-404.
- [24].-V Jacobson "Congestion avoidance and control" Proceedings ACM SIGCOMM'88, Stanford, aug. 1988, pp. 314-329.
- [25].-R Ramakrishnam and R Jain "A Binary Feedback Scheme for Congestion Avoidance in Computers Networks with a Connectionless Network Layer" Proceedings ACM SIGCOMM'88, Stanford, Aug. 1988, pp.303-313.
- [26].-J Pearson "Basic Communication Theory" Prentice Hall, Englewood Cliffs,NJ,1992.
- [27].-C Bohm, M Hidell, P Landgren, L Ramfelt and P Södin "Fast Circuit Switching for the Next Generation of High Performance Networks" IEEE Journal on Selected Areas of Communications, Vol. 14, N° 2, February 1996.
- [28].-S Ghosh & T Lee "Modeling and Asynchronous Distributed Simulation" IEEE Press, 2000.

# Cómputo Científico: Algoritmos Paralelos para Aprovechar Redes Locales Instaladas

Fernando G. Tinetti, Andrés Barbieri, Mónica Denham

Centro de Técnicas Analógico-Digitales (CeTAD)<sup>1</sup>  
Laboratorio de Investigación y Desarrollo en Informática (LIDI)<sup>2</sup>  
Laboratorio de Química Teórica (LQT)<sup>3</sup>  
{fernando, barbieri, mdenham}@lidi.info.unlp.edu.ar

## 1. Introducción

De acuerdo con la evolución en el tiempo, la potencia de cálculo disponible de las computadoras de escritorio crece a medida que el costo se reduce. Puesto de otra manera, la relación costo/potencia de cálculo o directamente costo/beneficio es más ventajosa para los usuarios a medida que transcurre el tiempo. De hecho, hasta las computadoras paralelas de mayor potencia de cálculo absoluta se están construyendo con hardware de procesamiento de uso masivo, por ejemplo, en las PCs y en las estaciones de trabajo que cada empresa dispone en el mercado de bajo costo [4]. Las instalaciones de redes locales para cómputo paralelo del tipo Beowulf [9] ya no son consideradas experimentales sino que se están transformando en las plataformas de cómputo paralelo de producción de varias instituciones [1] [5] [6] [7].

A medida que crece el la cantidad de instalaciones de redes locales (computadoras que se interconectan en una red local), se incrementa aún más la posibilidad de utilizarlas en paralelo para cómputo científico, dado que se pueden configurar y utilizar como una máquina paralela. Desde este punto de vista, las redes locales o redes de estaciones de trabajo (NOW: Networks of Workstations) instaladas tienen la capacidad de ser computadoras paralelas con “costo cero” al menos en términos de hardware. A pesar de la existencia de bibliotecas de software de uso libre para desarrollar y ejecutar programas paralelos en redes de computadoras [12] [11] [10], el costo de hardware no es todo el costo que se tiene que considerar, dado que:

- Estas herramientas de desarrollo y ejecución de programas paralelos se tienen que instalar y mantener, con sus costos asociados.
- La paralelización de aplicaciones sobre estas plataformas de cómputo paralelo no ha sido ni es algo inmediato.
- El aprovechamiento de la capacidad de cómputo y de comunicaciones de las redes locales para cómputo paralelo no necesariamente es óptimo (al menos el que se tiene a partir de las bibliotecas mencionadas).

En este contexto, es necesario investigar y explicitar cuáles son los problemas y las soluciones posibles que se pueden encontrar [8], específicamente en cuanto a la paralelización de aplicaciones y a la utilización optimizada de los recursos disponibles.

Las aplicaciones de cómputo científico provienen de muchas áreas y tienen múltiples características. En particular, las aplicaciones de álgebra lineal constituyen una de las grandes áreas de problemas que tradicionalmente han sido resueltos aprovechando el rendimiento que proporcionan las arquitecturas de cómputo paralelo disponibles. Dentro de las aplicaciones del álgebra lineal se han identificado un conjunto de operaciones o directamente rutinas de cómputo que se han considerado básicas y de utilización extensiva en la mayoría de los problemas incluidos dentro de esta área. Tales rutinas se han denominado BLAS (Basic Linear Algebra Subroutines) y tanto para su clasificación como para la identificación de requerimientos de cómputo y de memoria de cada una de ellas se las

---

1 Facultad de Ingeniería, Universidad Nacional de La Plata

2 Facultad de Informática, Universidad Nacional de La Plata

3 Facultad de Ciencias Exactas, Universidad Nacional de La Plata

divide en tres niveles: nivel 1, nivel 2 y nivel 3 (Level 1 o L1 BLAS, Level 2 o L2 BLAS y Level 3 o L3 BLAS) [2]. Desde el punto de vista del rendimiento, las rutinas de nivel 3 (L3 BLAS) son las que se deben optimizar para obtener rendimiento cercano al óptimo de cada máquina y de hecho, muchas empresas de microprocesadores estándares proveen bibliotecas BLAS con marcado énfasis en la optimización y el consiguiente rendimiento de las rutinas incluidas en BLAS de nivel 3.

## 2. Paralelización de Aplicaciones

En el contexto de cómputo paralelo en redes de computadoras, el modelo de programación más ampliamente elegido ha sido el de pasaje de mensajes. Inicialmente, la distribución y bajo acoplamiento del hardware de cómputo orienta esta elección. Además, la disponibilidad de herramientas de uso libre en este tipo de arquitecturas paralelas tales como PVM y algunas implementaciones de MPI también están orientadas en este sentido.

Los algoritmos paralelos que se han diseñado para este modelo de programación son los propuestos en general para las computadoras paralelas estándares pertenecientes a la clase MIMD (Multiple Instruction-Multiple Data Stream) de memoria distribuida o multicomputadoras [3]. Aunque con similitudes en cuanto a características generales de hardware con las redes locales que se pueden utilizar para cómputo paralelo, las máquinas paralelas clásicas han impuesto al menos dos características subyacentes en el hardware que se asumen en la mayoría (sino todos) los algoritmos paralelos propuestos:

- Interconexión de los procesadores en formas relativamente complejas. Quizás las más clásicas son las mallas o toros bidimensionales, arreglos de árboles o hipercubos. Esto significa tener la posibilidad de múltiples conexiones punto a punto y múltiples caminos opcionales de la red de interconexión para la transferencia de datos entre dos procesadores. Tampoco se han descartado las redes dinámicas de interconexión de procesadores, que tienen una flexibilidad y rendimiento total de comunicaciones aún mayor.
- Elementos de procesamiento homogéneos. Esto implica que para los problemas clásicos que se pueden resolver bajo el modelo de ejecución SPMD (Single Program-Multiple Data) el balance de carga es trivial y directamente dado por la distribución de la misma cantidad de datos involucrados a todos los procesadores.

Ninguna de las dos características anteriores es posible de mantener en las redes locales de computadoras heterogéneas. Por lo tanto, es necesario desarrollar algoritmos que hagan uso eficiente de las características de estas *nuevas* arquitecturas paralelas. Por un lado, estos algoritmos deben estar preparados para las diferencias de capacidad de cómputo de las máquinas interconectadas por las redes locales y por el otro deben aprovechar al máximo el rendimiento y las características de las redes de interconexión Ethernet. Y estas son las dos bases sobre las que se deberían apoyar los algoritmos paralelos que se propongan para resolver los problemas de cómputo científico sobre las redes locales de computadoras.

## 3. Antecedentes y Trabajo Futuro

Toda el área de aplicaciones de álgebra lineal y específicamente las subrutinas de LAPACK y BLAS de nivel 1, 2 y 3 tienen características muy específicas y comunes en cuanto a procesamiento de datos. Por esta razón ya se ha comenzado a trabajar en esta área y actualmente se dispone de un algoritmo de multiplicación de matrices más una rutina de mensajes optimizada para el aprovechamiento óptimo de las redes locales instaladas. Si bien el desarrollo e implementación de rutinas de comunicación entre procesos de un programa paralelo no estaba en los planes originales se debió hacer para no perder rendimiento ni escalabilidad del algoritmo propuesto.

El algoritmo desarrollado para la multiplicación de matrices en paralelo propuesto y con el que se ha experimentado ha demostrado varias cosas:

Es posible tener rendimiento aceptable y escalable con cómputo paralelo en las redes de computadoras instaladas a pesar de su heterogeneidad, bajo rendimiento de la red de interconexión y procesamiento con acoplamiento relativamente alto del problema a resolver.

Es posible aprovechar de manera optimizada los recursos de cómputo y comunicaciones que disponen

las redes locales instaladas aunque su objetivo principal no sea el de cómputo paralelo.

Aunque confiables, las herramientas (básicamente bibliotecas) disponibles para desarrollo y ejecución de programas paralelos en redes locales no necesariamente ponen a disposición del usuario todos los recursos. Expresado de otra forma: es muy difícil obtener rendimiento aceptable.

La extensión inmediata con respecto al algoritmo paralelo de multiplicación de matrices es la aplicación de los principios de paralelización (balance de carga heterogéneo y aprovechamiento de las capacidades de las redes Ethernet) al resto de las aplicaciones de álgebra lineal. Actualmente se está investigando y experimentando con el problema específico de sistemas de ecuaciones lineales. Las extensiones no tan inmediatas tienen dos orientaciones:

- De hardware: la utilización de más de una red local. Es muy interesante/importante el impacto de las comunicaciones entre máquinas que no están en una misma red local, dado que el rendimiento se verá afectado por factores no necesariamente controlables por las aplicaciones (tráfico en la red).
- De áreas de aplicación: la utilización de los principios de paralelización para otros problemas numéricos no necesariamente pertenecientes al área de álgebra lineal. Un ejemplo lo constituyen los métodos de programación no lineal y procesamiento de señales.

## Referencias

[1] Becker D. J., T. Sterling, D. Savaresse, J. E. Dorband, U. A. Ranawak, C. W. Packer, "Beowulf: A Parallel Workstation for Scientific Computation", Proc. of the International Conference on Parallel Processing, vol. 1, pp. 11-14, Boca Raton, Florida, Aug. 1996.

[2] Dongarra J., D. Walker, "Libraries for Linear Algebra", in Sabot G. W. (Ed.), High Performance Computing: Problem Solving with Parallel and Vector Architectures, Addison-Wesley Publishing Company, Inc., pp. 93-134, 1995.

[3] Flynn M, "Very High Speed Computing Systems", Proc. IEEE, Vol. 54, 1966.

[4] Meuer H., E. Strohmaier, J. Dongarra, H. Simon, "TOP500 Supercomputer Sites", 18th Edition, Nov. 2001. Disponible en <http://www.top500.org>

[5] Radajewski J., D. Eadline, Beowulf HOWTO, Nov 1998, disponible en <http://www.linuxdoc.org/HOWTO/Beowulf-HOWTO.html>

[6] Ridge D., D. Becker, P. Merkey, T. Sterling, "Beowulf: Harnessing the Power of Parallelism in a Pile-of-PCs, Proceedings IEEE Aerospace, 1997, disponible en <http://www.beowulf.org/papers/papers.html>

[7] Sterling T., J. Salmon, D. Becker, D. Savaresse, How to build a Beowulf: a guide to the implementation and application of PC clusters, The MIT Press, Cambridge, Massachusetts, 1999.

[8] Tinetti F., A. Quijano, A. De Giusti, Heterogeneous Networks of Workstations and SPMD Scientific Computing, 1999 International Conference on Parallel Processing, The University of Aizu, Aizu-Wakamatsu, Fukushima, Japan, September 21 - 24, 1999.

[9] Beowulf Home Page <http://www.beowulf.org>

[10] LAM/MPI (Local Area Computing / Message Passing Interface) Home Page <http://www.mpi.nd.edu/lam>

[11] MPICH Home Page <http://www-unix.mcs.anl.gov/mpi/mpich/>

[12] PVM Home Page [http://www.emm.ornl.gov/pvm/pvm\\_home.html](http://www.emm.ornl.gov/pvm/pvm_home.html)

## PROCESAMIENTO PARALELO DISTRIBUIDO HETEROGÉNEO EN APLICACIONES ESTRUCTURALES Y NUMÉRICAS

**Gustavo E. Vazquez y Nélica B. Brignole**

Grupo de Investigación y Desarrollo en Computación Científica GIDeCC  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur, Avda. Alem 1253, 8000 Bahía Blanca, Argentina

Planta Piloto de Ingeniería Química PLAPIQUI (UNS-CONICET)  
Comp CRIBABB, km 7 Camino La Carrindanga, 8000 Bahía Blanca, Argentina  
Tel.: 0291 4861700 - Fax: 0291 4861600 - e-mail: [dybrigno@criba.edu.ar](mailto:dybrigno@criba.edu.ar)

### Resumen

El objetivo global de esta línea de investigación es diseñar e implementar nuevas estrategias de procesamiento paralelo en entornos de cómputo distribuido heterogéneo para facilitar la resolución de problemas tanto estructurales como numéricos cuya resolución mediante técnicas secuenciales resulta costosa en tiempos de ejecución debido a la magnitud o complejidad de las instancias que se desea resolver. Como resultado de estas investigaciones se ha logrado el desarrollo de técnicas robustas y eficientes aplicables a un amplio espectro de problemas de búsquedas en grafos y de optimización con función objetivo y restricciones no lineales.

En términos generales, es posible distinguir dos ramas de investigación para el desarrollo de algoritmos paralelos distribuidos: la paralelización de algoritmos secuenciales existentes y la creación de alternativas intrínsecamente paralelas. En el caso de problemas estructurales, se estudiaron los métodos secuenciales clásicos de búsqueda en grafos y se establecieron las limitaciones para su uso en redes de estaciones de trabajo. Sobre esta base se propuso un nuevo método de distribución semi-dinámica y se lo aplicó al algoritmo GS-FLCN para análisis de observabilidad. Por otra parte, en la línea de los algoritmos intrínsecamente paralelos se desarrolló un nuevo algoritmo de búsqueda totalmente distribuido con el objeto de aumentar la eficiencia de los recorridos para esta aplicación específica. En vez de la estrategia tipo Master-Worker de la propuesta anterior se diseñó otra mucho más eficiente que sigue un esquema descentralizado tipo Master-Supervisor-Worker. En cuanto a los problemas numéricos, se consideraron estrategias para aplicar el paralelismo a las secciones de cómputo intensivo de los algoritmos secuenciales existentes para optimización no lineal con restricciones conocidos como GRG y SQP. Asimismo se desarrolló una nueva técnica de descomposición de dominio con el objeto de ampliar el rango de aplicabilidad de un algoritmo intrínsecamente paralelo concebido originalmente para problemas sin restricciones de modo que se lo pudiera utilizar en forma eficiente para el tratamiento de los problemas de optimización no lineal con restricciones.

En cuanto a las verificaciones de desempeño, las implementaciones se testearon sobre redes de estaciones de trabajo homogéneas y heterogéneas pequeñas y se hicieron estudios de escalabilidad. Las métricas clásicas de speed-up debieron ser ajustadas con el objeto de tener en cuenta la heterogeneidad de los procesadores y así poder asegurar comparaciones justas. En tal sentido, todos los nuevos algoritmos propuestos lograron muy buen desempeño en cuanto al tiempo de ejecución en comparación con los algoritmos secuenciales correspondientes. Se analizaron casos de estudio académicos y problemas industriales reales de mediano y gran tamaño pertenecientes al área de ingeniería de procesos.

## Resultados Alcanzados

### 1. Procesamiento Paralelo Distribuido Aplicado al Desarrollo de Algoritmos para Reordenamiento Estructural de Matrices Ralas.

Los grafos poseen una estrecha relación con la teoría de matrices ralas y en particular con los métodos de reordenamiento y manipulación estructural de dichas matrices. De hecho, una de las formas más utilizadas para representar grafos es justamente a través de matrices. El reordenamiento estructural de matrices a patrones de ralidad específicos puede efectuarse empleando búsquedas arbóreas. La permutación de matrices ralas a formas especiales permite ahorrar tiempos de cómputo y espacio de memoria. En particular, nosotros hemos trabajado en el reordenamiento estructural a forma triangular en bloques (FTB), el cual ha sido usado con éxito en la resolución de sistemas lineales, problemas de mínimos cuadrados, particionamiento de matrices para procesamiento paralelo, entre otras aplicaciones.

El problema específico seleccionado para efectuar nuestras investigaciones consiste en el desarrollo de algoritmos para el reordenamiento estructural de las matrices de ocurrencia ralas asociadas a los sistemas de ecuaciones que modelan procesos industriales con vistas a lograr la FTB requerida para resolver problemas de diseño de instrumentación. En este ámbito, las técnicas combinatoriales poseen un elevado costo computacional, el cual puede ser disminuido mediante la aplicación criteriosa del concepto de paralelismo. Se trabajó sobre la base del mejor algoritmo secuencial de particionamiento existente, denominado GS-FLCN [1] y se desarrolló una versión paralela que se basa en búsquedas arbóreas tipo depth-first search (DFS). En cuanto a la metodología de trabajo la tarea se realizó en dos etapas, paralelizándose primero la técnica DFS genérica [2] para luego encarar específicamente la paralelización de GS-FLCN [3]. La etapa preliminar permitió fijar pautas vinculadas a aspectos esenciales tales como el estudio de los umbrales a partir de donde empezar a paralelizar y la definición de una métrica apropiada para cálculos de speed-up en arquitecturas heterogéneas que condujeran a comparaciones justas. Se obtuvo un software eficiente, flexible y escalable sobre clusters de estaciones de trabajo heterogéneas interconectadas por redes locales de comunicación.

En cuanto a la verificación de desempeño de la versión paralela, se testeó el algoritmo DFS paralelo sobre un cluster de tres estaciones de trabajo que consta de una computadora Digital Alpha, una PC Pentium II de 400 MHz y una PC Pentium de 133 MHz. Luego, el algoritmo GS-FLCN paralelo se probó con dos casos de estudio industriales en un cluster homogéneo de computadoras Pentium de 200 MHz disponibles en el Centro de Cómputos del Departamento de Ciencias de la Computación de la UNS. Se comenzó trabajando con dos máquinas, para luego efectuar un escalado incremental por agregado de nuevos nodos de procesamiento al sistema hasta totalizar doce procesadores.

Si bien este algoritmo paralelo distribuido permite ganancias significativas en tiempo, el enfoque no explota todo el potencial de paralelización del problema pues hay subgrafos que son recorridos más de una vez. El principal inconveniente es que el método no aprovecha la posibilidad de factorizar la detección de subcamino repetidos. Cada vez que la tarea Master llega a una misma hoja en su recorrido a través de un camino distinto, se genera una exploración repetida a lo largo del subgrafo cuya raíz es ese nodo. Esta deficiencia es mas notoria a medida que crece el tamaño del grafo, tanto en cantidad de nodos como en densidad de arcos. Para evitar la reiteración de exploraciones fue necesario replantear el algoritmo desde su concepción. La nueva propuesta [4], que se basa en una arquitectura Master-Supervisor-Worker descentralizada especialmente diseñada para búsquedas en profundidad, evita la exploración repetida de subcamino. El trabajo se efectuó en varias etapas. En primer lugar se desarrollaron los pseudoalgoritmos para cada una de las tareas que componen el sistema. Se determinaron las estructuras de datos más adecuadas para lograr la

funcionalidad esperada, así como también los momentos y las formas de cada una de las interacciones entre las tareas. Se trabajó, además, en un protocolo de terminación distribuido del algoritmo. El nuevo algoritmo demostró ser mas eficiente que la estrategia paralela centralizada, lográndose sustanciales mejoras en los tiempos de cómputo.

## **2. Paralelización en Entornos de Procesamiento Distribuidos de Algoritmos Secuenciales Clásicos para la Resolución de Problemas de Optimización con Función Objetivo y Restricciones No Lineales.**

En esta línea se desarrollaron algoritmos paralelos basados en métodos tipo gradiente, de probada robustez. Implementaciones secuenciales de estos métodos se han empleado con éxito en numerosos problemas de ingeniería. Las aplicaciones más interesantes desde el punto de vista de las ventajas que se derivarían de su paralelización son aquellas cuya función objetivo y restricciones son no lineales y costosas de evaluar. Dado que no existen antecedentes en la literatura sobre implementaciones de estas metodologías en ambientes distribuidos, este trabajo constituye un interesante aporte para la resolución eficiente de una amplia variedad de problemas de interés.

Dentro de la familia de los métodos tipo gradiente existen dos grandes técnicas: Programación Cuadrática Sucesiva (SQP) y Gradiente Reducido Generalizado (GRG). SQP resuelve internamente un problema con función objetivo cuadrática y restricciones no lineales para generar cada dirección de búsqueda. GRG, en cambio, resuelve un sistema lineal de ecuaciones. Por otra parte, en el SQP la función objetivo del problema de búsqueda en línea es un Lagrangiano aumentado mientras que para GRG la evaluación de dicha función objetivo exige resolver otro sistema de ecuaciones lineales. La tarea consistió básicamente en introducir paralelismo en aquellas etapas de cada algoritmo base que resultan críticas desde el punto de vista de los tiempos de cómputo. Los métodos tipo gradiente se basan en la generación de direcciones de búsqueda, para lo cual se emplea información sobre las derivadas primeras y segundas de las funciones involucradas. La tarea de evaluación de gradientes de funciones objetivo y las distintas técnicas alternativas para búsqueda en línea son los dos aspectos que concentran la mayor carga de cómputo de estos métodos. Por lo tanto, se consideró la paralelización del cálculo de gradientes y/o Hessianos, así como también del proceso de búsqueda en línea. Cada etapa de la búsqueda en línea consiste en encontrar un óptimo dentro de una dirección definida previamente. Por lo tanto, dado el vector dirección, se paralelizó el proceso de selección del tamaño de paso adecuado. De este modo se generaron versiones paralelas de los métodos SQP [5] y GRG [6]. En ambos casos se paralelizó la evaluación de todas las derivadas parciales y el proceso de búsqueda en línea. La única diferencia conceptual entre la filosofía con que se paralelizaron ambas estrategias está centrada en la búsqueda en línea. Para SQP, se evaluó la función objetivo simultáneamente en varios puntos, mientras que GRG involucró la evaluación paralela de las restricciones requeridas por la rutina de resolución de sistemas no lineales.

La implementación y chequeo de los algoritmos adecuadamente paralelizados permitió definir la ganancia que puede esperarse gracias al empleo de esta metodología. Con este propósito se implementó un conjunto de casos de prueba que corresponden a problemas clásicos empleados para la evaluación de desempeño de algoritmos de optimización [7]. Se seleccionaron problemas representativos de diferentes tipos de dificultades que pueden presentarse para el cálculo del óptimo de modo que sirvieran para evaluar la calidad del algoritmo desarrollado.

En cuanto al ambiente de cómputo utilizado, se trabajó sobre una red Ethernet heterogénea pequeña, con una estación de trabajo Digital Alpha, una PC Pentium II de 400 MHz y una PC Pentium de 133 MHz. Se estudió en detalle el desempeño en cuanto a la cantidad de nodos de procesamiento usados, speed-up y eficiencia. Desde un punto de vista más general, el diseño y programación de estas estrategias permitió extraer conclusiones acerca de la utilización óptima de CPUs y de los efectos provocados por el uso de redes de comunicación de datos.

### 3. Desarrollo e Implementación de Algoritmos Intrínsecamente Paralelos para la Resolución de Problemas de Optimización con Función Objetivo y Restricciones No Lineales.

En este campo se propuso un nuevo algoritmo de optimización paralelo que se basa en el algoritmo de distribución paralela de variables (PVD) [8]. Este enfoque está basado en la descomposición del dominio del problema, lo cual básicamente se efectúa particionando el vector de variables de optimización. La técnica original sólo contemplaba problemas de optimización sin restricciones. Esto es claramente insuficiente para el tratamiento de la mayoría de los problemas prácticos complejos. Por lo tanto, se decidió modificar la técnica para poder considerar problemas restringidos. El principal aporte de la nueva metodología es la política incorporada para manejar restricciones no lineales y la propuesta de una nueva política de particionamiento de dominio.

El algoritmo paralelo se implementó en un entorno de procesamiento distribuido y el código se empleó en primer término para resolver varios modelos no lineales pequeños asociados a problemas clásicos de ingeniería [7]. A continuación, se encaró un problema industrial más grande y más complejo, correspondiente al modelo riguroso de una planta de expansión existente. Se lograron valores satisfactorios de speed-up y eficiencia. Una descripción detallada del algoritmo y el análisis de desempeño sobre todos estos ejemplos puede consultarse en [9].

#### Referencias

[1] **Ponzoni I., M.C. Sanchez, N.B. Brignole** "A New Structural Algorithm for Observability Classification" *Ind. Eng. Chem. Res.* (ISSN: 0888-5885), 38, 8, 1999, 3027-3035.

[2] **Vazquez G.E., I. Ponzoni, N.B. Brignole** "Parallel Depth-First Search on Clusters of Workstations", *PS SIAM Annual Meeting 1999*, Atlanta, USA, 12-15 Mayo 1999, 198

[3] **Ponzoni I., G.E. Vazquez, M.C. Sánchez, N.B. Brignole.** "Parallel Observability Analysis on Networks of Workstations", *Comp. & Chem. Eng.* (ISSN: 0098-1354), 25/7-8, 997-1002, 2001

[4] **Fapitalle, F., G.E. Vazquez, I. Ponzoni, N.B. Brignole** "A Decentralized Parallel-Distributed Observability Algorithm", aceptado *V WCCM (Fifth World Congress on Computational Mechanics)*, Viena, Austria, Julio 7-12, 2002

[5] **Vazquez G.E., Brignole N.B.** "Parallel NLP Strategies using PVM on Heterogeneous Distributed Environments", *Lecture Notes in Computer Science* (ISSN: 3-540-66549-8), 1697: *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, 533-540, Springer-Verlag, Berlin, 1999.

[6] **Vazquez G.E., R. Rainoldi, N.B. Brignole** "Non-Linear Constrained GRG Optimization under Heterogeneous Parallel-Distributed Computing Environments", *Computer-Aided Chemical Engineering* (ISBN: 0-444-82409-X), Vol. 8, Elsevier, 2000, 127-132.

[7] **Hock W., Schittkowsky K.**, "Test Examples for Nonlinear Programming Codes", Springer-Verlag, New York, 1981.

[8] **M.C. Ferris, O.L. Mangasarian** "Parallel Variable Distribution", *SIAM Journal of Optimization*, 4, 1994, 815-832.

[9] **Vazquez G.E., S. Diaz, N.B. Brignole, J.A. Bandoni** "Optimization of Industrial Problems using Parallel Processing under Distributed Environments", *Chemical Engineering Communications* (ISSN: 0098-4645), 189, 6, 2002.

## ***Integración de bloques IP en diseños SoC***

**Horacio Villagarcía Wanza, Oscar Bria<sup>1</sup>**

*LIDI. Laboratorio de Investigación y Desarrollo en Informática.  
Facultad de Informática. UNLP.  
50 y 115. 1er Piso. La Plata*

### **Palabras Clave**

Arquitectura de Computadoras, SoC -System on a Chip-, bloques IP -Intellectual Property-, VHDL.

### **Resumen**

La metodología de diseño de sistemas SoC basándose en bloques IP (Hard o Soft) presenta diversos retos, desde las cuestiones relativas a la síntesis y validación de cada bloque IP, hasta las características de integración física y verificación funcional del sistema completo. Nos propusimos abordar la temática del diseño de bloques IP basándonos en descripciones de hardware por lenguaje (en particular VHDL) y avanzar en los mecanismos para re-utilizar dichos bloques como parte de distintos sistemas.

### **Introducción**

La complejidad del hardware ha crecido de modo sostenido en los últimos años a punto tal que es posible diseñar -y fabricar- computadores digitales completos (CPU, Memoria y E/S) o colocar múltiples procesadores en un único chip de silicio con diversas soluciones tecnológicas (por ejemplo: ASIC's, CPLD's, FPGA's y Systems-on-Chip entre otras) y se proponen metodologías de diseño con distintos niveles de abstracción – flexibilidad que se basan en esas nuevas soluciones tecnológicas [1,2].

En la metodología de diseño System-on-a-Chip (SoC) el diseñador combinará bloques pre-diseñados y pre-verificados en un chip para implementar funciones complejas, poseyendo un limitado conocimiento de la estructura interna de esos bloques. Debe asegurarse entonces, el correcto diseño físico (temporizados, potencia, tamaño) y un comportamiento funcional verificable del sistema.

### **Temas de Investigación y Desarrollo**

Son de interés los siguientes temas:

- **Integración de bloques IP en diseños SoC.**

El ensamble de un conjunto de bloques en un diseño jerárquico presentará varias opciones y por lo tanto podremos encontrar diferentes problemas que deberán ser administrados con distintas estrategias. La selección de un bien diseñado y bien documentado bloque IP reduciría el esfuerzo de integración [5,7].

- **Verificación a nivel sistema.**

Las estrategias de verificación a nivel de sistema en un diseño SoC usan el esquema “divide y conquista” de un sistema jerárquico. Se comienza con la verificación de los

---

<sup>1</sup> Profesores Ordinarios Dedicación Exclusiva

bloques elementales, luego las interfaces entre bloques y finalmente el sistema completo (chip). En particular éste último paso es una verificación funcional que se realiza con aplicaciones de complejidad creciente [6].

- **Re-usabilidad de bloques IP.**

El temporizado, el consumo de potencia y las estructuras de test de los bloques “Hard IP” son tópicos definidos en la etapa de diseño del bloque propiamente dicho y que facilitan o no su re-usabilidad. Los mecanismos de interface previstos para cada bloque también constituyen un elemento definitorio de su re-usabilidad [6].

En todos los casos asumimos que la herramienta básica para generar un bloque IP será un Lenguaje como VHDL. En particular, utilizaremos para la ‘compilación a silicio’ dispositivos lógicos programables complejos (CPLD).

### **Algunos resultados obtenidos. Líneas de Investigación en curso.**

Además de los primeros estudios académicos mencionados en [1] y en los que se utilizó VHDL como herramienta de descripción, trabajamos en la idea de incorporar múltiples procesadores (bloques IP) en un único chip [3,4], surgiendo diferentes inconvenientes y por consiguiente problemas a resolver o, al menos, abordar. En particular nos centramos en el estudio de políticas para la verificación funcional de SoC, las características de interface entre bloques IP y el análisis más profundo de las especificaciones de diseño para facilitar la aplicación de test a los bloques IP.

### **Bibliografía**

- 1- H. Villagarcía, O. Bria “Diseño de bloques IP: Programabilidad y Re-utilización”, Anales III Workshop de Investigadores en Ciencias de la Computación, pág.2-5. Mayo 2001.
- 2- S. Wilton, R.Saleh, “Programmable Logic IP Cores in SoC Design: Opportunities and Challenges”, Proc. IEEE Custom Integrated Circuit Conference, San Diego, CA., pág. 63-66. Mayo 2001.
- 3- G. Jaquenod, H. Villagarcía, O. Bria, M. De Giusti “Adaptación del núcleo IP de un procesador tipo MC6805 para operar en un ambiente multiprocesador y multitarea”, Anales VII Congreso Argentino de Ciencias de la Computación, Tomo II, pág. 1323-1332, Octubre 2001.
- 4- G. Jaquenod, H. Villagarcía, M. De Giusti “Towards a Field Configurable non-homogeneous Multiprocessors Architecture”, Proc. V World Multiconference on Systemics, Cybernetics and Informatics, Vol XIV, pág. 248-253, Julio 2001
- 5- J. Plantin, E. Stoy, “Aspects on System-Level Design”, Proc. VII Int’l Workshop on Hardware Software Codesign, pág.209-210. Mayo 1999.
- 6- M. Keating, P. Bricaud, “Reuse Methodology Manual for System-On-a-Chip Designs” 2<sup>nd</sup> Ed, Kluwer Academic Publishers 2001, USA. ISBN 0-7923-8558-6.
- 7- W. Wolf, “Computers as Components: Principles of Embedded Computer Systems Design”, Morgan Kaufmann Publishers 2000, USA. ISBN 155860541X



# Modelando la Performance de Aplicaciones en un Clusters de PC's

Villalobos M., Flores S., Piccoli F., Printista M. \*

Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950  
5700 - San Luis  
Argentina

e-mail: {mavi, sflores, mpiccoli, mprinti}@unsl.edu.ar

## 1 Introducción

Un *cluster* de computadoras es una máquina que consiste de un número de workstations o PC's de bajo costo interconectados por una red para actuar como un único recurso de computación.

Los multiprocesadores simétricos, *SMP*, corrientemente tienen mejor relación precio-performance que una única workstation y por lo tanto son muy atractivos para ser incorporados en un cluster.

Resolver problemas en máquinas paralelas introduce intercambio de datos. El volumen de datos intercambiados crece con el número de procesadores intervinientes en la aplicación. Por lo cual, para construir clusters escalables, la capacidad de la red debe escalar con el número de nodos en el cluster.

A medida que el *cluster* ha ganado popularidad como plataforma adecuada para computaciones paralelas, ha surgido la necesidad de predecir la performance para el análisis de aplicaciones escalables que corren sobre dicha plataforma. La estimación de la performance facilita actividades tales como mejorar el rendimiento (*tuning*) de las aplicaciones, selección de las mejores técnicas de implementación y comparación de la performance entre distintas plataformas de multicomputadoras.

La performance de un multicomputador es una función de la performance de los procesadores o *SMP* utilizados, pero el *tiempo de pasaje de mensajes* es el tiempo más influyente en aplicaciones con pasaje de mensajes de gran volumen. Estimar este tiempo es un problema complejo. El tiempo depende de las características de los procesadores, de la red de interconexión, de la librería de pasaje de mensajes utilizada, de los patrones de comunicación, de los tamaños de los mensajes, del número de nodos y de la distancia entre los nodos de procesadores.

El objetivo de este proyecto es analizar y desarrollar un modelo que caracterice la performance de aplicaciones paralelas en un cluster de PC's. En particular, nuestros desarrollos se están realizando en un cluster de 14 *SMP* (duales), conectadas mediante una *Fast Ethernet*. Este cluster pertenece a la Unidad de Servicio Computacional (para aplicaciones científicas) del Instituto de Matemática Aplicada San Luis (IMASL), de la U.N.S.L.

---

\*Grupo subvencionado por la UNSL y ANPCYT (Agencia Nacional para Promover la Ciencia y Tecnología )

## 2 Modelo de Computación

Para modelar los costos de Computación, nos basamos en medidas que son dependientes de la aplicación. Este modelo propone un límite superior para la performance que puede ser alcanzado por una aplicación particular sobre ciertos procesadores. Dicho límite es encontrado contando las operaciones esenciales en la aplicación. El inconveniente radica en que algunas operaciones tienen diferencias significativas en sus costos. Por ejemplo una iteración que opera sobre un arreglo de números enteros es significativamente más costosa que una iteración que opera sobre un arreglo de números de punto flotante. Basándonos en esto, trabajamos sobre “bloques conceptuales de computación”. Para el análisis de performance nos concentramos en estos bloques asignándoles un factor de costo. Cada bloque conceptual,  $\Gamma^j$ , tiene un conjunto de instrucciones asociadas denominado  $\Upsilon_{\Gamma^j}$ . Notar que en base al número de veces que cada operación de  $\Upsilon_{\Gamma^j}$  se ejecuta se pueden formar  $k$  subconjuntos que cumplen:

$$\bigcup_{i=1}^k \Upsilon_{\Gamma^{j_i}} = \Upsilon_{\Gamma^j} \text{ y } \bigcap_{i=1}^k \Upsilon_{\Gamma^{j_i}} = \emptyset \quad (1)$$

Asociando un costo a cada subconjunto, se puede expresar el tiempo de computación insumido en cada bloque conceptual como:

$$T_{comp}^j = \sum_{i=1}^k C_i * N_{\Gamma^{j_i}} \quad (2)$$

donde  $C_i$  es el factor de costo del subconjunto  $\Gamma^{j_i}$  y  $N_{\Gamma^{j_i}}$  es el número de veces que las operaciones del subconjunto se ejecutan durante el programa. La expresión final para el tiempo de computación insumido durante la ejecución del programa se obtiene por la composición del tiempo de todos los bloques:

$$T_{comp}(p) = \sum_{j=1}^{n_{bcomp}} T_{comp}^j \quad (3)$$

## 3 Modelo de Comunicación

El enfoque presentado aquí se basa en medir y modelar la performance de un conjunto de patrones básicos de comunicación. Estos patrones fueron seleccionados de forma tal que otros patrones de comunicación más complejos puedan ser construidos desde los patrones básicos analizados. Se ha utilizado como librería de comunicación el estándar *MPI* [4], ya que sus primitivos de comunicación cubren la totalidad de primitivos de intercambio de datos globales y de primitivos de sincronización.

Similar al esquema de bloques conceptuales utilizado en las computaciones, para el análisis de las comunicaciones nos concentramos en cada comunicación en particular (la que será considerada un bloque conceptual de comunicación).

Para el cálculo de los tiempos de cada patrón de comunicación se realizaron repetidamente diferentes experimentos de los algoritmos que los implementan. En cada experimento se varía la longitud de los mensajes y el número de procesadores intervinientes. Se utilizó como medida estándar el *RoundTripTime* de la red [1] y los patrones de comunicación básicos considerados fueron: *Point-to-Point*, *Exchange*, *One-to-Many*, *Many-to-One*. Para desarrollar las fórmulas que den el tiempo como una función del número de procesadores ( $p$ ), los tiempos muestrales obtenidos serán analizados mediante la Técnicas de Ajuste de Multivariables [3]. Para modelar el tiempo de comunicación de cada bloque conceptual nos basamos en dos componentes: *tiempo de setup* de la red y *tiempo de transferencia*. En tiempo de comunicación para un bloque  $j$ , se puede expresar con la siguiente fórmula:

$$T_{comm}^j(p) = t_{comm}(p) + tt_{comm}(p) * longmens \quad (4)$$

Donde  $t_{comm}$  y  $tt_{comm}$  corresponden al tiempo de setup y de transferencia respectivamente. Estos tiempos deberán ser instanciados al patrón de comunicación particular involucrado en cada bloque conceptual de comunicación.

De lo anterior resulta que el tiempo total insumido en comunicaciones durante la ejecución de una aplicación paralela se obtiene por la siguiente fórmula:

$$T_{comm}(p) = \sum_{j=1}^{n_{bcomm}} T_{comm}^j(p) \quad (5)$$

## 4 Integración del Modelo

En nuestro enfoque para modelar la performance del cluster se han ejecutado dos aplicaciones intensivas de pasaje de mensajes: un algoritmo pipeline que resuelve el problema de los  $N$ -cuerpos en el espacio [2] [5] y el algoritmo de *Fox* [4] para resolver la multiplicación de matrices. El tiempo de ejecución de estas aplicación es la suma de sus tiempos de computación y de comunicación. El límite superior para el tiempo de computación se encuentra cronometrando los "bloques conceptuales de computación" existente en el código fuente de las aplicaciones secuencializado (con cantidad de procesadores  $NP = 1$ ), como especifica el modelo en ( 2) y ( 3). El tiempo de comunicación se calcula identificando las llamadas explícitas a las rutinas de pasaje de mensajes de MPI. Luego, se debe relacionar estas llamadas a un patrón básico de comunicación y finalmente de aplica el modelo de comunicación dado en ( 4) y ( 5).

## 5 Conclusiones y Próximos Pasos

El resultado del trabajo preliminar realizado, prueba la utilidad del modelo para ajustar aplicaciones de pasaje de mensajes, permitiendo seleccionar las mejores técnicas de implementación.

