

**RedUNCI**

RED DE UNIVERSIDADES CON CARRERAS EN INFORMÁTICA

# Computer Science & Technology Series

**XVIII Argentine Congress of Computer Science  
Selected Papers**

**Armando De Giusti / Guillermo Simari / Patricia Pesado  
(Eds)**



## **Computer Science & Technology Series**

---

XVIII ARGENTINE CONGRESS OF COMPUTER SCIENCE  
SELECTED PAPERS





# **Computer Science & Technology Series**

---

XVIII ARGENTINE CONGRESS OF COMPUTER SCIENCE  
SELECTED PAPERS

ARMANDO DE GIUSTI / GUILLERMO SIMARI / PATRICIA PESADO  
(Eds)



---

De Giusti, Armando

Computer science & technology series / Armando De Giusti ; Guillermo Simari ; Patricia Pesado ; edición literaria a cargo de Armando De Giusti ; Guillermo Simari ; Patricia Pesado. - 1a ed. - La Plata : EDULP, 2013.

306 p. ; 24x16 cm.

ISBN 978-987-1985-20-3

I. Informática. I. Simari, Guillermo II. Pesado, Patricia III. De Giusti, Armando, ed. lit. IV. Simari, Guillermo, ed. lit. V. Pesado, Patricia, ed. lit.

CDD 004

---

## **Computer Science & Technology Series**

XVIII ARGENTINE CONGRESS OF COMPUTER SCIENCE

SELECTED PAPERS

---

**Diseño y diagramación:** Andrea López Osornio y Julieta Lloret



### **Editorial de la Universidad de La Plata (Edulp)**

Calle 47 N° 380 / La Plata B1900AJP / Buenos Aires, Argentina

+54 221 427 3992 / 427 4898

[editorial@editorial.unlp.edu.ar](mailto:editorial@editorial.unlp.edu.ar)

[www.editorial.unlp.edu.ar](http://www.editorial.unlp.edu.ar)

Edulp integra la Red de Editoriales Universitarias Nacionales(REUN)

1° edición, 2013

ISBN N.º 978-987-1985-20-3

Queda hecho el depósito que marca la Ley 11.723

© 2013 - Edulp

Impreso en Argentina

# TOPICS

## XIII Intelligent Agents and Systems Workshop

Chairs Guillermo Leguizamón (UNSL) Alejandro García (UNS) Laura Lanzarini (UNLP)

## XII Distributed and Parallel Processing Workshop

Chairs Armando De Giusti (UNLP) Marcela Printista (UNSL) Javier Balladini (UNCOMA)

## X Information Technology Applied to Education Workshop

Chairs Cristina Madoz (UNLP) Zulma Cataldi (UTN –UBA) Marcela Chiarani (UNSL) Alejandra Malberti (UNSJ)

## X Graphic Computation, Images and Visualization Workshop

Chairs Silvia Castro (UNS) Roberto Guerrero (UNSL) María José Abásolo (CIC – UNLP)

## IX Software Engineering Workshop

Chairs Patricia Pesado (UNLP) Elsa Estévez (UNS-UNU) Horacio Kuna (UNM) Marcelo Estayno (UNLZ)

## IX Database and Data Mining Workshop

Chairs Olinda Gagliardi (UNSL) Rodolfo Bertone (UNLP) Hugo Alfonso (UNLPam)

## VII Architecture, Nets and Operating Systems Workshop

Chairs Jorge Ardenghi (UNS) Nelson Acosta (UNCPBA) Hugo Padovani (UMorón)

## IV Innovation in Software Systems Workshop

Chairs Pablo Fillotrani (UNS) Pablo Cosso (UBA) Osvaldo Sposito (UNLaM)

## III Computer Science Theoretical Aspects Workshop

Chairs Susana Esquivel (UNSL) Marcelo Falappa (UNS) Raul Kantor (UNR)

## III Signal Processing and Real-Time Systems Workshop

Chairs Oscar Bría (INVAP) Horacio Villagarcía Wanza (UNLP) Hugo Ramón (UNNOBA)

## **I Computer Security Workshop**

Chairs Javier Diaz (UNLP) Antonio Castro Lechtaller (UTN) Javier Echaiz (UNS)

## **I Innovation in Computer Science Education Workshop**

Chairs Sonia Rueda (UNS) Cecilia Sanz (UNLP) Beatriz Depetris (UNTDF)

## **II ETHICOMP LatinAmerica**

Chairs Guillermo Feierherd (Universidad Nacional de Tierra del Fuego - Argentina) Simon Rogerson (Monfort University - Reino Unido) William Fleischman (Villanova University – EE. UU.) Mario Arias Oliva (Universitat Rovira i Virgili – España)

## SCIENTIFIC COMMITTEE

Coordination: Guillermo, Simari - Loui, Ronald Prescott (USA)  
Armando, De Giusti - (Argentina) Luque, Emilio (Spain)  
Madoz, Cristina (Argentina)  
Manresa-Yee, Cristina (Spain)  
Marín, Mauricio (Chile)  
Marquez, María Eugenia (Argentina)  
Naiouf, Marcelo (Argentina)  
Navarro Martín, Antonio (Spain)  
Neil, Carlos (Argentina)  
Olivas Varela, José Ángel (Spain)  
Padovani, Hugo (Argentina)  
Pandolfi, Daniel (Argentina)  
Pesado, Patricia (Argentina)  
Piattini, Mario (Spain)  
Piccoli, María Fabiana (Argentina)  
Printista, Marcela (Argentina)  
Puppo, Enrico (Italy)  
Ramón, Hugo (Argentina)  
Rogerson, Simon (United Kingdom)  
Rossi, Gustavo (Argentina)  
Rueda, Sonia (Argentina)  
Santos, Juan Miguel (Argentina)  
Sanz, Cecilia (Argentina)  
Steinmetz, Ralf (Germany)  
Suppi, Remo (Spain)  
Tarouco, Liane (Brazil)  
Tirado, Francisco (Spain)  
Utreras, Florencio (Chile)  
Vendrell, Eduardo (Spain)  
Villagarcia Wanza, Horacio (Argentina)  
Vizcaino, Aurora (Spain)  
Zamarro, Jose Miguel (Spain)

Abásolo, María José (Argentina)  
Acosta, Nelson (Argentina)  
Alba Torres, Enrique (Spain)  
Alfonso, Hugo (Argentina)  
Ardenghi, Jorge (Argentina)  
Arias Oliva, Mario (Spain)  
Bertone, Rodolfo (Argentina)  
Bría, Oscar (Argentina)  
Brisaboa, Nieves (Spain)  
Bynum, Terrell Ward (USA)  
Cabero, Julio (Spain)  
Cancela, Héctor (Uruguay)  
Casali, Ana (Argentina)  
Castro Lechtaller, Antonio (Argentina)  
Castro, Silvia (Argentina)  
Cechich, Alejandra (Argentina)  
Chiarani, Marcela (Argentina)  
Coello Coello, Carlos (Mexico)  
Collazos Ordóñez, César Alberto (Colombia)  
Cukierman, Uriel (Argentina)  
Diaz, Javier (Argentina)  
Dix, Juerguen (Germany)  
Doallo, Ramón (Spain)  
Esquivel, Susana (Argentina)  
Estayno, Marcelo (Argentina)  
Estevez, Elsa (United Nations)  
Falappa, Marcelo (Argentina)  
Fillottrani, Pablo (Argentina)  
Fleischman, William (USA)  
Gagliardi, Olinda (Argentina)  
García, Alejandro (Argentina)  
Gröller, Eduard (Austria)  
Hernández, Gregorio (Spain)  
Janowski, Tomasz (United Nations)  
Kuna, Horacio (Argentina)  
Leguizamón, Guillermo (Argentina)

## ORGANIZING COMMITTEE

School of Computer Science and  
Engineering at Universidad Nacional del  
Sur – Argentina



# PREFACE

## **CACIC Congress**

CACIC is an annual Congress dedicated to the promotion and advancement of all aspects of Computer Science. The major topics can be divided into the broad categories included as Workshops (Intelligent Agents and Systems, Distributed and Parallel Processing, Software Engineering, Architecture, Nets and Operating Systems, Graphic Computation, Visualization and Image Processing, Information Technology applied to Education, Databases and Data Mining, Innovation in Software Systems, Security, Innovation in Computer Education, Computer Science Theory, Signal Processing, Real time Systems and Ethics in Computer Science).

The objective of CACIC is to provide a forum within which to promote the development of Computer Science as an academic discipline with industrial applications, trying to extend the frontier of both the state of the art and the state of the practice.

The main audience for, and participants in, CACIC are seen as researchers in academic departments, laboratories and industrial software organizations.

CACIC started in 1995 as a Congress organized by the Network of National Universities with courses of study in Computer Science (RedUNCI), and each year it is hosted by one of these Universities. RedUNCI has a permanent Web site where its history and organization are described: <http://redunci.info.unlp.edu.ar>.

## **CACIC 2012 in Bahía Blanca**

CACIC'12 was the eighteenth Congress in the CACIC series. It was organized by the School of Computer Science and Engineering at the Universidad Nacional del Sur.

The Congress included 13 Workshops with 178 accepted papers, 5 Conferences, 2 invited tutorials, different meetings related with Computer Science Education (Professors, PhD students, Curricula) and an International School with 5 courses. (<http://cs.uns.edu.ar/cacic2012/>).

CACIC 2012 was organized following the traditional Congress format, with 13 Workshops covering a diversity of dimensions of Computer Science Research. Each topic was supervised by a committee of 3-5 chairs of different Universities.

The call for papers attracted a total of 302 submissions. An average of 2.5 review reports were collected for each paper, for a grand total of 752 review reports that involved about 410 different reviewers.

A total of 178 full papers, involving 496 authors and 83 Universities, were accepted and 27 of them were selected for this book.

### **Acknowledgments**

CACIC 2012 was made possible due to the support of many individuals and organizations. The School of Computer Science and Engineering at Universidad Nacional del Sur, RedUNCI, the Secretary of University Policies, the National ministry of Science and Technology, CIC and CONICET were the main institutional sponsors.

This book is a very careful selection of best qualified papers. Special thanks are due to the authors, the members of the workshop committees, and all reviewers, for their contributions to the success of this book.

**ING. ARMANDO DE GIUSTI**

RedUNCI



## TABLE OF CONTENTS

### 17 **XIII Intelligent Agents and Systems Workshop**

Belief Revision and Defeasible Logic Programming Meet for Merging Ontologies

*Gómez, Sergio Alejandro (UNS) - Simari, Guillermo Ricardo (UNS)*

CLUIN-A New Method for Extracting Rules for Large Databases

*Hasperué, Waldo (UNLP) – Corbalán, Leonardo (UNLP)*

An approach for an algebra applied to a Defeasible Logic Programming

*Budán, Maximiliano (UNSE-CONICET), Gómez Lucero, Mauro (UNSE-CONICET), Simari, Guillermo Ricardo (UNS)*

### 53 **XII Distributed and Parallel Processing Workshop**

Power Characterisation of Shared-Memory HPC Systems

*Balladini, Javier (UNCOMA) - Rucci, Enzo (UNLP) - De Giusti, Armando E. (UNLP-CONICET) - Naiouf, Marcelo (UNLP) - Suppi, Remo (UAB)*

*– Rexachs, Dolores (UAB) – Luque, Emilio (UAB)*

A new approach for Analyzing I/O in parallel scientific applications

*Sandra Méndez, Sandra (UAB) - Panadero, Javier (UAB) - Wong, Alvaro (UAB) - Rexachs, Dolores (UAB) - Luque, Emilio (UAB)*

Analysis of Pure and Hybrid Parallel Solutions in a Simulation Problem.

*Gallo, Silvana Lis (UNLP) - Chichizola, Franco (UNLP) - De Giusti, Laura (UNLP) - Naiouf, Marcelo (UNLP)*

### 91 **X Information Technology Applied to Education Workshop**

PH-Helper: a Syntax-Directed Editor for Hoshimi Programming Language

*Luzza, Mariano (UNSL) - Berón, Mario Marcelo (UNSL) - Peralta, Mario (UNSL) - Salgado, Carlos (UNSL)*

ParasitePics: An Animal Parasitology Image Repository Prototype for Teaching and Learning.

*Martorelli, Sabrina Lorena (UNLP) - Sanz, Cecilia (UNLP) - Giacomantone, Javier (UNLP) - Martorelli, Sergio R. (UNLP-CONICET)*

Discovering Group Roles by Analyzing Collaborative Skills

*Ozán, Victor (UNSE) - Costaguta, Rosanna N. (UNSE) - Missio, Daniela (UNSE)*

Voice Recognition as Interaction Paradigm for People with Motor Difficulties

*Moralejo, Lucrecia (UNLP) - Sanz, Cecilia (UNLP) - Pesado, Patricia (UNLP-CIC)*

### **135 X Graphic Computation, Images and Visualization Workshop**

Techniques for an Image Occlusion Culling Engine

*Barbagallo, Leonardo R (UTN-FRBA) - Leone, Matias N (UTN-FRBA) - Banquero, Mariano M (UTN-FRBA) - Agromayor, Diego (UTN-FRBA) - Bursztyn, Andres (UTN-FRBA)*

A Virtual Reality Computing Platform for Real Time 3D Visualización

*Alvarado, Yoselie (UNSL) – Moyano, Manquel (UNSL) - Quiroga, Diego (UNSL) – Fernandez Jacqueline (UNSL) - Guerrero, Roberto (UNSL)*

### **159 IX Software Engineering Workshop**

ISO 9001:2008 Certification: Impact on the Requirements Engineering Process

*Arruzazabala, Mariana Cecilia (UNNE) - Dapozo, Gladys N. (UNNE) - Thomas, Pablo (UNLP)*

Using of knowledge representation techniques as a way to support requirements engineering

*Dessbesell Jr, Gilberto (UNISC) - Kroth, Eduardo (UNISC)*

ISO 9001 in software-developing VSEs assisted by the COMPETISOFT Model

*Esponda, Silvia (UNLP) - Pasini, Ariel (UNLP) - Pesado, Patricia (UNLP-CIC) - Boracchia, Marcos (UNLP)*

### **191 IX Database and Data Mining Workshop**

Histogram-based intrinsic plagiarism detection

*Funez, Dario (UNSL) - Errecalde, Marcelo (UNSL)*

Automatic Text Classification through Word Similarity and Hidden Meanings

*Catae, Fabricio S. (USP) - Rocha, Ricardo L. A. (USP)*

### **203 VII Architecture, Nets and Operating Systems Workshop**

Histogram-based intrinsic plagiarism detection

*Funez, Dario (UNSL) - Errecalde, Marcelo (UNSL)*

Automatic Text Classification through Word Similarity and Hidden Meanings

*Catae, Fabricio S. (USP) - Rocha, Ricardo L. A. (USP)*

**233 IV Innovation in Software Systems Workshop**

Assessment of Writing Text in Mobile Devices

*Capa-Arnao, Rodrigo (UIB) - Manresa-Yee, Cristina (UIB) - Mas Sanso, Ramon (UIB) - Larrea, Martín (UNS)*

**241 III Computer Science Theoretical Aspects Workshop**

Expressibility of the Logic SOF on Classes of Structures of Bounded FO Types

*Grosso, Alejandro (UNSL) - Turull Torres, Jose María (UNSL-UCuenca del Plata)*

**251 III Signal Processing and Real-Time Systems Workshop**

Real time images processing using embedded technology. Application in visual rehabilitation.

*Raponi, Marcelo (UNSAM) - Pawelko, Ezequiel (IUPFA) - Arcusin, Leandro (UTN-FRBA) - Delrieux, Claudio (UNS-CONICET)*

Preliminary Work to Get Synchronized Times in Multicore Node

*Romero, Fernando (UNLP) - Villagarcía Wanza, Horacio (UNLP-CIC) - Tinetti, Fernando G. (UNLP-CIC)*

**265 I Computer Security Workshop**

An analysis of network traffic characteristics for Botnet detection

*Erquiaga, María José (UNCUYO) - Catania, Carlos (UNCUYO) - Garcia Garino, Carlos (UNCUYO)*

**275 I Innovation in Computer Science Education Workshop**

Using Facebook as a Support Tool in an Undergraduate Computer Science Course

*Martinez de la Cruz, Victoria (UNS) - Chesñevar, Carlos Iván (UNS)*

A multiplatform interpreter to introduce structured and concurrent programming

*Depetris, Beatriz (UNTDF) - Feierherd, Guillermo (UNTDF) - Aguil Mallea, Daniel (UNTDF) - Tejero, Germán (UNTDF)*

**295 II ETHICOMP LatinAmerica**

Open Access to Scientific Literature and Research Data: A Window of Opportunity for Latin America

*Molfino, María del Rosario (UNLP) - González, Claudia M (UNLP)*



# Belief Revision and Defeasible Logic Programming Meet for Merging Ontologies<sup>\*</sup>

SERGIO A. GÓMEZ AND GUILLERMO R. SIMARI

Artificial Intelligence Research and Development Laboratory (LIDIA),  
Department of Computer Science and Engineering, Universidad Nacional del Sur,  
Av. Alem 1253, (8000) Bahía Blanca, Argentina,  
Email: {sag,grs}@cs.uns.edu.ar

***Abstract.** In this paper we combine argumentation, belief revision and description logic ontologies for extending the  $\delta$ -ontologies framework in order to merge two ontologies in which the union of the strict terminologies could lead to inconsistency. To do this, we revisit a procedure presented by Falappa et al. that allows to convert part of the offending terminologies into defeasible by using a kernel revision operator. We present a set of definitions to formalize the process along with a case study.*

## 1. Introduction

The confluence of Description Logics and argumentation is an important research topic as shown by the ever growing list of publications that can be found on the subject [1-5]. Description Logics [6] constitute the semantic substrate of the Web Ontology Language OWL [7], which is at the core of the Semantic Web initiative. The Semantic Web [8] is a vision of the current Web where resources have exact meaning assigned in terms of knowledge bases called *ontologies* [9], enabling agents to reason about them. Argumentation [10,11] is a form of non-monotonic reasoning that allows to obtain consequences from possibly inconsistent knowledge bases. On the other hand, belief revision is the process of changing beliefs to take into account a new piece of information; in spite of the union of argumentation and belief revision not being new it can be regarded as a live research topic (see [12-14]).