Es importante mencionar que el proyecto se dividió en las 3 etapas siguientes:

- Definición del Modelo de Performance
- Análisis de Performance de los patrones básicos de Comunicación
- Análisis y Predicción de la Performance

Este reporte de investigación incluye los resultados de la primera etapa del proyecto. La segunda etapa está en desarrollo. Una vez finalizada, quedar a definido el conjunto de parámetros claves que determinan la performance de comunicación del cluster especificado. Con esta información disponible, será posible realizar predicciones basadas en el modelo analítico presentado en este reporte.

El objetivo último, es que los parámetros obtenidos del modelo resulten lo suficientemente robustos para realizar la comparación de la performance entre distintas plataformas de multicomputadoras.

## Referencias

- [1] Abandah G. and Davinson E.S. *Modeling the Communication Performance of the IBM SP*", (10th. International Parallel Processing Symposium, 1996).
- [2] Marciniak, A. *Numerical solutions of the N-body Problem* (D. Reidel Publishing Co., Dordrecht, 1985).
- [3] Mendenhall W., Wackerly D. D. and Scheaffer R.L. *Mathematical Statistic with Applications*, (Duxbury Press, 1989).
- [4] Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J. *MPI: The complete Reference* (Cambridge, MIT Press, 1996).

- [5] Wilkinson B. and Allen M. *Parallel Programming: Techniques and Application using Networked Workstations and Parallel Computers* (Prentice-Hall, 1999).

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

# WICC 2002

*16 y 17 de mayo de 2002*

## Sistemas Inteligentes. Metaheurísticas



# Agentes y aprendizaje de máquina en optimización, control y entornos de enseñanza

Guillermo Aguirre      Marcelo Luis Errecalde      Guillermo Leguizamón

LIDIC \*  
Departamento de Informática  
Universidad Nacional de San Luis(UNSL)  
Ejército de los Andes 950 - Local 106  
5700 - San Luis. Argentina  
e-mail:{gaguirre,merreca,legui}@unsl.edu.ar

## Resumen

Este trabajo describe el estado actual de las tareas de investigación realizadas en forma conjunta por un grupo de investigadores pertenecientes al LIDIC.

Las problemáticas abordadas por los integrantes de este grupo son disímiles y abarcan temas como optimización, control y entornos de enseñanza. Sin embargo, tienen como elementos en común la utilización de agentes y el aprendizaje de máquina.

Un aspecto interesante de este trabajo conjunto es que permitió detectar problemáticas comunes que son abordadas con técnicas diferentes. Estas técnicas son usualmente investigadas en forma independiente y existen pocos trabajos que analizan sus similitudes y diferencias. A nuestro entender esto constituye una seria limitación ya que un mejor entendimiento de estos aspectos permitiría que cada una de estas áreas se nutriera con conceptos y resultados desarrollados en las restantes.

Con este objetivo en mente, estamos actualmente trabajando en forma conjunta en tres líneas principales de investigación que podemos referenciar en términos generales como: a) agentes, aprendizaje y optimización, b) control de agentes mediante técnicas de aprendizaje híbridas y c) aprendizaje en agentes de interfaz para entornos de enseñanza. Cada una de estas líneas de investigación se describen en forma general en este trabajo.

**Palabras Claves:** Agentes Inteligentes, Aprendizaje de Máquina, Optimización, Control, Entornos de Enseñanza-Aprendizaje.

## 1 Agentes, aprendizaje y optimización

La técnica ACO (Ant Colony Optimization) ha surgido recientemente como una nueva meta-heurística para atacar problemas duros de optimización combinatoria. Los algoritmos ACO se definen como instancias de la meta-heurística ACO, los cuales son básicamente un sistema multi-agente donde la interacción a muy bajo nivel de sus agentes componentes (llamados hormigas

---

\*El laboratorio es dirigido por el Dr. Raúl Gallard y subvencionado por la UNSL y la ANPCYT (Agencia Nacional para la Promoción de la Ciencia y la Tecnología)

artificiales) resulta en un comportamiento complejo del sistema en su totalidad. Dichos algoritmos han sido inspirados por las colonias de hormigas reales [5] las cuales depositan una sustancia química llamada feromona sobre el terreno por donde caminan. Esta sustancia influye en la elección que ellas hacen al momento de decidir que paso o camino seguir en su ruta de exploración (generalmente hacia y desde la fuente de alimento), así, mientras mayor sea la acumulación de feromona sobre un paso particular, mayor será la probabilidad de elegir dicho paso. De esta manera, las hormigas artificiales de un algoritmo ACO se comportan de manera similar.

Las primeras aplicaciones de la meta-heurística ACO están vinculadas con el Problema del Viajante de Comercio [5, 7, 6, 8], siendo tal vez el problema más estudiado en el contexto de estos algoritmos. Sin embargo, existe un importante conjunto de estudios experimentales que incluye los siguientes problemas: Problema de Asignación Cuadrática, Ruteo, Coloreo de grafos, Redes de Telecomunicación, Scheduling [5], como así también, el Problema de Múltiples Mochilas, Máximo Conjunto Independiente [15, 16], entre otros.

Es importante destacar que los algoritmos ACO pueden fácilmente aplicarse a problemas de optimización discreta que puedan ser caracterizados como grafos (aunque hay excepciones demostradas para otro tipo de problemas [15, 16]). Así, la solución a un problema de optimización puede ser expresada en término de pasos factibles sobre dicho grafo.

Uno de los principios más importantes que guían la búsqueda de estos algoritmos está directamente vinculado con la cooperación indirecta, denominada stigmergy, de los agentes que conforman el sistema. Dicha cooperación se refleja a través del depósito de una cierta cantidad de feromona (refuerzo) sobre los pasos transitados por los agentes en su recorrido por el grafo proporcional a la calidad de la solución encontrada por cada uno de los agentes. La cantidad acumulada de feromona durante los sucesivos ciclos de ejecución del algoritmo conformará el bloque de construcción a partir del cual nuevas soluciones puedan ser generadas. En consecuencia, dicho bloque de construcción determinará en subsecuentes iteraciones la región del espacio de búsqueda a ser considerada. Más allá que un algoritmo ACO incluya información específica del problema e inclusive alguna forma de hibridización para mejorar su performance, nuestro interés está principalmente centrado en los aspectos relacionados a la forma de actualización del refuerzo y la manera en que dicha información es usada al momento de decidir cual es el paso a seguir. Recientes trabajos de algoritmos ACO aplicados a problemas de scheduling [17, 18] están dirigidos principalmente a formas alternativas de usar la información proveniente de la cooperación (feromona sobre los pasos) para mejorar la performance del algoritmo, aunque aplicado a un problema de scheduling particular.

Nuestra propuesta de trabajo actual pretende enfocar y fundamentar el comportamiento de los algoritmos ACO desde la perspectiva del aprendizaje por refuerzo. Aun más, dada ciertas semejanzas entre los distintos enfoques, un estudio posible está relacionado con la transferencia de resultados desde el área de aprendizaje por refuerzo que puedan mejorar y/o entender con mayor profundidad el funcionamiento de los algoritmos ACO.

Con respecto a este último punto, ya hemos realizado una experiencia similar estudiando las similitudes entre Sistemas Clasificadores (SC) y el método de AR denominado Q-Learning. A partir de este estudio logramos adaptar otros métodos de AR (p. ej. SARSA) al contexto de los SC. Esto permitió derivar un nuevo esquema dentro de los SC, que además de lograr un buen comportamiento tiene garantías de convergencia al óptimo que ya han sido demostradas con SARSA en el área del AR.

## 2 Control de agentes mediante técnicas de aprendizaje híbridas

En el grupo de investigación se han realizado distintos trabajos que utilizan el Aprendizaje por Refuerzo (AR) como mecanismo principal para resolver el problema de controlar agentes autónomos [10, 12, 9, 11, 20].

El AR se basa en la idea central del aprendizaje *por prueba y error* [21], un concepto originalmente analizado en estudios relacionados con el comportamiento animal.

En su concepción más pura el AR plantea, como solución al problema del control de agentes autónomos, distintas técnicas de aprendizaje que se basan exclusivamente en la señal de recompensa que recibe desde el ambiente. Dentro de este esquema, no se asume ningún tipo de conocimiento previo sobre el ambiente o la tarea a resolver. Estas características llevaron a que muchos investigadores plantearan al AR como una solución general para aquellos dominios desconocidos o poco entendidos para el diseñador de un agente artificial.

En este contexto, una pregunta interesante que se plantea es: ¿puede un agente de AR, con tan poca información disponible, escalar a problemas generales y complejos del mundo real?

En este sentido, trabajos relacionados a la complejidad del AR elemental, han demostrado que el problema de AR es en general intratable si no se recurre a variantes que permiten incorporar conocimiento en el proceso de exploración del ambiente.

Por otro lado, no es realista pensar que todos los ambientes en que se desempeñará el agente deban ser completamente desconocidos. En algunos casos puede existir información previa disponible que sirva para un mejor comportamiento inicial del agente. Esta información cumplirá un rol similar al de los *reflejos*, que permiten mantener orientado al agente en la dirección correcta mientras intenta aprender.

En otros casos, el agente podrá convivir con otros agentes y el proceso de aprendizaje puede involucrar el descubrimiento por prueba y error, pero también cumple un rol fundamental la información obtenida desde otros agentes con quienes convive. En estas situaciones, y en concordancia con lo expresado en [22], *el aprendizaje es más a menudo una cuestión de transferencia que de descubrimiento*. Dentro de este contexto social, el agente puede mejorar su comportamiento mediante distintos procesos de aprendizaje. Una forma de hacerlo es observando e *imitando* el comportamiento de otros agentes más experimentados. Los otros agentes también pueden actuar como *consejeros* o *instructores* quienes proveen consejos, órdenes de ejecución obligatoria o simplemente ejemplos del comportamiento deseado. Es interesante notar que estas formas de aprendizaje, ya han sido analizadas en distintos trabajos como una forma de extender y superar las limitaciones del AR. Sin embargo, estos enfoques han sido incorporados por separado y sin un marco general que permita integrarlos en forma coherente a todos ellos.

En este sentido, nuestra hipótesis actual de trabajo, plantea que para construir agentes artificiales basados en aprendizaje, que se desempeñen en forma efectiva en la resolución de problemas complejos, es necesario extender el modelo del AR con nuevas formas de experiencia que puedan servir para mejorar el proceso de aprendizaje. Nuestro punto de partida, consiste en considerar al aprendiz como parte de un contexto social donde puede aprender no sólo desde las recompensas recibidas por sus acciones, sino también desde las múltiples fuentes de conocimiento que puedan estar socialmente disponibles, incluyendo agentes humanos y artificiales.

Nuestro objetivo es poder integrar en forma coherente en el proceso de aprendizaje estas múltiples fuentes de información a las que denominaremos *fuentes de experiencia*. En el contexto del aprendizaje de políticas de control, es decir, el problema de aprender a mapear situaciones en

acciones, consideramos como *f fuente de experiencia a cualquier dispositivo computacional que sirva para expresar sus preferencias sobre qué acción tomar en distintas situaciones*. Este dispositivo podrá tomar la forma de un sistema de aprendizaje por refuerzo, de aprendizaje supervisado, de aprendizaje por consejos, un planner deliberativo u otro cualquiera que sirva en la toma de decisiones sobre la acción a tomar.

Para poder integrar las diversas fuentes de experiencia en el proceso de decisión del agente, se pueden identificar al menos 3 tareas fundamentales a realizar:

1. **Definición de un lenguaje común de preferencias de acciones:** La salida de cada fuente, deberá ser traducida a este lenguaje común de preferencias antes de poder ser utilizadas en la toma de decisiones del agente.
2. **Definición de los traductores para cada fuente de experiencia:** Para cada fuente de experiencia se deberá asociar un traductor particular que convierta la salida producida por la fuente en el lenguaje común de preferencias de acciones.
3. **Definición de un mecanismo de consenso:** Las preferencias expresadas por las distintas fuentes en el lenguaje común deberán ser integradas en forma coherente para lograr como resultado una única política de control del agente. Será necesario por lo tanto, definir un mecanismo que establezca las reglas mediante las cuales se realiza esta integración.

Actualmente se ha definido un lenguaje común de preferencias de acciones que permite expresar políticas de control estocásticas. También se ha definido un mecanismo de consenso híbrido que combina un esquema de votación ponderado con un sistema jerárquico dictatorial para aquellas situaciones que requieren un tratamiento prioritario.

### 3 Aprendizaje en agentes de interfaz para entornos de enseñanza

La investigación sobre esta temática gira principalmente en torno de la problemática del diseño y la realización de una interfaz que facilite y haga más efectiva la interacción Hombre-Máquina, logrando comprender y controlar relaciones establecidas para resolver problemas en forma colaborativa sobre entornos virtuales basados en texto (MOO's). Los principales tópicos de investigación abarcan los agentes de interfaz, trabajo colaborativo soportado por computadoras y aprendizaje automático [4, 2]. El trabajo es abordado en el marco de conocimiento distribuido considerando varias aproximaciones que otorgan a los aspectos cognitivos y sociales una ponderación particular considerando que el medio de comunicación utilizado es la computadora. Esta ponderación particular implica emplear enfoques especiales para los términos *conocimiento* y *colaboración*: no se restringe a un conocimiento *en la cabeza* sino que se incluyen varios agentes y los elementos que estos emplean para hacer su tarea. En relación al trabajo colaborativo se lo considera como una actividad sincrónica y coordinada que es el resultado de un intento continuo de construir y mantener una concepción compartida de un problema. Esto se consigue creando una estructura de conocimiento compartida que integre metas, descripciones del estado actual del problema, conocimiento de acciones disponibles para la resolución de problemas y asociaciones que vinculan estos tres elementos. Esta estructura de conocimiento es una pieza central para conseguir un comportamiento como el esperado. El trabajo colaborativo sólo puede progresar en la medida que se logre

mantener y aumentar una plataforma de conocimiento compartida (ground) construida a partir de coincidencias establecidas entre los pares que intervienen en la actividad colaborativa [19].

El medio de comunicación condiciona el volumen del ground requerido; así en el caso de la comunicación a través de computadoras, se requiere un ground más grande que en una comunicación cara a cara. En nuestro caso los recursos de software que los usuarios pueden emplear para desarrollar su tarea son: un MOO en el cual se desarrollan comunicaciones basadas en texto y alguna aplicación específica de acuerdo a la problemática sobre la que se está trabajando [3]. A partir de la observación de las actividades que los pares realizan mientras usan estos recursos de software una interfaz inteligente puede construir el ground necesario para permitir una labor colaborativa. La interfaz, por medio de agentes computacionales irá monitoreando a los usuarios y recolectando información.

La aproximación socio-cultural al conocimiento humano ha ganado influencia en la tecnología educativa siguiendo la corriente encabezada por Vigotsky. Es así que hoy en día se otorga bastante importancia al papel que juegan los compañeros y los docentes en el aprendizaje. En particular la posibilidad de trabajar con ellos colaborativamente, usando computadoras como soporte, es una manera de lograr aprendizajes con un mínimo de enseñanza [1]. El aspecto social es muy importante en este tipo de entornos de aprendizaje por lo que nuestro trabajo le concede un lugar primordial. Usando técnicas de aprendizaje automático se pretende conseguir conocimiento sobre la forma en la cual los grupos interactúan y sobre el desempeño de cada usuario individualmente. El centro de la actividad social será el MOO, desde allí se puede conseguir información sobre los encuentros entre pares, los horarios y lugares dentro del MOO en que han coincidido, etc. Nuestro trabajo toma en consideración los logros conseguidos por el agente **Cobot**, que recolecta información social dentro de *LambdaMOO* [14, 13] para conseguir aprender cual es el comportamiento que prefieren los usuarios. Las acciones que Cobot realiza por su propia iniciativa resultan de lo aprendido sobre las preferencias de los usuarios que en determinado momento estén conectados al MOO. La técnica de aprendizaje por refuerzo fue escogida por las particulares características que presenta el entorno. En el MOO, las fuentes de aprendizaje son múltiples, ocasionalmente los usuarios se comportan de manera inconsistente o contradictoria, los experimentos son irreproducibles ya que se consideran entre otras cosas quienes son los usuarios conectados, la mayoría de los usuarios pierde interés en proveer rewards a Cobot por lo que las chances de aprendizaje son escasas y surge la posibilidad que un grupo reducido de usuarios condicione el comportamiento de Cobot. Por estas razones el algoritmo de aprendizaje considera para cada usuario una función de valuación individual basada sobre su feedback particular, así dentro del espacio de estado considerado se combinan las funciones de los usuarios presentes.

## 4 Conclusiones

Esta presentación describe los trabajos que actualmente se están realizando en tres temáticas en apariencia diferentes pero que, sin embargo, tienen características en común. Estas incluyen el uso de agentes, aprendizaje de máquina y los aspectos sociales y colaborativos de la sociedad de agentes como un todo. A partir de algunas experiencias que hemos realizado en base a este trabajo interdisciplinario hemos podido transferir resultados teóricos y prácticos entre las áreas que han llevado a un mejor entendimiento y aprovechamiento de trabajos realizados en forma aislada.

Es nuestro objetivo seguir aplicando la misma metodología de trabajo para poder justificar técnicas existentes u obtener nuevas propuestas tanto en las áreas de optimización, control como de entornos de enseñanza.

## Referencias

- [1] G. Aguirre and M. Lucero. Aprendizaje en entornos virtuales basados en texto. In *Congreso de Informatica en las universidades InfoUni'2001*, 2001.
- [2] Guillermo Aguirre. Creacion de entornos de aprendizaje en mundos virtuales. In *CACiC 2001*, pages 15–26, 2001.
- [3] Guillermo Aguirre. Interfaces inteligentes para resolución de problemas en entornos virtuales. In *WICC 2001*, pages 80–82, 2001.
- [4] C. Buiu and G. Aguirre. Learning interface for collaborative problem solving. In *Advanced Research in computers and communications in education*, pages 301–303, 1999.
- [5] D. Cornea, M. Dorigo, and F. Glover, editors. *New Ideas in Optimization*. McGraw-Hill International, 1999.
- [6] M. Dorigo, G. Di Caro, and L.M. Gambardella. Ant Algorithms for Discrete Optimization. *Artificial Life*, 5(2):137–172, 1996. Also available as Tech. Rep. IRIDIA/98-10, Université Libre de Bruxelles, Belgium.
- [7] M. Dorigo and L.M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66, 1997.
- [8] M. Dorigo, V. Maniezzo, and A. Colomi. Positive feedback as a search strategy. Technical Report Tech. Rep. No. 91-016, Politecnico di Milano, Italy, 1991.
- [9] Marcelo Errecalde. Marcos teóricos del aprendizaje por refuerzo multiagente. In *Proceedings del Workshop de Investigadores en Ciencias de la Computación 2001 - WICC 2001*, pages 77–79, San Luis, Argentina, 2001.
- [10] Marcelo Errecalde, Maria Liz Crespo, and Cecilia Montoya. Aprendizaje por refuerzo: Un estudio comparativo de sus principales métodos. In *Proceedings del Segundo Encuentro Nacional de Computación*, México, 1999.
- [11] Marcelo Errecalde and Alfredo Muchut. Exploración dirigida por el objetivo en aprendizaje por refuerzo basado en modelo para ambientes no estacionarios. In *Proceedings del Congreso Argentino de Ciencias de la Computación 2001 - CACIC 2001*, pages 1117–1129, El Calafate, Argentina, 2001.
- [12] Marcelo Errecalde, Alfredo Muchut, Guillermo Aguirre, and Montoya Cecilia. Aprendizaje por refuerzo aplicado a la resolución de problemas no triviales. In *Proceedings del Workshop de Investigadores en Ciencias de la Computación 2000 - WICC 2000*, pages 49–51, La Plata, Argentina, 2000.
- [13] Charles Lee Isbell, Chistian Shelton, and Peter Stone Michael Kearns, Satinder Singh. Cobot in lambdamoo: A social statistics agent. In *Fifth International Conference on Autonomous Agents*, 2000.
- [14] Charles Lee Isbell, Chistian Shelton, and Peter Stone Michael Kearns, Satinder Singh. A social reinforcement learning agent. In *Fifth International Conference on Autonomous Agents*, 2000.
- [15] G. Leguizamón and Z. Michalewicz. A New Version of Ant System for Subset Problems. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1459–1464. IEEE Press, Piscataway, NJ, 1999.
- [16] G. Leguizamón and M. Schütz. An ant system for the maximum independent set problem. In *VII Congreso Argentino de Ciencias de la Computación*, El Calafate, Santa Cruz, Argentina, 2001.
- [17] D. Merkle and M. Middendorf. An ant algorithm with a new pheromone evaluation rule for total tardiness problems. In *Proceeding of the EvoWorkshops 2000*, number 1803 in Lectures Notes in Computer Science, pages 287–296. Springer Verlag, 2000.
- [18] D. Merkle, M. Middendorf, and H. Schmeck. Ant colony optimization for resource constrained project scheduling. In D. Whitley et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, pages 893–900, Las Vegas, Nevada, USA, 10-12 July 2000. Morgan kaufmann.
- [19] D. Traum P. Dillenbourg and D. Shneider. Grounding in multi-modal task-oriented collaboration. In *EuroAI and Education Conference*, 1996.
- [20] Marcela Printista, Marcelo Errecalde, and Cecilia Montoya. Una implementación paralela del algoritmo de q-learning basada en un esquema de comunicación con caché. In *Proceedings del Congreso Argentino de Ciencias de la Computación 2000 - CACIC 2000*, Ushuaia, Argentina, 2000.
- [21] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. The MIT Press, 1998.
- [22] Steven D. Whitehead. A complexity analysis of cooperative mechanisms in reinforcement learning. In *Proceedings of the AAAI*, pages 607–613, 1991.

# Evolución de Controladores Basados en Redes Neuronales

Javier Apolloni, Carlos Kavka y Patricia Roggero

LIDIC

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950

D5700HHW - San Luis - Argentina

Tel: 02652-420823

Fax: 02652-430224

e-mail: {javierma,ckavka,proggero}@unsl.edu.ar

## Resumen

En los últimos años se han utilizado los algoritmos evolucionarios con mucho éxito para la obtención de controladores, en particular controladores representados como redes neuronales. La principal desventaja de estos métodos es que la solución obtenida no se puede adaptar a cambios en el sistema. Con el objeto de resolver este problema, se propone en este trabajo el almacenamiento de poblaciones intermedias obtenidas durante el proceso evolucionario. De esta manera, frente a cambios en el sistema bajo control, se puede iniciar una nueva evolución a partir de una población intermedia, con lo cual el tiempo necesario para obtener un nuevo controlador se reduce drásticamente.

## 1 Introducción

El diseño de controladores es una tarea difícil. En algunos casos los controladores pueden ser definidos a mano ajustando un conjunto de parámetros. Sin embargo, si el sistema a ser controlado, o una aproximación de él, puede ser simulado en una computadora, es conveniente usar un algoritmo para determinar automáticamente la mayor cantidad de parámetros posible.

Un ejemplo de un sistema típico que debe ser controlado es el de un tanque en el que se desea mantener un nivel constante de combustible. El sistema consta de una válvula a través de la cual se regula el flujo del combustible que ingresa al tanque y un flotante (sensor) que permite medir el nivel actual de combustible. El controlador debe determinar el grado de apertura de la válvula para mantener el nivel constante, en base a la información suministrada por el flotante. En términos más formales, debe aprender los parámetros de las ecuaciones diferenciales que controlan la relación entre la apertura de la válvula y el nivel indicado por el flotante.

Otro ejemplo es el control de temperatura de un líquido a través de un calentador eléctrico. El controlador debe aprender a mantener constante la temperatura en base a la información recibida por un sensor (termómetro) determinando el tiempo que el calentador debe permanecer encendido o apagado. Esta relación puede ser particularmente compleja si el calentador requiere tiempo para comenzar a calentar efectivamente el líquido, o si sigue aún caliente luego de que es apagado.

La principal desventaja de estos métodos es que la solución obtenida no se puede adaptar a cambios en el sistema. Por ejemplo, si la válvula de control de entrada de líquido sufre un desgaste por uso, la relación entre su apertura y el ingreso de líquido cambia, por lo que un controlador estático no funcionará adecuadamente.

En los últimos años se han propuesto muchos algoritmos para generar controladores. Algunos de los más exitosos están basados en algoritmos genéticos, principalmente basados en una técnica conocida como computación evolucionaria [5, 4, 1]. A través de esta técnica se pueden evolucionar estructuras y obtener la más adecuada para resolver una tarea dada. En particular, en muchas situaciones la estructura evolucionada es una red neuronal.

Las principales ventajas del uso de las redes neuronales como controladores son [6]: (a) las redes neuronales son resistentes al ruido, un problema que está siempre presente en el ámbito de control; (b) sus primitivas son del más bajo nivel posible y (c) las redes neuronales pueden aprender a través de entrenamiento.

Sin embargo, el uso de técnicas evolucionarias no resuelve el problema de la adaptabilidad. Con el objeto de resolver este problema, se propone en este trabajo el almacenamiento de poblaciones intermedias obtenidas durante el proceso evolucionario. De esta manera, frente a cambios en el sistema bajo control, se puede iniciar una nueva evolución a partir de una población intermedia, con lo cual el tiempo necesario para obtener un nuevo controlador se reduce drásticamente. Esta estrategia se justifica dado que los controladores de poblaciones intermedias han sido seleccionados porque logran resolver relativamente bien el problema, aunque no completamente, y es en las poblaciones intermedias en las cuales existe diversidad genética, la cual se pierde en las poblaciones finales en las que ya se ha producido la convergencia.

## 2 Modelo de Red RBF

El modelo de red neuronal seleccionado es RBF. Esta red consta de tres capas [3, 2]. La primera capa está compuesta por unidades de entrada en un número igual a la cantidad de entradas del controlador. La segunda capa es una capa oculta cuyas unidades no lineales están conectadas directamente a todas las unidades de entrada. La capa de salida consiste de una unidad de salida por cada salida del controlador, conectada a todas las unidades de la capa oculta.

La función de activación de las unidades de la capa oculta es la Función de Base Radial (RBF) definida por:

$$y = \sum_{i=1}^n e^{-\frac{(x_i - c_i)^2}{\sigma^2}} \quad (1)$$

donde  $n$  es el número de entradas,  $x$  es el vector de entrada,  $c$  y  $\sigma$  son parámetros de cada unidad. Esta función tiene la forma de una campana de Gauss cuyo centro está representado por el vector  $c$  y su amplitud por  $\sigma$ .

La función de activación de las unidades de la capa de salida es una función lineal definida por:

$$z = \sum_{i=1}^m w_i * y_i \quad (2)$$

donde  $m$  es el número de unidades de la capa oculta,  $y_i$  es la salida de la  $i$ -ésima unidad de la capa oculta y  $w_i$  es el peso asociado a la conexión que une la  $i$ -ésima unidad de la capa oculta con esta unidad de salida.

Este modelo de red tiene tres propiedades importantes [3]: (a) es un aproximador universal dado que puede aproximar arbitrariamente bien cualquier función continua; (b) tiene la propiedad de mejor aproximación debido a que dada una función no lineal desconocida, siempre existe una elección de coeficientes que aproxima esta función mejor que todas las otras elecciones posibles y (c) provee una solución óptima dado que minimiza una función que mide cuanto se desvía la solución de su valor real.

### 3 Algoritmo Evolucionario

Los algoritmos evolucionarios son una técnica de búsqueda global en la que las soluciones posibles son codificadas como strings (usualmente llamados cromosomas). Cada solución posible se evalúa en el problema específico. A través del operador de crossover se pueden combinar strings y obtener strings con mejor fitness. El operador de mutación puede ser utilizado para aplicar pequeñas perturbaciones a los strings.

Si las redes neuronales se representan como strings, el algoritmo evoluciona poblaciones de redes neuronales, evaluándolas y asignándoles un valor de fitness. Las mejores redes neuronales se combinan y nuevas redes son creadas.

Este esquema es particularmente interesante, dado que la mayoría de los algoritmos para redes neuronales sólo buscan valores adecuados para un conjunto de parámetros (usualmente llamados pesos), pero sobre una estructura de red definida a priori.

La codificación que se propone es la siguiente: un individuo  $Ind$  de tamaño  $m$ , con  $m \geq 1$  se especifica formalmente como una secuencia (lista) de unidades RBF:

$$Ind = (R_1, R_2, \dots, R_m) \quad (3)$$

Cada unidad  $R_i$  se especifica formalmente como una tripla de la siguiente forma:

$$R_i = (c_i, \sigma_i, w_i)$$

donde  $c_i$  representa el centro de la unidad RBF,  $\sigma_i$  es la amplitud y  $w_i$  el peso asociado.

El algoritmo evolucionario puede obtener redes RBF arbitrarias dado que esta representación codifica tanto la estructura de la red como los pesos. Se usan operadores de crossover y mutación estándar para individuos de longitud variable.

## 4 Un ejemplo

Esta sección presenta un ejemplo en el que se utiliza el algoritmo evolucionario descrito anteriormente, para obtener un controlador para una función  $f$  y para una función  $g$  (que es  $f$  levemente modificada). Las funciones  $f$  y  $g$  se definen de la siguiente manera:

$$f(x, y) = \sin(2\pi x) + \sin(2\pi y) \quad 0 \leq x \leq 1, 0 \leq y \leq 1 \quad (4)$$

$$g(x, y) = \begin{cases} \sin(2\pi x) + \sin(2\pi y) & 0 \leq x \leq 0.8, 0 \leq y \leq 0.8 \\ (\sin(2\pi x) + \sin(2\pi y)) * 1.1 & 0.8 < x \leq 1, 0.8 < y \leq 1 \end{cases} \quad (5)$$

La figura 1 (izquierda) muestra el error con respecto al número de generaciones en una ejecución típica del algoritmo para la obtención de un controlador para la función  $f$ . Las poblaciones intermedias obtenidas en las generaciones 20 y 50 son almacenadas para su posterior uso como poblaciones iniciales para la evolución del controlador para la función  $g$ .

La figura 1 (derecha) muestra el error con respecto al número de generaciones para obtener un controlador para la función  $g$  partiendo de: (a) una población aleatoria, (b) la población intermedia obtenida en la generación 20 y (c) la población intermedia obtenida en la generación 50.

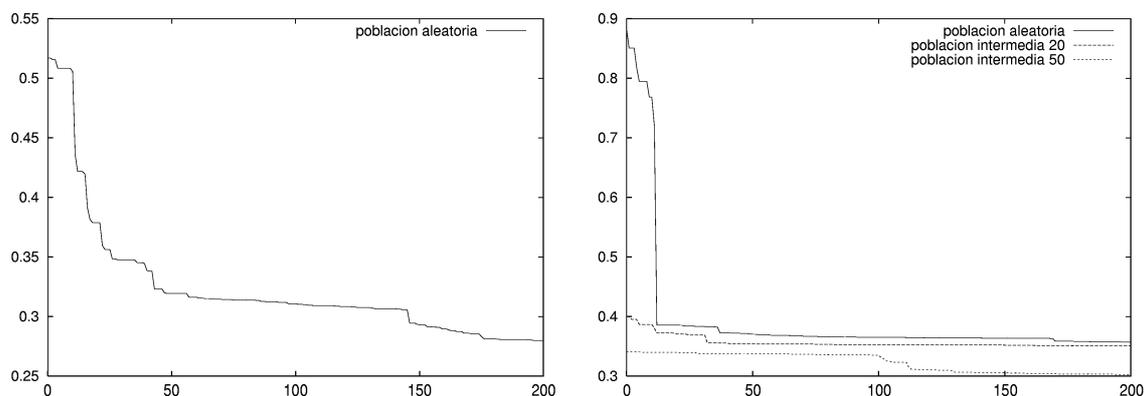


Figura 1: Error con respecto al número de generaciones para la función  $f$  (izquierda) y para la función  $g$  (derecha) partiendo de: (a) una población aleatoria, (b) una población intermedia obtenida en la generación 20 y (c) una población intermedia obtenida en la generación 50.

## 5 Conclusiones

El algoritmo evolucionario propuesto permite obtener controladores basados en redes neuronales. Para resolver el problema de cambios en los sistemas a ser controlados se propone el almacenamiento de poblaciones intermedias.

El gráfico de la derecha en la figura 1 muestra que el algoritmo evolucionario puede obtener más rápidamente un controlador, para una función levemente modificada, partiendo de poblaciones intermedias (obtenidas durante la evolución del controlador para la función original).

Estas poblaciones intermedias pueden ser almacenadas durante el proceso evolucionario sin incrementar la complejidad del algoritmo.

Actualmente se está trabajando en la aplicación de este concepto a problemas de control como los mencionados en la sección 1.

## Referencias

- [1] Wu Geng feng Carlos Kavka, María Liz Crespo and Fu Zhong quian. Fuzzy systems generation through symbiotic evolution. In *Symposium on Engineering of Intelligent Systems*, February 1998.
- [2] Emile Fiesler and Russel Beale, editors. *Handbook of Neural Computation*. Institute of Physics Publishing, 1997.
- [3] Simon Haykin. *Neural Networks, A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.
- [4] Risto Miikulainen and David Moriarty. Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, (22):11–33, 1996.
- [5] Risto Miikulainen and David Moriarty. Forming neural networks through efficient and adaptive coevolution. *Evolutionary Computation*, (5), 1998.
- [6] Stefano Nolfi. Evolving non-trivial behaviors on real robots: a garbage collecting robot. Technical Report 96-04, Institute of Psychology C.N.R. - Rome, April 1996.

## OPTIMIZACION EVOLUTIVA EN AMBIENTES DINAMICOS

Aragón V., Esquivel S.

LIDIC<sup>1</sup>

Facultad de Ciencias Físico, Matemática y Naturales

Universidad Nacional de San Luis

{aragonv,esquivel}@unsl.edu.ar

### Resumen

Los problemas del mundo real están, por lo general, asociados con grandes espacios de búsqueda, no lineales, funciones de evaluación multimodales y, en muchos casos, no existen métodos matemáticos para encontrar sus soluciones y, aun cuando existan, a menudo son computacionalmente intensivos para resultar prácticos. Su complejidad computacional aumenta tan rápidamente que cuando se trata de aplicarlos a problemas que no sean sencillos resultan prácticamente inútiles. Con la finalidad de transformarlos en útiles, por lo general, se aceptan simplificaciones para poder aplicar la evaluación de funciones lineales, con restricciones lineales, valores enteros u otros dispositivos matemáticos comunes, sacrificando la respuesta correcta para obtener una respuesta posible.

Actualmente la situación es peor porque en el proceso de resolución de problemas del mundo real no es suficiente con solucionar el problema. El problema debe ser solucionado, una y otra vez, en función de que el mundo está en permanente cambio.

Los métodos clásicos que generan soluciones para problemas combinatorios son matemáticos. Ellos computan solamente una solución y cuando las condiciones del problema cambian, la nueva mejor solución debe determinarse a partir de cero, es decir, el trabajo realizado hasta el momento, para resolver el problema, se pierde.

Por el contrario, los métodos que buscan soluciones, en vez de computarlas, parecen más convenientes para adaptarse a circunstancias cambiantes.

En este trabajo se presenta una línea de investigación en esta última dirección utilizando un enfoque evolutivo.

### 1. INTRODUCCION

Como se expresó anteriormente, la investigación presente parte del supuesto de que los métodos que buscan soluciones, más que calcularlas, a menudo parecen más aptos para trabajar en circunstancias cambiantes.

Cuando las condiciones de un problema cambian, en general, puede pensarse que es por dos razones: 1) cambia el landscape, o 2) cambian las restricciones de la región factible de las soluciones posibles. También puede suponerse que ambos cambios sean simultáneos [1].

Un cambio en el landscape se presenta cuando la función de evaluación se modifica, esto es totalmente posible si se considera que la definición de dicha función es algo subjetivo que expresa lo que se desea lograr ( y, por ende, lo que debe ser evitado). Las restricciones de la región factible pueden considerarse relacionadas con los recursos del problema y esta relación cambia acorde con la disponibilidad de los mismos, y también cuando cambian los propósitos o condiciones del problema, cambian el landscape y las restricciones.

Por otro lado no todos los cambios son pequeños o suaves y cuando cambios muy grandes se producen la similitud entre soluciones previamente encontradas y las nuevas pueden ser de poco o ningún valor. Independientemente de esta situación la computación evolutiva provee varias ventajas sobre otras heurísticas cuando trata de solucionar problemas no estacionarios.

<sup>1</sup> Laboratorio de Investigación y Desarrollo en Inteligencia Computacional, Director Dr. Raúl Gallard

La ventaja principal es que los algoritmos evolutivos mantienen una población de soluciones y, en consecuencia, permiten moverse de una solución a otra dentro de la población para determinar si alguna de ellas aun sigue siendo interesante y, ante el cambio, continuar la búsqueda desde soluciones ya encontradas y no desde cero.

Adicionalmente, a su habilidad para readaptarse a los ambientes dinámicos, los algoritmos evolutivos ofrecen el potencial para incorporar posibles cambios en las condiciones directamente en la función de evaluación de las soluciones. Si se puede limitar el posible efecto de los cambios y asignarle probabilidades razonables entonces se podrá optimizar soluciones respecto de diferentes criterios: 1) una esperanza pesada de efectos basada en sus probabilidades, 2) el efecto de máxima probabilidad, 3) el efecto del peor caso, y así siguiendo.

Por ejemplo, si se está diseñando una red de comunicaciones con muchos nodos conectados, se podrá evaluar cualquier solución candidata no sólo en términos de su funcionalidad, cuando la red funciona perfectamente, sino también cuando las conexiones fallan. La performance de una solución, bajo estas posibles fallas en el sistema, puede ser incorporada dentro del mérito general de las soluciones. Al final, la solución evolucionada habrá sido evaluada sobre muchas fallas posibles diferentes y se podrá confiar en que será capaz de responder apropiadamente [2].

## 2. PROPUESTA DE INVESTIGACION

El grupo ha comenzado a trabajar en estos tópicos a partir del año 2001, atacando inicialmente el primero de los problemas, es decir, los cambios que se producen a nivel de landscapes.

Para ello ha contado con un generador de problemas de test para ambientes no estacionarios facilitado por el Dr. De Jong, de la George Mason University [3], que propone un generador de funciones de prueba para algoritmos evolutivos que trabajen sobre cambios en la función de evaluación. Dicho generador permite introducir cambios en las alturas, en la forma y en la ubicación de los picos de la función de fitness y, además, controlar cómo se producen los cambios.

El generador trabaja con una morfología básica de landscape que denomina “campo de conos”, descripta, para el caso de dos dimensiones (pero puede escalar a un número mayor de ellas) por:

$$f(x,y) = \max_{i=1..N} [H_i - R_i * \text{RaizCuadrada}(x-x_i)^2 + (y - y_i)^2)], (x,y) \in [-1, 1]$$

Cada vez que el generador se invoca una morfología de este tipo se crea, generada aleatoriamente, en la cual valores random para cada cono se asignan tomando como base rangos definidos por el usuario:

$$H_i \in [H_{\text{base}}, H_{\text{base}} + H_{\text{rango}}]$$

$$R_i \in [R_{\text{base}}, R_{\text{base}} + R_{\text{rango}}]$$

El uso de esta función presenta varias ventajas como base para ambientes no estacionarios:

- habilidad para representar un rango amplio de landscapes complejos,
- la superficie contiene regiones no diferenciables,
- las características del landscape son identificadas paramétricamente,
- los valores de fitness pueden restringirse fácilmente a valores positivos o cualquier otro valor mínimo que se desee,
- la función es extendible a espacios de mayores dimensiones,

- la función ofrece tres características que pueden cambiarse: altura, ubicación y pendiente.

Los resultados obtenidos hasta el momento son muy promisorios. Algunos de los resultados obtenidos para dos y cinco dimensiones se muestran a continuación.

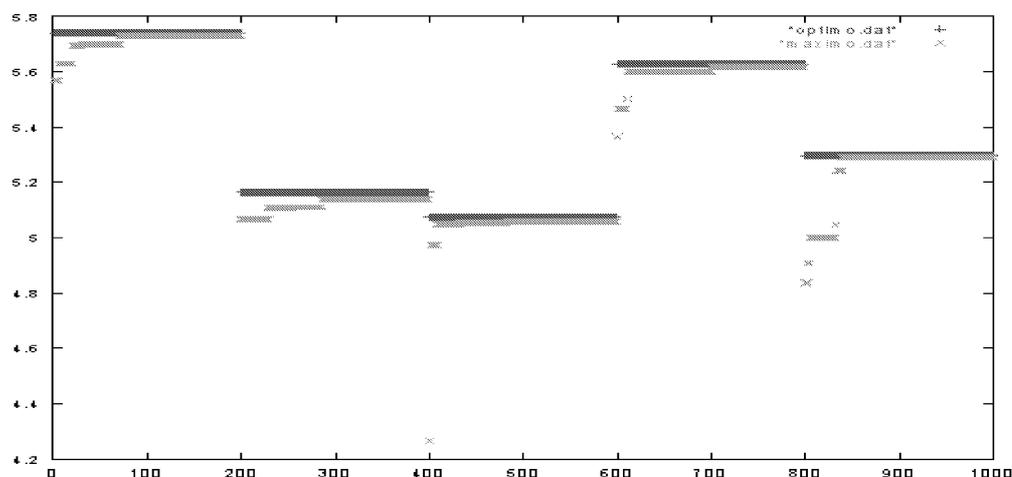
Tanto en el caso de dos como de cinco dimensiones los cambios realizados fueron en la altura y coordenadas (ubicación de los picos) en forma conjunta.

**Dos dimensiones:** los cambios se produjeron cada 200 generaciones. Todas las corridas evolucionaron 1000 generaciones. Las probabilidades utilizadas y los respectivos valores encontrados junto con las gráficas correspondientes fueron:

- Probabilidad de Crossover: 0.25
- Probabilidad de Mutación: 0.25
- Probabilidad de Transposición: 0.005
- Probabilidad de Transición: 0.01
- Probabilidad de Recrudescencia: 0.2
- Incremento de Probabilidad de Mutación: 0.4
- Incremento de Probabilidad de Crossover: 0.5
- Incremento de Probabilidad de Transposición: 0.009

Valor Optimo	Valor Máximo Obtenido	Error	Generación del Cambio	Generación del Valor Máximo
5.740429	5.731305	0.158951	0	71
5.165117	5.138240	0.520348	200	286
5.075247	5.060013	0.300174	400	484
5.628804	5.617080	0.208296	600	700
5.298064	5.292124	0.112109	800	840

Error Promedio de la Corrida : 0.259976



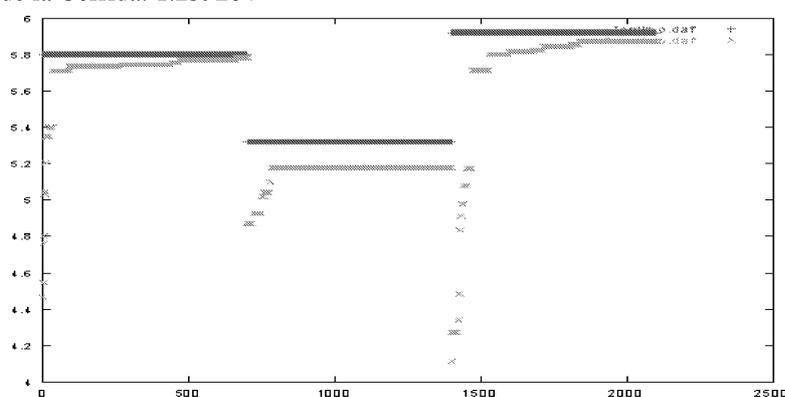
**Cinco Dimensiones:** Los cambios se produjeron cada 700 generaciones. La corrida evolucionó 2100 generaciones. Las probabilidades utilizadas y los respectivos valores encontrados junto con la gráfica correspondiente se detallan a continuación:

- Probabilidad de Crossover: 0.25

- Probabilidad de Mutación: 0.2
- Probabilidad de Transposición: 0.01
- Probabilidad de Transición: 0.01
- Probabilidad de Recrudescencia: 0.3
- Incremento de Probabilidad de Mutación: 0.6
- Incremento de Probabilidad de Crossover: 0.6
- Incremento de Probabilidad de Transposición: 0.05

Valor Optimo	Valor Máximo Obtenido	Error	Generación del Cambio	Generación del Valor Máximo
5.801404	5.785739	0.270012	0	661
5.321322	5.178497	2.684021	700	782
5.921322	5.876108	0.763579	1400	1847

Error Promedio de la Corrida: 1.239204



- Máximo Valor Encontrado
- Valor Optimo

Ante estos resultados alentadores, se propone continuar trabajando en esta dirección, perfeccionando el modelo de algoritmo evolutivo propuesto, escalando a un número mayor de dimensiones y trabajando en los otros tipos de cambios (restricciones y ambos simultáneamente).

### 3. REFERENCIAS BIBLIOGRÁFICAS

- [1] Michalewicz Z. and Fogel D.B. - "How to Solve It: Modern Heuristics", Springer, 2000.
- [2] Sebald, A.V and Fogel, D.B. - "Design of Fault Tolerant Neural Networks for Pattern Classification", Proceedings of the 2<sup>nd</sup> Annual Conference on Evolutionary Programming, Evolutionary Programming Society, La Jolla, CA, pp. 90 – 99, (1994).
- [3] Morrison R.W. and De Jong K.A. - "A Test Problem Generator for Non Stationary Environments", Congress on Evolutionary Computation, CEC99, Washington D.C. USA, July 1999, Vol. I, section Dynamic Fitness, pp. 2047 – 2053, IEEE Publisher Co. (1999).

# Aplicación de una técnica de visión a la representación de una superficie de terreno reconstruido a partir de contornos planares

**Cagnina Leticia, Guerrero Roberto**

Lab. de Investigación y Desarrollo en Inteligencia Computacional (LIDIC) \*

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950

5700 - San Luis

Argentina

e-mail: {lcagnina, rag}@unsl.edu.ar

## Abstract

Existe una diversidad de situaciones prácticas en las cuales la reconstrucción tridimensional de un objeto se realiza a partir de un stack de capas o rebanadas en las cuales el objeto fue descompuesto, denominadas contornos planares. Una aplicación concreta es la reconstrucción de la superficie de un terreno a partir de curvas de nivel. Dada la variedad de situaciones emergentes al momento de la reconstrucción, es necesaria la selección del método más adecuado a la mayoría de las situaciones. De la misma manera, una adecuada visualización en 3D depende de una buena percepción de la profundidad y perspectiva. La presente propuesta tiene como objetivo el análisis de diferentes métodos de reconstrucción de un terreno a partir de curvas de nivel y su posterior visualización en 3D mediante la técnica de estereografía. En la misma se estudiarán diferentes combinaciones de métodos de reconstrucción con la técnica estereográfica que permitan aumentar la información transmitida.

## 1 Introducción

La generación de imágenes ha evolucionado ampliamente en los últimos años, particularmente en áreas tales como el tratamiento de información de tipo geológico. Los datos geológicos pueden ser utilizados para el análisis, diseño y evaluación de recursos naturales, planeamientos regionales, etc.. Los datos superficiales de un terreno frecuentemente constituyen la información más relevante con la que se cuenta, por consiguiente, el estudio de la superficie del terreno y la inferencia de información que se pueda realizar a partir de ella juegan un rol fundamental.

---

\*El laboratorio es dirigido por el Dr. Raúl Gallard y subvencionado por la Univ. Nac. de San Luis y la ANPCyT (Agencia Nac. para la Prom. de la Ciencia y la Tecnología)

El desarrollo y análisis de la información geológica está comúnmente sustentado en diversos tipos de representaciones gráficas que muestran conceptualmente un significado geológico particular. Se puede definir un Modelo Digital de Terreno (MDT) como una representación digital de la variación continua del relieve en el espacio. Los MDT son fundamentales para el cálculo de las variaciones del terreno, para la evaluación del mismo en áreas montañosas, y la generación de visualizaciones que permitan tener una idea completa del relieve de la zona en estudio [4].

En la cartografía convencional la descripción de las elevaciones del terreno a través del mapa topográfico constituye la infraestructura básica de los mapas. El papel equivalente en los MDT lo desempeña el Modelo Digital de Elevaciones (MDE), que describe la altimetría de una zona mediante un conjunto de cotas. Estas entidades describen áreas o polígonos donde la localización de un elemento está representada por una cadena de coordenadas espaciales  $(x,y,z)$ . Los datos poligonales entonces están asociados con áreas superficiales sobre un espacio definido.

Siguiendo la analogía cartográfica, es posible construir un conjunto de modelos derivados, elaborados a partir de la información contenida explícita o implícitamente en el MDE. Los modelos derivados más sencillos pueden construirse exclusivamente con la información del MDE y reflejan características morfológicas simples (pendiente, orientación, etc.) [9]. Uno de ellos es el método de curvas de nivel. El mismo consiste en registrar una sucesión de contornos planares que representan la intersección de la superficie del terreno con una serie de planos horizontales que la cortan.

El proceso de reconstrucción consistirá de conectar contornos adyacentes por medio de una malla de triángulos [12] [11]. No obstante, y dependiendo de las características del terreno analizado pueden surgir una variedad de subproblemas (el problema de seccionado, de la correspondencia, etc.) que dificulten la formulación de un único método automático general a todas las situaciones. Existen actualmente diferentes métodos de reconstrucción de objetos a partir de datos desorganizados que solucionan un subconjunto de dichos problemas pero no todos al mismo tiempo [8] [5] [3]. En consecuencia, al momento de la visualización de la superficie, ésta reflejará en mayor o menor medida las características morfológicas del terreno dependiendo de los subproblemas específicos solucionados.

Bajo este contexto, una técnica de visualización debe crear representaciones gráficas de datos críticos, difíciles de interpretar a partir de mediciones, que destacan relaciones entre los datos que de otra manera pasarían desapercibidos para el usuario y, más aún, que resalten aquellas áreas donde el análisis de los datos debe ser re-evaluado [1] [14].

La estereografía es un método para la visualización de la información en 3D que intenta mejorar la percepción de la profundidad y perspectiva. Si bien éste es adecuado para la representación de datos complejos, se caracteriza por la dificultad en la generación de las imágenes y el costo inherente en la interpretación por parte de un usuario común. No obstante, existen diferentes técnicas que intentan reproducir el efecto visual de la estereografía mediante el uso de simulaciones [13].

## 2 Métodos de Reconstrucción de Superficies

Existen en la bibliografía numerosos algoritmos de reconstrucción de superficies [5] [8] [3] [6]. Todos ellos se basan en procesar un conjunto cualquiera de puntos  $\{x_1, \dots, x_n\} \subset R^3$  que

representan una copia de un objeto desconocido  $M$  y producen como salida una superficie simple que aproxima a  $M$ . Se asume no poseer ningún tipo de conocimiento respecto de la topología, la presencia de límites o la geometría de  $M$ ; todo ello debe ser inferido automáticamente a partir de los datos. La mayoría de los algoritmos desarrollan sus habilidades basándose en un análisis detallado de cada uno de los casos de prueba con el objeto de explotar la estructura parcial de los datos. Y es precisamente el conocimiento previo de la existencia de una estructura entre los puntos de datos el que conlleva a dejar de lado la consideración de situaciones particulares. Para el caso en particular considerado los datos poseen una estructura previa definida por contornos planares (polígonos cerrados) en planos paralelos. Una solución temprana de reconstrucción llevaría a pensar en realizar una triangulación por capas, lo cual produce como consecuencia un error en el tratamiento de quiebres del relieve (quebradas) o más formalmente el problema de la bifurcación de capas cuando se analizan capas adyacentes [12]. Es prioritario entonces la evaluación de los posibles problemas que se pueden presentar y la determinación de prioridades de reconstrucción acorde con necesidades particulares, para luego establecer el método más correcto.

### 3 La Técnica Estereográfica

La mayoría de las veces una fotografía captura información acerca del mundo como si éste fuera visto desde un único punto de vista. La lente de una cámara de fotos estática registra una escena tal cual ésta sucede en un instante preciso de tiempo, mientras que una filmadora captura una secuencia de imágenes que varían en el tiempo. En la vida real, nuestros ojos, debido a que están colocados con una cierta distancia de separación, no ven exactamente la misma vista. Los ojos perciben dos vistas similares pero no iguales de la misma escena y luego es el cerebro quien las compone. En el proceso de composición el cerebro recupera la profundidad de la imagen observada por medio de una triangulación. A este proceso se le denomina visión Estereoscópica.

Los algoritmos de computación gráfica se han concentrado mayormente en producir imágenes desde un único punto de vista, análogas a fotos estáticas registradas fotográficamente. Es decir, una gran parte de ellos modelan cámaras de un único ojo [1]. Nuestra propia experiencia como organismos de dos ojos confirma el beneficio del uso de la coherencia de perspectiva en escenas típicas. Surge entonces la propuesta de explotar la coherencia espacial y temporal de una escena a modo de beneficiar la visualización de imágenes fijas y más aún, de animaciones. La explotación de la coherencia espacial de dos imágenes permite la generación de una tercer imagen, denominada imagen Ciclopeana que permite crear una ilusión de profundidad, es decir, de tercera dimensión. A este proceso se le denomina visión Estereográfica.

Existen numerosos algoritmos para la generación de imágenes estereográficas que explotan ya sea la coherencia espacial o temporal de diferentes maneras [10] [13] [2] [7]. Algunos de ellos son computacionalmente costosos de implementar, otros reducen la calidad de las imágenes resultantes.

## 4 Propuesta en desarrollo

El presente trabajo pretende desarrollar una aplicación para la visualización de mapas geográficos en 3D mediante la simulación de la técnica de estereografía. Para ello el proceso se divide en dos partes: una primer etapa de reconstrucción del terreno mediante el uso de diferentes técnicas de reconstrucción, y posteriormente la visualización del modelo reconstruido en 3D mediante la simulación de la técnica estereográfica.

La primer etapa se basa en datos reales relevados en el terreno que constituyen una serie de polígonos cerrados determinando contornos planares del mismo, denominados curvas de nivel. El proceso de reconstrucción consistirá del análisis y testeo de diferentes técnicas que permitan recuperar en un alto porcentaje el aspecto real del relieve del terreno.

En la segunda etapa los modelos reconstruidos a través de las diferentes técnicas serán testeados mediante una simulación de la técnica estereográfica. Dicha simulación tiene por objeto brindar al modelo una visión en 3D sin tener que recurrir al uso de costosa tecnología ni grandes procesos computacionales. La recreación de la tercera dimensión se realizará mediante el uso de la profundidad cromática, es decir la utilización de color como inductivo de la profundidad.

## References

- [1] Watt A. and Policarpo F. *The Computer Image*. Addison-Wesley, 1998.
- [2] Stephen J. Adelson, Jeffrey B. Bentley, In Seok Chong, Larry F. Hodges, and Joseph Winograd. Simultaneous generation of stereoscopic views. In *Computers Graphics Forum*, pages 3–10, May 1991.
- [3] Fausto Bernardini, Chandrajit L. Bajaj, Jindong Chen, and Daniel R. Schifore. A triangulation-based object reconstruction method. In *6th Annual Video Review of Computational Geometry*, 1998. Proceeding of the 13th Annual ACM Symposium on Computational Geometry.
- [4] Tomlin D. *Geographic Information System and Cartographic Modeling*. Prentice-Hall, 1990.
- [5] Tamal K. Dey and Naven Leekha. *Surface Reconstruction Simplified*. Manuscrito, December 1999.
- [6] Bartels R. H., Beatty John C., and Barsky Brian A. *An Introduction to the Use of Splines in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Palo Alto, CA, 1987.
- [7] Michel Hall. Holographic stereograms as discrete imaging systems. In *Proc. SPIE on Practical Holography VIII*, Feb 1994.
- [8] Huges Hope, Tony de Rose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Computer Graphics*, pages 71–78. ACM, July 1992.

- [9] Wood J. and Fisher P. Assessing interpolation accuracy in elevation models. In *Proc. IEEE Computer Graphics and Applications*, volume 13, pages 48–56, March 1993.
- [10] Shaun Love and David F. McAllister. Computer generated lenticular stereograms. In *Three-dimensional visualization and display technologies, SPIE Proceeding*, page 102, 1989.
- [11] De Berg M., Van Kreveld M., Overmars M., and Schwarzkopf O. *Computational Geometry*. Springer-Verlag Berlin Heidelberg, 1997.
- [12] David Meyers. *Reconstruction of Surfaces From Planar Contours*. Dissertation for Doctor of Philosophy, University of Washington, 1994.
- [13] A. Michel Noll. Stereographic projections by digital computer. In *Computers in Automation*, page 32, May 1965.
- [14] T. Strothotte, B. Preim, J. Schuman A. Raab, and D. Forsey. How to render frames and influence people. *Computer Graphics Forum*, 13:455–466, Sept. 1994.

## INTRODUCIENDO MEJORAS EN LOS ALGORITMOS EVOLUTIVOS

ESQUIVEL S. C., GALLARD R.H.

<sup>1</sup>Laboratorio de Investigación y Desarrollo en Inteligencia Computacional,  
Universidad Nacional de San Luis, Argentina  
{esquivel, rgallard}@unsl.edu.ar

### RESUMEN

Hoy día se conocen como algoritmos evolutivos (AEs), a una familia de algoritmos basados en los principios de reproducción, variación aleatoria, competencia y selección de individuos. Una de las clases más exitosas la constituyen aquellos basados en lo que originalmente se conoció como algoritmos genéticos. Estos tienen en particular el uso de operadores genéticos (crossover y mutación) no siempre presentes en otras clases de algoritmos evolutivos. Esta presentación muestra variantes introducidas en esta clase de AEs para mejorar su habilidad en problemas de optimización.

### INTRODUCCIÓN:

En los últimos cinco años se han producido avances en las técnicas de la Computación Evolutiva. Estos avances tienden esencialmente a mejorar la calidad de las soluciones provistas por un algoritmo evolutivo y se han logrado a través de la implementación del paralelismo, la creación de nuevos métodos de recombinación, la hibridización y la inserción de conocimiento específico del problema en la representación de las soluciones y/o en el proceso evolutivo. Esta presentación discute cada uno de estos enfoques y los avances logrados en algoritmos evolutivos basados en algoritmos genéticos.

### PARALELISMO

Las implementaciones paralelas de los algoritmos evolutivos tienen como objetivo mejorar su performance. Una forma naturalmente distribuida de implementar algoritmos evolutivos paralelos es a través del modelo *isla*. Aquí se produce la migración de individuos entre poblaciones que evolucionan independientemente, favoreciendo de este modo el proceso evolutivo. Este intercambio se efectúa escogiendo individuos desde una población fuente y exportándolos a poblaciones receptoras. De esta forma se mejora la diversidad genética. A la llegada del individuo, es usual que éste se acepte sin ejercer política alguna sobre su aceptación. Esta falta de discriminación de individuos provenientes de poblaciones externas puede, en muchos casos, producir convergencia prematura (un super-individuo puede arrastrar hacia sí a los demás individuos de la población por selección repetida para el apareamiento).

En trabajos previos [4,8] hemos implementado tres estrategias de control de migración. La primera *Maximum Gap Allowed (MGA)*, intenta prevenir la propagación desbalanceada de genotipos utilizando un umbral de aceptación de los cromosomas inmigrantes. La segunda, *Dynamic Arbiter Strategy (DAS)*, permite la evolución independiente de las poblaciones y actúa cuando se detecta un posible estancamiento en la evolución. En esta situación se evade el óptimo local insertando un individuo de esperada disímil composición genética para proveer diversidad en la población. En este caso el sistema intercambia información asociada con el mejor y peor individuos globales y la media poblacional. La tercera alternativa, *Combined MGA-DAS Strategy (CMGA-DAS)*, combina ambas estrategias. Los ensayos preliminares se hicieron utilizando un algoritmo genético simple optimizando aquellas funciones de testing (uni y multimodales) que resultaron más dificultosas, utilizando nodos reales y

virtuales. Posteriormente los algoritmos evolutivos paralelos con 4 a 10 islas se implementaron para resolver el problema del flow shop [5].

## RECOMBINACIÓN

En los AEs basados en AGs, la recombinación es el mecanismo básico para la generación de nuevos individuos. Esto se logra a través del intercambio genético de los padres. Los métodos convencionales SCPC (Single Crossover Per Couple), aplican el crossover sólo una vez sobre un par de padres generando dos hijos que conformarán la nueva población. En 1995 (1er CACIC), en un trabajo realizado conjuntamente con Michalewicz [3], se presenta por vez primera MCPC (Multiple Crossovers Per Couple), un método de aplicación repetitiva del crossover sobre un par de padres. Aquí los  $2n_1$  (siendo  $n_1$  el número de crossovers) hijos generados se insertaban en la nueva población. Los primeros resultados mostraban una mejor performance que la de SCPC si  $n_1$  se mantenía entre 2 y 4. En otro caso existía el riesgo de convergencia prematura, originada en la presión selectiva emanada del método de selección (proportional selection) y la superexplotación de las características de buenos padres (alto fitness).

Para solucionar el problema, se implementaron AEs con MCPC, utilizando mecanismos alternativos de selección (ranking selection, dynamic ranking selection, fitness proportional couple selection). Todos estos métodos aunque más costosos mejoraron sensiblemente la performance de MCPC. Implementaciones actuales de MCPC, seleccionan uno, algunos pero no todos los hijos provenientes de una misma pareja para su inserción en la nueva generación. La elección del hijo a insertar se hace según algún criterio preestablecido. Todas ellas resultan en un menor esfuerzo computacional y no muestran convergencia prematura.

Una última familia de métodos recombinativos se conoce como MCMP (Multiple Crossovers on Multiple Parents). Estos métodos agregan a la visión de padres múltiples de Eiben [1,2], la de la repetitividad de crossovers. De esta forma se logra un balance entre exploración aportada por la multiplicidad de padres (una muestra mayor del espacio de búsqueda) y la explotación repetitiva de las buenas características de esos padres, evitándose la convergencia prematura. MCMP, con sus nuevas variantes (uso del semental e inmigrantes aleatorios, Fig. 1) ha provisto muy buenos resultados en problemas de optimización de simple y múltiples objetivos en el área de scheduling [6, 7]. Aquí, el proceso de creación de nuevos hijos es el siguiente. Desde la vieja población se escoge un individuo como semental aplicando selección proporcional. El número  $n_2$  de padres en el buffer de apareamiento se completa con individuos aleatoriamente generados (inmigrantes random). El semental se aparea con cada uno de los padres restantes y las parejas son sometidas a crossover. El mejor de los  $2n_2$  hijos generados se inserta temporariamente en una estructura transitoria. La operación de crossover se realiza  $n_1$  veces con un punto distinto de corte cada vez, hasta completar la estructura transitoria. Los hijos no son sometidos a mutación, pues los inmigrantes proveen alta diversidad genética. Finalmente el mejor de los hijos en la estructura transitoria se inserta en la nueva población.

Una característica valiosa de los métodos de multirecombinación es la agrupación de los valores de fitness de los individuos de la población final alrededor del fitness del mejor individuo. O sea, los algoritmos proveen soluciones alternativas distintas de muy cercano valor objetivo, las que pueden ser utilizadas en caso que las condiciones del sistema varíen dinámicamente.

## HIBRIDIZACIÓN E INSERCIÓN DE CONOCIMIENTO ESPECÍFICO DEL PROBLEMA

Las técnicas de hibridización combinan, en el proceso de búsqueda, el accionar de distintas heurísticas con el algoritmo evolutivo. En nuestro caso hemos estudiado el comportamiento de AEs hibridizados en distintas etapas del proceso evolutivo. Esto va desde la inserción de semillas (buenos individuos

provistos por otras heurísticas) en la población inicial o durante la evolución o alternativamente mejorar las soluciones provistas por el algoritmo evolutivo al iniciarse el proceso, durante el mismo o a su finalización. Dado que las heurísticas propias de un problema se basan en ciertas características del mismo, estamos insertando un conocimiento que es dependiente del problema y que el AE intrínsecamente ignora (la búsqueda se basa sólo en el fitness de una solución).

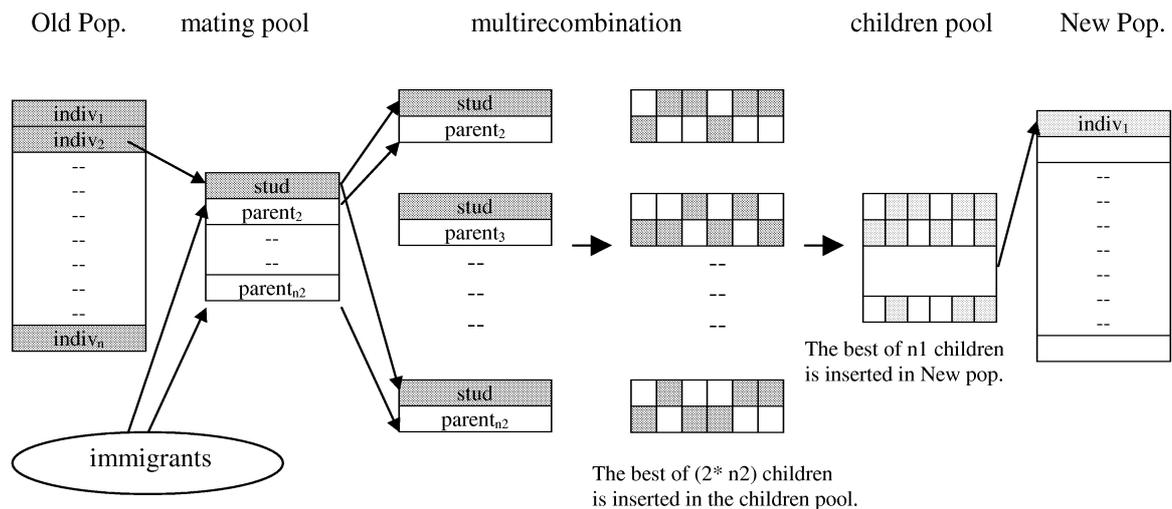


Fig. 1. The stud and random immigrants multirecombination process.

Otra forma de insertar conocimiento específico del problema es conociendo de antemano ciertas condiciones de optimalidad. Lamentablemente en la mayoría de los problemas complejos del mundo real esta información no se encuentra disponible. Pero en los casos como el de earliness/tardiness de scheduling de máquina única, donde una solución óptima tiene una forma V de ordenamiento, puede reducirse el esfuerzo de búsqueda limitándonos a buscar soluciones que cumplan tal propiedad. Como consecuencia de ello, métodos de multirecombinación con inserción de conocimiento específico han producido nuevos upper-bounds para tamaños considerables del problema en cuestión, para instancias propuestas de la OR-Library [9].

En caso de no conocer condiciones de optimalidad, es recomendable insertar repetitivamente durante la evolución, individuos generados por varias heurísticas en el pool de padres para forzar la recombinación de su material genético con el de padres provenientes de la evolución y de otros aleatoriamente generados. Esto ha producido muy buenos resultados en problemas de weighted tardiness de máquina única y en flow shop.

Por último, dada la naturaleza autoadaptativa de los AEs, podemos explotar esta propiedad y conservar las formas (subsecuencias del cromosoma) comunes en los padres diseñando operadores genéticos que las preserven. Aquí el conocimiento específico del problema es detectado por el AE.

## CONCLUSIONES Y TRABAJOS FUTUROS

Cuando los AEs se utilizan como algoritmos para la búsqueda de soluciones óptimas en distintos problemas, debe tenerse en cuenta que su oportunidad de supervivencia está basada, sólo en la bondad relativa de la solución. Otros factores como la naturaleza del espacio de soluciones, o su vecindario no son tenidos en cuenta. En consecuencia, dependiendo del seteo de los parámetros del algoritmo

(tamaño de población, probabilidad de operadores, límite para el proceso de búsqueda, etc) el algoritmo podrá converger a un óptimo local o global.

Este comportamiento (convergencia prematura o no) será consecuencia del balance entre la exploración y explotación en el espacio de soluciones. Las técnicas aquí enumeradas tienden a proveer una mejora en ese balance, con la consecuente mejora en la calidad de soluciones.

Los trabajos actuales y futuros, estudian nuevas estrategias para la inserción de conocimiento a través del uso de semillas, preservación de formas, multirecombinación y heterosis. Este último enfoque sugiere el cruzamiento de distintas líneas reproductivas. Implica evolucionar endogámicamente varias poblaciones y luego mezclarlas.

## REFERENCIAS

- [1] Eiben A.E., Raué P-E., and Ruttkay Zs., *Genetic algorithms with multi-parent recombination*. In Davidor, H.-P. Schwefel, and R. Männer, editors, *Proceedings of the 3rd Conference on Parallel Problem Solving from Nature*, number 866 in LNCS, pages 78-87. Springer-Verlag, 1994
- [2] Eiben A.E. and. Bäck Th. An empirical investigation of multi-parent recombination operators in evolution strategies. *Evolutionary Computation*, 5(3):347-365, 1997.
- [3] Esquivel S., Gallard R., Michalewicz Z., - "MCPC: Another Approach to Crossover in Genetic Algorithms" *Proceedings del Primer Congreso Argentino de Cs. de la Computación*, pp 141-150, Universidad Nacional del Sur, Octubre 1995.
- [4] Esquivel S., Leguizamon G., Ochoa C., Gallard R. - "An Overview Of A Symbiotic Coalition: Evolutionary Algorithms And Distributed Systems"- *Intelligent Information Systems*, Bahamas, pp 550-556, IEEE press, ISBN 0-8186-8218-3, Diciembre 1997.
- [5] Esquivel, S., Printista M., Zuppa F. y Gallard R. - "Paralell and Sequential Evolutionary Algorithms for the Flow Shop Scheduling Problem", expuesto y publicado en los Anales del VI Congreso Argentino de Ciencias de la Computación, CACIC, Universidad Nacional de la Patagonia San Juan Bosco, 2-7 de Octubre de 2000, pag. 1311.
- [6] Esquivel S., Gatica C., Gallard R. *Conventional and Multirecombinative Evolutionary Algorithms for the Parallel Task Scheduling Problem*. *Lecture Notes in Computer Science*, LNCS 2037, pp 223-232, Springer Verlag 2001
- [7] Esquivel S., Ferrero S., Gallard R., Salto C., Alfonso H. Schütz M., "Enhanced evolutionary algorithms for single and multiobjective optimization in the job shop scheduling problem". *Journal of Knowledge Based Systems*, Vol 15/1-2, pp 13-25, Elsevier, January 2002.
- [8] Ochoa C., Gallard R. -"Controlling Asynchronous Migration in Parallel Genetic Algorithms"- Presentado y publicado en los *Proceedings del International Symposium on Engineering of Intelligent Systems*, University of La Laguna, Tenerife, España, Vol. 1, pp 242-248, ISBN 3-906454-12-6, Febrero 1998.
- [9] 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" . Aceptado para publicación en *Cybernetics and Systems del Taylor and Francis Journal*, a aparecer (U.K.) en abril 2002.

## ENFOQUES HEURISTICOS PARA PROBLEMAS DE SCHEDULING ESTATICOS Y DINAMICOS

Esquivel S.C, Gallard R., Gatica C., Ferrero S., Leguizamón G., Zuppa F.

LIDIC<sup>1</sup>

Facultad de Ciencias Físicas, Matemáticas y Naturales

Universidad Nacional de San Luis

{esquivel, rgallard, gaticac, swf, legui, fede}@unsl.edu.ar

### Resumen

El presente trabajo describe una propuesta de investigación para atacar diferentes problemas de scheduling. Dicha tarea es realizada por investigadores de distintas líneas del LIDIC, y en consecuencia, con técnicas diferentes las cuales pueden usarse solas y/o combinadas.

En general, dentro del ámbito de scheduling, los modelos más estudiados fueron los modelos conocidos como clásicos o estáticos, es decir, donde las actividades, los recursos, los tiempos de procesamiento están predefinidos, no se modifican durante el proceso, y con un objetivo involucrando la minimización del tiempo de finalización y los costos de operación. Pero en los problemas del mundo real existen otra serie de decisiones que interactúan con el modelo clásico, si algunas de estas decisiones se adicionan a dicho modelo se obtienen los modelos de scheduling dinámicos.

Debido a la naturaleza de estos problemas, que pertenecen a la clase de problemas NP-duros, las tendencias más modernas tratan de hallar heurísticas con estrategias de explotación/exploración mixtas. Dentro de ellas cabe mencionar Tabu Search, Simulating Annealing, Colonia de Hormigas y Algoritmos Evolutivos.

### 1. INTRODUCCION

Un proceso de scheduling implica seleccionar y secuenciar actividades tal que ellas cumplan uno o más objetivos y satisfagan un conjunto de restricciones del dominio del problema. Durante este proceso se deberá seleccionar entre schedules (planes o planificaciones) alternativos y asignar recursos y tiempos a cada actividad de manera tal que dichas asignaciones respeten las restricciones temporales de las actividades (jobs) y las capacidades limitadas de un conjunto de recursos compartidos, de manera que ciertas funciones objetivo (por ejemplo tardiness, makespan, etc.) sean minimizadas. Los problemas que pertenecen a esta categoría son conocidos como pertenecientes a la clase de los problemas NP-duros.

En general, dentro del ámbito de scheduling, los modelos más estudiados fueron los modelos estáticos, es decir, donde las actividades, los recursos, los tiempos de procesamiento están predefinidos, no se modifican durante el proceso, y con un objetivo involucrando la minimización del tiempo de finalización y los costos de operación.

En los problemas del mundo real existen otra serie de decisiones que interactúan con el modelo clásico, si algunas de estas decisiones se adicionan a dicho modelo se obtienen los modelos de scheduling dinámicos. Por ejemplo, algunas de estas decisiones pueden ser cambiar la cantidad y/o configuración interna de los recursos mientras el proceso de scheduling está en avance para balancear los cambios en la carga de los jobs que arriban al sistema, o bien, puede aumentar o disminuir la cantidad de operarios en el sistema, en distintos momentos del día, o uno o más recursos pueden deshabilitarse temporariamente por razones de falla o mantenimiento.

<sup>1</sup> Laboratorio de Investigación y Desarrollo en Inteligencia Computacional, Director Dr. Raúl Gallard

Los modelos de este tipo implican una reconfiguración interna dinámica del proceso de scheduling para adaptarlo a la nueva situación del contexto. También pueden existir causas externas que necesiten una reconfiguración, por ejemplo cambio, por parte de los clientes, en las fechas de entrega en función de sus stocks y demandas[1].

Para encarar los problemas de scheduling históricamente se usaron diferentes enfoques, entre ellos cabe mencionar:

**Simulación por Computadoras:** Entre los años 50 y 60, con la presencia de las computadoras se hizo posible representar el sistema a planificar por medio de modelos que permitían reflejar su estructura, los jobs y sus operaciones, los recursos y una serie de restricciones con algún nivel de detalle tal que, dados los datos de entrada y reglas de heurísticas simples (dispatching), en determinados puntos de decisión, el programa de simulación podía extrapolar un schedule a un costo relativamente bajo. La literatura al respecto es relativamente amplia puede recomendarse especialmente [2], [3], [4], [5], [6] y [7]. Si bien la simulación puede representar modelos muy realísticos a un costo computacional modesto tiene la desventaja de que los resultados obtenidos no son ni siquiera aproximados al óptimo y, en consecuencia, no resultan en buenos benchmarks para establecer cuán buenas son las soluciones encontradas o cómo encontrar otras más eficientes.

**Modelos Matemáticos:** En la década del 60 diferentes investigadores [8], [9] desarrollaron las técnicas de programación entera, aprovechando la mayor potencia computacional ya disponible en ese tiempo, que permite que problemas, más realistas fuesen formulados de manera que teóricamente permita su resolución exacta. Esto dio lugar a una serie de trabajos siguiendo esta dirección [10], [11], [12], [13] y [14].

También otra técnica ampliamente utilizada es la de branch-and-bound y su variante branch-and-prune. En los años 70 Srinivasen [15] y otros aplicaron la técnica de la programación dinámica para resolver problemas de secuenciamiento usando todos los subconjuntos posibles de jobs como elementos del espacio de estados.

Tanto en el caso de la programación entera como en el de la programación lineal mientras problemas muy pequeños pueden resolverse encontrando la solución óptima, esto es prioridades perfectas, ruteo y demás, los problemas complejos (de tamaño grande) permanecen intratables y, es de esperar que los problemas del mundo real permanezcan así, puesto que ellos pertenecen a la clase de los problemas NP-duros, lo cual expresado muy simplemente significa que no se conoce algoritmo que no crezca en tiempo exponencial acorde con el tamaño del problema [16].

**Heurísticas:** Las razones explicadas, en el párrafo anterior, dieron lugar al desarrollo de diferentes heurísticas para resolver, no ya de manera exacta sino aproximada, tales tipos de problemas.

Estas heurísticas pueden clasificarse dentro de las categorías de explorativas, explotativas o explorativas/explotativas.

Dentro de las heurísticas explotativas (llamadas así porque inspeccionan el espacio de búsqueda en el vecindario de una solución posible) las búsquedas en el vecindario son una de las técnicas usadas en este tipo de problemas [17]. Esta técnica es un ejemplo particular del método de programación no lineal conocido como hill-climbing.

Otra técnica, Lagrangian Relaxation [18] soluciona problemas de programación lineal simples a través de la eliminación de algunas restricciones y pagando penalidades proporcionales a las cantidades en que dichas restricciones se violan. Este método de relajación es semejante a la heurística dinámica de cuello de botella [19] que si bien son, a menudo, poderosas son a la vez complejas de usar y no son aun métodos de propósito general.

Dentro de las heurísticas explorativas que pueden ser pensadas que actúan, en determinado momento del proceso al no avanzar hacia nuevas soluciones, diciendo “es tiempo de un cambio” [20] lo cual se traduce en realizar un salto a otro punto del espacio de búsqueda (no del vecindario). Una de las primeras en ser utilizadas fue la conocida como muestra random.

Las tendencias más modernas tratan de hallar heurísticas con estrategias de explotación/exploración mixtas. Dentro de ellas cabe mencionar Tabu Search, Simulating Annealing, Colonia de Hormigas y Algoritmos Evolutivos.