In [1], Gómez *et al.* developed a framework called  $\delta$ -ontologies that allows to reason in the presence of inconsistent description logic ontologies by using Defeasible Logic Programming [15], which is an argumentative framework based on logic programming. In a  $\delta$ -ontology the terminology defining the

---

<sup>\*</sup> This research is funded by the Project *Representación de Conocimiento y Razonamiento Argumentativo: Herramientas Inteligentes para la Web y las Bases de Datos Federadas (24/N030)*, Agencia Nacional de Promoción Científica y Tecnológica and Universidad Nacional del Sur.

vocabulary is separated in strict and defeasible, the former is inconsistency free but the latter could be not. While joining two defeasible terminologies is trivial, joining two strict terminologies can lead to inconsistency if done careless. In this article, we combine argumentation, belief revision and description logic ontologies for extending the  $\delta$ -ontologies framework to show how to merge two ontologies in which the union of the strict terminologies could lead to inconsistency. To solve this problem, we revisit a procedure presented by Falappa *et al.* [12] in which part of the offending terminologies are turned *defeasible* by using a kernel revision operator.

The rest of this paper is structured as follows. In Section 2 we briefly present the fundamentals of Description Logics and Defeasible Logic Programming. Section 3 briefly recalls the framework of  $\delta$ -ontologies for reasoning with possibly inconsistent ontologies. In Section 4, we extend the  $\delta$ -ontologies framework to allow for merging strict terminologies while conserving consistency. Finally Section 5 concludes.

## 2. Background

### 2.1 Fundamentals of Description Logics

*Description Logics* (DL) are a family of knowledge representation formalisms [6]. They are based on the notions of *concepts* (unary predicates, classes) and *roles* (binary relations) that allow building complex concepts and roles from atomic ones. Let  $C, D$  stand for concepts,  $R$  for a role and  $a, b$  for individuals. Concept descriptions are built from concept names using the constructors conjunction ( $C \sqcap D$ ), disjunction ( $C \sqcup D$ ), complement ( $\neg C$ ), existential restriction ( $\exists R.C$ ), and value restriction ( $\forall R.C$ ). To define the semantics of concept descriptions, concepts are interpreted as subsets of a domain of interest, and roles as binary relations over this domain. Further extensions are possible including inverse and transitive roles, noted as  $P^-$  and  $P^+$ , resp.

A DL ontology consists of two finite and mutually disjoint sets: a *Tbox* which introduces the *terminology* and an *Abox* which contains facts about particular objects in the application domain. Tbox statements have the form  $C \sqsubseteq D$  (*inclusions*) and  $C \equiv D$  (*equalities*), where  $C$  and  $D$  are (possibly complex) concept descriptions. Objects in the Abox are referred to by a finite number of *individual names* and these names may be used in two types of assertional statements: *concept assertions* of the type  $a:C$  and *role assertions* of the type  $\langle a, b \rangle : R$ , where  $C$  is a concept description,  $R$  is a role name, and  $a$  and  $b$  are individual names.

## 2.2 Defeasible Logic Programming

*Defeasible Logic Programming* (DeLP) [15] provides a language for knowledge representation and reasoning that uses defeasible argumentation [10] to decide between contradictory conclusions through a *dialectical analysis*. Codifying knowledge by means of a DeLP program provides a good trade-off between expressivity and implementability for dealing with incomplete and potentially contradictory information. In a defeasible logic program  $P=(\Pi, \Delta)$ , a set  $\Pi$  of strict rules  $P \leftarrow Q_1, \dots, Q_n$ , and a set  $\Delta$  of defeasible rules  $P \prec Q_1, \dots, Q_n$ , can be distinguished. An *argument*  $\langle A, H \rangle$  is a minimal non-contradictory set of ground defeasible clauses  $A$  of  $\Delta$  that allows to derive a ground literal  $H$  possibly using ground rules of  $\Pi$ . Since arguments may be in conflict (concept captured in terms of a logical contradiction), an attack relationship between arguments can be defined. A criterion is usually defined to decide between two conflicting arguments. If the attacking argument is strictly preferred over the attacked one, then it is called a *proper defeater*. If no comparison is possible, or both arguments are equi-preferred, the attacking argument is called a *blocking defeater*. In order to determine whether a given argument  $A$  is ultimately undefeated (or *warranted*), a dialectical process is recursively carried out, where defeaters for  $A$ , defeaters for these defeaters, and so on, are taken into account. Given a DeLP program  $P$  and a query  $H$ , the final answer to  $H$  w.r.t.  $P$  takes such dialectical analysis into account. The answer to a query can be: *yes*, *no*, *undecided*, or *unknown*.

## 3. Reasoning with Inconsistent Ontologies in DeLP

In the presence of inconsistent ontologies, traditional DL reasoners issue an error message and the knowledge engineer must then debug the ontology (*i.e.* making it consistent). In [1], Gómez *et al.* showed how DeLP can be used for coping with inconsistencies in ontologies such that the task of dealing with them is automatically solved by the reasoning system. We recall some of the concepts for making the article more self-contained.

Not all DL ontologies can be translated into logic programming. For being able to be translated, they must respect certain constraints.  $L_b$ -classes are class descriptions that can occur only in the body of Tbox inclusion axioms,  $L_h$ -classes are class descriptions that can occur only in the head of Tbox inclusion axioms. Finally  $L_{hb}$ -classes can occur in both head and body and therefore they can be used in Tbox equivalence axioms.

**Definition 1 ( $\delta$ -Ontology).** Let  $C$  be an  $L_b$ -class,  $D$  an  $L_h$ -class,  $A, B$   $L_{hb}$ -classes,  $P, Q$  properties,  $a, b$  individuals. Let  $T$  be a set of inclusion and equality sentences in the language of description logics  $L_{DL}$  of the form  $C \sqsubseteq D$ ,  $A \equiv B$ ,  $\top \sqsubset \forall P.D$ ,  $\top \sqsubseteq \forall P^-.D$ ,  $P \sqsubseteq Q$ ,  $P \equiv Q$ ,

$P \equiv Q^-$ , or  $P^+ \sqsubseteq P$  such that  $T$  can be partitioned into two disjoint sets  $T_S$  and  $T_D$ . Let  $A$  be a set of assertions disjoint with  $T$  of the form  $a:D$  or  $\langle a,b \rangle:P$ . A  $\delta$ -ontology  $\Sigma$  is a tuple  $(T_S, T_D, A)$ . The set  $T_S$  is called the *strict terminology* (or Sbox),  $T_D$  the *defeasible terminology* (or Dbox) and  $A$  the *assertional box* (or Abox).

*Example 1.* Figure 1 presents a very simple  $\delta$ -ontology  $\Sigma_1 = (\emptyset, T_D^1, A^1)$  that expresses that every man is a mortal unless he is a Highlander. Socrates is a man and McLeod is both a man and a Highlander.

<b>Ontology <math>\Sigma_1 = (\emptyset, T_D^1, A^1)</math></b>	
<b>Defeasible terminology <math>T_D^1</math>:</b>	<b>Assertional box <math>A^1</math>:</b>
Man $\sqsubseteq$ Mortal	SOCRATES : Man
Man $\sqcap$ Highlander $\sqsubseteq$ $\neg$ Mortal	MCLEOD : Man
	MCLEOD : Highlander

*Fig. 1.* A very simple  $\delta$ -ontology

For assigning semantics to a  $\delta$ -ontology, two translation functions  $T_\Delta$  and  $T_\Pi$  from  $L_{DL}$  to the DeLP language  $L_{DeLP}$  were defined based on the work of [16] (for details see [1]).

**Definition 2 ( $T_\Pi^*$  mapping from DL sentences to DeLP strict rules).** Let  $A, C, D$  be concepts,  $X, Y$  variables,  $P, Q$  properties. The  $T_\Pi^* : 2^{LDL} \rightarrow 2^{LDeLP}$  mapping is defined in Fig. 2. Besides, intermediate transformations of the form “ $(H_1 \wedge H_2) \leftarrow B$ ” will be rewritten as two rules “ $H_1 \leftarrow B$ ” and “ $H_2 \leftarrow B$ ”. Similarly transformations of the form “ $H_1 \leftarrow H_2 \leftarrow B$ ” will be rewritten as “ $H_1 \leftarrow B \wedge H_2$ ”, and rules of the form “ $H \leftarrow (B_1 \vee B_2)$ ” will be rewritten as two rules “ $H \leftarrow B_1$ ” and “ $H \leftarrow B_2$ ”.



$$\begin{aligned}
T_{\Pi}^* (\{C \sqsubseteq D\}) &=_{df} \{ T_h(D, X) \leftarrow T_b(C, X) \}, \\
&\quad \text{if } C \text{ is an } \mathcal{L}_b\text{-class and } D \text{ an } \mathcal{L}_h\text{-class} \\
T_{\Pi}^* (\{C \equiv D\}) &=_{df} T_{\Pi}^* (\{C \sqsubseteq D\}) \cup T_{\Pi}^* (\{D \sqsubseteq C\}), \\
&\quad \text{if } C \text{ and } D \text{ are } \mathcal{L}_{h,b}\text{-classes} \\
T_{\Pi}^* (\{\top \sqsubseteq \forall P.D\}) &=_{df} \{ T_h(D, Y) \leftarrow P(X, Y) \}, \\
&\quad \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
T_{\Pi}^* (\{\top \sqsubseteq \forall P^-.D\}) &=_{df} \{ T_h(D, X) \leftarrow P(X, Y) \}, \\
&\quad \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
T_{\Pi}^* (\{a : D\}) &=_{df} \{ T_h(D, a) \}, \\
&\quad \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
T_{\Pi}^* (\{\langle a, b \rangle : P\}) &=_{df} \{ P(a, b) \} \\
T_{\Pi}^* (\{P \sqsubseteq Q\}) &=_{df} \{ Q(X, Y) \leftarrow P(X, Y) \} \\
T_{\Pi}^* (\{P \equiv Q\}) &=_{df} \left\{ \begin{array}{l} Q(X, Y) \leftarrow P(X, Y) \\ P(X, Y) \leftarrow Q(X, Y) \end{array} \right\} \\
T_{\Pi}^* (\{P \equiv Q^-\}) &=_{df} \left\{ \begin{array}{l} Q(X, Y) \leftarrow P(Y, X) \\ P(Y, X) \leftarrow Q(X, Y) \end{array} \right\} \\
T_{\Pi}^* (\{P^+ \sqsubseteq P\}) &=_{df} \{ P(X, Z) \leftarrow P(X, Y) \wedge P(Y, Z) \} \\
T_{\Pi}^* (\{s_1, \dots, s_n\}) &=_{df} \bigcup_{i=1}^n T_{\Pi}^* (\{s_i\}), \text{ if } n > 1 \\
&\quad \text{where:} \\
T_h(A, X) &=_{df} A(X) \\
T_h((C \sqcap D), X) &=_{df} T_h(C, X) \wedge T_h(D, X) \\
T_h((\forall R.C), X) &=_{df} T_h(C, Y) \leftarrow R(X, Y) \\
T_b(A, X) &=_{df} A(X) \\
T_b((C \sqcap D), X) &=_{df} T_b(C, X) \wedge T_b(D, X) \\
T_b((C \sqcup D), X) &=_{df} T_b(C, X) \vee T_b(D, X) \\
T_b((\exists R.C), X) &=_{df} R(X, Y) \wedge T_b(C, Y)
\end{aligned}$$

Fig. 2. Mapping from DL ontologies to DeLP strict rules

**Definition 3 (Transposes of a strict rule).** Let  $r = H \leftarrow B_1, B_2, B_3, \dots, B_{n-1}, B_n$  be a DeLP strict rule. The set of transposes of rule  $r$ , noted as “Trans( $r$ )”, is defined as:

$$\text{Trans}(r) = \left\{ \begin{array}{l} H \leftarrow B_1, B_2, \dots, B_{n-1}, B_n \\ \overline{B_1} \leftarrow \overline{H}, B_2, B_3, \dots, B_{n-1}, B_n \\ \overline{B_2} \leftarrow \overline{H}, B_1, B_3, \dots, B_{n-1}, B_n \\ \overline{B_3} \leftarrow \overline{H}, B_1, B_2, \dots, B_{n-1}, B_n \\ \dots \\ \overline{B_{n-1}} \leftarrow \overline{H}, B_1, B_2, B_3, \dots, B_{n-1}, B_n \\ \overline{B_n} \leftarrow \overline{H}, B_1, B_2, \dots, B_{n-1} \end{array} \right\}$$

**Definition 4 ( $T_{\Pi}$  mapping from DL sentences to DeLP strict rules).** The mapping from DL ontologies into DeLP strict rules is defined as  $T_{\Pi}(T) = \text{Trans}(T_{\Pi}^*(T))$ .

**Definition 5 (Interpretation of a  $\delta$ -ontology).** Let  $\Sigma = (T_S, T_D, A)$  be a  $\delta$ -ontology. The interpretation of  $\Sigma$  is a DeLP program  $P = (T_{\Pi}(T_S) \cup T_{\Pi}(A), T_{\Delta}(T_D))$ .

Notice that in order to keep consistency within an argument, some internal coherence between the Abox and the Tbox must be enforced: given a  $\delta$ -ontology  $\Sigma = (T_S, T_D, A)$ , it must not be possible to derive two complementary literals from  $T_{\Pi}(T_S) \cup T_{\Pi}(A)$ .

**Definition 6 (Potential, justified and strict membership of an individual to a class)** Let  $\Sigma=(T_S, T_D, A)$  be a  $\delta$ -ontology,  $C$  a class name,  $a$  an individual, and  $P = (T_{\Pi}(T_S) \cup T_{\Pi}(A), T_{\Delta}(T_D))$ .

1. The individual  $a$  potentially belongs to class  $C$ , iff there exists an argument  $\langle A, C(a) \rangle$  w.r.t.  $P$ ;
2. the individual  $a$  potentially belongs to class  $C$ , iff there exists a warranted argument  $\langle A, C(a) \rangle$  w.r.t.  $P$ , and,
3. the individual  $a$  potentially belongs to class  $C$ , iff there exists and argument  $\langle \emptyset, C(a) \rangle$  w.r.t.  $P$ .

*Example 2 (Continues Ex. 1).* Consider again the  $\delta$ -ontology  $\Sigma_1$ , it is interpreted as the DeLP program  $P_1$  according to Def. 5 as shown in Fig. 3. From  $P_1$ , we can determine that Socrates justifiedly belongs to the concept **Mortal** in  $\Sigma_1$  as as there exists a warranted argument structure  $\langle A_1, mortal(socrates) \rangle$  where

$$A_1 = \{ mortal(socrates) -< man(socrates) \}.$$

Likewise, we can determine that Mcleod justifiedly belongs to the concept  $\neg$ **Mortal** in  $\Sigma_1$ . We can see that Mcleod potentially belongs to **Mortal**, as in the case of Socrates, for there is an argument  $\langle B_1, mortal(mcleod) \rangle$  where

$$B_1 = \{ mortal(mcleod) -< man(mcleod) \}.$$

This argument  $B_1$  is however defeated by  $\langle B_2, \sim mortal(mcleod) \rangle$ , where

$$B_2 = \{ \sim mortal(mcleod) -< man(mcleod), highlander(mcleod) \}.$$

Notice that  $B_2$  is more specific than  $B_1$  (see [17] for details), thus  $B_1$  cannot defeat  $B_2$ .

<b>DeLP program <math>P_1=(\Pi_1, \Delta_1)</math> obtained from <math>\Sigma_1</math>:</b>	
<b>Facts <math>\Pi_1</math>:</b> <i>man(socrates).</i> <i>man(mcleod).</i> <i>highlander(mcleod).</i>	<b>Defeasible rules <math>\Delta_1</math>:</b> <i>mortal(X) <math>\multimap</math> man(X).</i> <i><math>\sim mortal(X) \multimap man(X), highlander(X).</math></i>

*Fig. 3. DeLP program  $P_1$  obtained from ontology  $\Sigma_1$*

## 4. Ontology Merging based on Belief Revision Theory

We now extend the  $\delta$ -ontologies framework to allow for ontology merging based on belief revision. First the fundamentals of belief revision are presented. Next we present the actual extension along with a running example.

## 4.1 Fundamentals of Belief Revision

*Belief revision* is the process of changing beliefs to take into account a new piece of information. Two kinds of changes are usually distinguished [18,19]: *update*, in which new information must be considered with respect to a set of old beliefs, then update refers to the operation of changing the old beliefs to take into account the change; and *revision*, where there are old beliefs and new information; in this case the new information is considered more reliable, then revision is the process of inserting the new information into the set of old beliefs without generating an inconsistency. Belief revision should produce minimal change, *i.e.* the knowledge before and after the change should be as similar as possible.

A revision operator is a function that maps sets of sentences  $K$  and  $A$  to a new set of sentences. In particular, in Falappa *et al.* [12] the mechanism of a revision operator  $K \circ A$  by a set of sentences with partial acceptance is defined as follows: first, the input set  $A$  is initially accepted, and, second, all possible inconsistencies of  $K \cup A$  are removed. The mechanism of this operator consists of adding  $A$  to  $K$  and then eliminating from the result all possible inconsistencies by means of an incision function that makes a “cut” over each minimally inconsistent subset of  $K \cup A$ .

In [12], beliefs are split into two distinguished sets: (i) *particular beliefs*  $K_P$ , that are represented by ground facts, and (ii) *general beliefs*  $K_G$ , that are represented by closed material implications. Thus, each belief base  $K$  has the form  $K_P \cup K_G$  where  $K_P \cap K_G = \emptyset$ . When doing a kernel revision by a set of sentences, an incision function is needed to make a cut upon every set; *i.e.* it is necessary to determine which beliefs must be given up in the revision process. They consider two possible policies: discard particular beliefs and discard general beliefs. In the latter, at least one sentence is discarded. Thus [12] propose a refined characterization of revision by preserving retracted beliefs with a different status: retracted general beliefs are preserved as *defeasible rules*. They also introduce a revision operator that generates defeasible conditionals from a revision operator upon belief bases represented in a first order language. It may be the case that in the revision process a conditional sentence of the form  $(\forall(X))(\alpha(X) \rightarrow \beta(X))$  has to be eliminated. This can occur because new incoming information results in an inconsistency. One of the following cases may occur: (i) there exists some individual satisfying  $\alpha$  but not satisfying  $\beta$ , and (ii) there exists some individual satisfying  $\neg\beta$  but not satisfying  $\neg\alpha$ . Eliminating  $(\forall(X))(\alpha(X) \rightarrow \beta(X))$  from the knowledge base produces too much loss of information. As an alternative, Falappa *et al.* propose a transformation to change it into  $\beta \prec \alpha$ . Formally:

**Definition 7 (Positive/negative transformation [12]).** Let  $\delta = (\forall X_1 \dots X_n)(\alpha \rightarrow \beta)$  be a material implication in DeLP. A positive transformation of  $\delta$ ,

noted by  $T^+(\delta)$ , is a sentence of the form  $\beta \text{ -< } \alpha$ ; a negative transformation of  $\delta$ , noted by  $T^-(\delta)$ , is a sentence of the form  $\text{-}\beta \text{ -< } \text{-}\alpha$ .

**Definition 8 (Kernel (partial meet) composed revision [12]).** Let  $(K, \Delta)$  be a knowledge structure,<sup>1</sup> ( $^\circ$ ) an operator of kernel (partial meet) revision by a set of sentences for  $K$  and  $A$  a set of sentences. The kernel (partial meet) composed revision of  $(K, \Delta)$  w.r.t.  $A$  is defined as:  $(K, \Delta) * A = (K', \Delta')$  such that  $K' = K \circ A$  and  $\Delta' = \Delta \cup \Delta_1' \cup \Delta_2'$  where:

$$\begin{aligned} \Delta_1' &= \{ \alpha \text{ -< true} \mid \alpha \in (K_P \setminus K^\circ A) \} \\ \Delta_2' &= \{ T^+(\alpha) \mid \alpha \in (K_G \setminus K^\circ A) \} \cup \{ T^-(\alpha) \mid \alpha \in (K_G \setminus K^\circ A) \}. \end{aligned}$$

The set  $K'$  contains the revised undefeasible beliefs,  $\Delta_1'$  is the transformation in defeasible rules of particular beliefs (also called *presumptions* [15, Section 6]) eliminated from  $K$  whereas  $\Delta_2'$  is the transformation of general beliefs eliminated from  $K$  into defeasible rules.

## 4.2 Merging of d-ontologies using Belief Revision

We now adapt the reasoning framework for  $\delta$ -ontologies to use it in ontology merging. Merging is the process of creating a new ontology from two or more existing ontologies with overlapping parts [20]. Suppose we have two *strict* ontologies<sup>2</sup> that we desire to reason with at the same time. However accepting both ontologies at once may generate inconsistencies. The simplest solution is to consider the ontologies as *defeasible information*. This solution is too simplistic, a smarter approach consists of *transforming* into defeasible the part of the ontologies producing the inconsistency and letting the part which is not in conflict as is.

For simplicity, in the following discussion we assume *unique name assumption* (UNA). If UNA could not be assumed, it would be always possible to use an ontology integration schema based on a mapping function as it was presented in Gómez *et al.*[1].

Let  $\Sigma_1$  and  $\Sigma_2$  be two  $\delta$ -ontologies. The merged ontology between  $\Sigma_1$  and  $\Sigma_2$  is noted as  $\Sigma_1 \oplus \Sigma_2$ . In the same way as with single  $\delta$ -ontologies, the merge of two  $\delta$ -ontologies will be interpreted as a DeLP program.

**Definition 9 (Interpretation of a merged  $\delta$ -ontology).** Let  $\Sigma_1$  and  $\Sigma_2$  be two  $\delta$ -ontologies such that  $\Sigma_1 = (T_S^1, T_D^1, A^1)$  and  $\Sigma_2 = (T_S^2, T_D^2, A^2)$ . The interpretation of the merged  $\delta$ -ontology  $\Sigma_1 \oplus \Sigma_2$ , noted as  $T(\Sigma_1 \oplus \Sigma_2)$ , is defined as the DeLP program  $(\Pi, \Delta)$  where

$$\begin{aligned} \Pi_1 &= T_\Pi(T_S^1) \cup T_\Pi(A^1); \\ \Delta_1 &= T_\Delta(T_D^1); \end{aligned}$$

<sup>1</sup> In [12], a DeLP program composed of material implications instead of derivation rules is called a *knowledge structure*.

<sup>2</sup> This situation can be modeled by two  $\delta$ -ontologies with non-empty Sbox, and empty Dbox and non-empty Abox.

$$\begin{aligned}
\Pi_2 &= T_{\Pi}(T_S^2) \cup T_{\Pi}(A^2); \\
\Delta_2 &= T_{\Delta}(T_D^2); \\
(\Pi, \Delta') &= (\Pi_1, \Delta_1) * \Pi_2, \text{ and} \\
\Delta &= \Delta_1 \cup \Delta_2 \cup \Delta'.
\end{aligned}$$

We now extend the reasoning tasks over ABoxes for the case of a merged ontology. In particular, we define the *instance checking operation* for merged ontologies.

**Definition 10 (Instance checking for a merged  $\delta$ -ontology).** Let  $\Sigma_1$  and  $\Sigma_2$  be two  $\delta$ -ontologies,  $C$  a concept name and  $a$  an individual name.

- The individual  $a$  is a *potential member* of the concept  $C$  w.r.t.  $\Sigma_1 \oplus \Sigma_2$  iff there exists an argument  $\langle A, C(a) \rangle$  w.r.t.  $T(\Sigma_1 \oplus \Sigma_2)$ .
- The individual  $a$  is a *justified member* of the concept  $C$  w.r.t.  $\Sigma_1 \oplus \Sigma_2$  iff there exists a warranted argument  $\langle A, C(a) \rangle$  w.r.t.  $T(\Sigma_1 \oplus \Sigma_2)$ .
- The individual  $a$  is a *strict member* of the concept  $C$  w.r.t.  $\Sigma_1 \oplus \Sigma_2$  iff there exists an argument  $\langle \emptyset, C(a) \rangle$  w.r.t.  $T(\Sigma_1 \oplus \Sigma_2)$ .
- The individual  $a$  is an *indeterminate member* of the concept  $C$  w.r.t.  $\Sigma_1 \oplus \Sigma_2$  iff there is no argument for the literal  $C(a)$  w.r.t.  $T(\Sigma_1 \oplus \Sigma_2)$ .

**Definition 11 (Set of justified and strict answers).** Let  $\Sigma$  be a  $\delta$ -ontology,  $a$  an individual and  $p$  a concept. The set of justified answers of  $\Sigma$  is the set of literals  $p(a)$  such that  $a$  belongs justifiedly to  $p$ . The set of strict answers of  $\Sigma$  stands for all the literals  $p(a)$  such that  $a$  belongs strictly to  $p$ .

*Example 3.* Suppose we are given the  $\delta$ -ontology  $\Sigma_1 = (T_S^1, \emptyset, A^1)$  where:

$$\begin{aligned}
T_S^1 &= \left\{ \begin{array}{l} \text{Penguin} \sqsubseteq \text{Bird} \\ \text{Bird} \sqsubseteq \text{Flies} \end{array} \right\}, \text{ and} \\
A^1 &= \left\{ \begin{array}{l} \text{TWEETY} : \text{Bird} \\ \text{OPUS} : \text{Penguin} \end{array} \right\}.
\end{aligned}$$

The set of strict answers of this ontology is the set of literals:  $\{bird(tweety), penguin(opus), bird(opus), flies(tweety), flies(opus)\}$ .

Let us suppose that we receive another  $\delta$ -ontology  $\Sigma_2 = (T_S^2, \emptyset, A^2)$ , viewed as an explanation for “OPUS:  $\neg$ Flies”, where:

$$\begin{aligned}
T_S^2 &= \{ \text{Bird} \sqcap \text{Penguin} \sqsubseteq \neg \text{Flies} \}, \text{ and} \\
A^2 &= \left\{ \begin{array}{l} \text{OPUS} : \text{Bird} \\ \text{OPUS} : \text{Penguin} \end{array} \right\}.
\end{aligned}$$

Suppose that we now desire to find the DeLP program  $P = (\Pi, \Delta) = T(\Sigma_1 \oplus \Sigma_2)$  that interprets the  $\delta$ -ontology which merges  $\Sigma_1$  and  $\Sigma_2$ . When we compute the interpretation of the merged ontology, we must perform a kernel revision by a set of sentences. We need to find the minimally inconsistent subsets of the set of DeLP sentences:  $T_{\Pi}(A^1) \cup T_{\Pi}(T_S^1) \cup T_{\Pi}(A^2) \cup T_{\Pi}(T_S^2)$ . The two sets in this condition are:

$$\begin{aligned}
& 1. \mathfrak{T}\text{rans}\left(\left\{ \begin{array}{l} \textit{bird}(\textit{opus}), \\ \textit{penguin}(\textit{opus}), \\ (\textit{flies}(X) \leftarrow \textit{bird}(X), \textit{penguin}(X)), \\ (\textit{flies}(X) \leftarrow \textit{bird}(X)) \end{array} \right\}, \text{ and} \right. \\
& 2. \left. \mathfrak{T}\text{rans}\left(\left\{ \begin{array}{l} \textit{penguin}(X), \\ (\textit{bird}(X) \leftarrow \textit{penguin}(X)), \\ (\textit{flies}(X) \leftarrow \textit{bird}(X)), \\ (\sim\textit{flies}(X) \leftarrow \textit{bird}(X), \textit{penguin}(X)) \end{array} \right\} \right\}.
\end{aligned}$$

To discard general beliefs, we must discard at least one sentence in each set above. As the sentence “ $\textit{flies}(X) \leftarrow \textit{bird}(X)$ ” is in both sets, it can be discarded. The set  $\Pi$  of strict rules of the revised ontology is then composed by:

$$\begin{aligned}
\Pi = & \left\{ \begin{array}{l} \textit{bird}(\textit{tweety}), \\ \textit{bird}(\textit{opus}), \\ \textit{penguin}(\textit{opus}) \end{array} \right\} \cup \mathfrak{T}\text{rans}(\{ \textit{bird}(X) \leftarrow \textit{penguin}(X) \}) \cup \\
& \mathfrak{T}\text{rans}(\{ \sim\textit{flies}(X) \leftarrow \textit{bird}(X), \textit{penguin}(X) \}).
\end{aligned}$$

In this case, the set of strict answers of the merged ontology  $\Sigma_1 \oplus \Sigma_2$  is  $\{\textit{bird}(\textit{tweety}), \textit{bird}(\textit{opus}), \textit{penguin}(\textit{opus}), \sim\textit{flies}(\textit{opus})\}$ . Nevertheless, the set of deleted sentences are not completely forgotten but stored as defeasible rules. That is, the set  $\Delta$  of defeasible rules in the interpretation of the merged  $\delta$ -ontology is  $\Delta = \{(\textit{flies}(X) \prec \textit{bird}(X)), (\sim\textit{bird}(X) \prec \sim\textit{flies}(X))\}$ . Then the set of justified answers of  $\Sigma_1 \oplus \Sigma_2$  is  $\{\textit{bird}(\textit{tweety}), \textit{bird}(\textit{opus}), \textit{penguin}(\textit{opus}), \sim\textit{flies}(\textit{opus}), \textit{flies}(\textit{tweety})\}$ . Notice that the literal “ $\textit{flies}(\textit{tweety})$ ” is present in the set of justified answers but it is not in the set of strict answers; *i.e.* we are now able to conclude that the individual Tweety is a justified member of the concept **Flies**.

## 5. Conclusion

We have presented an approach for merging ontologies based on Belief Revision and Defeasible Logic Programming. We have combined argumentation, belief revision and description logic ontologies for extending the  $\delta$ -ontologies framework and thus showing how to merge two ontologies in which the union of the strict terminologies could lead to inconsistency. To solve this problem, we revisited a procedure presented by [12] in which part of the offending terminologies are turned *defeasible* by using a kernel revision operator. We have presented a framework for characterizing the behavior of the proposed approach and an example scenario. Future work includes characterizing mathematical properties of the approach.

## References

1. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: Reasoning with Inconsistent Ontologies Through Argumentation. *Applied Artificial Intelligence* **24**(1) (2010) 102-148.
2. Zhang, X., Zhang Z., Lin, Z.: An Argumentative Semantics for Paraconsistent Reasoning in Description Logic ALC (2009).
3. Bassiliades, N., Antoniou, G. Vlahavas, I.P.: DR-DEVICE: A Defeasible Logic System for the Semantic Web. In: PPSWR 2004. (2004) 134-148.
4. Wang, K., Billington, D., Blee, J., Antoniou, G.: Combining Description Logic and Defeasible Logic for the Semantic Web. In: Rule ML 2004. (2004) 170-181.
5. Grigoris Antoniou, Antonis Bikakis, G.W.: A System for Nonmonotonic Rules on the Web. In: RuleML 2004. (2004) 23-36.
6. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: *The Description Logic Handbook - Theory, Implementation and Applications*. Cambridge University Press (2003).
7. McGuinness, D.L., van Harmelen, F.: *OWL Web Ontology Language Overview* (2004).
8. Berners-Lee, T., Hendler, J., Lassila, O.: *The Semantic Web*. Scient. American (2001).
9. Gruber, T.R.: A translation approach to portable ontologies. *Knowledge Acquisition* **5**(2) (1993) 199-220.
10. Chesñevar, C.I., Maguitman, A., Loui, R.: Logical Models of Argument. *ACM Computing Surveys* **32**(4) (December 2000) 337-383.
11. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in Artificial Intelligence. *Artificial Intelligence*. **171** (10-15) (2007) 619-641.
12. Falappa, M.A., Kern-Isberner, G., Simari, G.R.: Explanations, Belief Revision and Defeasible Reasoning. *Artificial Intelligence* **141** (2002) 1-28.
13. Falappa, M.A., García, A.J., Kern-Isberner, G., Simari, G.R.: On the evolving relation between Belief Revision and Argumentation. *The Knowledge Engineering Review* **26**(1) (2011) 35-43.
14. Falappa, M.A., Kern-Isberner, G., Simari, G.R.: Belief Revision and Argumentation Theory. In Rahwan, I., Simari, G.R., eds.: *Argumentation in Artificial Intelligence*, Springer (2009) 341-360.
15. García, A., Simari, G.: Defeasible Logic Programming an Argumentative Approach. *Theory and Practice of Logic Programmming* **4**(1) (2004) 95-138.
16. Grosz, B.N., Horrocks, I., Volz, R., Decker, S.: *Description Logic Programs: Combining Logic Programs with Description Logics*. WWW2003, May 20-24, Budapest, Hungary (2003).
17. Stolzenburg, F., García, A., Chesñevar, C., Simari, G.: Computing Generalized Specificity. *J. of N. Classical Logics* **13**(1) (2003) 87-113.
18. Alchourron, C., Gärdenfors, P., Makinson, D.: On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic* (50) (1985) 510-530.

19. Falappa, M.A.: Teoría de Cambio de Creencias y sus Aplicaciones sobre Estados de Conocimiento. PhD thesis, Universidad Nacional del Sur (1999).
20. Klein, M.: Combining and relating ontologies: An analysis of problems and solutions. In Gomez-Perez, A., Gruninger, M., Stuckenschmidt, H., Uschold, M., eds.: Workshop on Ontologies and Information Sharing, IJCAI'01, Seattle, USA (2001).



# CLUIN – A New Method for Extracting Rules for Large Databases

WALDO HASPERUÉ<sup>1,2</sup> AND LEONARDO CORBALAN<sup>1</sup>

<sup>1</sup>Instituto de Investigación en Informática LIDI - School of Computer Science - National University of La Plata, Argentina.

<sup>2</sup> CONICET scholarship

{whasperue, corbalan}@lidi.info.unlp.edu.ar

***Abstract.** When there is a need to understand the data stored in a database, one of the main requirements is being able to extract knowledge in the form of rules. Classification strategies allow extracting rules almost naturally. In this paper, the CLUHR classification strategy is extended to work with databases that have nominal attributes. Finally, the results obtained using the databases from the UCI repository are presented and compared with other existing classification models, showing that the algorithm presented requires less computational resources and achieves the same accuracy level and number of extracted rules.*

***Keywords:** Rule extraction, classification, large datasets, supervised learning.*

## 1. Introduction

The extraction of knowledge is a process that combines machine learning, statistics and pattern recognition techniques, among others, used to assist the decision making process, understand the data and explain certain situations or phenomena. Various data mining techniques have been successfully applied to various areas that handle or have large volumes of data, as tools to model the available information and thus obtain knowledge [1] [2] [3].

Among the tasks that can be carried out with data mining techniques, clustering and classification are of great interest. Clustering involves techniques capable of clustering data in different groups by means of a similarity measurement [4] [5] [6] [7] [8]. On the other hand, the classification task includes techniques that know the class of each data element and whose purpose is establishing common patterns that allow explaining or summarizing the data belonging to each class [9] [10] [3].

IF-THEN rules are the most common way of passing knowledge, since they are easy to understand. Additionally, adaptive techniques allow adding new rules and removing or modifying existing ones. This is why rules are used by most existing techniques to produce knowledge [10] [11] [12] [13] [14] [15].

In [16], a powerful extraction strategy that extracts knowledge as IF-THEN rules is presented, called CLUHR. This strategy uses hyper-rectangles as tool to describe the characteristics of the various data classes. The greatest disadvantage of CLUHR is that it can only be used in databases whose attributes are numeric, since hyper-rectangles can only be handled in numeric domains. This disadvantage greatly reduces the scope of the strategy, since there are not many databases with only numeric attributes.

In this paper, an extension of the CLUHR technique, called CLUIN, is presented; CLUIN uses hyper-rectangles and overlap indexes for handling numeric attributes, adding the ability of dealing with nominal or categoric dataset attributes.

The same as CLUHR [16], this technique is deterministic because it no longer depends on absolute randomness, and also, response time in model building is lower than with other classification techniques, achieving similar results regarding the accuracy of the model built and the number of rules extracted.