Dentro de las heurísticas Evolutivas y de Colonia de Hormigas es que encuadra el desarrollo la presente investigación. Es importante destacar que los distintos problemas han sido estudiado por diferentes investigadores y desde diferentes disciplinas, especialmente en el modelo clásico o estático. Se han desarrollado diferentes algoritmos evolutivos abarcando distintos aspectos: representaciones directas vs. indirectas, enfoques puros vs. híbridos, algoritmos seriales vs. paralelos, operadores genéticos generales vs. específicos, y así siguiendo [22], [23], [24], [25], [26] y [27]]. Mientras que la heurística de la Colonia de Hormigas ha sido aplicada en pocos casos a problemas de scheduling. Dorigo y otros en [28,29] y Zwaan y otros[30] reportaron interesantes resultados para el problema de Job Shop Scheduling. Stützle en [31], desarrolló un algoritmo eficiente para resolver el problema de Flow Shop a través de la incorporación de búsqueda local. Otras aplicaciones incluyen diferentes algoritmos de Colonia de Hormigas para el problema de Single Machine Total Tardiness. En estas aplicaciones esta heurística es mejorada a través de formas alternativas de evaluación de feromona para guiar la búsqueda de las soluciones.

Con respecto al problema Scheduling Dinámico pocos investigadores lo han encarado usando algoritmos evolutivos. Dentro de los resultados publicados recientemente cabe mencionar [32], [33], [34]. Mientras que con respecto a la heurística de la Colonia de Hormigas el grupo no tiene conocimiento de que haya sido aplicada en ambientes no estacionarios.

El grupo tiene en su background un importante trabajo realizado sobre este tipo de problemas, esencialmente para los casos de Job Shop, Flow Shop y Máquinas Paralelas para el caso estático, atancado aspectos relacionados con la representación y multirecombinación que ha permitido obtener resultados interesantes. También se ha iniciado el estudio de esta heurística aplicada al problema de Flow Shop e hibridizadas con heurísticas greedy a fin de mejorar su performance [35].

## 2. PROPUESTA ACTUAL DE INVESTIGACION

La investigación actual se divide en las siguientes líneas:

1. **Job Shop:** Por un lado, se está trabajando en la búsqueda de soluciones dirigidas a determinadas regiones del espacio de búsqueda de todos los posibles schedules[36] usando un enfoque hibridizado de colonias de hormigas. Por otro lado, a partir de la experiencia obtenida [37], [38], [39] y [40], en el caso estático, se ha comenzado a encarar el diseño de heurísticas evolutivas alternativas para el problema del job shop dinámico adaptando las técnicas evolutivas aplicables a problemas generales de ambientes dinámicos, para ello se usarán distintas funciones objetivo y las características del problema inicialmente se restringen a: los jobs tienen distintos tiempos de arribo, cada job tiene asociado un peso y una fecha de entrega, las máquinas son reentrantes, no todos los jobs tienen la misma cantidad de operaciones y los jobs están agrupados en clusters que siguen las mismas rutas.
2. **Flow Shop:** En este momento se está trabajando en la optimización multiobjetivo, con un enfoque evolutivo, ámbito en el cual no hay mucho trabajo realizado.
3. **Máquinas Paralelas:** En este caso el problema más serio que se presenta es la falta de benchmarks y el no conocimiento de valores óptimos ni de upper bounds para instancias de tamaño de interés (medias y grandes), en consecuencia, si bien se trabaja en la incorporación de técnicas alternativas de multirecombinación, representación e hibridización simultáneamente se están implementando nuevas heurísticas más modernas y de ser posible algún método analítico que

permita obtener valores de referencia y óptimos, respectivamente, a los efectos de poder validar los resultados obtenidos.

### 3. REFERENCIAS BIBLIOGRÁFICAS:

- [1] Morton T.E., Pentico D.W. – Heuristic Scheduling Systems – John Wiley & Sons Inc., New York, 1993.
- [2] Baker, K. R. – Introduction to Sequencing and Scheduling – John Wiley & Sons Inc., New York, 1974.
- [3] Baker C.T y Dzienlinski B.P. – Simulation of a Simplified Job Shop – Management Science 6, págs. 311 – 323, 1960.
- [4] Moore J. M y Wilson R. C. - A Review Of Simulation Research in Job Shop Scheduling, Production and Inventory Management, 8, págs. 1 –10, 1967.
- [5] Pai A. R. y Mc Roberts K. L. – Simulation Research in Intercangeable Part Manufacturing, Management Science, 17, págs. 732 - 743, 1971.
- [6] Glaser R. u Hottenstein M. – Simulation Study of a Close-Loop Job Shop, Journal of Operations Management, 3, págs. 155 - 166, 1982.
- [7] Barret, R. T. y Barman S – A SLAM II Simulation of a Simplified Flow Shop, Simulation, 47, págs. 181 - 189, 1986.
- [8] Balas, E. - Discrete Programming by the Filter Method, Operations Research, 15, págs. 915 - 957, 1967.
- [9] Gomory, R. E. - Faces of an Integer Polyhedron, Proceedings of the National Academic of Science, 57, págs. 16 - 18, 1967.
- [10] Brown, A. y Lomnicki Z. A. Some Applications of the Branch and Bound Algorithm to the Machine Sequencing Problem, Operationional Research, 17, págs. 173 – 186, 1966.
- [11] Florian, M. P. et. al - An Implicit Enumeration Algorithm for the Machine Sequencing Problem, Management Science, 17, págs. 782 – 792 , 1971.
- [12] Greenberg, H. - A Branch and Bound Solution to the General Scheduling Problem, Operations Research, 16, págs. 353 - 361, 1968.
- [13] McMahan, G. B. y Burton P. G. – Flow Shop Scheuling with the Branch and Bound Method, Operations Research, 15, págs. 473 - 481, 1967.
- [14] Schwimer, J. - On the n-Job, One Machine, Sequence Independent Scheduling Problem with Tardiness Penalties: A Branch and Boun Solution, Management Science, 18, págs. 301 - 313, 1967.
- [15] Srinivasan, V. - A Hybrid Algorithm for the One Machine Sequencing Problem to Minimize Total Tardiness, Naval Research Logistics, 18, págs. 317 - 327, 1971.
- [16] Garey, M. R. - the complexity of Flowshop and Jobshop Scheduling, Matehematics of Operations Research, 1, págs. 117 - 129, 1976.
- [17] Wilkerson, L. J. e Irwin, J. D . – An Improved Algorithm for Scheuling Independent Tasks, AOOE Transactions, 3, págs. 239 - 245, 1971.
- [18] Van De Velde, S. L. - Machine Scheduling and Lagrangian Relaxation, PhD Thesis, Technische universiteit Eindhoven, 1991.
- [19] Glover, F. - Tabu Search: A Tutorial, Interfaces, 20, págs. 74 - 94, 1990.
- [20] Glover, F. y Laguna, M. – Target Analysis to Improve a Tabu Search Method for Machine Scheduling, Advanced Knowledge Research Group, US West Advanced Technologies, Boulder, CO., 1989.
- [21] Kirkpatrick, S.C. y Vecchi, M. P. – Optimazation by Simulated Annealing, Science, 220, págs. 671 – 680, 1983.
- [22] Whitley, et. al. – Sequence Scheduling ith Genetic Algorithms – New Directions for Operation Research in Manufacturing, Springer Verlag, 1991.

- [23] Hart, E, et. al – A Heuristic Cpbomatopm Method for solving Job Shop Scheduling Problems – Parallel Problem Solving from Nature V, Springer Verlag, 1998.
- [24] Lin, S., et. al. – Investigating Parallel Algorithms on Job Shop Scheduling Problems – VI Annual Conference on Evolutionary Programming, 1997.
- [25] Yamaha, T., et. al. – A Genetic Algorithm Applicable to Large Scale Job Shop Problems – Parallel Problem Solving from Nature II, 1992.
- [26] C. Bierwirth, et. al – Production Scheduling and Rescheduling with Genetic Algorithms – Evolutionary Computation, 7, N° 1, págs. 1 - 17, 1999.
- [27] Fang H. – Genetic Algorithms in Timetabling and Scheduling – Department of Artificial Intelligence, University of Edinburg, Scotland, 1994.
- [28] M. Dorigo and V. Maniezzo and A. Colorni - "Ant System for the job-shop scheduling". JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science, 1996.
- [29] M. Dorigo and V. Maniezzo and A. Colorni - "The ant system: Optimization by a colony of cooperating agents". IEEE Trasn. Systems, Man, and Cybernetics. Part B, pp 29-41, 1996.
- [30]S. van der Zwaan and C.Marques - "Ant colony optimization for job job scheduling". Proceedings of the Third Workshop on Genetic Algorithms and Artificial Life (GAAL '99), 1999.
- [31]T. Stützle - "An ant approach for the flow shop problem". Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT '98), Vol 3, pp 1560-1564, 1998. Verlag, Aachen.
- [32] Vazquez M. y Whitley , D – A Comparision of Genetic Algorithms for the Dynamic Job Shop Scheduling Problem – Parallel Problem Solving from Nature VI, 2000.
- [33] Lin S., et. al – A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problems – International Conference of Genetic Algorithms, 1997.
- [34] Whitley, D. y Kauth, G. – GENITOR: A Different Genetic Algorithm – Proceedings Rocky Mountain Conference on Artificial Intelligence, 1988.
- [35] Esquivel, Gallard, Leguizamón, Zuppa – “A Comparative Performance of Alternative Heuristics for the Flow Shop Scheduling Problem”, Evocop 2002, aceptado para su publicación en Lecture Notes in Computer Science, Springer Verlag.
- [36]G. Leguizamón and M. Schütz - "An improved ant colony algorithm for the job shop scheduling problem". 2002. En preparación.
- [37] Esquivel S., Ferrero S. y Gallard R. – “Multirecombined Evolutionary Algorithm to solve Multiobjective Job Shop Scheduling”, expuesto y publicado en los Anales del VI Congreso Argentino de Ciencias de la Computación, CACIC, Universidad Nacional de la Patagonia San Juan Bosco, 2-7 de Octubre de 2000, pag. 1331.
- [38] Esquivel S. C., Ferrero S.W., Gallard R.H. “A multiobjective optimization via multistage evolution for the job shop scheduling problem”. Second International ICSC Symposium of Intelligent Systems, EIS 2000, **presentado y publicado** June 27-30 , 2000, University of Paisley, Scotland U.K.
- [39] Esquivel S., Ferrero S., Gallard R., “Multirecombination in Multiobjective Evolutionary Algorithms for Job Shop Scheduling”, CEC’2000, Evolutionary Computation, San Diego, Usa, **IEEE Publishing Co**, aceptado para su publicación.
- [40] Esquivel S., Ferrero S., Gallard R. – “Multirecombination and Local Search in Multiobjctive Evolutionary Algorithms for the Job Shop Scheduling”, Third Iberoamerican Workshop on Distributed Artificial Intelligence and Multiagent Systems, 3WDAIMAS, **IEEE Publsishing Co.**, Sao Paulo, Brasil, November 2000, a aparecer.

## CONTROL, PLANNING Y NAVEGACION BASADOS EN INTELIGENCIA COMPUTACIONAL

Susana Esquivel, Carlos Kavka, Guillermo Leguizamón  
LIDIC<sup>1</sup>  
Facultad de Ciencias Físico, Matemáticas y Naturales  
Universidad Nacional de San Luis  
Ejército de los Andes 950 -(5700) San Luis - Argentina  
Tel/Fax: +54 2652 420823/430224  
{esquivel, ckavka, legui}@unsl.edu.ar

### RESUMEN

El presente trabajo describe una propuesta de investigación para atacar el problema de agentes (robots) móviles autónomos. Dicha tarea es realizada por investigadores de distintas líneas del LIDIC, y en consecuencia, con técnicas diferentes las cuales pueden usarse solas y/o combinadas.

Dentro de los distintos aspectos que están involucrados dentro de esta problemática, la investigación tiene como objetivo centrarse en dos temáticas bien delimitadas, por un lado el control y, por otro, en algoritmos de planning y navegación.

Dentro del campo conocido con el nombre de Robótica Evolutiva, en la actualidad el enfoque más innovativo parece ser la combinación de la Computación Evolutiva y las Redes Neuronales Artificiales [1] para enfrentar los aspectos de control. Al aplicar el enfoque evolutivo y las redes neuronales en los sistemas de control de los robots se han usado dos métodos básicos: El primero, consiste en fijar la topología de la red y luego utilizar un algoritmo evolutivo para determinar los pesos mientras que, el segundo, implica usar un algoritmo evolutivo para evolucionar la topología de la red y, luego, los pesos podrán determinarse por un algoritmo de aprendizaje separado o también obtenerse a través de un nuevo proceso evolutivo.

El problema de planning para agentes móviles [2] puede ser formulado típicamente como sigue: para un agente y un ambiente determinado planificar un camino del agente entre dos puntos especificados que sea libre de colisiones y satisfaga ciertos criterios de optimización.

El problema de navegación del agente en un ambiente con obstáculos desconocidos puede ser descrito como un problema con una función de evaluación no estacionaria y un conjunto de restricciones variante (crecientes) [3].

En consecuencia, inicialmente, las heurísticas con las que se propone trabajar son los algoritmos evolutivos, las redes neuronales y las colonias de hormigas.

**Palabras Claves:** agentes móviles, control, planning, navegación, algoritmos evolutivos, redes neuronales, colonia de hormigas.

---

<sup>1</sup>Laboratorio de Investigación y Desarrollo en Inteligencia Computacional. Director Dr. Raúl Gallard

## 1.INTRODUCCION

La Robótica Evolutiva es una nueva tendencia que incluye el uso de nuevas tecnologías de la Inteligencia Computacional para la creación automática de agentes (robots) móviles autónomos. Es un enfoque nuevo que considera a los robots como organismos artificiales que desarrollan sus propias destrezas en interacción con su ambiente, sin intervención humana. Esta nueva dirección en la robótica hace uso de tecnologías tales como redes neuronales, algoritmos evolutivos y otras heurísticas.

La idea básica detrás de la Robótica Evolutiva es la siguiente: Una población inicial de cromosomas artificiales diferentes, cada uno codificando el sistema de control (y en otras ocasiones la morfología) del agente, se crea aleatoriamente y se la deja interactuar con el ambiente. Cada agente (físico o simulado) se deja en libertad para actuar (moverse, identificar objetos y manipularlos) de acuerdo con el controlador genéticamente especificado mientras su comportamiento, al realizar una cierta tarea, se evalúa automáticamente.

A los robots con mayor adaptabilidad (fitness) se les permite aparearse y reproducirse, en la práctica esto se logra con el operador de crossover (intercambio de parte de sus cromosomas) y a los hijos se les cambia algo de la información genética recibida de los padres usando el operador de mutación. Este proceso se repite un cierto número de generaciones hasta que nace un individuo (robot) que satisface el criterio de performance (función de fitness) establecido por el experimentador.

En la actualidad el enfoque más innovativo parece ser la combinación de la Computación Evolutiva y las Redes Neuronales Artificiales [1]. Al aplicar el enfoque evolutivo y las redes neuronales en los sistemas de control de los robots se han usado dos métodos básicos: El primero, consiste en fijar la topología de la red y luego utilizar un algoritmo evolutivo para determinar los pesos mientras que, el segundo, implica utilizar un algoritmo evolutivo para evolucionar la topología de la red y, luego, los pesos podrán determinarse por un algoritmo de aprendizaje separado o también obtenerse a través de un nuevo proceso evolutivo.

En cuanto al movimiento de los agentes móviles existen diferentes estrategias para solucionar el problema de planning. Independientemente de sus muchas diferencias, las mismas están basadas en unos pocos enfoques generales, tales como mapas de rutas (roadmap), descomposición de celdas (cell decomposition) y campo potencial (potential field).

Aunque una gran cantidad de investigación ha sido realizada para el tratamiento de este problema los enfoques tradicionales tienden a ser inflexibles para:

- cambios de objetivos,
- cambios e incertidumbre en el ambiente y
- diferentes restricciones sobre los recursos computacionales, tales como tiempo y espacio.

Los planificadores tradicionales (off-line) a menudo asumen que el ambiente es perfectamente conocido y tratan de buscar el camino óptimo basado en algún criterio de costo fijo [2,4] (el más común es el de camino más corto). Por otro lado los planificadores on-line son a menudo puramente reactivos [5,6,7,8,9] y no tratan de optimizar un camino.

La mayoría de los enfoques actuales de agentes móviles usa una combinación de ambos. Usualmente, una vez que el agente ha determinado su ubicación y su meta se genera un camino. El agente sigue este camino pero periódicamente estudia sus sensores para determinar la presencia de nuevos obstáculos u otras restricciones que pueden requerirle desviarse del camino generado. En este último

caso usa un esquema de navegación reactivo para moverse alrededor del obstáculo y retomar nuevamente (si es posible) el camino [10].

En los últimos años han surgido planificadores basados en el paradigma de computación evolutiva. Sin embargo, los primeros planificadores basados en este paradigma usaron algoritmos standard [11,12,13], que asumían ambientes totalmente conocidos y, por lo tanto, también presentaban la poca flexibilidad de los planificadores tradicionales. Más recientemente, se han desarrollado planificadores evolutivos para tratar con ambientes dinámicos [14]. Sin embargo, aun existe la necesidad de desarrollar planificadores más generales, flexibles y adaptativos, capaces de tratar cualquier tipo de cambios en los requerimientos y/o ambientes.

Por otro lado, aplicaciones de la heurística basada en colonias de hormigas [17,18] en el ámbito de navegación de robots sólo han sido realizadas desde la perspectiva de "swarm intelligence" (o inteligencia colectiva) en donde una determinada actividad es llevada a cabo por un grupo de robots que proceden en forma cooperativa/descentralizada [15,16,19]. Sin embargo, y dada la flexibilidad de los algoritmos basados en colonias de hormigas, es posible diseñar planificadores off-line de inmediata implementación en este contexto dado que se traduce en un problema de optimización cuya solución es representada por un paso factible del ambiente (estático) en estudio. Considerando además que este enfoque está basado en poblaciones de agentes, la planificación on-line, característica de un ambiente dinámico (o combinación de estático y dinámico) podría ser lograda añadiendo atributos reactivos a los agentes que componen el algoritmo junto con capacidades de memoria (explícita o implícita) que permitan realizar una adecuada replanificación según los eventuales cambios producidos en el ambiente bajo consideración.

## **2.PROPUUESTA DE INVESTIGACION**

Las direcciones de la investigación propuesta pueden sintetizarse en las siguientes:

- a)Análisis de las arquitecturas de control de robots móviles propuestas en investigaciones recientes centrándose principalmente en los enfoques de redes neuronales, y entre ellas las de redes neuronales recurrentes y de sus respectivos algoritmos de aprendizaje.
- b)Comprensión del concepto subyacente en el desarrollo del robot Khepera como una metodología integrada basada en software y hardware creado con la finalidad de investigar algoritmos adaptativos y evolutivos en robótica.
- c)Desarrollo de esquemas de navegación en ambientes estacionarios y no estacionarios usando algoritmos evolutivos y de colonia de hormigas. Comparación de los mismos tratando de efectuar análisis de sus similitudes y diferencias para profundizar en el conocimiento acerca de su funcionamiento.
- d)El diseño de experimentos iniciales de simulación del robot Khepera, usando el software disponible en Internet.
- e)Para la última etapa se espera contar con un robot Khepera (cuyo trámite de compra ya ha sido iniciado por el LIDIC) y se tratará de implementar sobre el mismo la arquitectura de control seleccionada y simulada previamente con la finalidad de que el robot adquiera estrategias para la navegación autónoma sobre diferentes ambientes estacionarios y no estacionarios.

### 3. REFERENCIAS BIBLIOGRAFICAS

- [1] Meeden L.A and Kumar D. -Trends in Evolutionary Robotics, Soft Computing for Intelligent Robotic Systems, Physica-Verlag, New York, pp 215-233, (1998).
- [2] Yap, C.K. - "Algorithmic Motion Planning", Advances in Robotics, Vol. 1: Algorithmic and Geometric Aspects of Robotics. J.T Schwartz and C.K. Yap Eds., Lawrence Erlbaum Associates, 1987, pp. 95 - 143.
- [3] Trojanowski, K - "Evolutionary Algorithms with Redundant Genetic Material for Non Stationary Environments", PhD. Thesis, Institute of Computer Science, Polish Academy of Sciences, Poland, May 2000.
- [4] Latombe, J.C. - "Robot Motion Planning", Kluwer Academic Publishers, 1991.
- [5] Arkin, R.C. - "Motor Schema-Based Mobile Robot Navigation", International Journal Robotics Research, August 1989, pp. 92 - 112.
- [6] Borenstein, J. and Koren Y. - "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", IEEE Transaction Robotics and Automation, 7(3), June 1991, pp. 278 - 287.
- [7] Brooks, R.A. - "A Robust Layered Control System for a Mobile Robot", IEEE Journal of Robotics And Automation, Vol. 2, 1986, pp. 14 -23.
- [8] Lumelsky, V.J. - "A Comparative Study on The Path Length Performance of Maze-Searching and Robot Motion Planning Algorithms", IEEE Transaction on Robotics and Automation, 7(1), 1991, pp. 57 -66.
- [9] Lumelsky V.J. and Stepanov, A.A. - "Path Planning Strategies for a Point Mobile Automaton Moving Unknown Obstacles of Arbitrary Shape", Algorithmica, Vol 2, 1987, pp. 403-430.
- [10] Tsubochi, T., Hirose, A. and Arimoto S. - "A Navigation Scheme with Learning for a Mobile Robot among Multiple Moving Obstacles", Proceedings of IROS'93, Intelligent Robots for Flexibility, Vol. 2317, 1993, pp. 2234 - 2240.
- [11] Page, W.C., McDonell, J.R. and Anderson, B. - " An Evolutionary Programming Approach to Multi-Dimensional Path Planning", Proceeding of the First Annual Conference on Evolutionary Programming, D. Fogel and J.W. Atmar Eds., Evolutionary Programming Society, 1992, pp. 63 70.
- [12] Shibata, T. and Fukuda, T. - " Intelligent Motion Planning by Genetic Algorithm with Fuzzy Critic", Proceeding of the 8<sup>th</sup> IEEE International Symposium on Intelligent Control, Chicago, August 25-27, 1993.
- [13] Sugihara K. and Smith J. - "Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot", Proceedings 1997 IEEE Intelligent Symposium on Computational Intelligence in Robotics and Automation", 1997, pp. 138 - 143.

- [14]Hocaoglu, C. and Sanderson, A.C. - "Planning Multiple Paths using Speciation in Genetic Algorithms", Proceedings of the IEEE International Conference on Evolutionary Computation, ICEC'96, IEEE Publishing Inc., 1996, pp. 378 -383.
- [15]Krieger, M. J. B., Billeter, J.-B., Keller, L. - "Ant-like task allocation and recruitment in co-operative robots". Nature, 406, 992-995, 2000.
- [16]E.Bonabeau and M.Dorigo and G.Theraulaz - "Swarm Intelligence - From Natural to Artificial Systems". Oxford University Press, 1999. Santa Fe Institute Studies in the Sciences of Complexity.
- [17]D.Cornea and M.Dorigo and F.Glover - "New Ideas in Optimization". McGraw-Hill International, 1999.

## EVOLUTIONARY ALGORITHMS TO SOLVE THE SCHOOL TIMETABLING PROBLEM

Fernández N., Alfonso H.  
Proyecto UNLPAM-09/F015<sup>1</sup>  
Departamento de Informática - Facultad de Ingeniería  
Universidad Nacional de La Pampa  
Calle 110 esq. 9  
(6360) General Pico – La Pampa – Rep. Argentina  
e-mail: { fernaty, alfonsoh }@ing.unlpam.edu.ar  
Phone: +54 2302 422780/422372, Ext. 6302

Gallard R.  
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)<sup>2</sup>  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106  
(5700) - San Luis -Argentina  
e-mail: rgallard@unsl.edu.ar  
Phone: +54 2652 420823  
Fax : +54 2652 430224

### ABSTRACT

Evolutionary Computation (EC) have been applied to a great number of optimisation problems with some success, specially in fields such as production operations in manufacturing industry, parallel and distributed systems, logistics and traffic.

The Timetabling problem is a kind of three-dimensional cutting problems, but it can be considered as a kind of scheduling problems too. The problem of creating a valid timetable involves scheduling classes, teachers and rooms into a fixed number of periods, in such a way that no teacher, class or room is used more than once per period.

The EC mimics the process of natural selection, with the effect of creating a number of potentially optimal solutions to some complex search problem. This work shows an evolutionary approach to solve the school-timetabling problem. To ensure the birth of valid offspring this approach uses a population of decoders. Implementation details for a set of timetabling instances of distinct complexity, using evolutionary approaches to the problem, are shown.

---

<sup>1</sup> The Research Group is supported by the Universidad Nacional de La Pampa.

<sup>2</sup> The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

## 1. INTRODUCTION

The problem of creating a valid timetable involves scheduling classes, teachers and rooms into a fixed number of periods, in such a way that no teacher, class or room is used more than once per period. For example, if a class must meet twice a week, then it must be placed in two different periods to avoid a clash. A class consists of a number of students. We assume that the allocation of students into classes is fixed, and that classes are disjoint, that is, they have no students in common. For them, a correct timetable is one in which a class can be scheduled concurrently with any other class. In each period, a class is a subject. It is possible for a subject to appear more than once in a period. A particular combination of a teacher, a subject, a room and a class is called a tuple. A tuple may be required more than once per week. Thus, the timetabling problem can be phrased as scheduling a number of tuples such that a teacher, class or room does not appear more than once per period. There are several works in the literature that attempt to solve this problem using different heuristics [1,2, 3,6, 7].

The task of combining a class, teacher and room into a tuple is handled as a separate operation. Tuples are formed with knowledge of the requirements of the various classes and teachers, as well as information on room availability. It is convenient to partition the problem in this way as it reduces the complexity of the scheduling task. In many cases it is possible to form the tuples without the need to use an optimisation scheme because the relationship between classes, teachers and rooms is often fixed.

It is possible to define an objective or cost function for evaluating a given timetable [8]. This function is an arbitrary measure of the quality of the solution. A convenient cost function calculates the number of clashes in any given timetable [2]. An acceptable timetable has a cost of 0. The optimisation problem becomes one of minimising the value of this cost function. The cost of any period can be expressed as the sum of three components corresponding to a class cost, a teacher cost and a room cost. The sum of the components is not strictly necessary but it can reflect the quality of the solution.

However, by using a sum, it is easy to weight the various costs so that one may be made more important than the others. In this way the optimisation process can be guided towards a solution in which some type of clashes are more important than others.

The class cost is the number of times each of the classes in the period appears in that period, less one if it is greater than zero. Thus if a class appears no times or once in a period, then the cost of it is zero. If it appears many times, the class cost for that class is the number of times less one. The class cost for a period is the sum of all class costs. The same computation applies for teachers and rooms. The cost of the total timetable is the sum of the period costs. Therefore, any optimisation technique should aim to find a configuration with the lowest possible cost.

## 2. APPLYING EVOLUTIONARY ALGORITHMS TO TIMETABLING

A timetable can be represented by a fixed set of tuples, each of which contains a class number, a teacher number and a room number. The scheduling algorithm must assign a period number to each of these tuples such that a given class, teacher or room does not appear more than once in a period.

In order to use an evolutionary algorithm to solve this problem, it is necessary to devise a representation for a timetable which maps onto chromosomes. Each chromosome is composed of genes, each of which represents some property of the individual timetable.

The specification of an appropriate schedule representation is a decisive factor to find a good performance of an evolutionary algorithm [4,5]. The principal difficulty present in scheduling problems is to determine which chromosome representation is to be used. We choose an indirect chromosome representation for the scheduler, where the transformation from a chromosome representation to a legal schedule had been done for a special decoder. With this kind of representations, the algorithm works on a population of encoded solutions [9]. Because the representation do not directly provides a schedule, a scheduler builder is necessary to transform a chromosome into a schedule, validate and evaluate it. The scheduler builder guarantees the feasibility of a solution and its work depends on the amount of information included in the representation.

## Decoders

Here a chromosome is an  $n$ -vector where the  $i^{\text{th}}$  component is an integer in the range  $1..(n-i+1)$ . The chromosome is interpreted as a strategy to extract items from an ordered list  $L$  and build a permutation. For example if  $L = (1, 2, 3, 4, 5, 6)$  then table 1, indicates the correspondence among chromosomes and permutations.

Chromosome	Permutation
1 1 1 1 1 1	1 2 3 4 5 6
4 3 4 1 1 1	4 3 6 1 2 5
4 2 3 2 1 1	4 2 5 3 1 6

Table 1. Chromosome-permutation correspondence

A decoder is a mapping from the representation space into a feasible part of the solution space, which includes mappings between representations that evolve and representation that constitute the input for the evaluation function. This simplifies implementation and produces feasible offspring under different conventional crossover methods, avoiding the use of penalties or repair actions.

Although the offspring was feasible, the schedule builder could not find a place in the timetable for some tuples then a repair action or penalty is necessary for this individual.

## 3. IMPLEMENTATION DETAILS

The experiments were developed for a set of selected instances of the Timetabling Problem, under permutation representation. For each experiment series of fifty runs were performed. Elitism to retain the best valued individual was implemented. The population size was fixed at 50 individuals. For crossover, we used the traditional *one-point* crossover operator. And for mutation, a *big-creep* operator was used, which consists in replacing the gene value by another in the permitted range. The algorithms evolved for a minimum of 500 generations, after that a control of the mean fitness population progress began: if this value remained within a determined range for 20 consecutive then the algorithm stops. Probabilities for crossover and mutation were fixed at 0.65 and 0.01, respectively. These values were determined as the best combination of probabilities after many initial trials. At the moment, we are testing the described algorithm on three instances [3]: *hdtt4*, *hdtt5* and *hdtt6*.

## 4. REFERENCES

- [1] Abramson D. "Constructing School Timetables Using Simulated Annealing: Parallel and Sequential Solutions", *Management Science*, pp 98-113. Jan 1991.
- [2] Abramson D., Abela J.: *A Parallel Genetic Algorithm for Solving the School Timetabling Problem*, 15 Australian Computer Science Conference, pp 1-11. Feb. 1992.
- [3] Abramson D.A. and Dang H. "School Timetables: A Case Study in Simulated Annealing", in *Lecture Notes in Economics and Mathematical Systems: Applied Simulated Annealing*, Rene V.V. Vidal (Ed.), 103-124, 1993.
- [4] Bäck T: *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.
- [5] Bruns R.: *Scheduling*, Handbook of Evolutionary Computation, 1997, chapter F1.5, pp F1.5:1-F1.5:9. 1997.
- [6] Burke E.K., Elliman D.G., Weare R.: *A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems*, Proceedings of the Sixth International Conference on Genetic Algorithms, 1995.
- [7] Caldeira J.P., Rosa A.C.: *School Timetabling using Genetic Search*. Laseeb - ISR - IST. 1997.
- [8] Goldberg D. E.: *Genetic Algorithms: In Search, Optimization and Machine Learning*. 1989, Addison-Wesley Publishing Co.

- [9] Salto C., Alfonso H., Gallard R.: *Performance Evaluation of Evolutionary Algorithms under Different Representations when Solving the JSS problem*, Proc. of 5<sup>th</sup> World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), pp. 424-428, 2001.

# Un enfoque evolutivo como método alternativo para la técnica de halftoning

**Roberto Guerrero, Guillermo Leguizamón y Hugo Viano**

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional\*

Departamento de Informática

Universidad Nacional de San Luis

San Luis, Argentina

{rag,legui,hviano}@unsl.edu.ar

## Abstract

Halftoning (mediotonar) es un proceso por el cual imágenes en tonos de grises son convertidas en patrones de puntos blancos o negros, de manera tal que éstas conserven las características perceptuales de la imagen original. Una imagen mediotonada (halftoned) reproduce en mayor o menor medida a la imagen original dependiendo de la técnica utilizada. La presente propuesta tiene como objetivo investigar la aplicabilidad de técnicas de inteligencia computacional como técnica alternativa para el proceso de halftoning. En dicho proceso distintas funciones de evaluación pueden ser consideradas de manera tal de automatizar la distribución de los pixels. Dado que dicha distribución puede ser formulada como un problema de optimización sobre un espacio de búsqueda conformado de matrices binarias, la aplicación de un enfoque evolutivo para su resolución es casi directa, pudiendo además, incorporar variadas funciones de importancia para obtener imágenes mediotonadas en las cuales se aumente la información transmitida desde distintas perspectivas. El enfoque evolutivo estudiado son los Algoritmos Genéticos. Los resultados de su aplicación serán contrastados con técnicas tradicionales de halftoning.

## 1 Introducción

Imagery [2] es un término general que permite englobar un conjunto de disciplinas relacionadas a la creación, almacenamiento y manipulación de modelos e imágenes de objetos. Una de ellas, el área de procesamiento de imágenes, trata el análisis y manipulación de imágenes ya existentes; donde la nueva imagen generada es de alguna manera diferente a la imagen original. En particular, el análisis de imágenes es importante para áreas como la biomedicina, aviación, geología, etc., en usos específicos tales como el reconocimiento aéreo, scanning de cromosomas entre otras [1]. En este contexto, las imágenes son potenciales herramientas para la toma de decisiones dado que permiten aumentar la información transmitida. Sin

---

\*El laboratorio es dirigido por el Dr. Raúl Gallard y subvencionado por la Universidad Nacional de San Luis y la ANPCyT (Agencia Nacional para la Promoción de la Ciencia y la Tecnología).

embargo, crear y reproducir imágenes presenta problemas específicos a la manera en que éstas son manipuladas. Por ejemplo, imprimir una imagen implica plasmar en papel los diversos detalles contenidos en la misma. No obstante, existen impresoras que sólo pueden trabajar con puntos sólidos de tinta, es decir imprimen un punto o no; por lo cual no podrán cumplir con su tarea si la imagen en cuestión se encuentra definida mediante tonos continuos de grises (desde el blanco al negro).

Para resolver este problema, las impresoras usan una técnica conocida como halftoning. Esta técnica es un proceso por el cual imágenes en tonos de grises son convertidas en patrones de puntos blancos o negros, de manera tal que éstas puedan ser luego impresas. Así, una mayor cantidad de grandes puntos ubicados en zonas aledañas representan áreas oscuras, mientras que pequeños puntos y dispersos representan áreas claras. Si bien la técnica de halftoning ha sido principalmente usada para lograr una reproducción lo más acertada posible del original [1] [9] [5] [11], diferentes resultados podrían obtenerse a través del uso de funciones de importancia [8]. Una función de importancia se define con el objeto de poder obtener reproducciones que resalten otros aspectos de la imagen original (aumentando o no la información transmitida): bordes, texturas, etc.

Recientemente han surgido enfoques alternativos basados en computación evolutiva [3] para ser usados en imagery en general y en halftoning en particular. Para el caso de halftoning la función de importancia se convierte en la función objetivo a ser optimizada por un Algoritmo Genético (AG) [4], lo que permitiría la automatización de la distribución de los pixels en la representación binaria de acuerdo al objetivo definido, es decir, de acuerdo a la función de importancia considerada. Los AGs han demostrado ser herramientas potentes y robustas para su aplicación en el campo de optimización [3]. La flexibilidad inherente de estos algoritmos ha permitido atacar la resolución de los más diversos problemas académicos y del mundo real. En el campo de Imagery ha habido un importante avance en la aplicación de enfoques evolutivos en general y AGs en particular para la resolución de problemas complejos y de gran interés práctico: halftoning, segmentación de imágenes de color, análisis de imágenes medicinales, análisis de escena (visión), y muchos otros [6] [10].

## 2 Funciones de importancia

La premisa subyacente al halftoning de importancia es distribuir los patrones de puntos hacia áreas de la imagen que poseen atributos importantes de la misma [8].

El sistema visual humano actúa como un analizador de Fourier en el dominio espacial. Debido a éste comportamiento, se podría decir que en una etapa inicial el mismo evalúa e interpreta una imagen desde un aspecto global y posteriormente se focaliza (a diferentes resoluciones) sobre ciertos atributos o áreas de la imagen. El halftoning de importancia consiste en construir un filtro de pasa banda a partir de una función de importancia definida por el usuario. La función de importancia identifica la presencia de atributos importantes en la imagen sobre un segmento de la misma.

Si la función de importancia fuese, por ejemplo, la intensidad  $g$  en una posición  $(x, y)$  de la imagen, luego la función de importancia sería simplemente:  $F(x, y) = g(x, y)$ . Adicionalmente, el usuario puede construir su propia función de importancia que combine varias funciones de importancia y pesarlas acorde con su criterio:

$$F(x, y) = w_1 \times f_1(x, y) + w_2 \times f_2(x, y) + \dots + w_n \times f_n(x, y)$$

donde  $w_i$  es el peso de cada función tal que  $\sum_{i=1}^n w_i = 1$ .

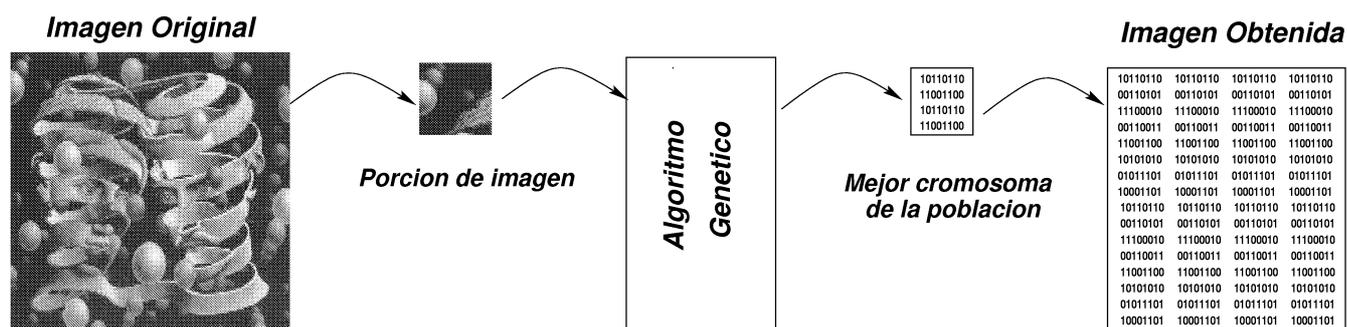


Figure 1: Proceso a través del cual se obtiene una imagen mediotonada usando un AG

### 3 El enfoque evolutivo

Nuestro objetivo está centrado en investigar acerca de distintas funciones de importancia de manera tal de automatizar la distribución de los pixels a través de un enfoque evolutivo con el objetivo de lograr una imagen que aumente o no la información transmitida.

Uno de los primeros trabajos en relación con nuestro estudio experimental [7] presenta un estudio de la aplicación de AGs para haltoning donde son consideradas dos funciones de evaluación para la generación de una imagen mediotonada acorde con las técnicas tradicionales. En términos de funciones de importancia se podría decir que la imagen es considerada como un todo donde se trata de conservar la intensidad promedio de la imagen original.

Las funciones de evaluación determinarán la calidad de la imagen obtenida y establecerán la necesidad o no de proseguir con el proceso.

En la figura 1 se muestra, en forma general, como se realiza el proceso de obtención de la imagen binaria a través de un AG. Inicialmente, la imagen original donde cada pixel representa un valor de tono de grises, se divide en varios bloques. Cada bloque es considerado de a uno por vez como entrada de un AG que se encargará de buscar un bloque binario que optimice la función o combinación de funciones de importancia dadas como parámetros del algoritmo. La salida del AG es un bloque binario que representa el mejor bloque obtenido entre las poblaciones procesadas a través de las sucesivas generaciones. Dicho bloque ocupará el correspondiente lugar en la imagen final (versión mediotonada). Es importante destacar que cada uno de los bloques puede ser procesado en forma independiente, lo que posibilita una directa paralelización del proceso previamente descrito.