The rest of the paper is organized as follows: in Section 2, the strategy proposed and the CLUIN algorithm are presented. In Section 3, the results obtained are detailed. Finally, the conclusions drawn from this work are discussed in Section 4.

## 2. CLUIN

The strategy proposed in this paper, called CLUIN, uses the same algorithm as CLUHR to work with numeric attributes by adding a new algorithm that is responsible for handling the nominal attributes in the database.

Each of the nominal attributes  $i$  in the database has its own domain represented by a set  $S_i$ . Thus, the data model of a given class is formed by a hyper-rectangle that describes the numeric attributes of the class and a set  $S$ , called set of nominal values, that is formed by sets  $S_i$ .

**Definition:** Be  $C$  a class of data and  $value(x,i)$  the nominal value of attribute  $i$  for element  $x \in C$ . Set  $S_i$  for  $C$  is formed as follows:  $S_i = \{ v \mid \exists x \in C, value(x,i) = v \}$

**Definition:** Be  $C$  a class of data and  $S_i, i=1..N$ , the corresponding sets of the  $N$  nominal attributes of class  $C$ . The set of nominal values  $S$  is formed by all  $S_i$ :  $S = \{ S_i, i=1..N \}$

**Definition:** Be  $C$  a class with  $N$  nominal attributes and  $M$  numeric attributes, the data in class  $C$  are represented by the set of nominal values  $S$  formed from the  $N$  nominal attributes and hyper-rectangle  $H$  formed from the  $M$  numeric attributes.

In CLUHR, two hyper-rectangles whose classes are different can present an overlap in space, the same as two sets of nominal values from different classes can intersect.

**Definition:** Be  $S$  and  $T$  two sets of nominal values in classes  $C$  and  $D$ , respectively. There is an intersection between both sets if the intersection of all pairs of sets  $S_i$  and  $T_i$  is different from the empty set.

From the definition above, it can be seen that with just one attribute  $i$  for which  $S_i \cap T_i = \emptyset$ , classes  $C$  and  $D$  will not intersect. Attribute  $i$  is precisely the one that allows splitting data from both classes.

**Definition:** Two classes  $C$  and  $D$  overlap if their respective hyper-rectangles overlap and their respective sets of nominal values also overlap.

Therefore, the overlapping classes can be split just by dividing the sets of nominal values or the hyper-rectangles.

The intersections detected in the sets of nominal values must be removed; to do so, similarly to CLUHR, certain indexes are calculated. In CLUHR, divisibility indexes  $Z_i$  are calculated, which are applicable to each numeric attribute. These indexes  $Z_i$  are used to calculate the divisibility index  $\Omega_i$ , which in turn determines the attribute to be used for the adjustment. CLUIN uses the same index  $\Omega_i$  for numeric values and also uses the divisibility indexes  $Y_j$  that are calculated for each nominal attribute and are used for calculating the divisibility index  $\Psi_j$ . The maximum value between  $\Psi_j$  and  $\Omega_i$  determines the attribute to be used for the *splitting* operation.

If the attribute is numeric, hyper-rectangles are divided as described for CLUHR. If the attribute is nominal, the division is done by modifying the sets of nominal values as detailed further on.

## 2.1 Indexes

In CLUHR, overlap indexes are calculated to measure how much two hyper-rectangles overlap in each space dimension. This index is calculated for all space dimensions and for each of the two hyper-rectangles. Thus, if two hyper-rectangles  $H$  and  $J$  overlap, the indexes  $Z_i(H)$  and  $Z_i(J)$  are calculated for  $i=1..M$ .

Similarly, CLUIN calculates the intersection indexes for each nominal attribute and for each of the intersected sets of nominal values  $S$  and  $T$ . Thus, indexes  $Y_j(S)$  and  $Y_j(T)$  are calculated for  $j=1..N$ , and are then used to calculate the divisibility index  $\Psi_j$ .

In this paper, we present the use of two indexes  $Y$  that have been successfully used in various classification algorithms. The first of these indexes is the Information Gain Ratio used in many decision trees [17] [18] [19] and the second one is Kolmogorov-Smirnoff distance, used in more recent works [20] [21].

**Information Gain Ratio:** This index determines the nominal value  $v$  in an attribute  $j$  that has the lowest entropy among all values of  $i$ . This index returns a value between 0 and 1, where 0 indicates the greatest possible entropy and the lowest entropy is indicated by the value tending to 1. As this index is calculated using the values from both sets  $S$  and  $T$ , a single calculation is done, and it is assigned to indexes  $Y_j(S)$  and  $Y_j(T)$ :

$$\begin{aligned}
Y_j(S) &= Y_j(T) = \text{IGR}(S \cup T, j) \\
\text{IGR}(ST, j) &= \text{IG}(ST, j) / \text{IV}(ST, j) \\
\text{IG}(ST, i) &= H(ST) - \\
& \sum_{v \in \text{valores}(i)} \frac{\# x \in ST \text{ valor } x, i = v}{\# ST} \cdot H(x \in ST \text{ valor } x, i = v) \\
\text{IV}(ST, i) &= - \sum_{v \in \text{valores}(i)} \frac{\# x \in ST \text{ valor } x, i = v}{\# ST} \cdot \log_2 \left( \frac{\# x \in ST \text{ valor } x, i = v}{\# ST} \right)
\end{aligned}$$

Where  $H(E)$  is the entropy of set  $E$ .

Once the attribute with the highest value of  $Y$  is selected, the nominal value  $v$  is determined; this value is used to carry out the *split* operation that has the lowest entropy.

**Kolmogorov-Smirnoff distance:** This distance is a statistical test that finds the maximum distance between two probability functions. When using this index, the division of the two sets of nominal values is carried out by means of the nominal value that generates the greatest division of data from both classes. This index tends to 1 when the use of a value generates a better division of the classes, and it tends to 0 in the opposite case.

$$\begin{aligned}
Y_i(S) &= \\
\max_{v \in \text{valores } i} (\# x \in S \text{ valor } x, i = v) - \# x \in S \text{ valor } x, i \neq v) \\
Y_i(T) &= \max_{v \in \text{valores}(i)} (\# x \in T \text{ valor } x, i = \\
& v) - \# x \in T \text{ valor } x, i \neq v)
\end{aligned}$$

Once the attribute  $j$  with the highest value of  $Y$  is selected, the nominal value  $v$  is determined; this value is used to carry out the split operation that generated the value of the index (that with the maximum distance).

These two indexes are used for calculating the divisibility index  $\Psi_i$ .

$$\Psi_i(H, S) = \frac{\sum_{i=1}^n Y_i S}{n}$$

Thus, the maximum between  $\Psi_j$  and  $\Omega_i$  determines the attribute to be used to split the classes. If the attribute is numeric, the split operation is carried out by modifying the respective hyper-rectangles from each class as in CLUHR, while if the attribute is nominal, then the split process is carried out by modifying the sets of nominal values.

If the attribute to be used for the division is nominal, then it is possible that each of the calculated indexes  $Y$  may have yielded a different value for the division. This strategy does not suggest the use of any specific value over the others; any of them can be used. The value chosen to be used for the *split* process will depend on the problem to be solved, and it is determined by the user.

To carry out a division by modifying a nominal attribute with a value  $v$ , one of the two sets  $S_j$  is arbitrarily selected and two new sets are formed for class  $C$ . One of them will be formed by all data such that for attribute  $j$  it is equal to  $v$ , and the other one will be formed by its complement.

$$S_{j1} = \{ x \mid \text{value}(x, j) = v \}$$

$$S_{j2} = \{ x \mid \text{value}(x, j) \neq v \}$$

## 2.2 Regrouping sets of nominal values and hyper-rectangles.

If the split operation is carried out using a numeric attribute, the hyper-rectangles from both classes are modified and new hyper-rectangles are generated. Each new hyper-rectangle has a new set of corresponding data from the class with which the minimum representative hyper-rectangle is adjusted (see [16] for more details). At the same time, with each data subset, the corresponding set of nominal values is calculated.

Similarly, if the division is carried out using a nominal attribute, the new sets of nominal values obtained will each have a subset of data. With these same data, the corresponding minimum representative hyper-rectangles are formed. After the adjustments are made, there is a new search for overlaps and indexes  $\Omega_i$  and  $\Psi_j$  are calculated once again.

The CLUIN algorithm is as follows:

```

Initialize hyper-rectangles and sets of nominal
values for each class of data
Detect overlaps and calculate indexes  $\Omega_i$ 
while there are overlaps
    Divide by hyper-rectangles or by sets of
    nominal values
    Adjust the new hyper-rectangles and sets of
    nominal values
    Calculate indexes  $\Omega_i$ 
end while
Extract resulting rules

```

## 2.3 Rule extraction

When the algorithm finishes, the result obtained are the hyper-rectangles and the sets of nominal values that form the entire data model. Each data class is mapped to one or more pairs of hyper-rectangles and sets of nominal values. Each pair of hyper-rectangle/set will be used to extract a classification rule. This rule will be formed by the limits of the hyper-rectangle itself for numeric attributes and the nominal values contained in the set of nominal values.

The clauses of the numeric attributes will be formulated as: (value\_attribute\_i  $\geq$  Hni) AND (value\_attribute\_i  $\leq$  hx1).

While the clauses of the nominal attributes can have either of two formulations: (value\_attribute\_i = nominal\_value) or (value\_attribute\_i  $\in$  subset\_of\_nominal\_values)

For high-dimension problem spaces, this procedure produces very complex and inadequate rules if the purpose is extracting knowledge from the base that the user can understand and explain knowledge through these rules. The conditions of these complex rules can be refined by simple inspection; the problem, however, is when there is a large number or dimensions of rules. The reader can find methods to automatically simplify rules in the bibliography ([22] [23]).

### 3. Results

CLUIN, the technique proposed in this paper, has been tested on 13 databases of the UCI repository [24]. Both performance and the computational effort required to build the model were compared with the results presented by [25] and [26]. The comparative analysis of the results obtained by CLUHR in [16] and CLUIN is not necessary, because the latter inherits the same classification strategy used by CLUHR for handling databases with numeric attributes, and therefore, the same results are obtained.

The test known as 10-fold cross validation was performed over each database; the test was run 10 separate times over each dataset and the final accuracy was determined as the average of these 10 runs. The comparative results of classification accuracy for CLUIN and the results presented by [25] and [26] are shown in Table 1. Table 2 shows the average number of rules that were created during the process.

The two-sided t-student test with a confidence level of 95% was carried out to determine if the differences between CLUIN and PSO/ACO2 are statistically significant. In tables 1 and 2, the signs “+” or “-” are used to indicate if CLUIN was better or worse, respectively, with a statistically significant difference, and the sign “=” is used to indicate that there was no difference. It should be noted that the work presented in [25] does not include statistical data and it was therefore impossible to compare it.

Based on tables 1 and 2, it can be concluded that, even though the efficacy of the method proposed is similar to those presented for the remaining techniques, the number of hyper-rectangle/set pairs generated is lower. This reduces the number of rules and facilitates understanding the knowledge extracted by the user.

The techniques proposed in [25] and [26], being based on evolutionary and optimization strategies, are able to find an individual from their respective populations that represents an optimal solution to the problem. Even if this is the case, it can be seen that CLUHR achieves similar results to those obtained by these techniques. The main disadvantage of the latter is the computational effort that they require to build the model.

In [25], an evolutionary algorithm is used to find a set of hyper-rectangles that represent a model of the data. From an initial set  $HS$  of hyper-rectangles, each individual of the evolutionary population represents a subset of  $HS$ , which are evolved until the best individual is found, after several generations.

This hyper-rectangle is built using both numeric and nominal attributes – when a dimension  $i$  of the hyper-rectangle corresponds to a nominal attribute, this dimension is built with a subset of values of attribute  $i$ , converting the hyper-rectangle into a structure that is not numerically complete. On the other hand, the fitness of an individual is assessed by searching, for each element  $x$  of the database used, the hyper-rectangle that is closest to  $x$ . This means that, for each assessment of the fitness of each individual, the entire database must be examined. In [25], it is mentioned that for the tests carried out, one run of the evolutionary algorithm consists in 10,000 assessments, which means that the entire database is examined 10,000 times. Optimistically, it can be assumed that this technique will achieve an optimal result in a shorter time. Either way, for a population of 100 individuals, the optimal result will hardly be achieved in less than 20 generations, which means that the lower limit would be 2,000 assessments of individuals to achieve the optimal result, which in turn means that the database has to be examined 2,000 times.

In [26], a hybrid algorithm between PSO and ACO is used to find rule clauses. ACO is used to find the clauses of the nominal attributes, while PSO is used to find the clauses of the numeric attributes. In ACO, each particle has a pheromone matrix for each nominal attribute, and in PSO, each particle vector has two values for each attribute – one for the lower limit and one for the upper limit. To test a particle in the pheromone matrix, the nominal clauses are chosen probabilistically and the vector in PSO is converted into clauses that are then tested.

The first time that the PSO algorithm tries to find a rule, it works with all the data in a class. Both in ACO and PSO, assessing the fitness of a particle implies exploring the entire database and measuring the accuracy of the rule represented by the vector of the particle. The result of this operation is a rule, and all the data that meet this rule are not analyzed when looking for a second rule.

Since there is no way of knowing which subset of data is assessed over and over again as the rules are generated, the lower limit is set by considering that for each class, a single rule is extracted. It is therefore established that for each class, the ACO algorithm is run once and the PSO algorithm is run once with the data from that class.

In [26], it is explained that the runs of ACO and PSO are carried out with a swarm of 100 particles, and that the algorithms are run a maximum of 100 iterations. In ACO, the 100 iterations are always run to obtain the pheromone matrixes that yield the best possible result; in each iteration rule quality has to be measured, which requires going through the entire database. Therefore, one run of ACO goes through the database  $100 \times 100 = 10,000$  times.

As regards PSO, in the best of cases PSO runs a single iteration to obtain the optimal result. To do so, it has to assess the fitness of 100 particles by going through all of the data in a class for each of these assessments. If for each class, a single PSO is run with only one iteration (hypothetical case that is almost impossible), then the database has to be explored in its entirety at least 100 times.

ACO's 10,000 times added to PSO's 100 times are the minimum number of times that the database is explored in its entirety. More realistically, assuming that two rules are extracted for each class in PSO, with the second rule being built only with 50% of the data from each class, and each PSO running 20 iterations, then the database would have to be explored 2,000 to find the first rule and 1,000 times to find the second one (since it would be working with half the data), for a total of 3,000 times that the database is explored in full. This plus the 10,000 times required for ACO adds up to a total of 13,000 times.

With CLUIN, the data is explored in its entirety only the first time to build the hyper-rectangles and the nominal data sets, a second time to determine how many data fall within each intersection, and a third time to calculate the indexes and determine how to remove the intersection selected. In summary, for each intersection that is to be removed, the database is explored three times. The total number of times that the database is explored will be  $3*Q$ , where  $Q$  is the number of intersections that is removed during the execution of the algorithm.

When an intersection is removed, only the hyper-rectangles and the sets of nominal data involved are modified, and therefore, only the data represented by such pairs of hyper-rectangle/set are explored. At each intersection  $q$ , these data represent a fraction  $f_q$  of the database, meaning that the database is explored  $3*Q*f_q$  times.

Table 3 shows, for each database used for the tests, the number of times that the database was explored.

**Table 1.** Accuracy of the method proposed versus the results presented in [25] and [26]. Standard deviation is indicated between brackets; statistical differences are shown in the last column.

Dataset	EHS-CHC	PSO/ACO2	CLUIN	
Contraceptive	0.4983		0.4852 (0.0265)	
Credit	0.8464	0.8560 (0.0284)	0.8497 (0.0541)	=
Zoo	0.9300	0.9718 (0.0625)	0.9621 (0.0357)	=
Balance scale		0.8272 (0.0477)	0.8236 (0.0219)	=
Australian credit		0.8531 (0.0414)	0.8479 (0.0387)	=
German credit		0.6790 (0.0582)	0.6802 (0.0468)	=
Statlog heart		0.8111 (0.0616)	0.8257 (0.0521)	=
Mushroom		0.9990 (0.0110)	0.9742 (0.0205)	-



Promoter	0.8100 (0.1212)	0.8351 (0.0981)	=
Soybean	0.8701 (0.0653)	0.8594 (0.0782)	=
Tic-Tac-Toe	1.000 (0.0000)	0.9863 (0.0029)	-
Chess (Kr vs. Kp)	0.9947 (0.0510)	0.9746 (0.0268)	=
Splice	0.9348 (0.0124)	0.9420 (0.0254)	=

**Table 2.** Number of hyper-rectangle/set pairs created for the data model by the strategies studied. Standard deviation is indicated between brackets; statistical differences are shown in the last column.

Dataset	EHS-CHC	PSO/ACO2	CLUIN	
Contraceptive	12.7		11.8 (3.27)	
Credit	6.1	22.5 (3.1)	7.9 (3.80)	+
Zoo	5.6	7.1 (0.32)	6.3 (1.32)	=
Balance scale		26.6 (1.07)	29.4 (2.92)	-
Australian credit		22.7 (2.0)	25.14 (4.84)	=
German credit		54.3 (1.89)	48.67 (1.51)	+
Statlog heart		9.7 (1.34)	10.8 (1.63)	=
Mushroom		8.7 (0.48)	6.5 (0.87)	+
Promoter		5.1 (0.32)	6.3 (0.49)	-
Soybean		24.2 (1.03)	22.1 (2.07)	+
Tic-Tac-Toe		9.0 (0.0)	8.06 (0.62)	+
Chess (Kr vs. Kp)		18.7 (2.0)	25.8 (3.98)	-
Splice		88.0 (2.91)	95.7 (4.79)	-

**Table 3.** Number of times the entire database is explored with algorithms EHS-CHC, PSO/ACO2 and CLUIN. For CLUIN, the average and the standard deviation for 10 runs are shown. For the other two techniques, the number of times estimated in section 4, Results, is used.

Dataset	EHS-CHC	PSO/ACO2	CLUIN
Contraceptive	2000		25 (5.4)
Credit	2000	13000	47 (4.2)
Zoo	2000	13000	15 (3.2)
Balance scale		13000	32 (1.8)
Australian credit		13000	54 (4.8)
German credit		13000	26 (5.4)
Statlog heart		13000	41 (3.9)

Mushroom	13000	35 (4.8)
Promoter	13000	28 (2.8)
Soybean	13000	52 (8.2)
Tic-Tac-Toe	13000	14 (3.7)
Chess (Kr vs. Kp)	13000	15 (5.3)
Splice	13000	61 (7.1)

## 5. Conclusions

A new knowledge extraction strategy has been presented, called CLUIN, that extends the strategy proposed by CLUHR [16] to work with nominal attributes. Each hyper-rectangle generated with the numeric attributes corresponds to a set of nominal values. This hyper-rectangle/set pairs are reduced to other, smaller ones as indicated by the result of calculating a battery of indexes to remove or minimize existing overlaps.

The use of indexes as criterion to select two overlapping hyper-rectangles or sets that intersect in order to be modified and thus remove such overlap turns the proposed strategy into a robust and efficient tool, in the sense that certain indexes can be calculated or not depending on the interests of the end user, including the possibility of adding new indexes resulting from new research activities or future experiences with problems that are solved using this method.

CLUIN improves the power of CLUHR by adding the possibility of handling nominal attributes. CLUIN follows the same line as CLUHR; namely, it uses the same definition of overlapping sets of nominal values and how to detect these overlaps, and calculates indexes to determine the degree of intersection between sets of different classes used together with the intersection indexes used in CLUHR to calculate the overlap index. The latter indicates the attribute that should be used to minimize the overlap between the data from both classes.

The results obtained were compared with an evolutionary technique and an optimization technique, and it was observed that a better accuracy and a slightly smaller number of extracted rules were achieved. The greatest advantage of CLUHR, when compared with the strategies mentioned, is that it requires a significantly lower computational effort to achieve similar results. This is very important when working with large databases.

## References

1. Bouguessa, M., Wang, S. Mining Projected Clusters in High-Dimensional Spaces. *IEEE Transactions on Knowledge and Data Engineering*, 21 (4), 507-522. (2009).

2. Hsu, C.-M., Chen, M.-S. On the Design and Applicability of Distance Functions in High-Dimensional Data Space. *IEEE Transactions on Knowledge and Data Engineering*, 21 (4), 523-536 (2009).
3. Koul, N., Caragea, C., Honavar, V., Bahirwani, V., Caragea, D. Learning Classifiers from Large Databases Using Statistical Queries. *Web Intelligence and Intelligent Agent Technology IEEE/WIC/ACM International Conference on*, 923-926 (2008).
4. Bandyopadhyay, S., Saha, S. A Point Symmetry-Based Clustering Technique for Automatic Evolution of Clusters. *IEEE Transactions on Knowledge and Data Engineering*, 20 (11), 1441-1457 (2009).
5. Chen, H.-L., Chuang, K.-T., Chen, M.-S. On Data Labeling for Clustering Categorical Data. *IEEE Transactions on Knowledge and Data Engineering*, 20 (11), 1458-1472 (2009).
6. Lu, J.-L., Wang, L.-Z., Lu, J.-J., Sun, Q.-Y. Research and application on KNN method based on cluster before classification. *Machine Learning and Cybernetics International Conference on*, 307-313 (2008).
7. Wang, Z., Qi, Q., Xu, L. Cluster Analysis Based on Spatial Feature Selecting in Spatial Data Mining. *Computer Science and Software Engineering International Conference on*, 386-389 (2008).
8. Aslanidis, T., Souliou, D., Polykrati, K. CUZ, An Improved Clustering Algorithm. *Computer and Information Technology Workshops IEEE 8th International Conference on*, 43-48 (2008).
9. Bakar, A.A., Othman, Z.A., Hamdan, A.R., Yusof, R., Ismail, R. An Agent Based Rough Classifier for Data Mining. *Intelligent Systems Design and Applications Eighth International Conference on*, 145-151 (2008).
10. Kamwa, I., Samantaray, S. R., Joos, G. Development of Rule-Based Classifiers for Rapid Stability Assessment of Wide-Area Post-Disturbance Records. *IEEE Transactions on Power Systems*, 24 (1), 258-270 (2009).
11. Martens, D., Baesens, B., Van Gestel, T. Decompositional Rule Extraction from Support Vector Machines by Active Learning. *IEEE Transactions on Knowledge and Data Engineering*, 21 (2). 178-191 (2009).
12. Shi, X.-J., Lei, H. A Genetic Algorithm-Based Approach for Classification Rule Discovery. *Information Management, Innovation Management and Industrial Engineering International Conference on*, 175-178 (2008).
13. Hasperu e, W., Osella Massa, G., Lanzarini, L. Obtaining a Fuzzy Classification Rule System from a non-supervised Clustering. *Information Technology Interfaces (ITI) 30th International Conference of*, 341-346 (2008).
14. Konig, R., Johansson, U, Niklasson, L. Genetic programming - a tool for flexible rule extraction. *Evolutionary Computation IEEE Congress on*, 1304-1310 (2007).

15. Hasperué, W., Lanzarini, L.C. A new clustering strategy for continuous datasets using hypercubes. 36th Conferencia Latinoamericana de Informática (2010).
16. Hasperué, W., Lanzarini, L., De Guisti, A. Rule Extraction on Numeric Datasets Using Hyper-rectangles. Computer and Information Science. Vol. 5, No 4, pp. 116-131 (2012).
17. Quinlan J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc. ISBN 1-55860-238-0 (1993).
18. Quinlan J. R. Induction of Decision Trees Machine Learning. 1: Vol. 1. - págs. 81-106 (1986).
19. Piao M., Li M. y Ryu K. Ho Using Significant Classification Rules to Analyze Korean Customers' Power Consumption Behavior: Incremental Tree Induction using Cascading-and-Sharing Method. págs. 1649-1653 (2010).
20. Mballo, C., Diday E. Kolmogorov-Smirnov for Decision Trees on Interval and Histogram Variables (2004).
21. Chao S., Wong Fai. An incremental decision tree learning methodology regarding attributes in medical data mining. International Conference on Machine Learning and Cybernetics. págs. 1694-1699 (2009)
22. Darrah, M., Taylor, B., Skias, S. Rule Extraction from Dynamic Cell Structure Neural Networks Used in a Safety Critical Application. Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, 629-634 (2004).
23. Ma, J., Guo, D., Liu, M., Ma, Y., Chen, S. Rules Extraction from ANN Based on Clustering. Computational Intelligence and Natural Computing International Conference on, 19-21 (2009).
24. A. Asuncion and D. Newman, UCI machine learning repository, Available: <http://archive.ics.uci.edu/ml/> (2007).
25. Garcia, S., Derrac, J., Luengo, J., Herrera, F. A First Approach to Nearest Hyperrectangle Selection by Evolutionary Algorithms. Intelligent Systems Design and Applications Ninth International Conference on, 517-522 (2009).
26. Holden, N., Freitas, A.A. A hybrid PSO/ACO algorithm for discovering classification rules in data mining. Journal of Artificial Evolution and Applications, 1-11 (2008).

# An approach for an algebra applied to a Defeasible Logic Programming

MAXIMILIANO C. D. BUDÁN<sup>1,2,3,4</sup>, MAURO J. GÓMEZ LUCERO<sup>1,2,3</sup>,  
GUILLERMO R. SIMARI<sup>2,3</sup>

<sup>1</sup> Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)

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

<sup>3</sup> Universidad Nacional del Sur

<sup>4</sup> Universidad Nacional de Santiago del Estero

{mcd, mjg, grs}@cs.uns.edu.ar

***Abstract.** In the last decade, several argument-based formalisms have emerged, with application in many areas, such as legal reasoning, autonomous agents and multi-agent systems.*

*In this work we present an approach applied to any concrete argumentation systems, called Argumentative Labels Algebra (ALA), in which propagate certain information through the internal structure of the argument and the existing relations between these. This information can be used for different proposed: (1) determine which argument defeat another, analyzing a feature that is relevant to the domain (eg. time availability, degree of reliability, particular characteristics, among others) and (2) define a acceptability threshold which will determine if the arguments are strong enough to be accepted.*

*For this way, we obtain an approach that strengthens the argumentations systems, increase the ability of real-world representation modeling different attributes associated to the arguments.*

***Keywords:** Concrete Argumentation Systems, Argumentative Labels Algebra, Defeasible Logic Programming.*

## 1. Introduction

Argumentation has contributed with a human-like mechanism to the formalization of commonsense reasoning. Argumentation is the process of defending a given affirmation by giving reasons for its acceptance. Both the original claim and its support are subject to consideration, since reasons supporting conflicting claims can be proposed. Several argument-based formalisms have emerged, with application in many areas such as legal reasoning, autonomous agents and multi-agent systems [3-5].

Besides abstract argumentation approaches, different more concrete argumentation systems exists, specifying a knowledge representation language, and how arguments are built. One of those systems is Defeasible Logic Programming (DeLP) [1], a formalism that combines results of Logic

Programming and Defeasible Argumentation. DeLP allows representing information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, and provides a defeasible argumentation inference mechanism for determining warranted conclusions. The defeasible argumentation basis of DeLP allows to build applications that deal with incomplete and contradictory information in dynamic domains.

In real application domains of argumentation may be require the explicit treatment of special characteristics (eg. time availability, degree of reliability, particular characteristics, among others), this information is not in general directly associated with arguments, but instead it is attached to the basic pieces of knowledge (in general logical rules) from which arguments are built.

In this paper we introduce a novel approach, called *Argumentation Labels Algebra (ALA)*, which can be applied to any concrete argumentation systems for modeling the capability of propagate information through the arguments. This information can be used to determine which argument defeat another, analyzing a feature that is relevant to the domain (eg. time availability, degree of reliability, particular characteristics, among others)<sup>1</sup> and define an acceptability threshold which will determine if the arguments are strong enough to be accepted.

The central contribution of this paper is increase the ability of real-world representation modeling different attributes associated to the arguments, using an algebra of labels for propagate this information.

## 2. Defeasible Logic Programming

Different argumentation systems exist, specifying a knowledge representation language, and how arguments are built. One of those systems is Defeasible Logic Programming (DeLP) [1], a formalism that combines results of Logic Programming and Defeasible Argumentation. DeLP allows representing information in the form of weak rules in a declarative way, from which arguments supporting conclusions are constructed, and provides a defeasible argumentation inference mechanism for determining warranted conclusions. The defeasible argumentation basis of DeLP allows to build applications that deal with incomplete and contradictory information in dynamic domains.

Below we present the definitions of program and argument in DeLP.

**Definition 1 (DeLP Program)** A DeLP program  $\Pi$  is a pair  $(\Pi, \Delta)$  where (1)  $\Delta$  is a set of defeasible rules of the form  $L \text{--} P_1, \dots, P_n$ , with  $n > 0$ , where  $L$  and each  $P_i$  are literals, and (2)  $\Pi$  is a set of strict rules of the form  $L \text{--} P_1, \dots, P_n$ , with  $n \geq 0$ , where  $L$  and each  $P_i$  are literals.  $L$  is a ground atom  $A$  or a negated ground atom  $\sim A$ , where ' $\sim$ ' represents the strong negation.

---

<sup>1</sup> The usefulness of some of these parameters were published in previous works [6, 7]

Pragmatically, strict rules can be used to represent strict (non-defeasible) information, whereas defeasible rules are used to represent tentative or weak information. In particular, a strict rule  $L \leftarrow P_1, \dots, P_n$  with  $n = 0$  is called fact and will be denoted just as  $L$ , and a defeasible rule  $L \leftarrow P_1, \dots, P_n$  with  $n = 0$  is called presumption and will be denoted just as  $L^-$ . It is important to remark that the set  $\Pi$  must be consistent as it represents strict (undisputed) information. In contrast, the set  $\Delta$  will generally be inconsistent, since it represents tentative information.

**Definition 2 (Argument)** Let  $L$  be a literal and  $\Pi = (\Pi, \Delta)$  be a DeLP program. An argument for  $L$  is a pair  $\langle A, L \rangle$ , where  $A$  is a set of defeasible rules of  $\Delta$ , such that:

1. there is a defeasible derivation for  $L$  from  $\Pi \cup A$ .
2.  $\Pi \cup A$  is not contradictory, and
3.  $A$  is a minimal, i.e., there exist no proper subset  $A'$ ,  $A' \subsetneq A$  satisfying conditions (1) and (2).

We say that an argument  $\langle B, Q \rangle$  is a sub-argument of  $\langle A, L \rangle$  iff,  $B \subseteq A$ .

DeLP provides an argumentation based mechanism to determine warranted conclusions. This procedure involves constructing arguments from programs, identifying conflicts or attacks among arguments, evaluating pairs of arguments in conflict to determine if the attack is successful, becoming a defeat, and finally analyzing defeat interaction among all relevant arguments to determine warrant.

Below we briefly present the formalization of the previously mentioned notions, as introduced in [1].

**Definition 3 (Disagreement)** Let  $\Pi = (\Pi, \Delta)$  be a DeLP program. Two literals  $L$  and  $L'$  are in disagreement iff the set  $\Pi \cup \{L, L'\}$  is contradictory.

**Definition 4 (Attack)** Let  $\Pi = (\Pi, \Delta)$  be a DeLP program. Let  $\langle A, L_1 \rangle$  and  $\langle A, L_2 \rangle$  be two arguments in  $\Pi$ . We say that  $\langle A, L_1 \rangle$  counter-argues, rebuts, or attacks  $\langle A, L_2 \rangle$  at the literal  $L$  iff there is a sub-argument  $\langle A', L' \rangle$  of  $\langle A, L_2 \rangle$  such that  $L$  and  $L'$  are in disagreement. The argument  $\langle A', L' \rangle$  is called disagreement sub-argument, and the literal  $L$  will be the counter-argument point.

In order to decide if a partial attack really succeeds, constituting a defeat, a comparison criterion must be used, establishing the relative strength of the arguments involved in the attack. In this work we will use the criterion adopted by default in DeLP, called specificity, which favors arguments based on more information or supporting their conclusions more directly.

**Definition 5 (Defeat)** Let  $\Pi = (\Pi, \Delta)$  be a DeLP program. Let  $\check{u}A, L_1\check{z}$  and  $\check{u}A, L_2\check{z}$  be two arguments in  $\Pi$ . We say that  $\check{u}A, L_2\check{z}$  defeats  $\check{u}A, L_1\check{z}$  iff exist a sub-argument  $\check{u}A, L\check{z}$  of  $\check{u}A, L_1\check{z}$  such that  $\check{u}A, L_2\check{z}$  counter-argues  $\check{u}A, L\check{z}$  at literal  $L$  and it holds that:

1.  $\check{u}A, L_2\check{z}$  is strictly more specific than  $\check{u}A, L\check{z}$  (proper defeater), or
2.  $\check{u}A, L_2\check{z}$  is unrelated to  $\check{u}A, L\check{z}$  (blocking defeater)

In DeLP a literal  $L$  will be warranted if there exists a non-defeated argument structure  $\check{u}A, L\check{z}$ . In order to establish whether  $\check{u}A, L\check{z}$  is non-defeated, the set of defeaters for  $A$  will be considered. Thus, a complete dialectical analysis is required to determine which arguments are ultimately accepted. Such analysis results in a tree structure called dialectical tree, in which arguments are nodes labeled as undefeated (U-nodes) or defeated (D-nodes) according to a marking procedure.

**Definition 6 (Dialectical tree [1])** The dialectical tree for an argument  $\check{u}A, L\check{z}$ , denoted  $\check{u}A, L\check{z}$ , is recursively defined as follows: (1) A single node labeled with an argument  $\check{u}A, L\check{z}$  with no defeaters (proper or blocking) is by itself the dialectical tree for  $\check{u}A, L\check{z}$ ; (2) Let  $\check{u}A_1, L_1\check{z}, \check{u}A_2, L_2\check{z}, \dots, \check{u}A_n, L_n\check{z}$  be all the defeaters (proper or blocking) for  $\check{u}A, L\check{z}$ . The dialectical tree for  $\check{u}A, L\check{z}$ ,  $T_{\check{u}A, L\check{z}}$  is obtained by labeling the root node with  $\check{u}A, L\check{z}$ , and making this node the parent of the root nodes for the dialectical trees of  $\check{u}A_1, L_1\check{z}, \check{u}A_2, L_2\check{z}, \dots, \check{u}A_n, L_n\check{z}$ .

For the marking procedure we start labeling the leaves as U-nodes. Then, for any inner node  $\check{u}A_2, Q_2\check{z}$ , it will be marked as U-node iff every child of  $\check{u}A_2, Q_2\check{z}$  is marked as a D-node. If  $\check{u}A_2, Q_2\check{z}$  has at least one child marked as U-node then it is marked as a D-node.

This marking allows us to characterize the set of literals sanctioned by a given DeLP, called warranted literals. A literal  $L$  is warranted iff exist an argument structure  $\check{u}A, L\check{z}$  for  $L$ , such that the root of its marked dialectical tree  $T_{\check{u}A, L\check{z}}^*$  is a U-node.

### 3. An Argumentative Labels Algebra

In any argumentation systems, can be of interest propagate certain information through the existing relations between the arguments.

In other words, can be useful the analysis of a characteristic associated with a certain argument dependent on the environment or a characteristic to reflect changes in the environment. This information can be applied for different proposed such as: (1) determine which argument defeat another, analyzing a feature that is relevant to the domain (eg. time availability, degree of reliability, particular characteristics, among others) and (2) define a



acceptability threshold which will determine if the arguments are strong enough to be accepted.

For the first postulate, we will use the labels associated to the arguments for comparison among them. For example, assume a scenario in which we must decide whether to invest in the company  $J$  or  $K$ , and we have the arguments proposed by the consultants  $A$  and  $B$ :

- $(Arg_A)$  Invest in the company  $J$  because is stable and safe.
- $(Arg_B)$  Invest in the company  $K$  because the shares are rising.

Our approach offers the possibility to handle extra information associated with the arguments, i.e., its possible associate a degree of reliability at the arguments  $(Arg_A)$  and  $(Arg_B)$ , that represent the reliability degree of the consultants  $A$  and  $B$ . The argument  $(Arg_A)$  has a degree of reliability  $[0.8]$  and  $(Arg_B)$  has a degree of reliability  $[0.5]$ , then the  $(Arg_A)$  defeat  $(Arg_B)$  such that  $[0.8] > [0.5]$  (or the argument  $Arg_A$  is more reliable than  $Arg_B$ ).