Los resultados reportados en [7] de acuerdo al proceso explicado previamente, mostraron que el enfoque evolutivo es capaz de obtener imágenes mediotonadas de calidad similar en comparación con el uso de técnicas tradicionales.

### 4 Propuesta en desarrollo

Siguiendo esta línea de investigación, están bajo consideración otras funciones de importancia y dentro de lo que es el enfoque evolutivo, se está trabajando en el estudio de la influencia de los distintos parámetros en la performance del AG, como por ejemplo, velocidad de convergencia, operadores genéticos, tamaño de

la población, etc.

En cuanto a las funciones de importancia, un enfoque original consistirá en combinarlas con las funciones de evaluación con el objeto de unificar los procesos de filtrado y evaluación en un solo paso.

Las funciones a aplicar en un comienzo son la varianza de tonos (contraste), y el gradiente, así como combinaciones entre ellas con la intensidad promedio de la imagen.

Dadas las características del problema y de los algoritmos evolutivos, un futuro estudio experimental involucrará la implementación de una versión paralela a los efectos de mejorar la performance, tanto en calidad como en el tiempo de computación insumido.

## References

- [1] Ulichney R. A. *Digital Halftoning*. MIT Press, 1987.
- [2] Watt A. and Policarpo F. *The Computer Image*. Addison-Wesley, 1998.
- [3] Thomas Bäck, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. Institute of Physics Publishing and Oxford University Press, 1997.
- [4] Goldberg D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Wesley Publishing, 1989.
- [5] Peli E. Multiresolution error-convergence halftone algorithm. *Op. Soc. Am. A.*, 8(4):625–636, 1991.
- [6] Egbert J.W. Boers et al., editor. *Application of Evolutionary Computation*, volume 2037 of *Lecture Notes in Computer Sciences*, Como, Italy, April 18-20 2001. Springer.
- [7] Saito H. and Kobayashi N. Evolutionary computation approach to halftoning algorithm. In *Proc. of the First IEEE Conf. on Evolutionary Computation*, volume 2, pages 787–791, June 1994.
- [8] Streit L. and Buchanan J. Importance driven halftoning. In *Proceedings of Eurographics 98*, pages 207–217, 1998.
- [9] Jarvis J. F. Judice C. N. and Ninke W. H. A survey of techniques for the display of continuous tone pictures on bilevel displays. In *Proceedings of Computer Graphics Image Processing*, pages 13–40, 1976.
- [10] Aguirre H. Tanaka K. Sugimura T. and Oshita S. Halftone image generation with improved multiobjective genetic algorithm. In *EMO*, pages 501–515, 2001.
- [11] Mitsa T. and Parler K. J. Digital halftoning using a blue noise mask. In *Proc. of 1991 International Conference on Acoustic. Speech and Signal Processing*, pages 2809–2812, 1991.

# Redes Neuronales aplicadas al Reconocimiento de Patrones

Lic. Laura Lanzarini<sup>1</sup>, Ing. A. De Giusti<sup>2</sup>

*Laboratorio de Investigación y Desarrollo en Informática<sup>3</sup>  
Facultad de Informática - Universidad Nacional de La Plata*

## Resumen

El principal objetivo del Reconocimiento de patrones es la clasificación ya sea supervisada o no supervisada. Aplicaciones como Data Mining, Web Searching, recuperación de datos multimedia, reconocimiento de rostros, reconocimientos de caracteres escritos a mano, etc., requieren de técnicas de reconocimiento de patrones robustas y eficientes.

Las redes neuronales, por su capacidad de generalización de la información disponible y su tolerancia al ruido, constituyen una herramienta muy útil en la resolución de este tipo de problemas.

Esta línea de investigación se centra en la aplicación de redes neuronales al reconocimiento de patrones. En esta dirección, se han resuelto problemas de clustering y clasificación.

Actualmente, se estudia la posibilidad de optimizar las soluciones propuestas incorporando estrategias evolutivas sobre la arquitectura y el aprendizaje de la red.

**Palabras Claves:** Redes Neuronales, Reconocimiento de Patrones Adaptivo, Técnicas de Clustering, Clasificación.

## 1. Introducción

Las Redes Neuronales, a diferencia de la programación convencional, permiten obtener soluciones para aquellos problemas en los que no se conoce el algoritmo a utilizar. Para ello se basa en la información contenida en los datos de entrada. Areas como Inteligencia Artificial, Reconocimiento de Patrones o Procesamiento de Señales plantean situaciones con estas características.

Además, las redes neuronales han demostrado tener un buen desempeño tanto en aprendizaje supervisado como no supervisado. En este último caso, tienen capacidad para generalizar la información contenida en los datos de entrada mostrando relaciones que a priori resultan complejas.

En este proyecto interesa especialmente la aplicación de las redes neuronales al reconocimiento de patrones [5]. En esta línea, se han realizado varios desarrollos orientados al reconocimiento de los elementos de muestras histológicas dentro del marco del convenio existente entre el LIDI y la Facultad de Ciencias Médicas de la UNLP [8][11].

En general, estas aplicaciones presentan un alto nivel de complejidad así como un alto tiempo de procesamiento. La necesidad de obtener respuestas en tiempos razonables justifican el análisis de la paralelización de las arquitecturas utilizadas [10][12].

---

<sup>1</sup> Profesor Adjunto Dedicación Exclusiva. Fac. de Informática. Universidad Nacional de La Plata. E-mail: laural@info.unlp.edu.ar

<sup>2</sup> Investigador Principal del CONICET. Profesor Titular Dedicación Exclusiva. Fac. de Informática. Universidad Nacional de La Plata. E-mail: degiusti@info.unlp.edu.ar

<sup>3</sup> Calle 50 y 115 1er Piso, (1900) La Plata, Argentina, TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

## 2. Temas de investigación y desarrollo

Los temas de investigación son los siguientes:

### A) Reconocimiento de patrones utilizando redes neuronales:

- Análisis de las arquitecturas existentes: redes feed-forward, SOM. Aprendizaje supervisado y no supervisado.
- Estudio de las diferentes redes neuronales que permiten resolver el problema de “clustering” [7].
- Desarrollo de la arquitectura CBE (Clustering basado en el Entorno) [4][9].

### B) Paralelización de la arquitectura CBE

- Evaluación de la solución monoprocesador.
- Análisis de las componentes de tiempo de la solución secuencial.
- Análisis del Speed Up de la arquitectura paralela propuesta.

## 3. Algunos resultados obtenidos en los trabajos experimentales

### A) Reconocimiento de patrones

Se ha desarrollado una nueva red neuronal para clustering y segmentación de los datos o patrones de entrada que, a diferencia de los modelos existentes, no requiere la indicación de parámetros de aceptación dependientes del problema [3]. Su comparación con las técnicas clásicas de reconocimiento de patrones ha dado resultados favorables.

Esta red fue pensada para la clasificación de pixels de imágenes a 256 colores y forma parte de un proceso para el reconocimiento de los elementos de una muestra de tejido .

Reducir la cantidad de parámetros de entrada implica un costo adicional que está dado por la evaluación del entorno de cada patrón. Esta evaluación puede tener un costo computacional alto por lo que resulta de interés analizar la paralelización de la red propuesta.

### B) Paralelización de la arquitectura CBE

Resultados experimentales sobre distintos tipos de imágenes han demostrado la efectividad de la arquitectura CBE para resolver el clustering de un conjunto de patrones de entrada. En la solución monoprocesador, el tiempo de procesamiento es del orden  $N \times N \times P$  donde  $N \times N$  es la resolución de la imagen y  $P$  es la cantidad de patrones diferentes utilizados.

Luego de analizar la solución no paralela, se ha concluido en que el tiempo más significativo es el empleado en realizar la transformación del espacio de pixels al espacio de patrones. Esta tarea es del orden  $N^2 \times P$ . El segundo factor es el análisis de similitud entre patrones de orden  $P^2$ .

Utilizando estos resultados se trabajó sobre una arquitectura paralela basada en procesadores homogéneos y memoria distribuida que permitió mejorar el speed-up de la solución [1].

#### 4. Líneas de Trabajo futuro

Resulta de interés incorporar mecanismos evolutivos al proceso de aprendizaje de las redes neuronales con el fin de mejorar su capacidad de adaptación.

Se ha comenzado a trabajar con Redes Neuronales Evolutivas para la resolución de problemas concretos con buenos resultados [2].

Los temas a estudiar son los siguientes:

##### 4.1. Evolución de los pesos de conexión:

- Analizar diferentes representaciones para codificar los pesos de conexión así como la aplicación de diferentes operadores genéticos de búsqueda [6].
- Comparar las soluciones evolutivas con las obtenidas a través de algoritmos de entrenamiento convencionales.

##### 4.2. Evolución de la arquitectura:

- Representación de la red neuronal.
- Analizar diferentes operadores de búsqueda.

##### 4.3. Evolución de las reglas de aprendizaje

#### 5. Bibliografía Básica

- [1] *Parallelizing a new neural network for environment based clustering*. Lanzarini, De Giusti. 5<sup>th</sup> World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001) and 7th International Conference on Information Systems Analysis and Síntesis (ISAS 2001); Orlando, USA, Julio 2001.
- [2] *Criaturas Virtuales especificadas a través de Redes Neuronales Evolutivas* Corbalán, Pisano, Osella Massa, Lanzarini. Jornadas Chilenas de Computación 2001; Pta.Arenas, Chile, Noviembre 2001 (Referato Internacional) y VII Congreso Argentino de Ciencias de la Computación. Calafate, Argentina. Octubre 2001. Vol.2, Pags. 1105-1115 (Referato internacional).
- [3] *Una Nueva Red Neuronal para Clustering y Segmentación basada en el entorno*. Lanzarini, De Giusti. VI Congreso Argentino de Ciencias de la Computación. Ushuaia, Argentina, Octubre 2000. Vol. 3, Pags. 1389-1398. ISBN: 950-763-033-3
- [4] *Environment based Clustering. A New Approach*. Lanzarini, De Giusti. IWISPA 2000, 1<sup>st</sup>. Int'l Workshop on Image and Signal Processing and Analysis. Croatia, Junio 14-15, 2000. *IEEE CAS & ASSP Croatia*. Vol.1, pags. 75-80. Año 2000; ISBN : 953-96769-1-6.
- [5] *Statistical Pattern Recognition*. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 22, Nro. 1, Enero 2000
- [6] *Evolving Artificial Neural Networks*. X. Yao. Proceedings of the IEEE. Vol 9. Pags 1423-1446. 1999

- [7] *Data Clustering. A review.* Jain, Murty and Flynn. ACM Computing Surveys. Vol 31, nro. 3, Sep 1999
- [8] *Reconocimiento y Clasificación de los elementos de una muestra de sangre utilizando Redes Neuronales.* Lanzarini, Vargas, Badrán, De Giusti. 6° Congreso Internacional de Nuevas Tecnologías y Aplicaciones Informáticas. Cuba.1998
- [9] *A New Neural Network for cluster-detection-and-labeling* Torbjorn Eltoft, IEEE Transactions on Neural Networks, Vol.9, No. 5, pp.1021 – 1035, 1998
- [10] *Parallel Computation. Models and Methods.* Akl S. Prentice Hall. 1997
- [11] *Caracterizacion de los Elementos de una Muestra Histológica utilizando Khoros.* Lanzarini, Castañeda, Badrán, De Giusti. II International Congress on Information Engineering. Buenos Aires. 1995.
- [12] *Studies in computational science: Parallel Programming Paradigms.* Brinch Hansen. Prentice Hall Inc. 1995

## HYBRID EVOLUTIONARY ALGORITHMS TO SOLVE SCHEDULING PROBLEMS

Minetti, G., Salto C., Bermúdez, C., Fernández, N., Alfonso H.  
Proyecto UNLPAM-09/F015<sup>1</sup>  
Departamento de Informática - Facultad de Ingeniería  
Universidad Nacional de La Pampa  
Calle 110 esq. 9  
(6360) General Pico – La Pampa – Rep. Argentina  
e-mail: { minettig,saltoc,bermudezc,fernaty,alfonsoh }@ing.unlpam.edu.ar  
Phone: +54 02302 422780/422372, Ext. 6302

Gallard R.  
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)<sup>2</sup>  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106  
(5700) - San Luis -Argentina  
e-mail: rgallard@unsl.edu.ar  
Phone: +54 2652 420823  
Fax : +54 2652 430224

### ABSTRACT

The choice of a search algorithm can play a vital role in the success of a scheduling application. Evolutionary algorithms (EAs) can be used to solve this kind of combinatorial optimization problems. Compared to conventional heuristics (CH) and local search techniques (LS), EAs are not well suited for fine-tuning those structures, which are very close to optimal solutions. Therefore, in complex problems, it is essential to build hybrid evolutionary algorithms (HEA) by incorporating CH and/or LS to provide fine-tuning.

EAs are good at global search but slow to converge, while local search is good for fine-tuning but often falls into local optima. The hybrid approach complements the properties of evolutionary algorithm and other techniques.

This research guide attempts to develop EAs hybridized with local search and conventional heuristics. They are incorporated at different stages of the evolutionary process. Either when the initial population is created, or in intermediate stages, or in the final population, or within the evolutionary process itself.

### 1. INTRODUCTION

Over the past decades, extensive research has been done on the scheduling problems: there is no easy algorithm, which can provide an optimal solution. Integer programming and branch and bound techniques can be used to find the optimal solution [3]. However, they are not very effective on large problems. Hence, many *heuristics* have been developed to provide a good and quick solution.

---

<sup>1</sup> The Research Group is supported by the Universidad Nacional de La Pampa.

<sup>2</sup> The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

Local search techniques try to continuously improve solutions initially obtained by constructive heuristics. Given a solution  $s$ , the *neighbourhood* of  $s$  is a set of solutions that can be derived by applying predefined slight modifications to  $s$ . The moving operators provide moves from one solution to another in the neighbourhood. The process continues until a termination criterion is fulfilled.

Many researchers have shown that EAs perform well for global searching because they are capable of quickly finding and exploiting promising regions of the search space, but they take a relatively long time to converge to a local optimum. The EA finds a good solution but requires many more generations to reach the optimal solution (or the best known solution). Local improvement procedures quickly find the local optimum of a small region of the search space, but are typically poor global searchers. These procedures do not guarantee optimality.

With the hybrid approach, evolutionary algorithms are used to perform global exploration among the population while other heuristic methods are used to perform local exploitation around chromosomes. Because of the complementary properties of evolutionary algorithms and local improvement procedures, the hybrid approach often outperforms either method operating alone.

## 2. HIBRYD EVOLUTIONARY ALGORITHMS

When designing a hybrid evolutionary algorithm, a fundamental principle is to hybridize where possible. This can be done in a variety of ways, including the following [2]:

- Incorporate heuristics into initialization to generate a well-adapted initial population. In this way, a hybrid evolutionary algorithm with elitism guarantees to do no worse than the conventional heuristic does.
- Incorporate a local search heuristic as an add-on to the basic loop of the genetic algorithm, working together with genetic operators, to perform quick and localized optimization in order to improve offspring before returning it to be evaluated.
- Improve the final population with local search heuristic.

## 3. SCHEDULING PROBLEMS

The above mentioned hybrid approaches have been used in the following scheduling problems:

- ✓ Flow shop scheduling (FSSP): consists of  $n$ -job and  $m$ -machine. A job is processed on one machine at a time without pre-emption, and a machine processes no more than one job at a time. Each job consists of  $m$  operations, and each operation requires a different machine.  $n$  jobs have to be processed in the same sequence on  $m$  machines. The processing time of job  $i$  on machine  $j$  is given by  $t_{ij}$  ( $i=1,\dots,n; j=1,\dots,m$ ). The objective is to find the sequence of jobs minimizing the maximum flow time, which is called *makespan* [2].
- ✓ Job shop scheduling (JSSP): consists of  $m$  different machines and  $n$  jobs. Only one job may execute on a machine at a time. All schedules and jobs are non-preemptive. Jobs can have distinct priorities and all of them are available at production initiating time. Each job visits all machines, only once, following a predetermined sequence of machines, called a route. Consequently a job can be seen as composed by various steps, called operations. The objective is to find a schedule of minimum makespan [2].
- ✓ Travelling salesman problem (TSP): attempts to find a minimum distance (or cost) for a hamiltonian circuit in a graph.

## 3. CONCLUSIONS AND FUTURE WORK

All the hybrid approaches improved results obtained by EAs with diverse extra effort. Besides different improvements to those evolutionary algorithms were incorporated through modified NEH [8] and CDS [1] heuristic algorithms, Simulated Annealing [6,5, 9, 10] and Tabu Search [4, 7, 11]. These hybridizations obtained whole populations closer to the optimum than the evolutionary algorithm alone. Further, the individuals in the final population represent different sequences. However, hybridized options needed a bigger number of evaluations of the objective function.

Further work is driven to the analysis of effects of the incorporation of new solutions into the population. These solutions can be obtained by the above mentioned techniques.

#### 4. References

- [1] Campbell, H., Dudek, R., Smith, M., A Heuristic Algorithm for the  $n$ -job  $m$ -machine Sequencing Problem, *Management Science*, vol. 16B, pp. 630-637, 1970.
- [2] Gen M., Cheng R., *Genetic Algorithms & Engineering Design*, Wiley Interscience, 1997.
- [3] Ignall, E., Schrage, L., Application of the Branch and Bound Techniques to Some Flow Shop Scheduling Problems, *Operations Research*, vol. 13, pp. 400-412, 1965.
- [4] Michalewicz, Z., Fogel, D.B., *How to Solve It: Modern Heuristics*, Germany: Springer-Verlag Heidelberg, 2000.
- [5] G. Minetti, R. Gallard, H. Alfonso, "Hybrid Multi-Inver-Over Evolutionary Algorithms for the Euclidean Travelling Salesman Problem", in proc. 4<sup>th</sup> International ICSC SOCO, Paisley, UK, 2001, pp. 134.
- [6] Minetti G., Gallard R., Alfonso H., Influence of Temperatures Settings for Simulated Annealing in a Hybrid Evolutionary Algorithm, in Proc. 5<sup>th</sup> World Multiconference on SCI 2001 and 7<sup>th</sup> International Conference ISAS, ISBN 980-07-7543-9, Orlando – USA July 2001, pp. 397-400.
- [7] Minetti G., Gallard R., Alfonso H., Bermudez C., Hybridizing Multi-Inver-Over Evolutionary Algorithms with Tabu Search for the Symmetric TSP, VII Congreso Argentino de Ciencias de la Computación – CACIC'2001, ISBN 987-96-288-6-1, Calafate – Argentina, October 2001, pp. 1141 a 1150.
- [8] Nawaz M., Enscore E., Ham I., A Heuristic Algorithm for the  $m$ -machine  $n$ -job Flow Shop Sequencing Problem, *Omega*, vol. 11, pp. 11-95, 1983.
- [9] Salto, C., Alfonso, H.A., and Gallard, R.H., Evolutionary Algorithms and Local Search to Solve Scheduling Problems, in proc. 4<sup>th</sup> International ICSC SOCO, Paisley, UK, 2001, pp. 131.
- [10] Salto, C., Alfonso, H.A., and Gallard, R.H., A Hybrid EA Approach Used to Solve the Job Shop Problem, Proc. 5<sup>th</sup> World Multiconference on SCI 2001 and 7<sup>th</sup> International Conference ISAS, Orlando, USA, 2001, pp. 424-428.
- [11] Fernandez, N., C. Salto, Alfonso, H. and Gallard, R.H., Incorporating Tabu Search for Local Search into Evolutionary Algorithms to Solve the Job Shop Scheduling Problem, in proc. VII Congreso Argentino de Ciencias de la Computación – CACIC'2001, Calafate, pp. 1151 a 1162.

## WEBEXPERT – TOOL FOR KNOWLEDGE ACQUISITION BY WEB

**Scholarship holder:Raphael Luiz Nascimento Advisor:Anita Ma. da Rocha Fernandes, Dr.**  
[raphael@inf.univali.br](mailto:raphael@inf.univali.br) [anita@inf.univali.br](mailto:anita@inf.univali.br)

UNIVALI – University of Itajaí Valley  
Uruguai Street 458, Itajaí city , Santa Catarina State - Brazil

### 1. Introduction

#### 1.1 Contextualization

Knowledge acquisition for systems based on knowledge can be called knowledge engineering, according to Feigenbaum (1980) who used the term to describe the reduction of a wide context of knowledge for a set of facts and rules. The term Knowledge Engineer started to be used to denote the responsible person by development of such systems.

The basic function of Knowledge Engineering is to play the role of mediator between the expert and the Knowledge Base, to extract the knowledge from the expert, to codify the same in contribution with the expert to reach an acceptable performance.

However, Knowledge Engineer in role of intermediary between the expert and the systems based on knowledge can face many problems, concerning the way as each expert reasons (Gaines, 1987). The computer appears as an excellent tool to help the expert to structuralize the knowledge domain and, the researches in knowledge acquisition have centralized the development of knowledge acquisition tools based on computer (Boose and Gaines, 1988; Boose, 1989). Many tools projected to be used directly by the expert with the minimum of intervention of Knowledge Engineer; they have emphasized the easiness for visualization of the domain concepts. The objective of systems such as PLANET (Shaw and Gaines, 1983; 1986; 1987), ETS (Boose, 1984; 1986), MOLE (Eshelman et al, 1987), SALT (Marcus, 1987), Kitten (Shaw and Gaines, 1987b), Knack (Klinker et al, 1987); Kriton (Direch, Ruhmann and May, 1987), KKSSO (Gaines, 1987b. Gaines and Shaw, 1986) is to fulfill the process of knowledge acquisition and to transfer it to the systems based on knowledge.

However, the tools of existing knowledge acquisition are widely based on PC'S and graphical Workstation, and the use of the same in a distributed community involves mobile software. The process of development and test of the models based on knowledge through a distributed community would be used sufficiently if the networks of great areas could be used to coordinate the activities in different sites. The initial objective of the paper that is presented here is to use the web to support the distributed knowledge acquisition by the development that operates in web. Then the technique of Personal Construction Psychology was used.

The option for this technique is due to Kelly's Personal Construction Psychology (1995) has been used for knowledge acquisition since the beginning of the development of systems based on knowledge (Shaw and Gaines, 1983; Boose, 1984) and has been refined throughout the years to support the increase of the complexity of knowledge structures (Gaines and Shaw, 1993).

#### 1.2 The problem

Despite the Knowledge Engineer has little knowledge domain that the expert, the communication problems prevent the transference process of knowledge for a program. The vocabulary used initially by the expert to speak about the domain with a layperson is generally inadequate for solution of problems, and then, the Knowledge Engineer and the expert have to work together to refine it. One of the most difficult aspects for the Knowledge Engineer is to help the expert to structuralize the knowledge domain, to identify and to legalize the concepts of the domain (Haynes – Roth, Waterman, Lenat, 1983).

The development of the systems based on knowledge involves knowledge acquisition starting from various sources generally different geographically. The sources include books, articles, manuals, videos with the performance of the expert, translation of protocols and interviews, and interactions man-machine with the expert. The time of the expert is generally a scarce resort and experts are, generally accessible in different places, particularly in international projects. The methodologies of knowledge acquisition, as well as the tools, have been developed to deal these problems by the use hypermedia systems to manage a wide amount of heterogeneous data, interactive graphical interfaces to present knowledge models in an understandable way to the experts. systems of

fast prototype production to test the models in operation, and systems of models comparison to call attention for the variations among experts.

Another important point in knowledge acquisition is the handling of incoherence. Depending on the form as the knowledge is acquired, errors of acquisition can have. These errors can result of own knowledge nature, as in data gotten through sensory that are subject to noise, or can be generated by the existing human interface between the real world and the representation system. Techniques had been developed to prevent errors of acquisition in what type of waited knowledge is defined. These techniques are common to the knowledge representation systems and to database management systems. On the other hand, a knowledge base can be examined periodically with the purpose to detect incoherence eventually introduced in the acquisition process. This method is limited by the fact of reasonable expressive representation languages do not count on full procedures of known verification. Finally, it must be observed that the adequacy of representation formalism to the type of knowledge of real world to be represented is basically for the efficiency of the acquisition process (Bittencourt, 1998).

The necessity of development of a knowledge acquisition tool is clear based previous displayed on, however several experts have different ideas and understandings for the same domain.

An application area of Personal Construction Psychology, which has been useful in many areas and professions, is social knowledge process modeling through comparison of developments and terminologies of different individuals in the same domain. The comparative analysis of repertory grid to become such similarities and differences exposed was one of the first computational applications of Personal Construction Psychology (Shaw, 1979, 1980) and the involved techniques have been presented throughout the years to supply more efficient tools of analysis (Shaw and Gaines, 1991a, 1991b, 1993).

Thus, this article presents the development of a knowledge acquisition tool for web, based on Kelly's Personal Construction Psychology, and it has as case study the areas of oceanography, biotechnology and environmental engineering.

So, some basic stages must be fulfilled: the study of Kelly's Personal Construction Psychology to establish the criteria to be used in repertory grid; establishment of the structure of database that will give support to the system; evaluation of interface diagramming in order to be friendly to the user originating from any area of knowledge; establishment of criteria for evaluation of prototype performance, during its case study in some domains: oceanography, biotechnology and environmental engineering.

## 1 Theoretical basis

A bibliography search about Personal Construction Psychology, knowledge acquisition, software utility engineering, PHP and MySQL was necessary to develop the Web Expert.

George Kelly created the Personal Construction Psychology in 1955 (Kelly, 1995). In a computational context, Kelly's approach is attractive because it directly deals with representations and algorithm, and its representation in terms of modern intentional logic leads to a model of human psychology process.

The methodology used by Kelly for the extraction of concept structures is the "repertory grid". Gaines and Shaw (1980) suggested that the repertory grid would be a useful technique for the development of systems based on knowledge. Many tools for knowledge acquisition have incorporated repertory grid as an extraction technique (Boose and Brash, 1987; Dierich, Rulmann and May, 1987; Garg-Janardan and Salvendy, 1987; Shaw and Gaines, 1987; Ford et al, 1990).

Regarding to the knowledge acquisition, it is not a phase of the systems development based on knowledge, but a component that enters in all phases, and since the definition of the problem it has persisted in the maintenance. Rook and Croghan (1989) show how the acquisition can be faced as being a component of engineering structuralized in knowledge systems. Therefore, it is also a very delayed task, which demands constant devotion of the expert and the knowledge engineer.

The knowledge acquisition typically involves actions to congregate information of one or more human experts and/or of documentary sources, arranging this information in some way and, then, translating it for one clear form by machine. Second (Breuker and Wielinga), knowledge acquisition is the process to transform data of expertise into implementation formalism.

The usability engineering was studied due to necessity of developing a friendly interface for the user. The development of Web Expert interface is based on Basteim & Scarpeen's software ergonomics concepts (1994).

To support the implementation of database and knowledge, as well as the interface by web, it was necessary a bibliographical study, beyond the study about PHP, MySQL and Flash.

PHP is a programming language called serve-side script to create dynamic sites. Using PHP, the direct interaction of the user with the site can be made, through forms, counters of access, statisticians, or creating

It is also possible to interact with database and existing applications in the server, with the advantage of no display the source code for the customer. This can be useful when the program is dealing with passwords or any type of confidential information.

Another advantage of PHP referring to other languages of this type is that, it is totally free, or better, we only must have an apache server to use it PHP makes interactions with several database and one of them is MySQL that is also free.

MySQL is a server of multi user database; multitask that works with one of the most popularized languages of manipulation of data of the world. According to Fernandes (2000), it has as features of MySQL: a) it supports different platforms: win32, Linux, FreeBSD, Unix, and others; b) it supports APIs of the following languages: PHP, Perl, C, C++, Java, Python, and others; c) it has support to multiprocessors; d) sophisticated system of flexible and safe cryptographic passwords; e) it has support to DDBC- the user can easily connect the Access to a database MySQL; f) it supports up to 16 index for table; g) source code is written in C and C++ and tested with a variety of different compilers.

Because it is a database totally compatible with language SQL Ansi, practical, fast, efficient and of easy handling and extremely reliable, maybe this is the reason that it is very used and it is being transformed quickly into the favorite server of many developers.

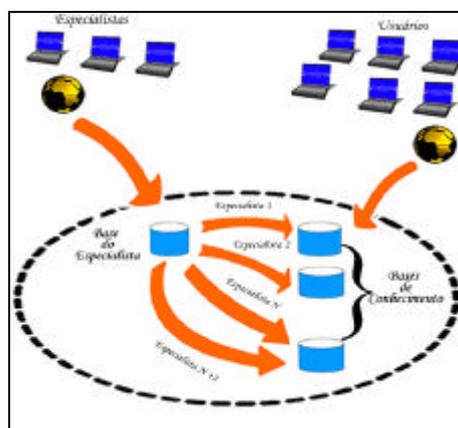
Flash graphical tool was also used because it is a tool of authorship and edition of vectorial pictures with animation, sound and interactive. Based on vectorial pictures, turning possible the creation of advanced effect in very small archives.

Beyond vectorial pictures, to Flash content can be added bitmap archives, digitalized sounds in .au and .wav, video formats, and even though archives with sequences of pictures – the animated GIFs.

The main advantage of Flash is the easy learning, and an enormous amount of tools enclosed in software, facilitating the programming.

## 2 The System

The basic architecture of the system consists of several experts connected in the system by Internet, set up in the base of experts, feeding the knowledge bases in order that the users in Internet can consult them. There will be two modules, the expert module and the user module. As show in figure 1.

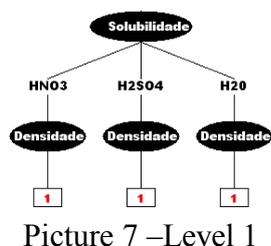


Picture1 – system structure

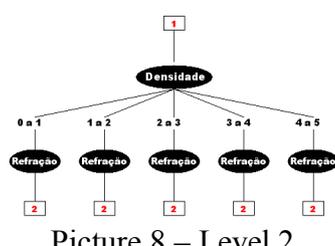
The current system is composed only by expert module. In this module the expert will be possible to set up in the base informing data referring his/her person, the expert can also inform his/her specialty and a brief professional curriculum. After the register is made the expert will be able to access the system using his/her login and his/her password. As shown in picture 2.

Expert's module possesses the following items: new Base: Where the Expert will be able to initiate his/her Knowledge base; my register: It will be able to make alterations in the register; accesses: Its actions of the system will be able all inside; chat: It will be able to debate subjects with other experts set up in register of the system; forum: It will be able to add questions or to answer questions about definitive subject; alter Password: It will be able to modify his/her password. The picture 3 shows Expert's menu interface.

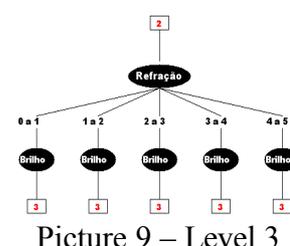




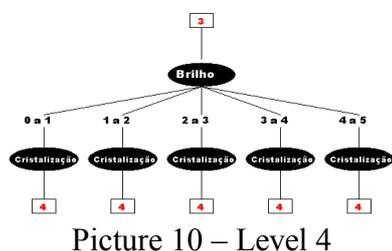
Picture 7 – Level 1



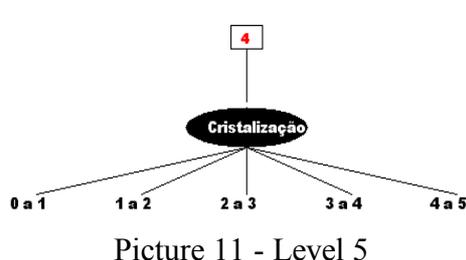
Picture 8 – Level 2



Picture 9 – Level 3



Picture 10 – Level 4



Picture 11 - Level 5

The knowledge tree will be shown in the picture format, made with the graphical library of PHP (gd-20.1). It will be set up independent of the number of variables. The root of the tree will contain the variable of bigger scale (5), and below it its sub-variables, after that one variable of scale will come to each one of sub-variables (4), after that the sub-variables will come, thus successively until arriving in the sub-variables of variable of scale (1).

In studies and tests made during the implementation of the tree it was verified that the same one was with its visualization harmed while the number of involved variables magnified. Being thus, for a better visualization of the tree, it was decided to break up it in n parts, where n is the number of involved variables in the process. To go from a level to another the user clicks in the connectors (squared).

### 3.2 Knowledge Modeling

After the construction of the tree the expert will be able to verify if his/her reasoning is correct through the rules of production generated with base in the tree of decision constructed.

The system has the tools of chat and forum in order to the expert of the same area can change ideas about the best refinement of knowledge in a determined area.

### 4 Conclusions

The system is still in development phase, being that only the expert module is constructed. The following stages say respect to the elaboration of content confrontation developed by experts in the same subject through Kelly's Personal Construction Psychology. Soon, after the ending of the construction of the production rules the using module will be constructed where students and professionals of the area will be able to visualize the rules and to use the system. And also the use of students of the discipline - Artificial Intelligence, to help in the understanding of the reasoning used for the specialist in determined subject.

### 5 Bibliographical References

- BITTENCOURT, G. **Inteligência Artificial – Ferramentas e Teorias**, Editora da Universidade Federal de Santa Catarina, 1998.
- DIEDERICH, J., RUHMANN, I. A knowledge acquisition tool for experts systems. **International Journal of Man-Machine Studies** 26 (1) 29-40. 1987
- ESHELMAN, L et all. MOLE: A tenacious knowledge acquisition tool. **International Journal of man-Machine Studies** 26 (1) 41-54, 1987
- GAINES, B.R. Induction of inference rules for expert systems. **Fuzzy Sets and Systems**, 18(3) 315-328, 1986
- HAYES-ROTH, F.; WATERMAN, D.A. and LENAT, D.B, Eds. **Building Expert Systems**. Reading, Massachusetts: Addison-Wesley. 1983
- MARCUS, S. Taking backtracking with a grain of SALT. **International Journal of Machine Studies** 26 (4) 393-398. 1987
- KLINKER, G et all. – report-driven knowledge acquisition. **International Journal of Man-Machine Studies** 26 (1) 65-79 1987

**INSERTING PROBLEM-SPECIFIC KNOWLEDGE IN MULTIRECOMBINED EVOLUTIONARY ALGORITHMS**

PANDOLFI D., DE SAN PEDRO M., VILLAGRA A. , VILANOVA G.

Proyecto UNPA-29/B032<sup>1</sup>

División Tecnología

Unidad Académica Caleta Olivia

Universidad Nacional de La Patagonia Austral

Ruta 3 Acceso Norte s/n

(9011) Caleta Olivia – Santa Cruz - Argentina

e-mail: {dpandolfi,gvilanov,edesanpedro,avillagra}@uaco.unpa.edu.ar

Phone/Fax : +54 0297 4854888

GALLARD R.

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)<sup>2</sup>

Departamento de Informática

Universidad Nacional de San Luis

Ejército de los Andes 950 - Local 106

(5700) - San Luis -Argentina

e-mail: rgallard@unsl.edu.ar

Phone: +54 2652 420823

Fax : +54 2652 430224

**ABSTRACT**

In the restricted single-machine common due date problem the goal is to find a schedule for the  $n$  jobs which jointly minimizes the sum of earliness and tardiness penalties, while for the weighted tardiness problem the goal is to find a schedule that minimizes the tardiness penalties. Both problems, even in their simplest formulations, are an NP-Hard optimization problem [4, 14]. This presentation discusses how problem specific knowledge is inserted into the evolutionary algorithm to enhance its performance.

**INTRODUCTION**

New trends to enhance evolutionary algorithms introduced *multiple-crossovers-on-multiple-parents* (MCMP) [5, 6, 7] a multirecombinative approach allowing multiple crossovers on the selected pool of (more than two) parents. MCMP-SRI [13] is a novel MCMP variant, which considers the inclusion of a stud-breeding individual in a pool of random immigrant parents. Members of this mating pool subsequently undergo multiple crossover operations.

The main objective of this new recombinative method is to find an equilibrium between exploration and exploitation in the search process. An individual of the old population is selected as the stud and subsequently mated with a set of new generated individuals (immigrants). The presence of the stud ensures to retain good features of previous solutions while the immigrants, as continuous source of genetic diversity, avoid premature convergence and make unnecessary to apply mutation.

Recent results are promising and showed its potential by finding good solutions, and in some instances improving the upper bound published by the OR-Library [2].

---

<sup>1</sup> The Research Group is supported by the Universidad Nacional de La Patagonia Austral.

<sup>2</sup> The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

This presentation shows different multirecombined approaches applied on two problems of single machine scheduling. In these approaches we use two different ways to introduce problem-specific knowledge to the algorithms. Both problems and details of implementation are discussed.

## 1. THE SINGLE-MACHINE COMMON DUE DATE PROBLEM

In the restricted single-machine common due date problem (GWET),  $n$  jobs with deterministic processing times must be processed attempting to conform a common due date, thus minimizing penalties imposed to early and tardy jobs.

The problem can be stated as follows: A set of  $n$  jobs with deterministic processing times  $p_i$  and a common due date  $d$  is given. The jobs have to be processed on one machine. For each of the jobs an individual earliness  $\alpha_i$  and tardiness  $\beta_i$  penalties are given. The goal is to find a schedule for the  $n$  jobs which jointly minimizes the sum of earliness and tardiness penalties.

Even simple in the formulation, this model leads to an optimization problem that is NP-Hard [4], and can be precisely stated as defined in [11]:

$$\min \sum_{i=1}^n (\alpha_i E_i + \beta_i T_i) \text{ where}$$

$$E_i = \max \{0, d - c_i\}, T_i = \max \{0, c_i - d\}$$

and  $c_i$  is the completion time of job  $j_i$

An optimal schedule for GWET problem [1] is V-shaped around the due date. A chromosome contains  $n$  binary bits, each of which indicates one of two sets (the tardy job set T and the nontardy job set E) a corresponding job belongs to.

## 2. THE WEIGHTED TARDINESS SCHEDULING PROBLEM

The single-machine total weighted tardiness problem [3] can be stated as follows:  $n$  jobs are to be processed without interruption on a single machine that can handle no more than one job at a time. Job  $j$  ( $j = 1, \dots, n$ ) becomes available for processing at time zero, requires an uninterrupted positive processing time  $p_j$  on the machine, has a positive weight  $w_j$ , and a due date  $d_j$  by which it should ideally be finished. For a given processing order of the jobs, the earliest completion time  $C_j$  and the tardiness  $T_j = \max\{C_j - d_j, 0\}$  of job  $j$  can readily be computed. The problem is to find a processing order of the jobs with minimum total weighted tardiness,

$$\sum_{j=1}^n w_j T_j$$

Even with this simple formulation, this model also leads to an optimization problem that is NP-Hard [3]. For this problem there are heuristics which provide good solutions. We decided to use three of the best ones [14]. These heuristics are described below.

- Rachamadagu and Morton Heuristic (R&M). This heuristic provides a schedule according to the following calculation.

$$\pi_j = (w_j / p_j) [\exp\{-(S_j)^+ / k p_{av}\}]$$

where  $S_j = [d_j - (p_j + Ch)]$  is the slack of job  $j$  at time  $Ch$ , and  $Ch$  is the total processing time of the jobs already scheduled,  $k$  is a parameter of the method (usually  $k = 2.0$ ) and  $p_{av}$  is the average processing time of jobs competing for top priority. In the R&M heuristic, also called the Apparent Tardiness Cost heuristic, jobs are scheduled one at a time and every time a machine becomes free

a ranking index is computed for each remaining job. The job with the highest ranking index is then selected to be processed next.

- The Covert rule. This heuristic provides a schedule according to the following calculation.

$$\pi_j = (w_j / p_j) \{1 - (S_j)^+ / kp_j\}^+$$

Under this heuristic the WSPT (Weighted Shortest Processing Time first) rule is modified by an slack factor, and processing times different than the job being considered are not taking into account. The Covert rule works, in the case of a single resource similar to R&M.

- Modified R&M heuristic. Here a logarithmic function of the slack of the job considered and the average processing times of remaining jobs is used,

$$\pi_j = (w_j / p_j) [\ln\{-(S_j)^+ / kp_{av}\}]$$

Note that  $k$  is a free parameter and it was experimentally determined for each problem class. The seeds (schedules) provided by each heuristic were determined for each instance before the EA begins running.

### 3. MULTIRECOMBINED APPROACHES

In our attempt to achieve a better balance between exploration and exploitation we devised MCMP-SRI[13, 10], where a permutation based representation was adopted. Here, the process for creating offspring is performed as follows. From the old population an individual, designated the stud, is selected by means of proportional selection. The number of  $n_2$  parents in the mating pool is completed with randomly created individuals (random immigrants). The stud mates every other parent, the couples undergo partial mapped crossover (PMX) and  $2 \cdot n_2$  offspring are created. After crossover the best of these  $2 \cdot n_2$  offspring is stored in a temporary children pool. The crossover operation is repeated  $n_1$  times, until the children pool is completed. Finally, the best offspring created from  $n_2$  parents and  $n_1$  crossover is inserted in the new population. Children are not exposed to mutation because the random immigrants provide the necessary genetic diversity to the mating pool.

The idea of inserting seeds (good solutions) in evolutionary processes was originally implemented by Reeves [15] as an alternative way to introduce problem-specific knowledge to the algorithm. In his approach Reeves, inserted one seed provided by a non evolutionary heuristic, only once in the initial population, expecting that its genetic material were occasionally exchanged by means of the selection mechanism. In the weighted tardiness problem the seeds are always present in the mating pool for each new generation and consequently, because they mate the stud, their contribution is guaranteed. The latter we called MCMP-SRSI. The three seed immigrants are obtained at the beginning of the process by means of the conventional heuristics described above in section 2.

Attempting to insert problem-specific-knowledge in the *single-machine common due date problem* we decided to follow Lee and Kim [12] proposal giving rise to MCMP-V[9] and MCMP-SRI-V [8]. Regarding representation, a chromosome can suitably be represented by a binary string of  $n$  bits. Here, bit indicates one of two sets a corresponding job belongs to (1 represents the tardy job set T, and 0 represents the non-tardy job set E). Afterwards the jobs are sequenced in set E in non-increasing order of  $p_i / \alpha_i$  and jobs in set T in non-decreasing order of  $p_i / \beta_i$  to form a V-shaped schedule. It is important to note that using the conventional permutation representation the algorithm is compelled to search in a problem space of size  $n!$ . In MCMP-V, the process for creating the new population from the old population is performed as follows. Each time an offspring is to be inserted in the new population,  $n_2$  parents selected from the old population undergo  $n_1$  crossover operations (Uniform Scanning Crossover). Each crossover operation generates a single offspring, which eventually undergoes mutation. The Uniform Scanning Crossover (USX) can be safely used under this representation and is performed as follows: each gene in the child is provided from any of the corresponding alleles in the parents with equal probability.

Finally, MCMP-SRI-V is combination of both previously explained approaches, using binary representation and Uniform Crossover(UX). From the recombination point of view this method is similar to MCMP-SRSI and from the representation point of view this method is similar to MCMP-V.

#### 4. CONCLUSIONS

In the single machine common due date problem inserting problem-specific-knowledge through this representation the problem space size is reduced from  $n!$  to  $2n$  and consequently obtained better results than other approaches with representation based in permutation.

In weighted tardiness problem inserting seeds in the mating pool combined with random immigrants also best results are obtained.

Inserting problem specific knowledge in evolutionary algorithms, not only improve the quality of results, but also with less computational effort leads the algorithm to promising search sub-spaces avoiding a large search.

#### 5. ACKNOWLEDGEMENTS

We acknowledge the co-operation of the project group for providing new ideas and constructive criticisms. Also to the Universidad Nacional de San Luis, the Universidad Nacional de La Patagonia Austral, and the ANPCYT from which we receive continuous support.

#### 6. REFERENCES

- [1] Baker K. and Scudder G., Sequencing with earliness and tardiness penalties: A review, *Operations Research*, Vol. 38, pp 22-36, 1990.
- [2] Beasley J. E. Scheduling Problems, OR Library, <http://mscmga.ms.ic.ac.uk/jeb/orlib>.
- [3] M. Pinedo, "Scheduling: Theory, Algorithms and System." First edition Prentice Hall, 1995.
- [4] Chen T. and Gupta M., Survey of scheduling research involving due date determination decision, *European Journal of Operational Research*, vol 38, pp. 156-166, 1989.
- [5] Eiben A.E., Raué P-E., and Ruttkay Zs., *Genetic algorithms with multi-parent recombination*. In Davidor, H.-P. Schwefel, and R. Männer, editors, Proceedings of the 3rd Conference on Parallel Problem Solving from Nature, number 866 in LNCS, pages 78-87. Springer-Verlag, 1994
- [6] Eiben A.E., van Kemenade C.H.M., and Kok J.N., *Orgy in the computer: Multi-parent reproduction in genetic algorithms*. In F. Moran, A. Moreno, J.J. Merelo, and P. Chacon, editors, Proceedings of the 3rd European Conference on Artificial Life, number 929 in LNAI, pages 934-945. Springer-Verlag, 1995.
- [7] Eiben A.E. and. Bäck Th., An empirical investigation of multi-parent recombination operators in evolution strategies. *Evolutionary Computation*, 5(3):347-365, 1997.
- [8] Pandolfi D.; Vilanova G.; De San Pedro M.; Villagra A. Gallard R.: Solving the Single-Machine Common Due Date Problem Via Stud and Immigrants in Evolutionary Computation vol III pp 409-413 World Multiconference on Systemics, Cybernetics and Informatics Orlando 2001.
- [9] Pandolfi D.; Vilanova G.; De San Pedro M.; Villagra A. Gallard R.: Evolutionary algorithms to minimize earliness-tardiness penalties from a common due date vol III pp 409-413 World Multiconference on Systemics, Cybernetics and Informatics Orlando 2001
- [10] Pandolfi D., De San Pedro M., Villagra A, Vilanova G., Gallard R., Studs and Immigrants in Multirecombined Evolutionary Algorithm to face weighted tardiness Scheduling Problems. vol II pp.1251 Proceedings of the 7th Argentine Congress of Computation Sciences 2001, El Calafate 2001.
- [11] Hall N. and Posner M., Earliness Tardiness Scheduling problems: weighted deviation of completion times about a common due date. *Operations Research* vol 39 pp 836-846 1991.

- [12] Lee C. and Kim S., Parallel Genetic Algorithms for the tardiness job scheduling problem with general penalty weights, *International Journal of Computers and Industrial Engineering*, vol. 28, pp. 231-243, 1995
- [13] Pandolfi D., Vilanova G., De San Pedro M., Villagra A. Multirecombining studs and immigrants in evolutionary algorithm to face earliness-tardiness scheduling problems. *Proceedings of the International Conference in Soft Computing*. University of Paisley, Scotland, U.K., June2001.
- [14] Morton T., Pentico D., *Heuristic scheduling systems*, Wiley series in Engineering and technology management. John Wiley and Sons, INC, 1993.
- [15] Reeves C., A genetic algorithm for flow shop sequencing, *Computers and Operations Research*, vol 22, pp5-13, 1995.

## PERFORMANCE EVALUATION OF SELECTION METHODS TO SOLVE THE JOB SHOP SCHEDULING PROBLEM

Stark N., Salto C., Alfonso H.  
Proyecto UNLPAM-09/F015<sup>1</sup>  
Departamento de Informática - Facultad de Ingeniería  
Universidad Nacional de La Pampa  
Calle 110 esq. 9  
(6360) General Pico – La Pampa – Rep. Argentina  
e-mail: { stark, saltoc, alfonsoh }@ing.unlpam.edu.ar  
Phone: +54 2302 422780/422372, Ext. 6302

Gallard R.  
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)<sup>2</sup>  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106  
(5700) - San Luis -Argentina  
e-mail: rgallard@unsl.edu.ar  
Phone: +54 2652 420823  
Fax : +54 2652 430224

### ABSTRACT

In evolutionary algorithms selection mechanisms aim to favour reproduction of better individuals imposing a direction on the search process. It does not create new individuals; instead it selects comparatively good individuals from a population and typically does it according to their fitness. The idea is that interacting with other individuals (competition), those with higher fitness have a higher probability to be selected for mating. In that manner, because the fitness of an individual gives a measure of its “goodness”, selection introduces the influence of the fitness function to the evolutionary process. Moreover, selection is the only operator of genetic algorithm where the fitness of an individual affects the evolution process. In such a process two important, strongly related, issues exist: selective pressure and population diversity.

In this work we are showing the effect of applying different selection mechanisms to a set of instances of the Job Shop Scheduling Problem, with different degrees of complexity. For these experiments we are using multiplicity features in the selection of parents for the reproduction with the possibility to generate multiple number of children too, because the results using these approaches outperform to those obtained under traditional evolutionary algorithms. This was shown in our previous works.

A description of each method, experiments and preliminary results under different combinations are reported.

---

<sup>1</sup> The Research Group is supported by the Universidad Nacional de La Pampa.

<sup>2</sup> The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

## 1. INTRODUCTION

By simulating evolution, an Evolutionary Algorithm (EA) maintains a population of individuals (*chromosomes*) which evolve throughout generations by reproduction of the fittest individuals. After initialisation, to create the original population of individuals, an EA consists of a selection-recombination-mutation cycle until a termination criterion holds.

*Selection*, *crossover* and *mutation* are the main operators repeatedly applied throughout the EA execution used to modify individual features. So, it is expected that evolved generations provide better and better individuals (searchers in the problem space).

A well known property of a selection operator is *selective pressure* which can be defined as the probability of the best individual being selected relative to the average probability of selection of all individuals.

During the selection step of an EA, copies of better ones replace worst individuals. Consequently, part of the genetic material contained in these worst individuals disappears forever. This *loss of diversity* is defined as the proportion of the population that is not selected for the next generation [3].

When the selection mechanism imposes a strong selective pressure then the loss of diversity can be high and, to prevent a premature convergence to a local optimum then, either a larger population size or adequate crossover and mutation operators are needed. On the other side of the coin a small selective pressure can excessively slow the convergence rate.

In the last years a variety of evolutionary schedulers based on genetic algorithms have been reported in the literature. In general, the task of scheduling is the allocation of jobs over time when limited resources are available, where a number of objectives should be optimized, and several constraints must be satisfied. A job is determined by a predefined set of operations, and the result of a scheduling algorithm is a schedule that contains the start times and allocation of resources to each operation [2].

Improvements in evolutionary algorithms have been found by using a multiplicity feature [6, 7, 8, 9], which allows multiple recombination on multiple parents. A multiplicity approach is *Multiple crossovers on multiple parents* (MCMP) [8,9] which supplies a balance in exploitation and exploration because the searching space is efficiently exploited (by the multiple application of crossovers) and explored (by a greater number of samples provided by multiple parents). MCMP provides a means to exploit good features of more than two parents selected according to their fitness by repeatedly applying one of the *scanning crossover* (SX) variants [4,5]: a number  $n_1$  of crossovers is applied on a number  $n_2$  of selected parents. From the  $n_2$  produced offspring a number  $n_3$  of them are selected, according to some criterion, to be inserted in the next generation. This is the method used in all our experiments.

MCMP was successfully applied to single and multicriteria optimisation and to the *job shop scheduling problem* (JSSP) under various representations, such as: decoders, dispatching rule representation and operation based representation [11,12, 13, 14, 15]. It was shown that a greater number of crossovers for a given number of parents provide better results. Another representation is *job based representation*, which consists of a list of jobs and a schedule is constructed according to the sequence of jobs. Here we deal with permutations, and then adequate genetic operators should be used. Due to the use of permutations in the chromosome representation and aiming to apply MCMP, we considered the modified SX methods [5] in order to produce feasible offspring. Furthermore, another crossover method, specially designed for this kind of representation, is the *Adjacency Based Crossover* (ABC). For the two crossover methods we used the variants *Uniform* (U-XX) (all the parents have the same chance to be donors) and *Occurrence-Based* (OB-XX) (the gene values is selected according to the value occurrence).

In next sections, the selection mechanisms used are described, and finally implementation details and some previous results are presented.

## 2. COMMONLY USED SAMPLING MECHANISMS

For the following discussion we concentrate on EAs applied to search (optimization) problems. Here it is convenient to adopt the notation used by Bäck [1]. Let us call  $I$  the space of individuals  $a \in I$  and  $f: I$

→  $\mathcal{R}$  a real-valued fitness function (maximization and minimization are equivalent). Let be  $\mu$  the population size and  $P(t) = (a_1^t, \dots, a_\mu^t) \in I^\mu$  a population at generation  $t$ .

### 2.1. Proportional Selection

In proportional selection, an individual  $a_i$  is chosen at random for mating from a population of size  $\mu$  according to the following probability:

$$P_{sel}(a_i) = \frac{f(a_i)}{\sum_{j=1}^{\mu} f(a_j)}$$

This is the simplest selection scheme also known as *roulette-wheel selection* or *stochastic sampling with replacement*. Here, individuals are mapped to contiguous segments in the real interval  $[0,1]$  in such a way that a segment corresponding to an individual has a size equal to the individual fitness. Then a random number in such interval is generated and the individual whose segment encompasses the random number is selected.

### 2.2. Rank-based selection

A first approach was called *linear ranking*, where the selective pressure can be controlled more directly than using the proportional selection and consequently the search process can be accelerated remarkably. Whitley [16] pointed out that ranking acts as a function transformation assigning a new fitness value to an individual based on its performance relative to other individuals. The Baker's original linear ranking method assigns a selection probability that is proportional to the individual's rank. Here, according to Bäck [1] the mapping  $rank: I \rightarrow \{1, \dots, \mu\}$  is given by:

$$\begin{aligned} \forall i \in \{1, \dots, \mu\}: rank(a_i) = i &\Leftrightarrow \\ \forall j \in \{1, \dots, \mu-1\}: f(a_j) &\leq f(a_{j+1}) \end{aligned}$$

where  $\leq$  denotes the  $\leq$  relation or the  $\geq$  relation for minimization or maximization problems, respectively. Consequently the index  $i$  of an individual  $a_i$  denotes its rank. Hence, individuals are sorted according to their fitness resulting  $a_1$  the best individual and  $a_\mu$  the worst one. Assuming that the expected value for the number of offspring to be allocated to the best individual is  $\eta_{max} = \mu P(a_1)$  and that to be allocated to the worst one is  $\eta_{min} = \mu P(a_\mu)$  then

$$P_{sel}(a_i) = \frac{1}{\mu} \left( \eta_{max} - (\eta_{max} - \eta_{min}) \cdot \frac{i-1}{\mu-1} \right)$$

As the following constraints must hold

$$\begin{aligned} P_{sel}(a_i) &\geq 0 \quad \forall i \\ \sum_{i=1}^{\mu} P_{sel}(a_i) &= 1 \end{aligned}$$

it is required that:  $1 \leq \eta_{max} \leq 2$  and  $\eta_{min} = 2 - \eta_{max}$

The selective pressure can be adjusted by varying  $\eta_{max}$ . As remarked by Baker [2] if  $\eta_{max} = 2.0$  then all individuals would be within 10% of the mean and the population is driven to convergence during every generation. To restrain selective pressure, Baker recommended a value of  $\eta_{max} = 1.1$ . This value for  $\eta_{max}$  close to 1 leads to  $P_{sel}(a_i) \cong 1/\mu$ , almost the case of random selection.

### 2.3. Tournament Selection

In tournament selection  $q$  individuals are randomly chosen from the population and then the best fitted individual, designated as the winner, is selected for the mating pool. The parameter  $q$  is known as the tournament size and usually it is fixed to  $q = 2$  (binary tournament). If  $q = 1$  then there is no selection at

all: each individual has the same probability to be selected. As long as  $q$  increases the selective pressure is augmented.

As Blicke [3] affirms, tournament selection can be implemented efficiently having the time complexity  $O(\mu)$  because no sorting of the population is necessary but, as a counterpart, this also leads to high variance in the expected number of offspring resultant from  $\mu$  independent trials.

As showed by Bäck [1], the selection probability for individual  $a_i$ , ( $i \in \{1, \dots, \mu\}$ ) for  $q$ -tournament selection is given by

$$P_{sel}(a_i) = \frac{1}{\mu^q} \left( (\mu - i + 1)^q - (\mu - i)^q \right)$$

#### 2.4. Stochastic Universal Sampling (SUS)

The idea, introduced by Baker [2], is to make a single draw from a uniform distribution and use it for determining the exact number of copies from each parent.

In this method the individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in proportional selection. Here equally spaced pointers are placed over the line as many as individuals have to be selected. Consider  $n$  the numbers of the individuals to be selected, then the distance between the pointers are  $1/n$  and the position of the first pointer is given by a randomly generated number in the range  $[0, 1/n]$ .

### 3. EXPERIMENT DESCRIPTIONS

The experiments were developed for a set of selected instances of the Job Shop Scheduling Problem (JSSP), under permutation representation. For each experiment series of fifty runs was performed. Experiments corresponded to different  $(n_1, n_2)$  combinations, crossover methods and selection mechanisms. Elitism to retain the best-valued individual was implemented. The population size was fixed at 100 individuals. USX, OBSX, OBABC and UABC were implemented and for insertion in the next generation the best child was chosen ( $n_3 = 1$ ). Number of crossovers and parents were set to:  $1 \leq n_1 \leq 4$  and  $3 \leq n_2 \leq 5$ , respectively. For mutation an *interchange* operator was used. The algorithms evolved for a minimum of 500 generations, after that a control of the mean fitness population progress began: if this value remained within a determined range for 20 consecutive generations the algorithm stops. Probabilities for crossover and mutation were fixed at 0.7 and 0.2, respectively. These values were determined as the best combination of probabilities after many initial trials. Proportional selection and SUS were applied in the conventional way. In the case of raking selection, setting value of  $\eta_{max}$  was 1.1. In the case of tournament selection, the size  $q$  of the set of competing individuals was fixed to 2. Four instances [10], with known optimal makespan were used: *la01*, *la06*, *la12* and *la15*.

### 4. PRELIMINARY RESULTS AND CONCLUSIONS

This contribution explained the behaviour of multirecombinative approaches (MCMP) and selection methods used to solve the JSSP for a set of instances of distinct complexity using a permutation representation.

After a long series of experiments a general overview indicates, independently of the crossover method used, a better performance of uniform approaches than occurrence approaches for any  $(n_1, n_2)$  association. Particularly, comparing results obtained on both USX and UABC, the first one exhibits a better quality of results.

Contrasting the different selection method the values achieved, show a better performance for both SUS and Proportional Selection over the other methods (without a clear conclusion on which one of these outperform the other).

In general, when 4 crossovers are applied better results than when a single crossover is applied are obtained.

The promising results on the smallest instances encourage us to deep investigation. To improve results in larger instances, further work include self adaptation of parameters such as  $(n_1, n_2)$  associations and study the adjustment of selection method according to the evolution.

## 5. REFERENCES

- [1] Bäck T: *Evolutionary algorithms in theory and practice*. Oxford University Press, 1996.
- [2] Baker J. E.: *Reducing bias and inefficiency in the selection algorithm*. Proc. of the 2<sup>nd</sup> International Conf. on Genetic Algorithms, pp. 14-21. J.J Grefenstette Ed, 1987.
- [3] Blickle T., Thiele L.: *A comparison of selection schemes used in genetic algorithms*. Technical Report 11 TIK, Swiss Federal Institute of Technology, December 1995.
- [4] Eiben A.E., Van Kemenade CHM, Kok J.N.: *Orgy in the computer: multi-parent reproduction in genetic algorithms*, in Moran F., Moreno A., Merlo J.J., and Chacon P., editors, Proc. of the 3<sup>rd</sup> European conference on Artificial Life, number 929 in LNAI, pp. 934-945. Springer Verlag, 1995.
- [5] Eiben A.E., Bäck T.: *An empirical investigation of multi-parent recombination operators in evolution strategies*, Evolutionary Computation, 5(3), pp.347-365, 1997.
- [6] Esquivel S., Leiva A., Gallard R.: *Multiple crossover per couple in genetic algorithms*. Proc. of the 4<sup>th</sup> IEEE International Conf. on Evolutionary Computation (ICEC'97), pp 103-106, Indianapolis, USA, April 1997.
- [7] Esquivel S., Leiva A., Gallard R.: *Couple Fitness Based Selection with Multiple Crossover Per Couple in Genetic Algorithms*. Proc. of the International Symposium on Engineering of Intelligent Systems, University of La Laguna, España, Tenerife, Vol. 1, pp. 235-241, ISBN 3-906454-12-6, Feb. 1998.
- [8] Esquivel S., Leiva A., Gallard R.: *Multiple Crossovers between Multiple Parents to Improve Search in Evolutionary Algorithms*, en Proc. of the 1999 Congress on Evolutionary Computation (IEEE), vol. 2, pp. 1589-1594, Washington DC., 1999.
- [9] Esquivel S., Leiva A., Gallard R.: *Multiplicity in Genetic Algorithms to Face Multicriteria Optimization*, Proc. of the 1999 Congress on E. C., IEEE Service Center, pp. 85-90, Washington, D.C., 1999.
- [10] Lawrence S.: *Resource Constrained Project Scheduling: an Experimental Investigation of Heuristic Scheduling Techniques*, (Supplement), Graduate School of Industrial Administration, Carnegie-Mellon University, Pittsburgh, Pennsylvania, 1984.
- [11] Minetti G., Salto C., Alfonso H., Gallard R.: *A Comparison of two Multirecombined Evolutionary Algorithms for the Job Shop Scheduling Problem*, Proc. VI Congreso Argentino de Ciencias de la Computación, CACIC 2000 Universidad Nacional de la Patagonia San Juan Bosco, 2000.
- [12] Salto C., Alfonso H., Gallard R.: *Multiplicity and Incest Prevention in Evolutionary Algorithms to Deal with de Job Shop Problem*, Proc. of the Second ICSC Symposium on Engineering of Intelligent Systems, University of Paisley, Scotland, pp. 451-457, 2000.
- [13] Salto C., Minetti G., Alfonso H., Gallard R.: *A Hybrid Evolutionary Algorithm: Multirecombination with Priority Rule Base Representation for the Job Shop Scheduling Problem*, Proc. VI Congreso Argentino de Ciencias de la Computación, CACIC 2000, Universidad Nacional de la Patagonia San Juan Bosco, pp. 1365-1376, 2000.
- [14] Salto C., Alfonso H., Gallard R.: *Breeding in multirecombined evolutionary algorithms with permutation based representation for the job shop scheduling problem*, Proc. of 4<sup>th</sup> International ICSC Symposium on soft Computing and Intelligent Systems for Industry (SOCO/ISFI2001), pp. 130, 2001.
- [15] Salto C., Alfonso H., Gallard R.: *Performance Evaluation of Evolutionary Algorithms under Different Representations when Solving the JSS problem*, Proc. of 5<sup>th</sup> World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2001), pp. 424-428, 2001.
- [16] Whitley D.: *The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best*. Proc. of the 3<sup>rd</sup> International Conf. on Genetic Algorithms, pp. 116-121. Morgan Kaufmann 1989.

MULTIRECOMBINING RANDOM AND SEED IMMIGRANTS IN  
EVOLUTIONARY ALGORITHMS TO FACE THE FLOW SHOP SCHEDULING PROBLEM

Vilanova G. Villagra A., Pandolfi D., De San Pedro M,  
Proyecto UNPA-29/B032<sup>1</sup>  
Unidad Académica Caleta Olivia  
Universidad Nacional de La Patagonia Austral  
(9011) Caleta Olivia – Santa Cruz - Argentina  
e-mail: [gvilanov@satlink.com](mailto:gvilanov@satlink.com), {[avillagra](mailto:avillagra@uaco.unpa.edu.ar),[dpandolfi](mailto:dpandolfi@uaco.unpa.edu.ar),[edesanpedro](mailto:edesanpedro@uaco.unpa.edu.ar)}@uaco.unpa.edu.ar  
Phone/Fax : +54 297 4854888

Gallard R.  
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)<sup>2</sup>  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106  
(5700) - San Luis – Rep. Argentina  
e-mail: [rgallard@unsl.edu.ar](mailto:rgallard@unsl.edu.ar)  
Phone: + 54 652 20823 Fax: +54 652 30224

## ABSTRACT

In an  $m$ -machines  $n$ -jobs *flow-shop sequencing problem* each job consists of  $m$  operations and each operation requires a different machine, so  $n$  jobs have to be processed in the same sequence on  $m$  machines. The processing time of each job on each machine is given. Frequently, the main objective is to find the sequence of jobs minimizing the maximum flow time, which is called the *makespan*. The flow-shop problem has been proved to be NP-complete.

Evolutionary algorithms (EAs) have been successfully applied to solve scheduling problems. Improvements in evolutionary algorithms consider multirecombination, allowing multiple crossover operations on a pair of parents (MCPC, *multiple crossovers per couple*) or on a set of multiple parents (MCMP, *multiple crossovers on multiple parents*). MCMP-STUD and MCMP-SRI are novel MCMP variants, which considers the inclusion of a *stud-breeding individual* as a *seed* in a pool of random immigrant parents. Random immigrants provide genetic diversity while seed-immigrants afford the knowledge of some conventional robust heuristics. Members of the mating pool subsequently undergo multiple crossover operations.

Another question in a multirecombined EA is the setting of parameters  $n_1$  (number of crossovers) and  $n_2$  (number of parents). In the experiments conducted they were empirically determined, by a deterministic rule or by self adaptation of parameters  $n_1$  and  $n_2$ . In the last case the idea is to code the parameters within the chromosome and undergo genetic operations. Hence it is expected that better parameter values be more intensively propagated.

**KEYWORDS:** Evolutionary algorithms, Multiple Crossovers, Multiple Parents, Flow Shop Scheduling Problem.

## 1. INTRODUCTION.

The flow-shop problem has been proved to be NP-complete [17]. Hence conventional and evolutionary heuristics have been developed by many researchers to solve the FSSP. Tsujimura et al [21] pro-

<sup>1</sup> The Research Group is supported by the Universidad Nacional de La Patagonia Austral.

<sup>2</sup> The LIDIC is supported by the Universidad Nacional de San Luis and the ANPCYT (National Agency to Promote Science and Technology).

vided evidence of the performance of (EAs) contrasted with conventional heuristics (CDS, Gupta, Palmer, etc.). [10,14,15].

By means of multirecombination (MCPC and MCMP), [6,7] better results were achieved. This implies higher quality of the best solution found throughout the evolutionary process, as well as an improved final population surrounding near optimal solutions. This later property also provides a sort of fault tolerance, because if eventually the dynamics of the system impedes using the best solution found then a better set of alternative solutions are available. The multirecombined methods were applied to FSSP and contrasted on a series of suitable experiments against previous successful approaches of Tsujimura and Reeves [1,21, 18].

Reeves proposed the idea of inserting seeds (good solutions) in evolutionary processes [18] as an alternative way to introduce problem-specific knowledge to the algorithm. In his approach Reeves, inserted one seed provided by a non evolutionary heuristic, only once in the initial population, expecting that its genetic material were occasionally exchanged by means of the selection mechanism.

Two novel MCMP variants were implemented, MCMP-STUD and MCMP-SRI [22,16], which considers the inclusion of a *stud-breeding individual* as a *seed* in a pool of random immigrant parents. In the case of MCMP-SRI, the stud (breeding individual) was generated by the CDS heuristics and the rest of the members of the parents pool are random immigrant. Random immigrants provide genetic diversity while the seed-immigrant afford the knowledge of some conventional robust heuristics. Members of the mating pool subsequently undergo multiple crossover operations.

In a multirecombined EA the setting of parameters  $n_1$  (number of crossovers) and  $n_2$  (number of parents) remained as an open question [5,8,9,11,19]. Two parameter control alternatives during the evolutionary processes were implemented: (DPC, *Deterministic Parameter Control*), the parameters are determined by applying a deterministic rule and (SPC Self Adaptation parameter Control) which advises to dynamically update parameters of the algorithm by evolving them as part of the chromosome structure [24].

In what follows we discuss these new multi-recombined methods, and show conclusions when they were applied to find the minimum makespan for selected instances of the FSSP. All approaches were tested for six Taillard's benchmarks [20] for FSSP. We selected four instances for each of the following problem sizes: 20x5, 20x10, 20x20, 50x5, 50x10, and three instances for the 50x20 problem size. For each instance a series of ten runs were performed.

## 2. MULTIRECOMBINATION OF RANDOM AND SEED IMMIGRANTS WITH THE STUD

Among other heuristics [13, 17], evolutionary algorithms have been successfully applied to solve scheduling problems [18, 21]. New trends in evolutionary algorithms make use of multiparent [3, 4] and multi-recombinative approaches [6, 7]. The later called, *multiple-crossovers-on-multiple-parents* (MCMP). Instead of applying crossover once on a pair of parents this feature applies  $n_1$  crossover operation on a set of  $n_2$  parents. Extreme exploitation can lead to premature convergence and intense exploration can make the search ineffective [12]. In order to improve the balance between exploration and exploitation in the search process was incorporated the use of a breeding individual (stud) which repeatedly mates individuals that randomly immigrates to a mating pool. The approaches are known as MCMP-STUD and MCMP-SRI respectively [22, 16, 24]. Under the last approach the random immigrants incorporate exploration (making unnecessary the use of mutation operations) and the multi-mating operation with the stud incorporates exploitation to the search process.

In MCMP-STUD [22, 23, 24], a mating pool is created by selection of  $n_2$  individuals from the old population. Then the parent with minimum makespan (stud) mates every other parent in the pool. At that time *partially mapped crossover* (PMX) is applied to each couple and from the new offspring, after eventual mutation, the best one is selected for insertion in the next generation. The members of this mating pool subsequently undergo multiple crossover operations. The setting of parameters  $n_1$  (number of crossovers) and  $n_2$  (number of parents) were determined during the processes evolution.

In MCMP-SRI [16,23,24], the process for creating offspring is performed as follows. From the old population an individual, assumed as the stud, is selected by means of proportional selection. The number of  $n_2$  parents in the mating pool is completed with randomly created individuals (random immigrants). The stud mates every other parent, the couples undergo crossover and  $2*n_2$  offspring are created. The best of these  $2*n_2$  offspring is stored in a temporary children pool. The crossover operation is repeated  $n_1$  times, for different cut points each time, until the children pool is completed. Finally, the best offspring created from  $n_2$  parents and  $n_1$  crossover is inserted in the new population. In MCMP-STUD and MCMP-SRI, following the Reeves's idea [14,18] of inserting seeds (good solutions), the first individual in the population was generated in each generation as a *seed* by the non evolutionary heuristic CDS.

## 5. CONCLUSIONS.

The main objective of the MCMP-STUD recombinative method is to improve quality of results including a significant set of schedules which their objective values are much closer to that corresponding to the best individual. This later feature also provides fault tolerance, because if eventually the dynamics of the system impedes using the best solution found then a better set of alternative solutions are available at the end of the evolution. Beside elitism, the presence of the stud ensures to retain good features of previous solutions. Results obtained are promising and showed its potential by providing new near-optimal solution for the whole set of instances selected for testing in different one-machine problems.

In all cases MCMP-STUD and MCMP-SRI improve their performance, and much better in smaller problem size. MCMP-STUD and MCMP-SRI are methods, which inherently balance exploration and exploitation in the searching space when parameters are adequately selected, and consequently when self adaptation is applied better results were obtained.

These results are promising and encourage us to deep forward investigation in MCMP-SRI approach, by incorporating more than one *seed* to the mating pool generated by different non evolutionary heuristics.

## 6. ACKNOWLEDGEMENTS

We acknowledge the cooperation of the project group for providing new ideas and constructive criticisms. Also to the Universidad Nacional de San Luis, the Universidad Nacional de La Patagonia Austral, and the ANPCYT from which we receive continuous support.

## 7. REFERENCES

- [1] Bain M., Ferreyra J., Pires A., Varas V., Vilanova G., Gallard R.H, "Enhanced Evolutionary Approaches To The Flow Shop Scheduling Problem", Second International ICSC Symposium of Intelligent Systems EIS 2000, June 27-30 , 2000, University of Paisley, Scotland U.K.
- [2] Campbell H., Dudek R., Smith M., *A heuristic algorithm for the n job m machine sequencing problem*, Management Science 16,1970, 630-637.
- [3] Eiben A.E., Raué P-E., and Ruttkay Zs., *Genetic algorithms with multi-parent recombination*. In Davidor, H.-P. Schwefel, and R. Männer, editors, Proceedings of the 3rd Conference on Parallel Problem Solving from Nature, number 866 in LNCS, pages 78-87. Springer-Verlag, 1994

- [4] Eiben A.E, van Kemenade C.H.M.,and Kok J.N., *Orgy in the computer: Multi-parent reproduction in genetic algorithms*. In F. Moran, A Moreno, JJ Merelo, and P. Chacon, editors. Proceedings of the 3<sup>rd</sup> European Conference on Artificial Life, number 929 in LNAI, pages 934-945. Springer-Verlag, 1995.
- [5] Eiben A. E., Hinterding R.,Michalewicz Z. – *Parameter Control in Evolutionary Algorithms*, IEEE Transactions on Evolutionary Computation, Vol. 3 No.2, 1.999.
- [6] Esquivel S., Leiva A., Gallard R., - *Multiple Crossover per Couple in Genetic Algorithms*, Proceedings of the Fourth IEEE Conference on Evolutionary Computation (ICEC'97), pp 103-106, ISBN 0-7803-3949-5, Indianapolis, USA, April 1997.
- [7] Esquivel S., Leiva H.,Gallard R., *Multiple crossovers between multiple parents to improve search in evolutionary algorithms*, Proceedings of the 1999 Congress on Evolutionary Computation (IEEE). Washington DC, pp 1589-1594.
- [8] Esquivel S., Leiva H.,Gallard R., *Self-Adaptation of Parameters for MCPC in Genetic Algorithms*, Proceedings of the 4<sup>th</sup> Congreso Argentino de Ciencias de la Computación (CACiC'98), pp 419-425. Universidad Nacional del Comahue, Argentina, Octubre 1998.
- [9] Grefenstette J.J., *Optimization of control parameters for genetic Algorithms*, IEEE Transactions on Systems, Man and Cybernetics, 16 (1):122-128 1988.
- [10] Gupta J., *A functional heuristic algorithm for the flowshop scheduling problem*, Operational Research Quarterly 22, 1971,39-48.
- [11] Hinterding R., Michalewicz Z., Eiben A. E. – *Adaptation in Evolutionary Computation – A survey*, Proceedings of the 4<sup>th</sup> IEEE International Conference on Evolutionary Computation, Indianapolis, April 13-16, 1.997, pp. 65-69.
- [12] Michalewicz, M., *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, third revised edition, 1996.
- [13] Morton T., Pentico D., *Heuristic scheduling systems*, Wiley series in Engineering and technology management. John Wiley and Sons, INC, 1993.
- [14] Nawaz M., Enscore E., Ham I., *A heuristic algorithm for the m-machine n-job flow shop sequencing problem*. Omega vol II, pp 11-95, 1983.
- [15] Palmer D., *Sequencing jobs through a multistage process in the minimum total time- A quick method of obtaining a near optimum*, Operational Research Quarterly 16, 101-107, 1965.
- [16] Pandolfi D., Vilanova G., De San Pedro M., Villagra A., Gallard R. *Solving the single-machine common due date problem via studs and inmigrants in evolutionary algorithms*. Proceedings of the SCI 2001. World Multiconference on Systemics, Cybernetics and Informatics. Orlando. USA, June 2001.
- [17] Pinedo Michael (1995) – *Scheduling- Theory , Algorithms, and Systems*. Prentice Hall International in Industrial and System Engineering.

- [18] Reeves C., *A genetic algorithm for flow shop sequencing*, Computers and Operations Research, vol 22, pp5-13, 1995.
- [19] William M. Spears: *Adapting Crossover in Evolutionary Algorithms*. Proceedings of the Evolutionary Programming Conference, 1995.
- [20] Taillard, E. *Benchmarks for basic scheduling problems*, European Journal of Operational Research, vol.64,pp.278-285,1993. <http://www.idsia.ch/~eric/problems.dir/ordonnancement.dir/ordonnancement.html>).
- [21] Tsujimura Y., Gen M., Kubota E., *Flow shop scheduling with fuzzy processing time using genetic algorithms*. The 11<sup>th</sup> Fuzzy Systems Symposium pp 248-252. Okinawa. 1995.
- [22] Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Gallard R. Multiple recombination and breeding in evolutionary algorithm for flow shop scheduling problems. Proceedings of the International Conference in Soft Computing. University of Paisley, Scotland, U.K., June2001.
- [23] Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Bain M, Gallard R. *Deterministic Parameter Control in Multirecombined Evolutionary Algorithms for the Flow Shop Scheduling Problem*. Proceedings of the SCI, World Multiconference on Systemics, Cybernetics and Informatics - 2001 Orlando. USA. June 2001
- [24] Vilanova G., Pandolfi D., De San Pedro M., Villagra A., Bain M, Gallard R. *Parameter Control in Multirecombined Evolutionary Algorithms for the Flow Shop Scheduling Problem*. Proceedings of the 7<sup>th</sup> Argentine Congress of Computer Sciences - 2001 El Calafate . Argentina.

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

# WICC 2002

*16 y 17 de mayo de 2002*

## Sistemas de Tiempo Real. Procesamiento de Señales



## Fuzzy Logic Controllers on Chip

Nelson Acosta & Daniel Simonelli  
*Universidad Nacional del Centro de la Provincia de Buenos Aires*  
*Departamento de Computación y Sistemas*  
*INCA / INTIA, Tandil, Argentina*  
*Email: nacosta@exa.unicen.edu.ar*  
*www.exa.unicen.edu.ar/inca*

### Abstract

This paper analyzes a fuzzy logic (FL) oriented instruction set (micro)controller and their implementations on FIPSOC<sup>1</sup>. VHDL code is synthesized using a small portion of FIPSOC FPGA<sup>2</sup>. These circuits are used from the mP8051 FIPSOC built-in microcontroller to provide efficient arithmetic operations such as multipliers, dividers, minimums and maximums.

### 1. Introduction

Fuzzy logic controllers (FLC) can be implemented by software running on standard hardware or on a dedicated microcontroller [Tog86, Wat90, Sas93, Ung93]. Controllers for high rates can be implemented by specific circuits or IC<sup>3</sup> [Hun95, Cos96, Aco97, Son00]. In order to process a high number of rules, optimization techniques must be applied; for example: a) reduction of the number of inference computing steps [Des96]; b) parallel inference execution [Aco00, Kov00, Mor00]; c) active rules processing [Aco98, Kas99, Pir01]; and d) standard microcontroller core with a dedicated FL instruction set [Wat93, Wat96].