For the second postulate, suppose an environment in which decisions must be critical or high-risk, in this way we must accept the arguments that remain under a reliability level above the threshold. That is, no argument can be accepted with a reliability level below the threshold. For example, if we have a recommendation system to decide on which company would be wise to invest (buy shares), is of interest determine the result using only those arguments that have a level of belief above the threshold. That is, only considered the information given by sources with some degree of reliability.

In this section we present a different approach to argumentation, that uses an algebra of labels for the propagation of meta-data through an argumentation system. Next we formalize the definition of *Argumentative Label Algebra (ALA)*, which provides the elements required for this proposed.

**Definition 7** (*Argumentative Labels Algebra*) An *Argumentative Labels Algebra* (or simply *ALA*) is a 4-tuplan  $\langle Arg, \exists, \bar{o}, N \rangle$  where:

– Let  $Arg$  be a set of arguments. An argument is a 3-tupla  $\langle \bar{u}A, L, E \rangle$ , where  $A$  the set of knows pieces that support the conclusion,  $L$  is the conclusion of the argument, and  $E$  is a labels that represent a particular characteristic.

– Let  $\exists$  be an operator of weakness affecting the labels associated with the arguments.

Let  $\langle \bar{u}A, L, E_A \rangle$  and  $\langle \bar{u}B, Q, E_B \rangle$  be two arguments. Let  $\langle \bar{u}B, Q, E_B \rangle$  disagreement  $\langle \bar{u}A, L, E_A \rangle$  where  $\langle \bar{u}A_i, L_i, E_{A_i} \rangle$  is a sub-argument of  $\langle \bar{u}A, L, E_A \rangle$ . We defined the operator  $\exists$  as:

$$\langle \bar{u}A_i, L_i, E_{A_i} \rangle \exists \langle \bar{u}B, Q, E_B \rangle = \langle \bar{u}A_i, L_i, E_{A_i} - E_B \rangle$$

– Let  $\bar{o}$  be an operator of fortress affecting the labels associated with the internal structure (rules or knows pieces) of each arguments. Let  $\langle \bar{u}A, L, E_A \rangle$  be an argument.

- If  $A = \bar{O}$  then  $E_A = \bar{y}_L$  where  $\bar{y}_L$  is the label associate with the literal  $L$ . (in some argumentative system represent a fact or presumption).

- If  $A \neq \bar{O}$ ,  $L$  is obtained through a derivation  $R$  and the sub-arguments  $\check{u}A_1, L_1, E_{A_1}\check{z}, \check{u}A_2, L_2, E_{A_2}\check{z}, \dots, \check{u}A_n, L_n, E_{A_n}\check{z}$  then  $E_A = \check{u}A_1, L_1, E_{A_1}\check{z}\check{O}\check{u}A_2, L_2, E_{A_2}\check{z}\check{O} \dots \check{O} \quad \check{u}A_n, L_n, E_{A_n}\check{z}\check{O} \check{y}_R = E_{A_1} + E_{A_2} + \dots + E_{A_n} + \check{y}_R$ , where  $\check{y}_R$  is the label associate with the derivation  $R$ .

– Let  $N$  be the neutral element for the operators  $\exists$  and  $\check{O}$ .

Below we will apply our algebra ALA to DeLP. Then, we analyze the effect of the postulates mentioned at the beginning of the section.

## 4. An Argumentative Labels Algebra Applied in DeLP: Examples and Analysis

In the previous sections, we present a formalism that provides the ability to manipulate extra information through models of argument and the useful points of view for handling this information associated with the arguments. In this section, we apply this theory in DeLP. The combination of the ALA and DeLP incorporates the ability to represent a particular characteristic associated with rules composing arguments. This information is then propagated to the level of arguments, and will be used to define the represented characteristics of an argument. The association of this information to DeLP clauses is formalized through the definition of  $\check{y}$ -program.

**Definition 8** ( $\check{y}$ -program) Let  $\Pi$  be a  $\check{y}$ -program. We say that  $\Pi$  is a set of clauses of the form  $(\gamma, \check{y})$ , called  $\check{y}$ -clause, where: (1)  $\gamma$  is a DeLP clause, (2)  $\check{y}$  is a labels that represent the information associated with the clause  $\gamma$ .

We will say that  $(\gamma, \check{y})$  is a strict (defeasible)  $\check{y}$ -clause iff  $\gamma$  is a strict (defeasible) DeLP clause. Then, given a  $\check{y}$ -program  $\Pi$  we will distinguish the subset  $\Pi$  of strict  $\check{y}$ -clauses, and the subset  $\Delta$  of defeasible  $\check{y}$ -clauses.

In the previous section, we will present a notion of arguments as a 3-tupla  $\check{u}A, L, E\check{z}$ , where  $A$  the set of knows pieces that support the conclusion,  $L$  is the conclusion of the argument, and  $E$  is a labels that represent a particular characteristics. Now, using the formalism DeLP, we can specified the set of knows pieces by a set of  $\check{y}$ -clauses.

Given a set  $S$  of  $\check{y}$ -clauses, we will use  $Clauses(S)$  to denote the set of all DeLP clauses involved in  $\check{y}$ -clauses of  $S$ .  $Clauses(S) = \{\gamma \mid (\gamma, \check{y}) \in A\}$ .

**Definition 9** ( $\check{y}$ -argument) Let  $L$  be a literal, and  $\Pi$  be a  $\check{y}$ -program. We say that  $\check{u}A, L, E_A\check{z}$  is a  $\check{y}$ -argument for a goal  $L$  from  $\Pi$ , if  $A \check{r} \Delta$ , where:

- (1)  $Clauses(\Pi \cup A) \check{r} L$ ;
- (2)  $Clauses(\Pi \cup A)$  is non contradictory;

(3)  $\text{Clauses}(A)$  is such that there is no  $A_1 \checkmark A$  such that  $A_1$  satisfies conditions (1) and (2); and

(4)  $E_A = \checkmark A_1, L_1, E_{A_1} \checkmark \checkmark A_2, L_2, E_{A_2} \checkmark \checkmark \dots \checkmark \checkmark A_n, L_n, E_{A_n} \checkmark \checkmark \checkmark R$ , where  $\checkmark A_i, L_i, E_{A_i} \checkmark$  is a sub-argument of  $\checkmark A, L, E_A \checkmark$  with  $1 \leq i \leq n$ , and  $\checkmark R$  is the label associate with the rule with head  $L$  and body  $L_1, L_2, \dots, L_n$ .

We say that  $\checkmark B, Q, E_B \checkmark$  is a sub-argument of  $\checkmark A, L, E_A \checkmark$  iff  $B \checkmark A$ .

The classical DeLP provides an argumentation based mechanism to determine warranted conclusions. Now we defined for this new DeLP version the argumentation mechanism to determine warranted conclusions. On the one hand, we preserve the definitions of disagreement (Definition 3) and attack (Definition 4). On the other hand, we defined the news concepts of weaken and defeat.

ALA in DeLP can be applying in order to decide if a partial attack really succeeds a defeat. The labels associated with the arguments defined the relative strength of the arguments involved in the attack. Now we present a new concept for DeLP, called weaken, this new concept add to DeLP the treatment of weakening between arguments using the operator of ALA.

**Definition 10 (Weaken)** Let  $\Pi$  be a  $\checkmark$ -program. Let  $\checkmark A, L, E_A \checkmark$  and  $\checkmark B, Q, E_B \checkmark$  be two arguments in  $\Pi$ . We say that  $\checkmark B, Q, E_B \checkmark$  weaken  $\checkmark A, L, E_A \checkmark$ , iff exist a sub-argument  $\checkmark A_1, L_1, E_{A_1} \checkmark$  of  $\checkmark A, L, E_A \checkmark$  such that  $\checkmark B, Q, E_B \checkmark$  counter-argues  $\checkmark A, L, E_A \checkmark$  at literal  $L_1$  and  $E_{A_1} > E_B$ . The force of the argument  $\checkmark A_1, L_1, E_{A_1} \checkmark$  decreases according to the force of the argument  $\checkmark B, Q, E_B \checkmark$ , formally:

$$\checkmark A_1, L_1, E_{A_1} \checkmark \exists \checkmark B, Q, E_B \checkmark = \checkmark A_1, L_1, E_{A_1} - E_B \checkmark, \text{ where } E_{A_1} > E_B.$$

In this work, we will use two criterion to determine which argument is defeated: (1) determine which argument defeat another, analyzing a feature that is relevant to the domain and (2) define a acceptability threshold which will determine if the arguments are strong enough to be accepted. The formal definition of this postulates are presented below:

**Definition 11 (Defeat and Argument Force)** Let  $\Pi$  be a  $\checkmark$ -program. Let  $N$  be the neutral element for the operator  $\exists$ . Let  $\checkmark A, L, E_A \checkmark$  and  $\checkmark B, Q, E_B \checkmark$  be two arguments in  $\Pi$ . We say that  $\checkmark B, Q, E_B \checkmark$  defeat  $\checkmark A, L, E_A \checkmark$ , iff exist a sub-argument  $\checkmark A_1, L_1, E_{A_1} \checkmark$  of  $\checkmark A, L, E_A \checkmark$  such that  $\checkmark B, Q, E_B \checkmark$  counter-argues  $\checkmark A, L, E_A \checkmark$  at literal  $L_1$  and  $E_{A_1} \leq E_B$ . The argument  $\checkmark A_1, L_1, E_{A_1} \checkmark$  is weakened and defeated for the argument  $\checkmark B, Q, E_B \checkmark$ , because the force of the argument  $\checkmark A_1, L_1, E_{A_1} \checkmark$  is weakened to a force equal to  $N$ , formally:

$$\checkmark A_1, L_1, E_{A_1} \checkmark \exists \checkmark B, Q, E_B \checkmark = \checkmark A_1, L_1, N \checkmark, \text{ where } E_{A_1} \leq E_B.$$

An intuitive notion for the definition 11 would be that an argument is defeated if one of its supports (or piece of knows that composed the argument) are weakened to a force equal to the neutral element  $N$ . In other

words, if one of the supports of the argument is defeated then the argument is defeated. Now if combine this concept with the notion of threshold, we obtain the following definition of defeat.

**Definition 12 (Defeat and Threshold)** Let  $\Pi$  be a  $\ddot{y}$ -program. Let  $N$  be the neutral element for the operator  $\exists$ . Let  $\check{u}A, L, E_A\check{z}$  and  $\check{u}B, Q, E_B\check{z}$  be two arguments in  $\Pi$ . We say that  $\check{u}B, Q, E_B\check{z}$  defeat  $\check{u}A, L, E_A\check{z}$ , iff exist a sub-argument  $\check{u}A_1, L_1, E_{A_1}\check{z}$  of  $\check{u}A, L, E_A\check{z}$  such that  $\check{u}B, Q, E_B\check{z}$  counter-argues  $\check{u}A, L, E_A\check{z}$  at literal  $L_1$  and  $E_{A_1}, E_B \geq T$ . We say that  $\check{u}A, L, E_A\check{z}$  is defeated iff:  $E_{A_1} - E_B = N$  or  $E_{A_1} - E_B < T$

The difference between definitions 11 and 12 is that, in the definition 12 we consider only those arguments which have a force greater than the threshold  $T$ . That is, all arguments that have a force below  $T$  are taken as defeated.

In this version of DeLP, a literal  $L$  will be warranted if there exists a non-defeated argument structure  $\check{u}A, L, E_A\check{z}$ . In order to establish whether  $\check{u}A, L, E_A\check{z}$  is non-defeated, the set of defeaters for  $A$  will be considered. Thus, a complete dialectical analysis is required to determine which arguments are ultimately accepted. Such analysis results in a tree structure in which arguments are nodes labeled as weakened ( $\bar{W}$ -nodes), undefeated ( $\bar{U}$ -nodes) or defeated ( $\bar{D}$ -nodes) according to two types of marking procedure.

On the one hand, we have the marking procedure which takes into account the force of the arguments according to their characteristics. We start labeling the leaves as  $\bar{U}$ -nodes. Then, for any inner node  $\check{u}A_2, L_2, E_{A_2}\check{z}$ , it will be marked as  $\bar{U}$ -node iff every child of  $\check{u}A_2, L_2, E_{A_2}\check{z}$  is marked as a  $\bar{D}$ -node. If  $\check{u}A_2, L_2, E_{A_2}\check{z}$  has at least one child marked as  $\bar{U}$ -node or  $\bar{W}$ -nodes then it is marked as a  $\bar{D}$ -node or  $\bar{W}$ -nodes, depending on the strength of the arguments attackers. So an argument is defeated only when its strength is equal to the neutral element  $N$  for the operator (Figure1).

On the other hand, we have the marking procedure which takes into account the force of the arguments according to their characteristics and a threshold. This marking procedure is the same as presented in the preceding paragraph, except that any argument has less force than the threshold or is weakened to a smaller force than the threshold is marked as defeated  $\bar{U}$ -node (Figure1).

**Example 1.** We will take based on the following program  $\Pi$  ( $\ddot{y}$ -program).

$$\left\{ \begin{array}{l} (a \prec s, c, 0.9) (\sim s \prec t, r, 0.5) (j, 0.3)(k \prec m, l, 0.5) (\sim b \prec w, g, 0.9) (t, 0.5) \\ (c \prec j, k, 0.8) (r \prec z, p, 0.4) (l, 0.7)(\sim k \prec b, n, 0.4) (b \prec z, 0.5) (z, 1.2) \\ (s \prec j, l, 0.4) (n \prec p, 0.9) (m, 0.4)(w, 0.1) (p, 0.1) (g, 0.3) \end{array} \right\}$$

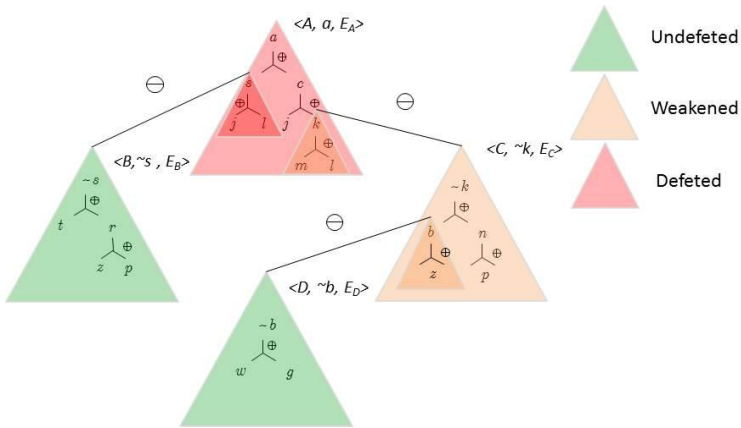


Fig. 1. Marking procedure and Arguments force.

In this program  $\Pi$  can be obtained the following arguments:

Now we calculate the label associated with the argument  $B$  through the operator  $\check{\circ}$ , for the same procedure can be obtained the labels for the others arguments.

$$E_B = \check{u}B_1, t, E_{B1} \check{\circ} \check{u}B_2, r, E_{B2} \check{\circ} \check{y}_{js-t, r} = E_{B1} + E_{B2} + \check{y}_{js-t, r} = 0.5 + 1.7 + 0.5 = 2.7$$

$E_{B1} = \check{y}_t = 0.5$  because the literal  $t$  is a fact (see in definition 7).

$$E_{B2} = \check{u}B_{21}, z, E_{B21} \check{\circ} \check{u}B_{22}, p, E_{B22} \check{\circ} \check{y}_{r-z, p} = E_{B21} + E_{B22} + \check{y}_{r-z, p} = 1.2 + 0.1 + 0.4 = 1.7$$

$$E_A = \check{u}A_1, s, E_{A1} \check{\circ} \check{u}A_2, c, E_{A2} \check{\circ} \check{y}_{a-s, c} = 5.0$$

$$E_C = \check{u}C_1, b, E_{C1} \check{\circ} \check{u}C_2, n, E_{C2} \check{\circ} \check{y}_{k-b, n} = 3.1$$

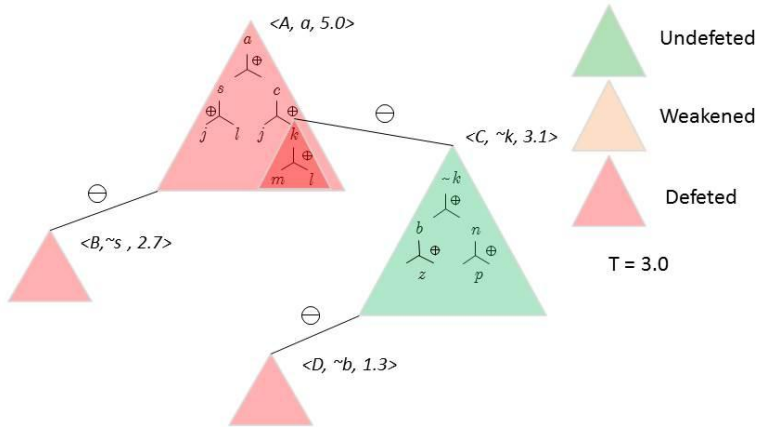
$$E_D = \check{u}D_1, w, E_{D1} \check{\circ} \check{u}D_2, g, E_{D2} \check{\circ} \check{y}_{b-w, g} = 1.3$$

Once obtained the arguments and the respective labels associated with them, we analyze the relationship between the arguments. In this example, there exist contradictions between the arguments so it continues applying the concept of weakness by definition 10 and defeat by definition 11, then we determine which arguments are warranted.

$\check{u}C_1, b, E_{C1} \check{\circ} \check{u}D_1, w, E_{D1} \check{\circ} \check{u}C_1, b, E_{C1} - E_{D1} \check{\circ} \check{u}C_1, b, 1.7 - 1.3 \check{\circ} \check{u}C_1, b, 0.4 \check{\circ}$ , where  $E_{C1} > E_{D1}$  (Weaken). Now, for this weakenes  $C_1$ , we will recalculate the strength of the argument  $C$ ,  $E_C = \check{u}C_1, b, E_{C1} \check{\circ} \check{u}C_2, n, E_{C2} \check{\circ} \check{y}_{k-b, n} = 1.8$

$\check{u}A_1, s, E_{A1} \check{\circ} \check{u}B_1, t, E_{B1} \check{\circ} \check{u}A_1, s, E_{A1} - E_{B1} \check{\circ} \check{u}A_1, s, 1.4 - 2.7 \check{\circ} \check{u}A_1, s, 0 \check{\circ}$  (Defeted). For that, the argument  $A$  is Defeted.