The digital system speed is limited by maximum total capacitive load of the I/O signals between the different cards and chip packages. Some systems manufacturers have developed a (multi)chip in a package. This approach is also limited by the I/O interface between the different IC in the same package. Other approach proposes to build all system in only one chip. Working inside a SOC the I/O IC system do not affect (greatly) the system performance. The I/O signals between the FPD<sup>4</sup> and the mP8051 are connections into the same IC. Systems on a Chip (SOCs) allow to easily prototype mixed signal problems through both hardware capabilities and CAE

software tools. FIPSOC includes a mixed signal programmable device with on-board microprocessor (the FIPSOC chip), a complete set of CAE software tools to manage it, and a set of library macros.

This paper aims to present an alternative scheme to compute the mP8051 controller functions by using a FL instruction set on the FPGA area. Main features of this approach are: a) to reduce the development time by using a well known controller architecture; b) to minimize the algorithm computing time getting faster arithmetic operations.

This paper continues the above research lines by exploring a standard microcontroller core with a dedicated FL instruction set in a SOC<sup>5</sup> platform. Some topological and architectural alternatives are analyzed. FIPSOC FPGAs series (provided by SIDA<sup>6</sup>) have been utilized as a technological framework. In section II, the main characteristics of the FLC algorithm are summarized. Section III shows some FIPSOC architecture features. Section IV presents an implementation analysis by the arithmetic and lattice units. The architecture are depicted in section V. Finally, the principal results are presented.

### 2. FLC: The Control Algorithm

An algorithm to compute the  $m$  functions  $f_0, f_1, \dots, f_{m-1}$  of  $n$  variables  $x_0, x_1, \dots, x_{n-1}$  is depicted. The value of  $x_i$  belongs to a set  $S_i$ , and the value of  $f_k$  to the set of the real numbers  $R$ . To each variable  $x_i$  correspond  $p_i$  membership functions:  $A_{ij}: S_i \rightarrow [0,1], 0 \leq j \leq p_i-1$ ; to each function  $f_k$  correspond  $q_k$  decoding functions:  $B_{kl}: [0,1] \rightarrow R, 0 \leq l \leq q_k-1$ ; the computation of  $f_k$  is based on sentences (inference rules) such as: "IF  $x_0$  is  $A_{0j_0}$  AND  $x_1$  is  $A_{1j_1}$  AND ... AND  $x_{n-1}$  is  $A_{n-1j_{n-1}}$  THEN  $f_k$  is  $B_{kl}$ ". A set of coefficients  $r$  is defined:  $r(j_0, j_1, \dots, j_{n-1}, k, l) = 1$  if

<sup>1</sup> FIPSOC: Field Programmable System On Chip.

<sup>2</sup> FPGA: Field Programmable Gate Array.

<sup>3</sup> IC: Integrated Circuit.

<sup>4</sup> FPD: Field Programmable Device (a FPGA like device).

<sup>5</sup> SOC: System On a Chip

<sup>6</sup> www.sida.es

the previously mentioned rule applies; else  $r(\mathbf{j}_0, \mathbf{j}_1, \dots, \mathbf{j}_{n-1}, \mathbf{k}, \mathbf{l}) = 0$ .

The computing algorithm of the  $\mathbf{f}_k$  functions is:

$$\begin{aligned} &\forall i: 0 \leq i \leq n-1, \forall j: 0 \leq j \leq p_i-1, \text{ compute} \\ &\quad \mathbf{y}_{ij} = \mathbf{A}_{ij}(\mathbf{x}_i); \\ &\forall k: 0 \leq k \leq m-1, \forall l: 0 \leq l \leq q_k-1, \text{ compute} \\ &\quad \mathbf{w}_{kl} = \vee r(\mathbf{j}_0, \mathbf{j}_1, \dots, \mathbf{j}_{n-1}, \mathbf{k}, \mathbf{l}) \cdot \\ &\quad \quad \mathbf{y}_{0j_0} \cdot \mathbf{y}_{1j_1} \\ &\quad \quad \dots \\ &\quad \quad \mathbf{y}_{n-1j_{n-1}}, \\ &\forall \mathbf{j}_0, \mathbf{j}_1, \dots, \mathbf{j}_{n-1} \\ &\quad \mathbf{v}_k = \mathbf{B}_k(\mathbf{w}_k); \\ &\forall k: 0 \leq k \leq m-1, \text{ compute} \\ &\quad \mathbf{N}_k = \mathbf{v}_{k0} + \mathbf{v}_{k1} + \dots + \mathbf{v}_{kq_k-1}, \\ &\quad \mathbf{D}_k = \mathbf{w}_{k0} + \mathbf{w}_{k1} + \dots + \mathbf{w}_{kq_k-1}, \\ &\quad \mathbf{f}_k = \mathbf{N}_k / \mathbf{D}_k \end{aligned}$$

This scheme needs to compute all the lattice (maximum, minimum) and arithmetic (add, multiplication, division) operations. The inference mechanism uses the lattice max-min operations to evaluate the fuzzy-rule set. The multiplication and division instructions are used during the defuzzification step and sometimes to compute the membership functions. This scheme uses a set of LUT to compute the membership functions (during the fuzzification and defuzzification).

This scheme needs the reduction of the lattice and arithmetic expressions to two-operand equivalent ones. This reduction is based on the properties of the domain space. It does not use the knowledge of fuzzy inference mechanisms.

### 3. FIPSOC Architecture Topics

This prototyping and integration system, consists of a mixed-signal Field Programmable Device (FPD) with a standard mP8051 core, a suitable core of CAD tools and a set of library macros and cells which support a number of typical applications to be easily mapped onto the FPD. A set of fixed-functionality yet configurable analog cells is also provided [Bae97, Fau97].

The FIPSOC chip includes a two-dimensional array of programmable DMCs (Digital Macro Cell). The DMC is a large granularity, Look Up Table (LUT) based, synthesis targeted 4-bit wide programmable cell.

Each DMC has two main blocks: a combinational part, composed of four 4-input LUTs, and a sequential block including four FFs. Any lookup table (LUT) can

implement any 4-input Boolean function. Every two 4-input LUTs share two inputs, and two LUTs can be combined to form a 5 input function or a 4 to 1 multiplexer (four inputs and two control bits). The four LUTs of a DMC can be combined to perform any 6-input Boolean function. (Figure 1)

The sequential part of the DMC includes four two-input flip-flops (FF), each of which can be independently configured as mux-type or enable-type, as latch or FF, and with synchronous and asynchronous set or reset.

To improve chip area, the number of shared segments in a point to point net should be maximized, as this consume less FPGA routing resources. This tradeoff forces the design of complex routing channels, with different length segments, which requires sophisticated Computer Aided Design (CAD).

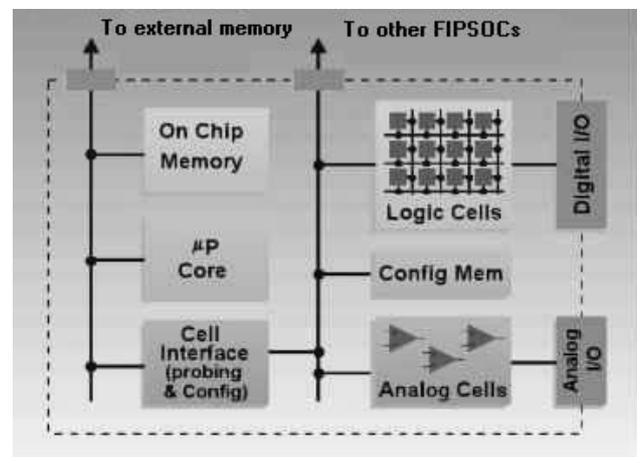


Figure 1 – FIPSOC Block Diagram

#### 3.1. 8051 Memory Organization

The memory map of the on-chip mP8051 keeps; by default, the memory organization of the original 8051. That is, it has a separate address space for Program Memory and Data Memory, distributed in the following four addresses spaces:

- Up to 64 Kbytes of Program Memory,
- Up to 64 Kbytes of external Data Memory,
- 256 bytes of internal Data Memory,
- 128 bytes of Special Function Registers area.

On the FIPSOC chip, two memory areas have been added, overlapped with the original memory areas, with the purpose of accessing to the different configuration memories and control registers of the on-chip peripherals of the mP8051. These locations, whose mapping is configured by Special Function Registers, are mapped in the following two spaces:

- 64 bytes of internal data memory, called buffer access area,
- Up to 16 Kbytes memory mapped on the lower external Data Memory (depending on device)

### 3.1.1. Program Memory.

The Program Memory of the on-chip mP8051 consists an internal and an external space. 512 bytes of Program Memory resides on-chip (dedicated to the FIPSOC boot program).

### 3.1.2. General Purpose Registers

The lower 32 positions of the internal RAM are grouped into four banks of 8 registers. Only one of the banks may be enabled at a time (two bits of PSW register are used to select the active bank). For indirect addressing, two of them are used as pointer or index registers.

### 3.1.3. Special Function Registers (SFR)

The SFR area is located in the upper 128 bytes of the internal memory. It has two important functions: Firstly, all CPU registers reside in this area (excepting PC and GPR banks). Secondly, a number of registers (not included in the SFR area of the original 8051) constitute the interface between the mP8051 and all the on-chip subsystems.

SFR, which are located in addresses which are multiple of eight, are both byte- and bit-addressable. All SFR can be accessed by direct addressing only.

Note that not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip (and they may be used in future). Read access to these addresses will return random data; write accesses will be ignored.

### 3.2. Subsystems Interface

The configuration and control of the subsystems that are included in the FIPSOC chip are realized through accesses to locations of the different memories of the mP8051. Some memory locations of the SFR area have been reserved for system configuration and general-purpose control.

The subsystems and the memory used for configuration are listed below:

a) The programmable blocks (CAB and PL block) are configured (and accessed) using special function registers and data memory (both internal and external) of the mP8051 memory organization. Thus, the configuration and data transfer of these blocks is made through positions located in both internal and external memories.

- **Programmable Logic Block:** Either internal or external memory locations may be used to configure, control and transfer data. In particular, both configuration memory and hardware outputs of the DMC are mapped in both internal (buffer access) and external; LUT memory data in DMCs can be accessed through internal memory only.

- **Configurable Analog Block (CAB):** The configuration of the block is made through the buffer access area only; Data transfer is made using memory locations of the SFR area.

b) Other subsystems are configured and controlled with special function registers only. These blocks are the Clock Generation Block (CGB), Interrupt Service Block, Serial Communication Block, Debugger Block, Watchdog Block and other mP8051 peripherals systems (Parallel & Serial I/O, Timers, etc.)

Configuration memories and outputs of the programmable blocks (PL and CAB) may be accessed through internal and external memory locations.

### 3.3. PL Subsystem Interface

**Regular Configuration Memory:** This memory may be mapped either through internal memory (data buffer access) or external memory. If a buffer access is done, row and column mask registers are needed, with the purpose of selecting the desired DMC(s), IO cell(s) or IIC (Internal Interface Cell) routing resources cell(s).

### 3.4. Extended SFR map

The new special function registers added to the original SFR map of the 8051 are dedicated to control, configure and data transfer purpose. Only 18 of them are related to the programmable blocks interface and the system mapping control.

### 3.5. Port Access

The original 8051 has 32 bi-directional and individually addressable I/O lines (grouped into four ports of 8 bits). Due to the fact that FIPSOC integrates some external peripherals on-chip, not all the 4 ports are accessible from the external system. In particular, port 2 and port 3 of the 8051 have been used for internal communication purposes (see figure 2)

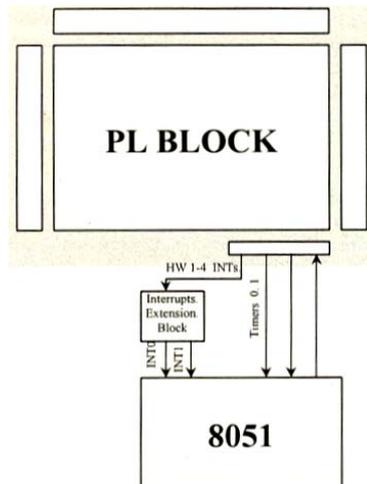


Figure 2 – Fast access bus interface

#### 4. Instruction Analysis

Multiplication is always an essential function in digital systems. This work is focused on parallel multipliers which are oriented to increase the mP8051 processing speed in a FL application.

Here are depicted the seven multipliers organized depending on the way they communicate their EPs (elementary processors) [Col99]. Every multiplier calculates  $A*B+C+S$ , generating a resulting number sized as  $\text{length}(A) + \text{length}(B)$ . The seven multipliers let to use different topologies and architectures.

• With Local Communication.	
	M <sup>c</sup> Canny - M <sup>c</sup> Whirter.
	De Mori.
• With Global Communication.	
	Hatamian & Cash.
	Ripple – Carry.
	Carry Save.
• With Mixed Communication.	
	Guild.
	De Mori – Guild.

The goal was to analyze FPGA occupation factors. With that idea in mind, 2-bit, 3-bit and 4-bit wide multipliers were used, fully combinational and with pipeline granularity 1. Although it may seem that these are very restrictive sizes, main idea is to use them for implementing iterative multipliers where, for example, an 8 x 8 multiplication can be obtained through 4 operations with a 4 x 4 multiplier.

The lattice operations involve a reduced arithmetical unit that computes the maximum and minimum values. Each of these units includes a set of registers to be used as local

RAM. The mP8051 saves the values in between the maximum or minimum must be computed. The unit output is registered too. This working scheme free the microcontroller to do any other operations while the computing is done.

For materializing these circuit structural RTL<sup>7</sup> VHDL<sup>8</sup> code was generated [BJM93, Boe96, Bue98, Col99] to configure the built-in FIPSOC FPGA [Fau97].

Other architectural option works by using a hardwired stack. The mP8051 push the two operands and the operator in the stack. The operator value activates the circuit starting to compute. When the unit finished the required operation, it pushes the resulting into the stack to be acquired by the mP8051.

The operations (maximum, minimum, multiplication and division) are implemented full combinational in this version; while the lattice operations are computed by a unit. The architecture design uses an 8-bit main bus joining the mP8051 I/O port to each computing unit. The right unit selection is done by a 2-bits selector:

0000	Write	1 <sup>st</sup> Op Multiplication
0001	Write	2 <sup>nd</sup> Op Multiplication
0010	Read	Low Byte Multiplication
0011	Read	High Byte Multiplication
0100	Write	1 <sup>st</sup> Op Division
0101	Write	2 <sup>nd</sup> Op Division
0110	Read	Integer Division
0111	Read	Remainder Division
1000	Unused	
1001	Write	1 <sup>st</sup> Op Maximum
1010	Write	2 <sup>nd</sup> Op Maximum
1011	Read	Resulting Maximum
1100	Unused	
1101	Write	1 <sup>st</sup> Op Minimum
1110	Write	2 <sup>nd</sup> Op Minimum
1111	Read	Resulting Minimum

The computing units let the programmer the responsibility to take into account the resulting value stability. This feature lets to save: (a) some mP8051 I/O port bits, and (b) computing time of a polling operation to check if the unit have finished.

#### 5. Architecture analysis

Parameters considered after 42 implementations were: minimum required LUT size, number of DMCs and FFs used. From routing resources point of view, congestion

<sup>7</sup> RTL: Register Transfer Level.

<sup>8</sup> VHDL: Very high speed IC Hardware Description Language.

and used-shared segments were taken into account. These quantities, give useful information about detailed routing algorithm, RAISE (Router using Adaptive Simulated Evolution) [Bae97] which yields an idea regarding efficient use of routing resources through the maximization of shared segments.

A crucial restriction is to provide FIPSOC with cheap FL core (in terms of programmable resources), therefore is very important to handle this information for not wasting too much of the interconnection resources.

## 6. Conclusions

Dedicated fast FL operations greatly increases the FLC system efficiency and performance. This feature let users to use parallelism between the mp8051 code, the lattice and the arithmetic operations. It also opens the door to many applications involving this kind of computation, where the previous standard microcontrollers were not fast enough or too inefficient. It would be necessary to develop a software to verify the possible timing violations.

Independently of the multiplier used, with a 3-bit input and no pipeline, the placement tool assigns a 5-bit LUT, as well as a low FFs usage from DMCs. As a matter of fact, except on the De Mori - Guild multiplier, FFs usage factor is less than 50%. Particularly, for Ripple - Carry multiplier, using pipeline with granularity 1 only implies adding one single DMC. Moreover, it improves FFs usage as well as diminish LUTs size.

A general consequence of this fact is that, regardless data size, in all those cases on which the combinational version had 5-bit LUTs, passing to a version with pipeline yielded a DMCs increment that never arrived to duplicate the original value, as long as in that cases with 4-bit LUTs, adding pipeline resulted on two times or even more DMCs.

This is a very relevant fact at the time of choosing a combinational scheme to save programmable resources: here is shown that depending on the situation, it is possible to obtain a pipelined scheme with a low impact additional cost.

As a future work, it is necessary to obtain the power consumption related to each FLC. An equation must describe area and speed behavior. This approach let the designer to estimate these metrics before the place and route. The parallel inference by using multiples max-min computing units is a key to try.

## References

- [Aco00] Acosta N. & Curti H. "*Motores de Inferencia para Controladores Difusos: análisis de materializaciones hardware*". ICIE, UBA, Bs. As. 26-28 abril 2000.
- [Aco97] Acosta N., Deschamps J-P. and Sutter G. "*Automatic Program Generator for Customized Fuzzy Logic Controllers*". Proc. IFAC AART'97, Vilamoura, Portugal, April 1997, pp: 259-265.
- [Aco98] N. Acosta, J-P Deschamps y J. Garrido. "*Optimized active rule fuzzy logic custom controller*", Int. Symposium EIS'98, Tennerife, España. Feb'98.
- [Bae97] V. Baena-Lecuyer, M. A. Aguirre, A. Torralba, L.G. Franquelo & J. Faura, "*RAISE: A Detailed Routing Algorithm for Field-Programmable Gate Arrays*", DCIS'97
- [BJM93] E.I. Boemo, E. Juárez y J. Meneses; "*Taxonomía de Multiplicadores*"; VIII DCIS Conference, Universidad de Málaga, Noviembre de 1993.
- [Boe96] E. Boemo; "*Contribution to the Design of Fine-Grain Pipelined Arrays*"; <http://www.ii.uam.es/~ivan/>; Jan 1996.
- [Bue98] S. López Buedo; "*The Webgen Project*"; <http://www.ii.uam.es/~eda/>; 1998.
- [Col99] C. Collado & K. Larsen; "*Sistema de Asistencia en el Diseño de Multiplicadores*"; UNCPBA, Fac. Ciencias Exactas, Tandil, 1999.
- [Cos96] Costa A., De Gloria A. and Olivieri M. "*Hardware design of Asynchronous Fuzzy Controllers*". IEEE Trans. on Fuzzy Systems, vol. 4, Nro. 3, Aug. 1996, pp.: 328-338.
- [Des96] Deschamps J-P and Acosta N. "*Algoritmo de optimización de funciones reticulares aplicado al diseño de Controladores Difusos*". 25as. JAIIO, 1996 pp: 1.17-1.27.
- [Fau97] Julio Faura, Miguel A. Aguirre, Juan M. Moreno, Phuoc van Duong, Josep M. Insenser, "*FIPSOC: A Field Programmable System On a Chip*", DCIS'97
- [Hun95] Hung D. L. (1995). "*Dedicated Digital Fuzzy Hardware*". IEEE Micro, vol. 15, nro. 4, pp: 31-39.
- [Kas99] Anastasios Kasiolas. "*Intelligent Control in Multimedia Traffic Policing, Shaping and Congestion Avoidance over Broadband Networks*", MSc. Thesis, Dep. Electrical & Comp. Eng, Univ. Ontario, Ontario, April, 1999.

- [Kov00] Z. Kovacic, S. Bogdan & T. Reichenbach. "NonLinear Position Control By Using Multiple Position-Dependent Self-Organizing Fuzzy Logic Controllers", 6th IFAC-Symposium on Robot Control SYROCO '00, pp.229-233, Vienna, Austria, 2000..
- [Mor00] Hipolito Moreno Llagostera. "Control of a Pneumatic Servosystem using Fuzzy Logic", 1<sup>st</sup>. FPNI - PhD Symposium. Hamburg, 2000. Pp. 189-201.
- [Pir01] Leonardo Pirrello, Leena Yliniemi & Kauko Leiviskä. "Development of a Fuzzy Logic Controller for a Rotary Dryer with Seft-Tuning of Scaling Factor", Oulu University, Control Eng. Laboratory, Report A17, Oulu, Finland. ISBN 951-42-6424-X. June 2001.
- [Sas93] Sasaki M. Ueno F. and Inoue T. (1993). "7.5 MFLIPS Fuzzy Microprocessor using SIMD and Logic-in-Memory Structure". Proc. of IEEE International Conference on Fuzzy Systems, pp: 527-534.
- [Son00] Feijun Song & Samuel M. Smith. "Takagi-Sugeno Type Fuzzy logic Controller with only 3 Rules for a 4 Dimensional Inverted Pendulum System", IEEE Int. Conf. System, Man & Cybernetics, pp.3800-3805, 2000.
- [Tog86] Togai M. and Watanabe H. (1986). "Expert System on a Chip: An Engine for Real-Time Approximate Reasoning". IEEE Expert, vol. 1, Nro. 3, pp: 55-62.
- [Ung93] Ungerling A., Thuener K. and Gosser K. (1993). "Architecture of a PDM VLSI Fuzzy Logic Controller with Pipelining and Optimized Chip Area". Proc. of IEEE International Conference on Fuzzy Systems, pp: 447-452.
- [Wat90] Watanabe H, Dettlof W. and Yount K. (1990). "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture". IEEE Journal on Solid-State Circuits, vol. 25, nro. 2.
- [Wat93] H. Watanabe y D. Chen. "Evaluation of fuzzy instructions in a RISC processor", Proc. IEEE Int. Conf on Fuzzy Systems, San Francisco, pp: 521-526, 1993.
- [Wat96] H. Watanabe, D. Chan y S. Konuri. "Evaluation of Min/Max instructions for fuzzy information processing", IEEE Trans. on Fuzzy Systems, Vol. 4, pp: 369-374, Aug. 1996.

# Travelling Source Estimation using Spatio-temporal Data

Oscar Norberto Bria  
onb@info.unlp.edu.ar

LIDI. Laboratorio de Investigación y Desarrollo en Informática.  
Facultad de Informática, UNLP.  
50 y 115. 1er Piso. La Plata

## Abstract

The estimation of travelling source parameters takes a considerable importance for many areas of sensor space-array processing, e.g., radar, underwater acoustic, non-invasive electro-medicine.

State-space models are a well suited framework for solving that dynamic estimation problem and they are in the core of our studies.

The parameter estimation problem is solved by analyzing spatio-temporal data, in applications where a relative large amount of noisy data is available.

Part of our project is to analyze the conjunction of state-space models and spatio-temporal data techniques for the estimation of the parameters of a travelling source. The technique will be used for the estimation of travelling brain sources from EEG/MEG measures.

## State-Space Models for a Travelling Source

State-space models are two related equations for modeling two related processes. One equation is a recursion describing the state transitions or travelling process. The other equation depicts the functional relation between the measurement process or observations and the states.

State-space models are easy to analyze when the two equations are linear because they lead, under certain noise assumptions, to the well known Kalman filter. When nonlinearities arise and can not be avoided, still the Kalman approach can be used with some modifications.

## Models for Propagation

A travelling source is a source with variant spatial characteristics. We postulate that the motion of the travelling source is governed by a linear dynamic multidimensional system driving by non-correlated zero-mean Gaussian noise.

$$\mathbf{x}_{t+1} = \mathcal{F}_t \mathbf{x}_t + \mathbf{w}_t \quad (1)$$

where  $\mathbf{x}_t$  is the unknown vector parameter of the source at time  $t$ ,  $\mathcal{F}_t$  is a known matrix describing the dynamics of the propagation, and  $\mathbf{w}_t$  is characterized by a known symmetric covariance matrix  $\Sigma_{\mathbf{w}_t}$ . Equation 1 defines a *linear propagation model*.

Now, suppose that  $\mathcal{F}_t$  depends upon an unknown vector of parameter or weights  $\mathbf{s}_t$ . We can compose a unique vector of unknown parameters  $\boldsymbol{\vartheta}_t = [\mathbf{x}_t \ \mathbf{s}_t]$ , and after grouping and extending with zeros, we obtain, in general, the following *nonlinear propagation model*.

$$\boldsymbol{\vartheta}_{t+1} = \Phi(\boldsymbol{\vartheta}_t) + \mathring{\mathbf{w}}_t \quad (2)$$

Above, to solve the problem of an unknown dynamics, the state vector is augmented by addition of the unknown parameters. This leads, in general, to the nonlinear dynamics equation (2) due to the occurrence of products between parameters and states [4]. But if we assume *polynomial dynamics* we can avoid the nonlinearity.

A particular case of the nonlinear propagation model is when  $\mathbf{s}_t$  is modelled using a parametric recursion. In such a case we have the following digression

$$\mathbf{x}_{t+1} = \mathcal{F}(\mathbf{s}_t) \mathbf{x}_t + \mathbf{w}_t \quad (3)$$

$$\mathbf{s}_{t+1} = B\mathbf{s}_t + \mathbf{v}_t \quad (4)$$

with symmetric covariance matrix  $\Sigma_{\mathbf{v}_t}$  for noise  $\mathbf{v}_t$ . Another particular case is when  $\mathbf{s}_t$  is a constant, i.e.,  $\mathbf{s}_{t+1} = \mathbf{s}_t = \mathbf{s}$ .

## Models for Observations

We have just indirect access to the source parameters  $\mathbf{x}_t$  and the travelling parameters  $\boldsymbol{\vartheta}_t$  through measures coming from a postulated observation process.

In most cases, the following nonlinear equation results for the observations,

$$\mathbf{z}_t = m(\mathbf{x}_t) + \mathbf{u}_t \quad (5)$$

where  $\mathbf{x}_t$  are the parameters of the source observed at time  $t$ ,  $m(\cdot)$  is the nonlinear relation between those parameters and the array observations  $\mathbf{z}_t$ , and  $\mathbf{u}_t$  represent the observation and modelling noise which is assumed Gaussian, non-correlated and zero-mean. Note that equation (5) defines the so called *nonlinear observation model*.

From observation model equation (5) we can obtain an a priori estimate of  $\mathbf{y}_t$  by applying any inverse technique for a single snapshot, e.g., single dipole analysis.

In the following,  $\mathbf{y}_t$  is any estimate of  $\mathbf{x}_t$  when only the information available at time  $t$  has been used in the estimation. Then a *linear observation model* is given by,

$$\mathbf{y}_t = \mathbf{x}_t + \mathbf{e}_t \quad (6)$$

with symmetric covariance matrix  $\Sigma_{\mathbf{e}_t}$  for the estimation error  $\mathbf{e}_t$ .

## State-Space Models

Our goal is either the estimation of the source position parameters  $\mathbf{x}_t$ , or the joint estimation of both, the source parameters and the travelling model parameters,  $\boldsymbol{\vartheta}_t = [\mathbf{x}_t \ \mathbf{s}_t]$ . In general<sup>1</sup>, that distinction is reflected in the nonlinearity or linearity of the propagation model.

We can estimate directly from the multiple snapshot process or we can re-estimate from the a priori computed single snapshot estimates. That distinction is reflected in the linearity or nonlinearity in the observation model.

Now we can combine any observation model with any dynamic model to build four state-space models with different complexity. For any one of them we can apply several Kalman-like techniques to solve the tracking/identification problems that arise [1], [2] [3], [4], [5], [6], [7].

Sumarizing, the state-space models we are analyzing are:

model I	$\begin{aligned} \mathbf{x}_{t+1} &= \mathcal{F}_t \mathbf{x}_t + \mathbf{w}_t \\ \mathbf{y}_t &= \mathbf{x}_t + \mathbf{e}_t \end{aligned}$
model II	$\begin{aligned} \mathbf{x}_{t+1} &= \mathcal{F}_t \mathbf{x}_t + \mathbf{w}_t \\ \mathbf{z}_t &= m(\mathbf{x}_t) + \mathbf{v}_t \end{aligned}$
model III	$\begin{aligned} \boldsymbol{\vartheta}_{t+1} &= \Phi(\boldsymbol{\vartheta}_t) + \mathring{\mathbf{w}}_t \\ \mathbf{y}_t &= \mathbf{x}_t + \mathbf{e}_t \end{aligned}$
model IV	$\begin{aligned} \boldsymbol{\vartheta}_{t+1} &= \Phi(\boldsymbol{\vartheta}_t) + \mathring{\mathbf{w}}_t \\ \mathbf{z}_t &= m(\mathbf{x}_t) + \mathbf{v}_t \end{aligned}$

---

<sup>1</sup>Recall that polynomial dynamics is an exception.

## Estimation from Spatio-Temporal Data

Classical array processing rely upon short time analysis. Whatever the considered processing, they try to optimize a a spatial contrast at a given instant [9]. The integration time for the estimation of fixed parameters is limited by the basic non-stationarity of the received signal due primarily to the travelling of the source.

When the temporal resolution is relatively large compared to the travelling characteristics of the source, then it is possible to consider relatively large integration times for detecting weak signals. The integration time defines the interval of the spatio-temporal data during which some parameters are consider fixed. That interval corresponds to a single snapshot in the state-space models. A spatio-temporal model describe those parameters of the source model that remain fixed and those which are unconstrained along the integration time.

Well known algorithms for the analysis of spatio-temporal data are the MUSIC-like scanning function [8]. Reconsider equation 5 dropping the temporal dependency. Assume that  $m(\cdot)$  is unknown since we do not yet know  $\mathbf{x}$ . A data record in a single trial is composed of  $N$  time samples  $Z = [z_1, \dots, z_N]$ . Define:

$$\bar{Z} = \sum_{k=1}^K Z_k / K \quad (7)$$

$$\hat{R} = \sum_{k=1}^K Y_k Y_k^T / NK \quad (8)$$

and let  $V_{nr}$  be a matrix whose columns are the (normalized) eigenvectors corresponding to the largest  $nr$  eigenvalues of  $(\hat{R}^{-1/2} \bar{Y} \bar{Y}^T \hat{R}^{-1/2}) / N$ . It can be shown that  $I_m - V_{nr} v_{nr}^T$  is the projection matrix onto the space orthogonal to the column space of  $\hat{R}^{-1/2} \hat{m}$ , where  $\hat{m}$  is a ML estimate of  $m$ . Then a scanning function can be easily constructed as

$$l(\theta) = \frac{1}{\lambda_{MIN} \left( m(\mathbf{x})^T \hat{R}^{-1/2} [I - V_{nr} V_{nr}^T] \hat{R}^{-1/2}, m(\mathbf{x})^T \hat{R}^{-1} m(\mathbf{x}) \right)} \quad (9)$$

where  $\lambda_{MIN}(\cdot, \cdot)$  denotes the smallest generalized eigenvalues of the matrices in the parenthesis and  $m(\mathbf{x})$  is a single source array response matrix.

When the signal is weak whereas the temporal resolution is relatively large it is on certain occasions likely to use MUSIC for single snapshot parameter pre-estimation by integrating over a sliding temporal window.

After pre-estimation it is feasible to smooth the estimation using Kalman filtering or smoothing techniques by incorporating dynamic information and/or constraints. Observe that the state-space model that applies is model I or model III.

## References

- [1] Digalakis, Rohlicek, Ostendorf, "ML Estimation of a stochastic linear system with the EM algorithm and its application to speech recognition," *IEEE Trans. Speech and Audio Proc.*, 1(4):431–442, 1993.
- [2] van der Merwe et.al, "The Unscented Particle Filter," *Advances in Neural Information Processing Systems (NIPS13)*. T.K. Leen, T.G. Dietterich and V. Tresp editors. December, 2000.
- [3] Ghahramani, Hinton, "Parameter Estimation for Linear Dynamical Systems," *Technical Report CRG-TR-96-2*, Department of Computer Science, University of Toronto, Canada, 1996.
- [4] Goodwin, Sin, *Adaptive Filtering, Prediction and Control*, Prentice Hall, 1984.
- [5] Johnston, Krishnamurthy, "Optimal MAP Estimation of Bilinear Systems via the EM Algorithm," *IEEE ICASSP*, 1998.
- [6] Mitchell, Tom, *Machine Learning*, McGraw-Hill, 1997.
- [7] Roweis, Ghahramani, "An Estimate for Identification of Nonlinear Dynamical Systems," *Kalman Filtering and Neural Networks*, to be published.
- [8] Stoica and Nehorai, "MUSIC, Maximum Likelihood and Cramér-Rao Bound," *IEEE Trans. Acoust., Speech, Sign. Proc.*, vol. ASSP-37, pp. 720-741, May 1989.
- [9] Zugmayer and Le Cadre, "On the optimization of Spatio-temporal Analysis for Source Motion Estimation," *IEEE*, 1992.

## **DISEÑO Y DESARROLLO DE UN SISTEMA DE SOPORTE DE DECISIÓN PARA ANÁLISIS DE INSTRUMENTACIÓN DE PLANTAS INDUSTRIALES**

**Ignacio Ponzoni**

Grupo de Investigación y Desarrollo en Computación Científica GIDeCC  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur, Avda. Alem 1253, 8000 Bahía Blanca, Argentina  
Telefax: 0291 4595100 - e-mail: [ip@cs.uns.edu.ar](mailto:ip@cs.uns.edu.ar)

**Gustavo E. Vazquez, Jessica A. Carballido y Nélide B. Brignole**

Grupo de Investigación y Desarrollo en Computación Científica GIDeCC  
Departamento de Ciencias e Ingeniería de la Computación  
Universidad Nacional del Sur, Avda. Alem 1253, 8000 Bahía Blanca, Argentina  
Planta Piloto de Ingeniería Química PLAPIQUI (UNS-CONICET)  
Comp CRIBABB, km 7 Camino La Carrindanga, 8000 Bahía Blanca, Argentina  
Tel.: 0291 4861700 - Fax: 0291 4861600 - e-mail: [dybrigno@criba.edu.ar](mailto:dybrigno@criba.edu.ar)

### **Resumen**

El objetivo global de estas investigaciones es desarrollar una metodología completa para el diseño de instrumentación de plantas de procesos químicos que sea robusta, rigurosa, independiente del punto de operación y por ende aplicable al diseño de plantas industriales grandes y/o complejas. Esto implica el desarrollo e implementación de nuevos algoritmos de clasificación de variables, la generación de herramientas computacionales de análisis como ayuda en la toma de decisiones y la integración de las mejores rutinas en un paquete de software adecuado.

La idea es trabajar sobre la base de una metodología para localización de sensores que tienda a ser perfecta, lo cual implica la creación de nuevos algoritmos para las etapas claves del diseño que son el análisis de observabilidad y de redundancias con el objeto de superar las limitaciones de las técnicas existentes.. El diseño involucra las siguientes tareas: desarrollo de una interfaz gráfica adecuada, generación automática de modelos matemáticos de plantas de procesos y de una configuración inicial de instrumentos, análisis de observabilidad, análisis de redundancias, reconciliación de datos y herramientas informáticas para soporte de decisión.

Con respecto a los objetivos específicos de diseño del DSS, se desea que el software cuente con las siguientes facilidades: interfaces amigables, seguras y confiables; modelado matemático riguroso y preciso de plantas reales; flexibilidad en las opciones de modelado; algoritmos eficaces, eficientes y robustos para análisis de observabilidad, análisis de redundancias y reconciliación de datos; chequeos de consistencia y validación de datos; herramientas inteligentes para ayudar en la toma de decisiones y capacidad para el tratamiento confiable de plantas de dimensión industrial.

El sistema completo constituirá un paquete novedoso y útil, distinto del software comercial existente tanto por el enfoque del análisis como por la rigurosidad del cálculo, constituyendo así un producto de calidad muy atractivo tanto para el ámbito académico como industrial.

## Metodología de Trabajo

En vista de las características propias de este tipo de proyectos se decidió diseñar un paradigma [1] especialmente adaptado para lograr máxima eficiencia en el desarrollo del software. El paradigma propuesto para el desarrollo de los diversos módulos de este proyecto se basa en el paradigma del modelo evolutivo, habiéndose modificado fundamentalmente la etapa de ingeniería, donde se realiza la implementación a partir de un prototipo evolutivo, y el concepto de análisis de riesgo que fue reemplazado por el de evaluación de estado del prototipo. Un esquema de este enfoque se muestra en la Figura 1. La Tabla 1 contiene la descripción de cada etapa.

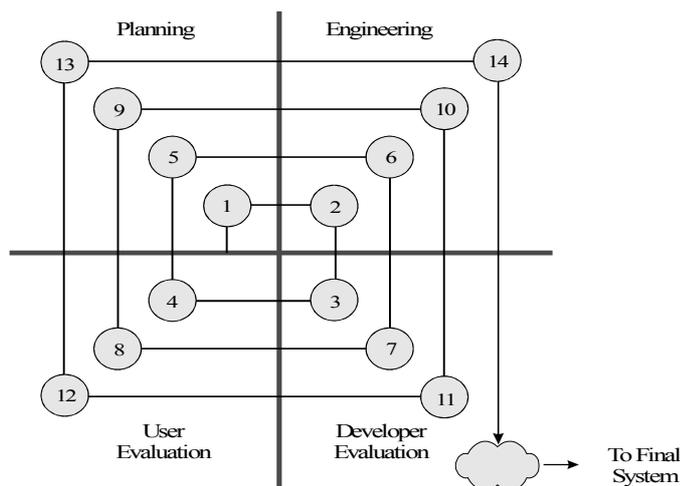


Figura 1: Paradigma de Desarrollo de este Software

Etapa	Descripción
1	Análisis inicial de requisitos para el proyecto
2	Prototipo inicial de software
3	Primera evaluación y revisión de los requisitos a partir del prototipo inicial
4, 8, 12	Evaluación del usuario
5, 9, 13	Nueva planificación del prototipo del siguiente nivel a partir de la evaluación del usuario y de la evaluación de desarrollo
6, 10, 14	Prototipo de siguiente nivel
7, 11	Evaluación del prototipo actual. (algoritmos, estructuras de datos, etc.)

Tabla 1: Etapas de Desarrollo de este Software según Figura 1

Un rasgo distintivo de este desarrollo es la disponibilidad permanente de usuarios finales quienes continuamente asisten a los diseñadores. Esto permite un control continuo de la calidad del producto, haciendo innecesaria la etapa de análisis de riesgo. Por otra parte, las especificaciones de los usuarios normalmente cambian en el transcurso de este proyecto. Por lo tanto, se realiza un refinamiento continuo del prototipo en todos los niveles de evolución, no solamente en los ciclos iniciales.

Los módulos que conformarán el DSS están siendo creados, desarrollados y chequeados individualmente empleando el paradigma arriba descrito. A medida que se van logrando módulos suficientemente refinados, estos se van integrando gradualmente a la estructura general del paquete,

verificándose nuevamente el desempeño conjunto de las partes ensambladas. Desde el punto de vista conceptual, se recurre a distintas áreas de computación científica. El análisis de observabilidad [2] está basado en técnicas de búsquedas en grafos, habiéndose diseñado algoritmos combinatorios y no combinatorios, secuenciales y paralelos. Para el análisis de redundancias se creó un nuevo enfoque estructural [3] basado en reglas que son deducciones a partir de la derivación simbólica de las ecuaciones del modelo. Se cuenta a la fecha con un prototipo experimental suficientemente refinado de esta técnica, restando efectuar su implementación definitiva. En cuanto a las herramientas para la toma de decisiones ya se han desarrollado varias para análisis de observabilidad [2], planeándose el desarrollo de otras nuevas para soportar los restantes módulos. Actualmente estamos comenzando a desarrollar el módulo de generación automática del conjunto inicial de mediciones, que utilizará algoritmos evolutivos. La última etapa será el desarrollo e incorporación del módulo de reconciliación de datos, que involucra el empleo de métodos para optimización con función objetivo y restricciones no lineales.