Now we can define a threshold  $T = 3.0$  necessary for the definition of defeat 12.



As in the conventional DeLP, a literal  $L$  is warranted iff exist an argument structure  $\langle A, L, E_A \rangle$  for  $L$ , such that its marked as  $\cup$ -node. Then, we can reach the conclusion that the argument  $\langle A, L, E_A \rangle$  is not accepted.

## 5. Conclusion. Related and Future

Argumentation has contributed with a human-like mechanism to the formalization of commonsense reasoning. In the last decade, several argument-based formalisms have emerged, with application in many areas, such as legal reasoning, autonomous agents and multi-agent systems.

In this work, increase the ability of real-world representation modeling different attributes associated to the arguments, using an algebra of labels (ALA) for propagate this information. We combined ALA and DeLP, introducing a rule-based argumentation framework considering different attributes represented by labels at the object language level. This information was used to two proposed: determine which argument defeat another, analyzing a feature that is relevant to the domain and define a acceptability threshold which will determine if the arguments are strong enough to be accepted which is a necessary in environments that require some degree of strength in their answers.

As future work we will develop an implementation of the application of ALA in the existing DeLP system<sup>2</sup> as a basis. The resulting implementation will be exercised in different domains requiring to model extra information associated to the arguments.

<sup>2</sup> See <http://lidia.cs.uns.edu.ar/delp>

## References

1. J. García and G. R. Simari, Defeasible Logic Programming: An Argumentative Approach (2004), Theory Practice of Logic Programming, Vol. 4, Num. 1, Pag. 95-138.
2. N. D. Rotstein, M. O. Moguillansky, A. J. García and G. R. Simari (2010), A Dynamic Argumentation Framework, COMMA 2010, Pag. 427-438.
3. P. Pasquier, R. Hollands, I. Rahwan, F. Dignum and L. Sonenberg. An empirical study of interest-based negotiation. AAMAS 2011. Vol:22, Num.2, Pag. 249-288.
4. I. Rahwan, S. D. Ramchurn, N. R. Jennings, P. Mcburney, S. Parsons, L. Sonenberg. Argumentation-based negotiation. KR 2003. Vol:18, Issue 4, Pag. 343-375.
5. T. L. van der Weide, F. Dignum, J. J. Ch. Meyer, H. Prakken, G. W. A. Vreeswijk. Multi-criteria argument selection in persuasion dialogues. AAMAS 2011. Taipei, Taiwan. 921-928.
6. M. C. D. Budán, M. Gómez Lucero, C. I. Chesñevar, G. R. Simari. An Approach to Argumentation Considering Attacks Through Time. SUM 2012 – Vol. 7520, pp. 99-112, Marburg, Germany, Septiembre 2012.
7. M. C. D. Budán, M. Gómez Lucero, C. I. Chesñevar, G. R. Simari. Modeling Time and Reliability in Structured Argumentation Frameworks. KR 2012, pp 578-582, Rome, Italy, June 2012.





# Power Characterisation of Shared-Memory HPC Systems<sup>†</sup>

JAVIER BALLADINI<sup>1</sup>, ENZO RUCCI<sup>2</sup>, ARMANDO DE GIUSTI<sup>2,4</sup>,  
MARCELO NAILOUF<sup>2</sup>, REMO SUPPI<sup>3</sup>, DOLORES REXACHS<sup>3</sup>  
AND EMILIO LUQUE<sup>3</sup>

<sup>1</sup> Department of Computer Engineering, Universidad Nacional del Comahue  
Buenos Aires 1400, 8300 Neuquén, Argentina  
[javier.balladini@fi.uncoma.edu.ar](mailto:javier.balladini@fi.uncoma.edu.ar)

<sup>2</sup> III LIDI, Facultad de Informática, Universidad Nacional de La Plata  
Calle 50 y 120, 1900 La Plata (Buenos Aires), Argentina  
{[erucci](mailto:erucci@lidi.info.unlp.edu.ar), [degiusti](mailto:degiusti@lidi.info.unlp.edu.ar), [mnaiouf](mailto:mnaiouf@lidi.info.unlp.edu.ar)}@lidi.info.unlp.edu.ar

<sup>3</sup> Department of Computer Architecture and Operating Systems, Universitat  
Autònoma de Barcelona  
Campus UAB, Edifici Q, 08193 Bellaterra (Barcelona), Spain  
{[remo.suppi](mailto:remo.suppi@uab.es), [dolores.rexachs](mailto:dolores.rexachs@uab.es), [emilio.luque](mailto:emilio.luque@uab.es)}@uab.es

<sup>4</sup> Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

***Abstract.** Energy consumption has become one of the greatest challenges in the field of High Performance Computing (HPC). Besides its impact on the environment, energy is a limiting factor for the HPC. Keeping the power consumption of a system below a threshold is one of the great problems; and power prediction can help to solve it. The power characterisation can be used to know the power behaviour of the system under study, and to be a support to reach the power prediction. Furthermore, it could be used to design power-aware application programs. In this article we propose a methodology to characterise the power consumption of shared-memory HPC systems. Our proposed methodology involves the finding of influence factors on power consumed by the systems. It is similar to previous works, but we propose an in-deep approach that can help us to get a better power characterisation of the system. We apply our methodology to characterise an Intel server platform and the results show that we can find a more extended set of influence factors on power consumption.*

***Keywords:** power characterisation, shared memory systems, microbenchmarks, green computing.*

## 1. Introduction

High Performance Computing (HPC) has had for decades the only goal of increasing the processing speed of computationally complex applications such as scientific applications. Supercomputers were designed exclusively with the aim of increasing the number of floating point operations per second

(FLOPS). This is reflected in the TOP500 list [13], which uses the FLOPS metric to determine the ranking of supercomputers. The performance and the trade-off price/performance were the most important objectives.

Thus, this led to the appearance of supercomputers that consume vast amounts of electrical power and produce so much heat that large cooling facilities must be constructed to ensure proper performance. According to the Lawrence Livermore National Laboratory (LLNL), for every watt (W) of energy consumed, 0.7 W is spent in cooling to dissipate the energy. The energy consumption of current supercomputers is so high that it produces a huge economic impact. In 2005, annual spending in electrical energy at LLNL was of 14.6 million dollars [8]. Currently, the fastest supercomputer in the world (according to TOP500) has a power of 7.7 MW. The energy consumption not only has an economic impact, it also affects the ecology and society due to the lack of exploitation of renewable and clean energy.

In 2007 the first list of the Green500 [9] was published, ranking the most energy-efficient supercomputers in the world. Thus, the new era of green computing began, avoiding the focus of performance-at-any-cost. Today, the TOP500 is not the only interesting ranking, but also the Green500.

Keeping the power consumption of a system below a threshold is a great challenge for HPC, motivated by the following reasons, among others. As an energy deficit can lead to service disruptions, the energy consumption below the available energy must be maintained. In addition, in order to improve system load factor, energy suppliers often provide electricity in low-load periods at a relatively low cost. They may also provide incentives, through conservation and load management programs, to encourage elimination or shifting of peak loads [5]. In case a computing centre is supplied by a intermittent renewable energy source (for example: wind or solar farms), the energy output from the power plant increases or decreases over time and the demand of the computing centre must change accordingly. All these reasons justify the necessity to accurately predict how changes in computing system parameters and utilisation will impact future power consumption.

It may be possible to predict the power using any of the following two approaches. One approach would be to perform an initial training phase in which the application is running at various system parameters and utilisation while the power is measured and recorded. This information can then be used to predict power in new application program executions. Another approach could be to identify different application phases and to search for historical power data of microbenchmarks that match with these identified phases. If the identified phases do not exist in the history, new microbenchmarks are added together with their power information.

The first approach does not seem to be difficult to achieve. However, unfortunately, the production systems do not enable (accurate and fine grained) power measurements (for the moment) because it increases costs due to board space constraints and the need for additional components. In absence of direct power measurement, the commonly used alternative is to make power models. The basic idea behind power modelling is to take as input some Performance Monitoring Counters (PMCs) and software counters and use those to calculate power consumption. Previous works [4,10,7,6,12]

calculate the total system power consumption using several learning techniques such as linear regression, recursive learning automata, and stochastic power models. There are many different hardware counters that can be tracked, while only a few can be tracked simultaneously. Thus, we have the problem to choose the best counters that will result in an accurate model. A power characterisation of computing system could be used to judiciously select these counters.

In the second approach, microbenchmarks must be generalised in order to match them with a major number of applications phases. A power characterisation of the system is also necessary to make this generalisation. Furthermore, a power characterisation could be used to design power-aware application programs.

In this article we propose a methodology to characterise the power consumption of shared-memory HPC systems. Our proposed methodology involves the finding of influence factors on power consumed by the systems, that is, a sensitivity analysis of workload properties and system parameters on the power behaviour. The workload considers the computation and communication aspects of applications while disk input/output operations are excluded due to being a large issue to be discussed beside computation.

Our methodology is similar to previous works [4,10], but we propose an in-deep approach of the impact of workload properties on power consumption. This study can help us to get a better power characterisation of system computation.

The remaining of this article is organised as follows. Section 2 exposes the methodology overview used to characterise the power consumption of shared-memory HPC systems. Sections 3, 4, 5 and 6 describe the methodology's phases and present a case of study. Finally, section 7 discusses the conclusions and future works.

## 2. Methodology Overview

Our methodology explores influence factors of workload properties and system parameters. The methodology consists of four phases:

1. *Identification of system architecture components and parameters.* The use of different parts of hardware and system parameters normally produce different power consumption, so it is necessary to first determine what are the components that make up the architecture and the configurable parameters of the system under study.
2. *Development of microbenchmarks.* Building of small synthetic applications, called microbenchmarks, whose operations stress and evaluate special features of each architectural component.
3. *Test Cases Creation and Electrical Power Measurement.* Instrumentation of the HPC system with a power meter, and measurement of the power used to compute the microbenchmarks at different system parameters.

4. *Finding of Power-Influence Factors.* Identification of influence factors based on measurement result analysis.

The phases are further explained and supported by a case of study in sections 3, 4, 5 and 6.

### **3. Phase I: Identification of System Architecture Components and Parameters**

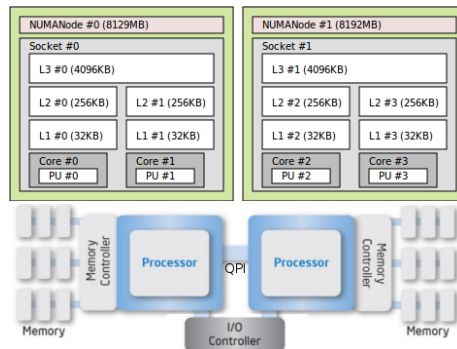
A computer consists of a set of components or modules of three basic types that communicate with each other: processor, memory and Input/Output (I/O). So, as we are focused on computation but not the I/O phases of programs, we can determine three system aspects (excluding I/O) to analyse:

1. CPU's functional units. Multi-core processors are composed of several cores (or CPUs, Central Processing Unit), and a core include several independent functional units such as Integer Unit, Floating-Point Unit, Branch Processing Unit, etc. These units inside a core are candidates to consume different power.
2. Data access. A shared memory system provides a global physical address space accessed from any core, and a design key issue of these systems is in the organisation of the memory hierarchy. The cores may have access to a central shared memory (UMA -Uniform Memory Access-), or may participate in a memory hierarchy with both local and shared memory (NUMA -Non-Uniform Memory Access-). Common memory organisations use shared caches, buses, and interconnection networks, and we need to evaluate the influence on power consumption of these parts.
3. System Parameters. The parameters of the system that can modify the power are mainly divided in two types: Resource Hibernation and Dynamic Voltage Scaling.
  - a. Resource Hibernation: The computer components consume power even when idle. Thus, the technique of resource hibernation turns off or disconnects components in idle moments. The components that can be hibernated depend on each system and can include: hard disks, cores, network interface cards, and memories.
  - b. Dynamic Voltage Scaling (DVS): Reducing the supply voltage reduces power consumption. However, it increases the delay of logic gates, so that the clock frequency should be reduced to allow the circuit to work properly. Current systems allow us to change the CPU's voltage/clock-frequency and it is clearly a factor to analyse.

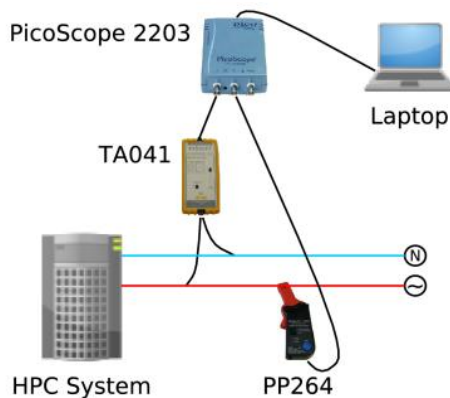
In particular, as a case of study, we evaluate the parallel platform Intel Server System SC5650BCDP, a dual socket with dual core Intel Xeon E5502 [1]

processors and 16GB of main memory (8GB per socket). Figure 1 shows the architecture and memory hierarchy, including memory sizes, of the parallel system. It is a NUMA system, each processor has an integrated memory controller, and the interconnection system between processors is the Intel QuickPath Interconnect (QPI), which provides high-speed, point-to-point links. The available CPU's clock frequencies are: 1.6, 1.73 and 1.86 GHz. For the best of our knowledge, the system does not support physical hot-plug of memories.

The processor supports low power states (C-states) at individual core. On the contrary, Intel Turbo Boost technology is not supported by E5502 processors.



**Fig. 1.** Architecture and memory hierarchy of the system under study.



**Fig. 2.** Measurement connection diagram.

## 4. Phase II: Development of Microbenchmarks

After identifying what are the components that make up the architecture under study, we developed a set of microbenchmarks that allow us to characterise it. The microbenchmarks were developed using language C, gcc compiler version 4.6.3, under GNU/Linux with Pthreads library for threads management. To guarantee that the compiler does not affect the microbenchmark's purpose, assembler codes generated by it were verified using the *objdump* command (with -d option).

Each microbenchmark launches four threads, where each thread runs a loop executing between 1 and 2.5 Gigaoperations. Taking into account the hardware characteristic of the support architecture, we consider the next factors at the time of developing the microbenchmarks:

**Operation and Data Type.** Different operations on different data types have different complexity hardware implementation. We developed microbenchmarks for add, multiply and division operations. Also, four special microbenchmarks were developed. The first one evaluates the cost of performing no specific operation. The rest of them evaluate the cost of performing a complex operation that involves other simpler ones. Data is read from three vectors accessed sequentially (stride = 1). The data types used are: int (32-bit integer), float (32-bit floating point) and double (64-bit floating point).

**Operands Accessing Mode.** Load and store instructions have different computational cost. It is interesting to study what occurs with power. We developed two microbenchmarks: one that only executes load instructions and one that only executes store instructions.

**Non-Uniform Memory Access.** The support architecture has a common memory address space, but the accesses may be local or foreign. The foreign access uses QPI interconnection link. Thus, we developed two microbenchmarks that write data in main memory: one with local access and one with foreign access.

**Resources Usage Efficiency.** Systems are often scheduled incorrectly and, as a result, resources usage efficiency decreases. We developed two microbenchmarks to study the power influence of this factor: an efficient one and an inefficient one. Both microbenchmarks write data in memory but they differ in the number of used cores. The efficient microbenchmark uses all the cores of the architecture used, while the inefficient microbenchmark use only one of them.

**Parallel Programming Model.** Programming models differ on the way of dealing process communication and synchronisation, either as shared memory or distributed memory. OpenMP is the most widely used model on shared memory, while MPI is the corresponding for distributed memory. We used the NAS Parallel Benchmarks (NPB) [2], implemented both in OpenMP and MPI, to evaluate the parallel application programming model influence on parallel architectures power. We selected the CG, IS and EP benchmarks which are computation bound, and the MG benchmark which is communication bound [11]. We selected only these benchmarks because the others (NPB) do not provide additional information for our objective. We determined the benchmarks problem size so that its main memory requirements for execution are met and

memory swapping never happens. Particularly, we chose a class C problem size defined by the specification of the NPB.

**Cache Friendliness.** The access to the memories of parallel machines (implemented with different technologies and placed at different locations) is a candidate to be a power influence factor. The developed microbenchmarks can be classified into two groups. The first group microbenchmarks are cache friendly, that is, they have a good cache hit rate. The second group microbenchmarks are not cache friendly, that is, they have bad cache performance at all the levels. The cache friendly microbenchmarks work with a data set smaller than the L1 cache size. The data set size of the no cache friendly microbenchmarks is bigger than the L3 cache size.

In a first stage, we focused on intensive use of the CPU, so we have left the analysis of the C-states (as a power influence factor) for the future. Table 1 shows a description of some of the developed microbenchmarks. Input parameters are: Characteristic Memory Access Pattern, Data Type, Characteristic Basic Operation, Cache Friendliness, Parallelism Level and Clock Frequency. The Characteristic Memory Access Pattern indicates the data structures used by the microbenchmark and how they are accessed by it. The Data Type parameter indicates the data type of the data structures that were defined in previous column. The Characteristic Basic Operation parameter represents the operation performed by each thread of the microbenchmark. The Cache Friendliness parameter indicates whether the microbenchmark is cache friendly or not. The Parallelism Level parameter is adjusted to the number of cores of the support architecture (one thread per core). The Clock Frequency parameter depends on the available processors clock frequencies. Different executions for different  $k_1$ ,  $k_2$  and  $f_i$  values were done. In all the cases, each thread performs the same basic operation on its own data set.