### Referencias

[1] **G.E. Vazquez, I. Ponzoni, M.C. Sanchez, N.B. Brignole**. “ModGen: a Model Generator for Instrumentation Analysis”, *Advances in Engineering Software* (ISSN: 0965-9978), 32, 1, 2001, 37-48.

[2] **I. Ponzoni** “Aplicación de Teoría de Grafos al Desarrollo de Algoritmos para Clasificación de Variables”. Tesis doctoral en Ciencias de la Computación. Universidad Nacional del Sur, Bahía Blanca, Argentina, Abril 2001.

[3] **S. Ferraro, I. Ponzoni, M.C. Sanchez, N.B. Brignole** “Symbolic Determination of Redundant Measurements for Plant Instrumentation Design”, *Proceedings ENPROMER III, 3er Congreso de Ingeniería de Procesos del Mercosur* (ISSN:1666-1621), 1, 415-420, Santa Fe, Argentina, 2001.

# Sistemas en Tiempo Real

Mg. Hugo Ramón<sup>1</sup>

Ing. Fernando Romero<sup>2</sup>

*Laboratorio de Investigación y Desarrollo en Informática<sup>3</sup>  
Cátedra de Sistemas de Tiempo Real - Facultad de Informática - UNLP*

## Resumen

El objetivo de esta línea de investigación es el estudio y desarrollo de aplicaciones que poseen restricciones temporales. Resulta de interés el estudio de diferentes metodologías para la especificación y desarrollo de aplicaciones en tiempo real, así como las plataformas donde dichas aplicaciones pueden ejecutarse.

Se trabaja experimentalmente con distintos tipos de hardware y software entre los que pueden mencionarse placas A/D, mini-robots, máquinas multimicroprocesadores y diferentes sistemas operativos.

Interesan especialmente las aplicaciones que involucren decisiones que van desde el hardware a utilizar hasta la plataforma sobre la que correrá dicha aplicación.

## Palabras Claves

Tiempo Real, Ingeniería de Software, Dispositivos A/D, mini-Robots, RT-LINUX, QNX

## Introducción

Los sistemas de tiempo real (STR) son sistemas de tiempo crítico, es decir, se trata de aplicaciones para las cuales el tiempo en que se produce la respuesta es un requerimiento esencial.

Por lo general, un STR incluye un conjunto de dispositivos independientes (hardware y software) que operan a diferentes velocidades.

---

<sup>1</sup> Profesor Adjunto Dedicación Exclusiva. Fac. de Informática. Universidad Nacional de La Plata. E-mail: hramon@lidi.info.unlp.edu.ar

<sup>2</sup> JTP Semidedicación. Fac. de Informática. Universidad Nacional de La Plata. E-mail: fromero@lidi.info.unlp.edu.ar

<sup>3</sup> Calle 50 y 115 1er Piso, (1900) La Plata, Argentina, TE/Fax +(54)(221)422-7707. <http://lidi.info.unlp.edu.ar>

En un STR se deben satisfacer requerimientos de performance. Comparando con un sistema tradicional, se debe tener especial control sobre la utilización de recursos, mejora en los tiempos de respuesta y manejo de distintas prioridades.

Existen algunas diferencias importantes entre un sistema de software tradicional y un STR:

- Control de dispositivos externos.
- Procesamiento de mensajes que pueden llegar en intervalos irregulares.
- Detección y control de condiciones de falla.
- Modelización de condiciones concurrentes.
- Alocaación y control de procesos concurrentes.
- Manejo las comunicaciones entre procesos.
- Protección de datos compartidos.
- Manejo de requerimientos de tiempos y performance.
- Testeo y debug de procesos concurrentes.
- Diseño de simuladores del hardware que no se encuentran disponibles durante la fase de pruebas.
- Selección del hardware adecuado para soportar el diseño del software.

Se espera que un STR se ejecute en forma continua, automática y segura, teniendo un impacto en los costos de desarrollo y la seguridad.

Esta caracterización indica claramente que un STR tiene que emplear herramientas de desarrollo y modelado diferentes a las utilizadas en el desarrollo de software tradicional.

### ***Temas de Investigación y Desarrollo***

Se plantean tres temas de estudios:

- Análisis de plataformas para desarrollo de aplicaciones de Tiempo Real
- Ingeniería de Software orientada a Tiempo Real
- Dispositivos de Tiempo Real

#### ***Análisis de plataformas para desarrollo en Tiempo Real***

En este sentido se está estudiando el sistema operativo RT-Linux, explorando sus posibilidades para el desarrollo de aplicaciones distribuidas en tiempo real.

Se han realizado benchmark sobre sistemas operativos tradicionales para evaluar la posibilidad de ejecutar sistemas con restricciones blandas.

***Ingeniería de Software orientada a Tiempo Real***

Se estudian métodos de Ingeniería de Software orientados específicamente a las características del estudio del tiempo [GOLDSMITH 1993].

En este momento se incluyen herramientas de especificación como DTE, Redes de Petri y metodologías alternativas de diseño [GOMAA 1993].

***Dispositivos para tiempo real***

Se han desarrollado drivers para diversas placas conversoras a fin de que funcionen sobre diferentes plataformas.

Otros trabajos abarcan el uso de agentes físicos (mini-robots) móviles y fijos para los cuales se han desarrollado interfaces en diferentes plataformas.

Se han desarrollado simulaciones en OpenGL (GLScene) con aplicación docente y para testeado de aplicaciones utilizando como base experimental los mini-robots.

***Líneas de Trabajo Futuras***

Testing orientado a aplicaciones distribuidas en tiempo real [BEIZER 99].

Modelización con UML para sistemas complejos de tiempo real [DOUGLASS 2000].

Sistema operativo QNX, como plataforma de STR. Comparación con otros RTOS.

Definición de benchmark para evaluación de RTOS.

## ***Bibliografía***

[BEIZER 99] Software Testing Techniques. B. Beizer. International Thomson Computer Press, 1999.

[BUHR 1999] An Introduction to Real-Time Systems. L. Buhr. Prentice Hall, 1999.

[BURN 1996] Real-Time Systems and Programming Languages. A. Burns & A. Wellings. Addison Wesley, ISBN 90-201-40365-x

[DOUGLASS 2000] Real-Time UML: Developing Efficient Objects for Embedded Systems 2/ed. Addison Weley, 2000.

[ELLISON 1994] Developing Real-Time Embedded Software. S. Ellison, Wiley, 1994

[GOLDSMITH 1993] Practical Guide to Real-Time Systems Development. S. Goldsmith. Prentice Hall, 1993.

[GOMAA 1993] Software Design Methods for Concurrent and Real-Time Systems. H. Gomaa. SEI Series, 1993.

[GREHAN 1998] Real-Time Programming, Grehan. Addison Wesley, 1998.

[HATLEY 1988] Strategies for Real-Time System Specification, Hatley & Pirbahai, 1988, Dorset House, ISBN 0-932633-11-0

[LEVI 1990] Real Time System Design. Mc. Graw Hill, Levi & Agrawala, 1990, ISBN 0-07-037491-0

[NIELSEN 1990] ADA in Distributed Real-Time Systems, Nielsen, 1990, Mc. Graw Hill, ISBN 0-07-046544-4

[SHUMATE 1992] Software Specification and Design. A Disciplined Approach for Real-Time Systems, Shumate & Keller, 1992, Wiley, ISBN 0-471-53296-7

[SILBERSCHATZ 1994] Sistemas Operativos. Conceptos Fundamentales, 3rd Ed A. Silberschatz, J. Peterson, P. Galvin, Addison-Wesley Iberoamericana, 1994.

[SIMON 1999] An Embedded Software Primer, D. Simon, 1999. Addison Wesley CEPUB

Artículos IEEE & ACM

---

# IV WORKSHOP DE INVESTIGADORES EN CIENCIAS DE LA COMPUTACIÓN

## WICC 2002

*16 y 17 de mayo de 2002*

### Teoría de Computación



# Herramientas de generación de procesadores de lenguajes para código móvil seguro.

J. Aguirre, M.Arroyo, N. Florio, J. Felippa, G. Gomez, F.Bavera, P. Caymes Scutari, D. Nordio<sup>1</sup>.

## RESUMEN

En este trabajo se presentan las líneas del grupo de investigación del Departamento de Computación de la UNRC constituido por los autores. Los aspectos fundamentales son la creación de modelos y herramientas de última generación para la generación de procesadores de lenguajes incluyendo la generación de Código Móvil Seguro. Se presenta el trabajo ya realizado que ha permitido construir un prototipo de ambiente de generación de procesadores de lenguajes basado en un modelo que soporta una integración más fuerte que las usadas, de las gramáticas de atributos y los esquemas de traducción, los trabajos realizados sobre evaluación concurrente de atributos y su inserción en el prototipo. Se analiza la nueva técnica para obtener Código Móvil Seguro (CMS) basado en la generación conjunta del código y de una prueba de seguridad – Proof Carrying Code PCC – y el proyecto de integrar en el ambiente un módulo que genere PCC basado en tipos dependientes. Es de destacar que el gran desarrollo de las redes de computadoras y de las actividades que se soportan sobre ellas y la aparición de lenguajes y herramientas, como Java, que soportan la migración de código entre distintas plataformas, ha puesto al CMS en un lugar relevante de tecnología informática.

## INTRODUCCIÓN

La especificación de lenguajes formales y el desarrollo de sus correspondientes procesadores, compiladores, intérpretes o interfaces con el usuario, ha ocupado un lugar preponderante en las Ciencias de la Computación, con muy importantes consecuencias sobre la industria de software. Pese a los grandes avances realizados en el área, aún quedan aspectos importantes por resolver en la problemática de la generación de procesadores de lenguajes y en particular en la de los compiladores de compiladores [Aho86][App98][Fra95][Wai92].

A su vez el gran desarrollo de las redes de computadoras y de las actividades que se soportan sobre ellas, ha impulsado el uso de código móvil. Por otra parte la aparición de lenguajes y herramientas, como Java, que soportan la migración entre distintas plataformas ha hecho que se torne fundamental poder contar con métodos y herramientas que garanticen la seguridad del entorno local en el que se ejecuta código migrado del exterior.

El campo de trabajo del grupo aborda aspectos de las problemáticas citadas, pudiéndose agrupar sus actividades en dos líneas: A) Modelos y herramientas que soportan la generación de procesadores de lenguajes y B) Generación de código móvil probado (Proof Carrying Code). Como resultado instrumental se pretende obtener un prototipo de ambiente de generación de procesadores de lenguajes que implemente los resultados obtenidos, del cual existe una primera versión *-japlage-* [Agu01]

---

<sup>1</sup> Universidad Nacional de Río Cuarto. e-mails: [jaguirre@dc.exa.unrc.edu.ar](mailto:jaguirre@dc.exa.unrc.edu.ar), [marroyo@dc.exa.unrc.edu.ar](mailto:marroyo@dc.exa.unrc.edu.ar), [nflorio@dc.exa.unrc.edu.ar](mailto:nflorio@dc.exa.unrc.edu.ar), [jfelippa@dc.exa.unrc.edu.ar](mailto:jfelippa@dc.exa.unrc.edu.ar), [ggomez@dc.exa.unrc.edu.ar](mailto:ggomez@dc.exa.unrc.edu.ar), [pancho@dc.exa.unrc.edu.ar](mailto:pancho@dc.exa.unrc.edu.ar), [pcaymesscutari@dc.exa.unrc.edu.ar](mailto:pcaymesscutari@dc.exa.unrc.edu.ar), [nordio@dc.exa.unrc.edu.ar](mailto:nordio@dc.exa.unrc.edu.ar).

Este trabajo ha sido realizado en el marco de proyectos subsidiados por la SECyT de la UNRC y por la Agencia Córdoba Ciencia.

## Descripción de las líneas de trabajo

A continuación se da una descripción sucinta de las dos líneas fundamentales:

A) Modelos y herramientas que soportan la generación de procesadores de lenguajes.

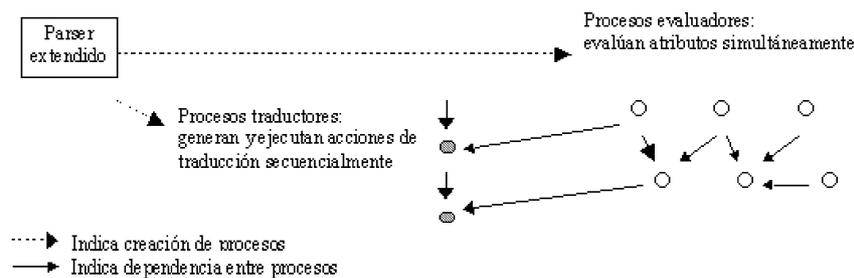
Los modelos teóricos que sustentan los lenguajes de especificación usados actualmente por los generadores de procesadores de lenguajes – y en particular de compiladores de compiladores – son las gramáticas de atributos y los esquemas de traducción. La herramienta más usada, yacc – originalmente desarrollada dentro del entorno de desarrollo de Unix y luego migrada a prácticamente todas las plataformas – brinda un mecanismo que resulta de una débil integración de ambos modelos. Con esta herramienta se puede trabajar con un conjunto restringido de esquemas de traducción – aquellos que no alteran el orden de los símbolos no terminales en la traducción–, a los cuales se agregan las facilidades de las gramáticas con atributos a izquierda – L-attributed grammars. Este modelo no permite resolver problemas que requieran gramáticas que escapen a este último tipo, tales como el de la verificación del sistema de tipos de Ada, en estos casos el usuario debe recurrir a la generación de un árbol sintáctico, usando las facilidades brindadas por yacc, y luego construir su propio evaluador que implemente las estrategias de recorrido necesarias. Por otra parte, como yacc está basado en un analizador ascendente –LALR(1)– la herencia debe ser emulada por el usuario, mediante la introducción de símbolos demarcadores y esta técnica conduce en muchos casos a que la gramática modificada deje de pertenecer a la clase LALR(1) quedando fuera del alcance de la herramienta. En los últimos años la aparición de JAVA, con la seguridad de su sistema de tipos fuerte y la posibilidad de realizar implementaciones independientes de la plataforma y de generar código móvil a través de redes, ha hecho que aparezcan nuevas herramientas de generación implementadas sobre él. Sin embargo, todas ellas siguen imponiendo al usuario restricciones sobre el tipo de la gramática de atributos a utilizar.

El grupo ha construido con un modelo teórico - Esquemas de Traducción con Atributos (ATS)- y un modelo de implementación [Agu98], que brinda una combinación más potente de los formalismos antes mencionados y se ha implementado un prototipo de ambiente *Japlage* [Agu01], que implementa los modelos desarrollados.

El sistema de tiempo de ejecución –RTS- de los procesadores generados según el modelo de implementación está compuesto por tres tipos de procesos:

1. El parser, proceso secuencial que a lo largo del análisis sintáctico inicializa los procesos de los otros dos tipos.
2. Los procesos de traducción, iniciados por un *scheduler* que determina el orden en que deben ser ejecutados. Estos procesos pueden correr en paralelo con el analizador sintáctico y los procesos del tercer tipo.
3. Los procesos de evaluación de atributos que son ejecutados concurrentemente con todos los procesos del sistema, incluso con los del mismo tipo, con la única restricción de la dependencia entre atributos.

Los procesos de tipo 3 se comunican entre sí y con los procesos de tipo 2 mediante objetos compartidos en los que se alojan los valores de los atributos. Cuando un proceso hace referencia a un atributo aún no evaluado, se bloquea hasta que el valor esté disponible, la figura 1 muestra la vinculación entre estos procesos.



**Figura 1:** Implementación de un ATS: componentes y su interrelación

Este tipo de evaluación llamado “bajo demanda” [Sar93] genera un importante costo adicional debido a la nutrida comunicación producida. El grupo ha comenzado a trabajar en la optimización de la evaluación concurrente de Gramáticas de Atributos, obteniendo un algoritmo de partición del grafo de dependencias que permite generar procesos con comunicación mínima para gramáticas de atributos de tipo NC(1) [Arr00]. Así mismo se ha extendido este algoritmo a las gramáticas NC(1) condicionales [Boy96] [Arr01].

Asimismo se pretende realizar un estudio sobre implementaciones con paralelismo real. Para lo cual se ha configurado un cluster de PC's en principio basado en la plataforma BeoWulf para emular una arquitectura paralela de alto rendimiento con bajo costo.

#### B) Generación de código móvil probado (Proof Carrying Code, PCC).

El avance hacia la sociedad informatizada y el consecuente uso masivo de las redes públicas de computadoras, impone la utilización de código que se mueve de un equipo a otro. Resulta por lo tanto de primordial importancia que se pueda garantizar que la ejecución del código importado no dañe la integridad y/o privacidad del ambiente local en el que se ejecuta, mediante el uso de código seguro. Un importante ejemplo de software inseguro es el código que se importa mediante browsers de Internet y un claro ejemplo de software invasor lo constituyen los virus que usan ese código como medio de propagación.

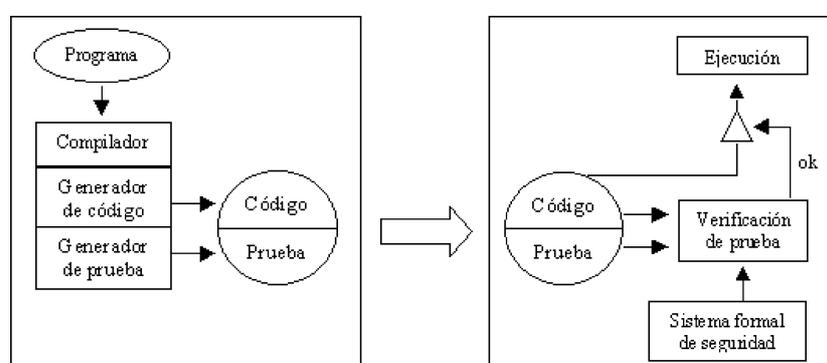
La técnica de Proof Carrying Code (PCC), iniciada por Necula y Lee [Nec96][Fei97] permite a un productor de software proveer un programa conjuntamente con la prueba formal de su seguridad. El destinatario puede especificar su política de seguridad mediante un sistema de axiomas para razonar sobre ella. La demostración construida por el productor debe basarse en dichos axiomas y será ejecutada en el ambiente local antes de que el código gane el control. El código importado sólo será ejecutado si la prueba resulta exitosa.

PCC se basa en los mismos métodos formales que la verificación de programas [Pon99], pero con la ventaja de que la demostración de condiciones de seguridad es mucho menos compleja que la demostración de la corrección de programas. La demostración del productor no asegura la corrección del programa pero sí que su ejecución no violará, bajo ninguna circunstancia, ninguna de las condiciones de seguridad impuestas. Al usarse PCC no es necesario verificar la autenticidad de la identidad del productor, ya que la validez de la prueba constituye una evidencia incontrovertible de su seguridad.

Los lenguajes de código intermedio constituyen un elemento capital en el diseño de compiladores. Ellos permiten modularizar el proceso descomponiéndolo en dos fases: el pasaje del len-

guaje fuente al lenguaje intermedio y la generación del código objeto a partir de este último. Los lenguajes intermedios también son centrales en el PCC, si la prueba se realiza sobre ellos, el desarrollo puede usarse para una diversidad tanto de lenguajes de programación, desde los cuales se genera dicho código intermedio, como de plataformas para las cuales se genera código a partir de él [App00].

En el proyecto se propone usar un sistema de tipos que corresponde a una avanzada tecnología, a efectos de disponer de la mayor información estática en la fase de la compilación correspondiente a la generación de código intermedio. Para ello se introducirá un nuevo lenguaje de código intermedio de última generación basado en los tipos dependientes (Dependent types). El uso de Tipos Dependientes en lugar de un sistema de tipos tradicional para el lenguaje intermedio, presenta la ventaja de que estos sistemas son pequeños pero altamente poderosos. Usando Tipos Dependientes se puede implementar un único formalismo para la generación de código intermedio y para la construcción de la prueba.



**Figura 2:** Visión global de Proof-Carrying Code

## Productos que se espera obtener.

Como resultados esperados se espera producir tres herramientas computacionales:

- Una herramienta de última generación para la producción de procesadores de lenguajes formales.
- Un evaluador de atributos paralelo eficiente para la familia más amplia de gramáticas de atributos (Well Formed Attributed Grammars). El evaluador será integrado a la herramienta mencionada anteriormente.
- Una herramienta para la generación de código PCC integrada al ambiente de desarrollo de procesadores de lenguajes.

## Referencias

[Agu98] J. Aguirre, V. Grinspan, Marcelo Arroyo, “Diseño e Implementación de un Entorno de Ejecución, Concurrente y Orientado a Objetos, generado por un Compilador de Compiladores”, Anales ASOO 98 JAIIO, pp. 187-195, Buenos Aires1998.

[Agui 01] JACC un entorno de generación de procesadores de Lenguajes, J. Aguirre, M. Arroyo, J. Felippa, G. Gomez. Anales del CACIC 2001, pp. 145-355, Calafate 2001, ISBN 987 96 288-6-1

[Aho86] A.V. Aho, R. Sethi, J.D. Ullman, "Compilers: Principles, Techniques, and Tools", Addison Wesley, 1986.

[App00] A. W. Appel, A. P. Felty, "A Semantic Model of Types and Machine Instructions for Proof-Carrying Code", 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages.(POPL '00), pp. 243-253, January 2000.

[App98] A. W. Appel, "Modern Compiler Implementation in Java". Cambridge University Press, ISBN: 0-521-58388-8, 1998.

[Arr00] M. Arroyo, N. Florio, J. Aguirre, "Un generador de evaluadores de gramáticas de atributos NC(1) de máximo paralelismo sin sincronización entre procesos", UNRC, Anales del CACIC 2000 pp. 523-526, Ushuaia 2000.

[Arr01] M. Arroyo, N. Florio, J. Aguirre, "Gramáticas de atributos NC(1) condicionales", UNRC, Anales del CACIC 2001 pp. 333-344, Ushuaia 2001, ISBN 987 96 288-6-1

[Boy96] J. T. Boyland, "Conditional Attribute Grammars", ACM Transactions on Programming Languages and Systems, Vol. 18, No. 1, January 1996, pages 73-108.

[Fei97] J. Feigenbaum, P. Lee. "Trust Management and Proof-Carrying Code in Secure Mobile-Code Applications", DARPA, Workshop on Foundations for Secure Mobile Code. 1997.

[Fra95] C. Fraser, "Retargetable C Compiler : Design and Implementation", Benjamin Cummings Publishing Company, 1995

[Nec96] G. Nacula, P. Lee. "Safe Kernel Extensions Without Run-Time Checking". Second Symposium on Operating Systems Design and Implementation. Seattle. 1996.

[Pon99] C. Pons, R.Giandini, G. Baum, M. Felder, "A dynamic logic theory describing the UML". Position paper for the OOPSLA'99 workshop on Rigorous Modeling and Analysis with the UML, Denver, Colorado, USA, November 1999.

[Sar93] J. Saravia, P. Henriques, "Concurrent Attribute Evaluation", technical report, Departamento de Informática, Univ. de Minho, Portugal, 1993.

[Wai92] Waite, Carter. "An Introduction to Compiler Construction". HaperCollins College Publishers. ISBN: 0-673-39822-6. 1992.

## *La Geometría Computacional a nuestro alrededor*

*Edilma Olinda Gagliardi*<sup>(1)</sup>  
*María Teresa Taranilla*<sup>(2)</sup>  
*Mario Marcelo Berón*<sup>(2)</sup>

*Gregorio Hernández Peñalver*

<sup>(1)</sup> LIDIC<sup>♦</sup>

<sup>(2)</sup> Departamento de Informática  
 Facultad de Ciencias Físico, Matemáticas y  
 Naturales  
 Universidad Nacional de San Luis, Argentina  
 {oli, tarani, mberon}@unsl.edu.ar  
 Fax: 54-2652-430224

Departamento de Matemática Aplicada  
 Facultad de Informática

Universidad Politécnica de Madrid, España  
 gregorio@fi.upm.es  
 Fax: 34-91-3367426

**Resumen:** La Geometría Computacional es una disciplina que brinda un marco teórico y formal para el diseño de estructuras y análisis de algoritmos requeridos para dar soluciones a problemas que surgen en las más diversas áreas de la Informática. Actualmente ha cobrado un gran interés debido a las numerosas aplicaciones que tiene en distintas líneas de investigación. En este trabajo se hace una introducción a la disciplina indicando su interés en ciertas áreas particulares, aportando enlaces de interés para la docencia y la investigación en la materia.

### **Introducción**

La Geometría Computacional se ocupa de resolver problemas geométricos de modo constructivo. Se interesa por demostrar la existencia de la solución de un problema y por encontrar los algoritmos y estructuras de datos eficientes, medidos respecto de su complejidad temporal y espacial respectivamente. Por lo tanto, podemos decir que esta disciplina forma parte de la teoría del diseño y análisis de algoritmos y estructuras de datos.

En ocasiones, la Geometría brinda soluciones más eficientes en problemas que no parecen geométricos. Descubrir que los datos de un problema verifican propiedades geométricas sirve para poder aplicar alguna técnica algorítmica o alguna estructura de datos especial, que nos permite describir una solución óptima.

Nuestra introducción al estudio de la disciplina fue basada en la relación existente con las bases de datos: un problema que se presenta a menudo es el estudio de los rangos y las consultas por rangos, denominado *Búsqueda por Rangos*. Este problema tratado desde una perspectiva geométrica nos permite diseñar y analizar los algoritmos y estructuras de datos utilizadas con herramientas propias de la Geometría Computacional.

Luego, los integrantes de esta línea de investigación fueron introduciéndose en el estudio de otros temas que abrían la interconexión con otras líneas de investigación. Para lo cual se tomaron dos temas relevantes: la Sumas de Minkowski y el Ruteo con brújula en redes geométricas.

Las sumas de Minkowski son utilizadas en un amplio rango de aplicaciones, tales como planificación de movimientos de robots, procesamiento de imágenes, sistemas de información geográfica y marcado de moldes, entre otras.

Con respecto al Ruteo con brújula en redes geométricas, es modelizado con teoría de grafos, cuyo problema de estudio es decidir cuáles grafos planares admiten una inmersión geométrica para los cuales el ruteo por brújula produce el paso más corto entre cualquier par de vértices de interés.

---

<sup>♦</sup> Laboratorio de Investigación y Desarrollo en Inteligencia Computacional. Director: Dr. Raúl H. Gallard.

## Búsqueda por Rangos

La *Búsqueda por Rangos* es uno de los problemas centrales en Geometría Computacional, no sólo por la variedad de aplicaciones que posee sino porque, además, una gran cantidad de problemas geométricos pueden resolverse observándolos como problemas relacionados a las búsquedas por rango.

Un *Espacio de Rangos* es un sistema  $\Omega = (U, F)$ , donde  $U$  es un conjunto de objetos geométricos y  $F$  es una familia de subconjuntos de  $U$ . Los elementos de  $F$  son llamados *Rangos* de  $\Omega$ . Por ejemplo:

- $\Omega_1 = (\mathbf{R}^d, \{ h/ h \text{ es un semiespacio en } \mathbf{R}^d \})$
- $\Omega_2 = (\mathbf{R}^d, \{ h/ h \text{ es una bola en } \mathbf{R}^d \})$

El sistema  $\Omega$  es llamado *Espacio de Rango Finito* si el conjunto  $U$  es finito.

Un problema de búsqueda por rangos puede expresarse del siguiente modo:

Dados un espacio de rangos  $\Omega = (U, F)$ ,  $S$  un subconjunto de objetos de  $U$  y  $R$  un rango de  $F$ , consultar los objetos geométricos que están en  $S \cap R$ .

En este caso, a  $R$  suele llamárselo *rango de consulta* (*query range*).

El problema real para el cual tiene sentido este estudio, es que el conjunto de objetos geométricos,  $S$ , es dado con anterioridad y luego, repetidas veces, variando el rango  $R$ , se consulta por la intersección. Entonces, no es suficiente con diseñar un buen algoritmo sino que también se debe diseñar una estructura de datos apropiada para almacenar el conjunto  $S$ , de forma tal que frente a cada consulta por rango  $R$ , ésta pueda ser respondida eficientemente.

La mayoría de las estructuras de datos para búsquedas o consultas por rango, son construidas de forma recursiva, dividiendo el espacio de objetos geométricos en varias regiones, con propiedades geométricas deseables sobre ellas. La búsqueda por rangos, esencialmente, consiste en buscar los objetos geométricos que contiene una determinada región (rango) del espacio de objetos geométricos.

Observando los rangos de consultas, podemos hacer una clasificación en cuatro tipos clásicos de búsqueda por rangos:

- *Búsqueda por Rangos Ortogonales (BRO)*: los rangos son  $d$ -rectángulos, cada uno de la forma  $\prod_{i=1..d} [a_i, b_i]$   $a_i, b_i \in \mathbf{R}^d$ , con lados paralelos a los ejes de coordenadas del espacio subyacente.
- *Búsqueda por Rangos Semi-Espaciales (BRSp)*: los rangos conforman el conjunto de todos los semiespacios cerrados de  $\mathbf{R}^d$ . Por ejemplo, en  $d=2$  tenemos semiplanos.
- *Búsqueda por Rangos Simpliciales (BRSp)*: los rangos son los simplices. En  $d=2$  tenemos triángulos, en  $d=3$  tetraedros, etc.
- *Búsqueda por Rangos Semi-Algebraicos (BRSa)*: los rangos están definidos por conjuntos semialgebraicos. Por ejemplo, el conjunto de todas las paraboloides en  $\mathbf{R}^3$ .

Dado un rango de consulta, puede interesarnos efectuar alguna de las siguientes consultas sobre él:

- *Consultas de Recuento (range counting)*: se pregunta por la cardinalidad del conjunto.
- *Consultas de Reporte (range reporting)*: se pide cuáles son los objetos geométricos que están contenidos en el rango.
- *Consultas Booleanas (range emptiness)*: se verifica si la intersección es vacía o no.

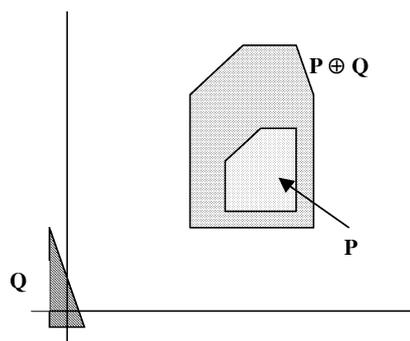
- *Consultas Especiales*: se pregunta por los objetos de  $S \cap R$  que satisfacen alguna determinada propiedad.

Los análisis de las complejidades tanto en tiempo de construcción de la estructura de datos, en espacio que ocupa y en tiempos de respuestas de las consultas, depende de la dimensión del espacio, de la cardinalidad de  $S$ , de la estrategia de particionamiento de  $S$  y del tamaño de la respuesta de la consulta.

## Sumas de Minkowski

Las sumas de Minkowski son usadas en una amplia gama de aplicaciones y juegan un rol importante en las más diversas áreas.

Se define del siguiente modo: Dados dos conjuntos  $P$  y  $Q \subset R_2$ , la suma de Minkowski de  $P$  y  $Q$ , denotada por  $P \oplus Q$  se define como  $P \oplus Q = \{ p + q : p \in P, q \in Q \}$ , donde  $p + q$  es un vector que representa la suma de los vectores  $p$  y  $q$ , es decir que dados los puntos  $p = (p_x, p_y)$  y  $q = (q_x, q_y)$  tenemos que  $p + q = (p_x + q_x, p_y + q_y)$ .



Suma de Minkowski de dos polígonos  $P$  y  $Q$

Las áreas de aplicación se han remitido principalmente a la robótica, específicamente en planificación de movimientos, donde son usadas para el cálculo del espacio *prohibido* según el enfoque del espacio de configuración.

Si tenemos un robot de traslación  $R$  en un ambiente donde los obstáculos son polígonos disjuntos, podemos calcular el espacio de obstáculos que corresponde a un obstáculo  $P_i$  como la suma de Minkowski de  $P_i \oplus (-R)$ , siendo  $-R$  una copia de  $R$  reflejada en el origen. Luego el espacio prohibido para el robot  $R$  será descripto como la unión de las sumas de Minkowski de cada uno de los obstáculos  $P_i$  con  $(-R)$ .

Otra área de aplicación es en el área de procesamiento de imágenes. La dilatación de un conjunto  $A$  con un supuesto elemento estructurado  $B$  es la suma de Minkowski de  $A$  y  $B$ . El elemento estructurado  $B$  es usualmente una figura simple en el plano como un cuadrado o un disco.

También en los Sistemas de Información Geográfica (GIS), el término *buffer* es comúnmente usado para denotar la suma de Minkowski de un conjunto de objetos geométricos dado con un disco.

Tanto la suma como la diferencia de Minkowski son herramientas poderosas de preprocesamiento para resolver problemas de intersección e inclusión de polígonos. Una consulta de solapamiento y/o inclusión de polígonos puede ser resuelta con estas operaciones, con una complejidad sublineal. Es de gran utilidad en problemas de empaquetamiento.

Por lo que podemos observar que las sumas de Minkowski son usadas en una amplia gama de aplicaciones, tienen propiedades interesantes que las hacen una herramienta útil que puede ser aplicada en la resolución de diversos problemas.

## Ruteo con brújula

Los problemas de ruteo en redes se han estudiado en Teoría de Grafos y Ciencias de la Computación. Inicialmente se describieron algoritmos para calcular el camino más corto entre los nodos de la red. El desarrollo tecnológico ha demandado nuevos algoritmos de ruteo sobre redes, que pueden modelarse informalmente como sigue:

Supongamos que un turista llega a una ciudad y desea encontrar el camino hasta la Catedral, cuya torre es visible desde cualquier punto de la ciudad. El turista no tiene plano de la ciudad y se encuentra en una plaza desde la que parten varias calles. La elección natural será caminar por la calle cuyos “puntos” sean lo más cercanos a su destino. Repitiendo este proceso en cada cruce de calles, ¿alcanzará la Catedral?

En términos matemáticos podemos modelizar el problema así: El sistema de calles de la ciudad se representa por un grafo geométrico plano, es decir, un grafo plano en el que los vértices se representan por puntos, y las aristas por segmentos de recta. El algoritmo que aplica el turista, designado por el término “ruteo con brújula”, se describe así: Desde un vértice inicial  $s$  se desea viajar hasta un vértice destino  $d$  en un grafo geométrico. La información disponible en cada nodo es local, sólo conoce las coordenadas del punto y las de sus vecinos en el grafo. Cuando alcanza un nodo  $v$  continua el camino por la arista incidente a  $v$  cuya pendiente es la más próxima a la pendiente del segmento que conecta  $v$  con el nodo destino  $d$ . Este algoritmo alcanza su objetivo en numerosas ciudades, pero en otras se necesita una estrategia diferente.

En los algoritmos de ruteo compacto (por ej. “interval routing” o “boolean routing”) cada nodo almacena una copia de un algoritmo distribuido, lo que puede llegar a ser muy costoso pues se necesita almacenar gran cantidad de información en cada nodo.

Se está trabajando ahora en algoritmos de ruteo con las siguientes características:

Información local. En cada nodo sólo se conoce la posición de sus vecinos.

Memoria limitada. El turista sólo recuerda un número constante de nodos ya visitados, así como las coordenadas del origen y del destino.

Ecológico. No se permite al turista dejar marcas en los nodos visitados.

Decisiones locales. La elección del camino a seguir en cada nodo se basa exclusivamente en la información local almacenada en el nodo y en la información que lleva el turista (de tamaño constante)

## Objetivos

Entre los objetivos de esta propuesta está la de desarrollar herramientas de aplicación que sirvan para ser utilizadas como herramientas educativas en la Geometría y que puedan ser incorporadas en aplicaciones futuras en donde estas operaciones resulten de fundamental importancia en otras líneas de investigación. También la de realizar un estudio que muestre el estado del arte del tema expuesto presentando los aspectos teóricos y prácticos relevantes para los temas presentados.

Como objetivos subyacentes a esta investigación, se busca consolidar y alimentar una línea de estudio, que brinde un puente a trabajos de investigación de mayor profundidad, que permitan posteriormente cubrir temas de tesis doctorales o de maestría.

## Conclusiones

La Geometría Computacional es una disciplina propia de la Matemáticas, la cual brinda un marco teórico y formal para el diseño de estructuras y análisis de algoritmos requeridos para dar soluciones a problemas que surgen en áreas de la Computación, tales como las bases de datos, recuperación de la información, computación gráfica, robótica, VLSI, etc. Las soluciones a tales problemas son halladas reduciendo los mismos a problemas propios de la Geometría Computacional. En nuestro trabajo nos proponemos observar las técnicas algorítmicas utilizadas, las estructuras de datos que surgen de la Geometría Computacional apropiadas para el manejo de datos geométricos y los algoritmos propuestos para resolver problemas presentados, con sus respectivos análisis y complejidades.

## Referencias Bibliográficas

- Aho, A.V.; Hopcroft, J. E.; Ullman, J. *The desing and analysis of computer algorithms*, Addison-Wesley Series in Computer Science and Information Processing, 1974.
- Berg, Kreveld, Overmars, Schwarzkopf. *Computational Geometry: algorithms and applications*, Springer Verlag, BH 1997
- Boissonnat, J.D.; Yvinec, M. *Algorithmic Geometry*, Cambridge University Press, 1998.
- Gonnet G.;Baeza Yates, R. *Handbook of Algorithms and Data Structures*
- Goodman, J.; O'Rourke, J. *Handbook of Discrete and Computational Geometry*. CRC Press 1997.
- Günther, O. *Efficient Structures For Geometric Data Management*. Springer Verlag 1988.
- I. Heywood, S. Corneluis, S. Carver, *Geographical Information Systems*, Addison-Wesley Longman, New York, 1998.
- J. Serra , *Image Analysis and Mathematical Morphology*, Academic Press, New York, 1982.
- J. Serra, *Image Analysis and Mathematical Morphology, Vol II: Theoretical Advances*. Academic Press, New York, 1988.
- Kranakis E., Singh H., J. Urrutia. *Compass rotuing on Geometric networks*
- Kurt Melhorn *Multidimensional Searching and Computational Geometry* Springer Verlag 1984
- Latombe J.C., *Robot Motion Planning*, Kluwer Academic Publisher, Boston, MA, 1991.
- Pach, J. *New trends in Discrete and Gomputational Geometry*. Springer Verlag NY 1993.
- Preparata, F.; Shamos,M. *Computational Geometry: an introduction*, /Springer Verlag, NY 1985
- Sack, J.R.; Urrutia, J.. *Handbook of Computational Geometry*. Elsevier Science B.V. 2000.
- Schneider, M. *Spatial Data Types For Database Systems* (corresponde a su tesis doctoral) 1995.
- Zhenyu Li, *Compation Algorithms for Non- Convex Polygons and their Applications*, 1994.