**Table 1.** Some developed microbenchmarks.

Microbenchmark	Characteristic Memory Access Pattern	Data Type	Characteristic Basic Operation	Cache Friendliness	Parallelism Level	Clock Frequency
addFloatCache	a,b,c: stride-1	Float	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = a_j + b_j$	Yes	4	$f_i$
divDoubleMem	a,b,c: stride-1	Double	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = a_j / b_j$	No	4	$f_i$
storeDoubleCache	a,b: stride-1	Double	$for_{i=0}^{k_1} for_{j=0}^{k_2} a_j = b_j = 0$	Yes	4	$f_i$
loadDoubleCache	a,b: stride-1	Double	$for_{i=0}^{k_1} for_{j=0}^{k_2} x = a_j, y = b_j$	Yes	4	$f_i$
noOpIntCache	a,b,c: stride-1	Int	$for_{i=0}^{k_1} for_{j=0}^{k_2} x = a_j, c_j = b_j$	Yes	4	$f_i$
multiOpIntCache	a,b,c: stride-1	Int	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = 8 * a_j + b_j + a_j * a_j * \frac{5}{b_j} + b_j * b_j * a_j$	Yes	4	$f_i$
multiAddIntCache	a,b,c: stride-1	Int	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = a_j + b_j + a_j + a_j + 5 + b_j + b_j + a_j$	Yes	4	$f_i$
umaFloatCache	a,b,c: stride-1	Float	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = a_j + b_j + c_j, a_j = c_j, b_j = c_j$	Yes	4	$f_i$
efficientFloatCache	a,b,c: stride-1	Float	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = a_j / b_j$	Yes	4	$f_i$
inefficientFloatCache	a,b,c: stride-1	Float	$for_{i=0}^{k_1} for_{j=0}^{k_2} c_j = a_j / b_j$	Yes	1	$f_i$
...	...	...	...	...	...	...

## 5. Phase III: Test Cases Creation and Electrical Power Measurement

Once the microbenchmarks are developed, the next step is the creation of test cases. Test cases are written using bash scripts that execute combinations of microbenchmarks and system parameters. Once the HPC system is instrumented with the power meter, the test cases are run and measured. Following, we explain how to scale the clock frequency using a bash command, and later we expose the power measurement methodology.

### 5.1 System Parameter: Clock Frequency Scaling

Modern general purpose processors can scale the frequency of each core individually. Access is through the Advanced Configuration and Power Interface (ACPI). It is possible to know, for a given core, the available frequencies and the frequency currently in use, respectively, reading the following two files in GNU/Linux:

```
/sys/devices/system/cpu/cpu0/cpufreq/scaling_available_frequencies
/sys/devices/system/cpu/cpu0/cpufreq/scaling_cur_freq
```



To change the frequency it is possible to use *cpufreq-selector* command. For example, running the command “`cpufreq-selector -c 0 -f 1000000`” the core number 0 is set to 1 GHz.

## 5.2 Power Measurement Methodology

This section explains some definitions about power and energy, and the methodology typically used to measure the electrical power of a whole system, detailing the instruments utilised by us.

Power is the rate at which the system consumes electrical energy. The watt (W) is the unit of real power, equivalent to 1 joule by second (1 J/s), and it is the product of current times voltage. Energy is the total amount of electrical energy that the system consumes over time, and is measured in joules or watt-hour (Wh).

We are interested in power; and energy can be calculated by integrating power over time. We measure the power consumption of the whole shared-memory HPC system. For this, we use the oscilloscope PicoScope 2203, the TA041 active differential oscilloscope probe, and the PP264 60 A AC/DC current clamp, all products of Pico Technology. The electrical signals captured by the dual-channel PicoScope 2203 are transmitted in real-time via USB to a laptop. The voltage is measured using the TA041 probe that is connected to one oscilloscope's input channel. The current of the phase conductor is measured using the PP264 current probe that is connected to the other input channel of the oscilloscope. Then, power is calculated as the product of measured voltage and current. The sample rate for the experiments was of 1000 Hz. Figure 2 shows the measurement connection diagram.

## 6. Phase IV: Finding of Power-Influence Factors

After running and measuring test cases, we proceed to find power-influence factors. Figure 3 shows the average power for the microbenchmarks developed to evaluate “Operation and Data Type” and “Cache Friendliness” factors. The microbenchmarks run at the maximum clock frequency. It can be observed that microbenchmarks with bad cache performance produce higher average power. Regarding the “Operation and Data Type” factor, it can be seen that the operation to perform has no significant impact on int data type, except when the operation is composed of multiple ones (multiOp, multiAdd and multiDiv). While floating point data types (float and double) have different size, their power behaviour are similar. Also, it can be seen that the operation to perform impacts on the average power produced, being these values generally lower than those for int microbenchmarks.

The influence of “Data Type”, “Operands Accessing Mode” and “Cache Friendliness” factors are analysed in Figure 4. This chart shows the average power for the microbenchmarks developed to evaluate those factors at the maximum clock frequency. It can be observed that average power for load

and store microbenchmarks are similar when the cache has a good performance but this similarity does not maintain when cache miss rate increases. It can be seen that store instructions performed in main memory produces higher average power, particularly when using int and double data types.

Figure 5 shows the average power for “Non-Uniform Memory Access” and “Cache Friendliness” microbenchmarks at the maximum clock frequency. It can be observed that average power hardly varies when the cache performance is good. In the opposite situation, that is when the cache has a bad performance, the use of the QPI interconnection link decreases average power.

Figure 6 allow us to evaluate the power influence of “Resources Usage Efficiency” and “Cache Friendliness” factors. This chart shows the average power for the microbenchmarks developed to evaluate those factors at the maximum clock frequency. It can be seen that average power increases when resources are used efficiently, beyond the cache performance.

The “Parallel Programming Model” influence factor is analysed in Figure 7. This chart shows the average power for each NAS benchmark executed at the maximum clock frequency. It can be observed that the programming model practically has no impact on the average power of each benchmark tested.

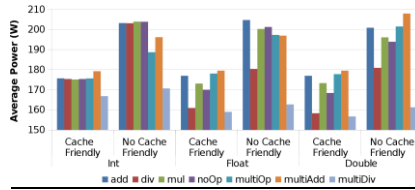


Fig. 3. Influence of “Operation and Data Type” and “Cache Friendliness”.

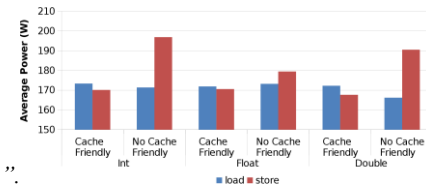


Fig. 4. Influence of “Operands Accessing Mode” and “Cache Friendliness”.

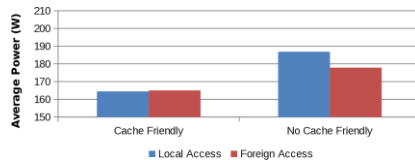
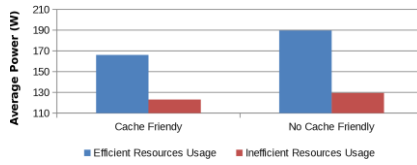
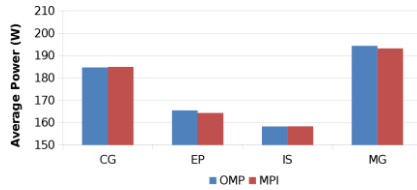


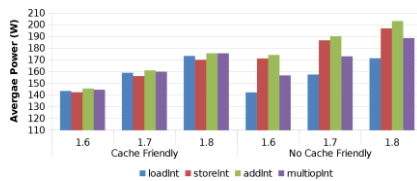
Fig. 5. Influence of “Non-Uniform Memory Access” and “Cache Friendliness”.



**Fig. 6.** Influence of “Resources Usage Efficiency” and “Cache Friendliness”.



**Fig. 7.** Influence of “Parallel Programming Model”



**Fig. 8.** Influence of “Voltage and Frequency Scaling” and “Cache Friendliness”.

Figure 8 allow us to assess the influence of “Cache Friendliness” and “Voltage and Frequency Scaling” factors. This chart shows the average power for int microbenchmarks with good and bad cache performance at different clock frequencies (for readability only four microbenchmarks are shown). It can be seen that average power increases when the clock frequency increases, regardless the cache performance.

From previous analysis, we can confirm the influence of the studied factors on the average power of the support architecture:

- When working with floating point data types (float or double), the operation to perform must be taken into account because it influences the produced average power. It does not occur the same with int data type. Beyond data type used, the ratio of the number of mathematical operations to the number of data read from main memory must be a factor to consider.
- When the cache has a good performance, local and foreign accesses have similar power behaviours. Nevertheless, when cache

- performance is bad, foreign accesses produce lower average power than local accesses.
- Increase the efficiency in resources usage produces higher average power.
  - The parallel programming model is not a power influence factor (although it is an energy influence factor as we analysed in [3]).
  - The cache performance has no influence on the average power produced by load instructions. However, store instructions produces higher average power when they are performed in main memory, particularly when int and double data types are used.
  - Beyond the operands accessing mode, the operation to perform, the data type, the parallel programming model and the cache performance, the average power increases when the clock frequency increases.

## 7. Conclusions and Future Works

In this work we present a methodology to characterise the power consumption of shared-memory HPC systems. The power characterisation can be used to know the power behaviour of the system under study in order to design power-aware application programs, and to be a support to reach the power prediction. We apply our methodology to characterise an Intel server platform and the results show that we can find an extended set of influence factors on power consumption.

As future works, we will analyse the influence of C-states on power consumption of our platform. Later, we plan to find a way to automatically characterize a system, following our methodology. Furthermore, we will continue working on power prediction of HPC systems using the information obtained with the power characterisations.

## References

1. Intel E5500 datasheet - Vol 1 (Accessed on 2012), <http://www.intel.com/content/www/us/en/processors/xeon/xeon-5500-vol-1-datasheet.html>.
2. NAS Parallel Benchmarks (Accessed on 2012), <http://www.nas.nasa.gov/publications/npb.html>.
3. Ballardini, J., Suppi, R., Rexachs, D., Luque, E.: Impact of parallel programming models and cpus clock frequency on energy consumption of hpc systems. In: AICCSA. pp. 16-21 (2011).
4. Bircher, W.L., John, L.K.: Complete system power estimation using processor performance events. IEEE Transactions on Computers 61, 563-577 (2012).
5. Capehart, B.L. (ed.): Encyclopedia of Energy Engineering and Technology. CRC Press (2007).

6. Contreras, G.: Power prediction for intel xscale processors using performance monitoring unit events. In: In Proceedings of the International symposium on Low power electronics and design (ISLPED). pp. 221-226. ACM Press (2005).
7. Economou, D., Rivoire, S., Kozyrakis, C.: Full-system power analysis and modeling for server environments. In: In Workshop on Modeling Benchmarking and Simulation (MOBS (2006).
8. Feng, W.C.: The importance of being low power in high-performance computing. *Cyberinfrastructure Technology Watch Quarterly* 1 (3) (August 2005).
9. The Green500 website (Accessed on 2012), <http://www.green500.org/>.
10. Jiménez, V., Cazorla, F.J., Gioiosa, R., Valero, M., Boneti, C., Kursun, E., Cher, C.Y., Isci, C., Buyuktosunoglu, A., Bose, P.: Power and thermal characterization of power6 system. In: Proceedings of the 19th international conference on Parallel architectures and compilation techniques. pp. 7-18. PACT '10, ACM (2010).
11. Jin, H., Hood, R., Chang, J., Djomehri, J., Jespersen, D., Taylor, K.: Characterizing application performance sensitivity to resource contention in multicore architectures. Tech. rep., NASA Advanced Supercomputing (NAS) Division (2009).
12. Qiu, Q., Wu, Q., Pedram, M.: Stochastic modeling of a power-managed system: construction and optimization. In: Proceedings of the 1999 international symposium on Low power electronics and design. pp. 194-199. ISLPED '99 (1999).
13. The TOP500 website (Accessed on 2012), <http://www.top500.org/>.



# A new approach for Analyzing I/O in parallel scientific applications<sup>1</sup>

SANDRA MÉNDEZ, JAVIER PANADERO, ALVARO WONG,  
DOLORES REXACHS AND EMILIO LUQUE.

Computer Architecture and Operating Systems Department (CAOS)  
Universidad UniversitatAutònoma de Barcelona, Barcelona, Spain

{javier.panadero,alvaro.wong}@caos.uab.es, {sandra.mendez, dolores.rexachs,  
emilio.luque}@uab.es

***Abstract.** The efficient use of high performance computing is usually focused on the use of computational resources. However, scientific applications currently produce a large volume of information. Therefore, the Input/Output (I/O) subsystem also should be used efficiently. In order to do so, it is necessary to know the application I/O patterns and establish a relationship between these patterns and the I/O subsystem configuration. To analyze the I/O behavior of applications, we propose use a library of the PAS2P (Application Signature for Performance Prediction) tool. Parallel applications typically have repetitive behavior, and the I/O patterns of parallel applications also have that behavior. We propose to identify the portions (I/O phases) where the application does I/O. From these I/O phases, we extract an application model that can be used to evaluate it in different I/O subsystems considering the I/O phases and compute-communication phases. In this paper, we present the concepts used in the PAS2P methodology, which have been adapted for MPI-IO applications. We have extracted the I/O model of applications. This approach was used to estimate the I/O time of an application in different subsystems. The results show a relative error of estimation lower than 10%.*

**Keywords:** Parallel I/O System, I/O Access Patterns, I/O Configuration, Modeling I/O, I/O phases.

---

<sup>1</sup>This research has been supported by the MICINN Spain under contract TIN2007-64974, the MINECO (MICINN) Spain under contract TIN2011-24384, the European ITEA2 project H4H, No 09011 and the AvanzaCompetitividad I+D+I program under contract TSI-020400-2010-120. Appreciation to The Centre of Supercomputing of Galicia (CESGA), Science and Technology Infrastructures (in spanish ICTS).

## 1. Introduction

Due to the historical "gap" between the computing and Input/Output (I/O) performance, in many cases, the I/O system becomes the bottleneck of parallel systems. In order to hide this "gap", the I/O factors with the biggest influence on performance must be identified. Furthermore, the increased computational power of processing units and the complexity of scientific applications, which use high performance computing, require more efficient Input/Output Systems.

The configuration of the I/O subsystem affects the application performance. It is important to understand the I/O subsystem structure: filesystem type, I/O devices, interconnection networks and I/O libraries. The I/O subsystem in computer clusters can have several I/O configurations and the user should select the configuration depending on the I/O requirements of his application. However, usually the user does not know the I/O subsystem and the I/O requirement of application.

We propose a methodology to extract the I/O requirements of the application expressed by a I/O abstract model that can be used in different I/O subsystem. Also, we propose a method to select the I/O configuration from set existing configurations, depending on the I/O abstract model of application.

Since the I/O operations are affected by time between operations, we need to analyze the parallel application, taking into account the computing and communication. For this reason, we selected the PAS2P library to trace the I/O operations of standard MPI-2.

In previous work [1], we have presented a methodology for performance evaluation of the I/O system, which is focused on I/O path.

In this paper, we explain the I/O analysis for the application's access pattern extraction, the I/O phase's identification and the I/O abstract model of application.

This article is organized as follows: in Section II we review the related work; Section III introduces our proposed methodology. In Section IV we review the experimental validation. Finally, we present conclusions and future work.

## 2. Related Work

There are other tools which are closely related to the I/O analysis of message-passing applications. Darshan[2] is a parallel I/O characterization tool designed to characterize the MPI-IO file access of HPC applications in a non-intrusive way. It characterizes the application by using statistics and cumulative timing information. The information obtained can be used to analyze the I/O behavior of a MPI-program. It is implemented as a set of user space libraries. These libraries require no source code modification and can be added in a transparent way.



This approach differs from PAS2P-I/O because Darshan provides statistical averages of I/O instead of information by I/O operation.

LANL-Trace [3] is a tracing framework that wraps the standard Unix library and system call tracing utility ltrace. LANL-Trace generates three types of outputs, which are useful for the I/O analysis.

One advantage of LANL-Trace is that it is simple to understand and use. Because of its simple nature, it is also easy to modify.

However, LANL-Trace's simplicity is a trade-off because it causes higher overhead. This work differs from our proposal because PAS2P-I/O intercepts the MPI functions imposing minimal overhead.

There are other general-purpose instrumentation tools that profile and trace general MPI and CPU activity. These tools allow the analysis of the I/O parallel applications. The most common tools used by the scientific community are: Jumpshot[4], TAU [5] and STAT [6]. The main difference with our approach lies in the fact that these tools are focused primarily on the general analysis of the application, without providing specific details and information about the MPI-I/O operations.

### **3. Proposed Methodology**

We propose a methodology to analyze the I/O requirements of parallel scientific applications, which is composed by three steps: Modeling Input/Output of Application, using the Application Input/Output Model, and Validation of Application I/O Model Applicability.

#### **3.1 Modeling Input/Output of Application**

The I/O model of applications is defined by three characteristics: metadata, spatial global pattern and temporal global pattern.

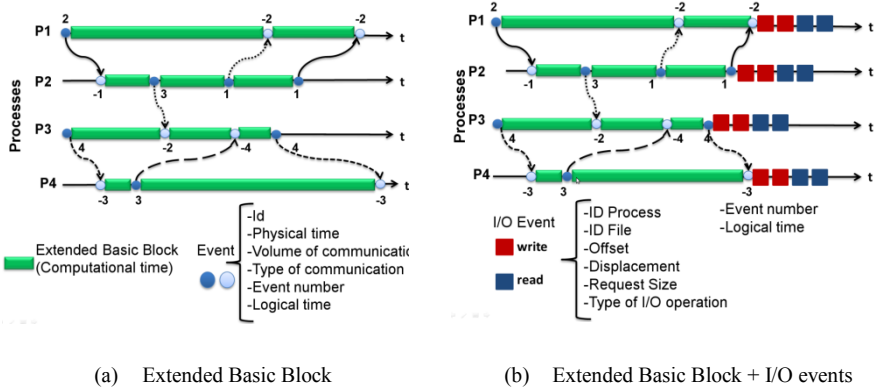
To obtain the application I/O model, we have implemented a library extension named libpas2p-io.so. This dynamic library will be used to instrument the application.

PAS2P tool implements the PAS2P methodology [7], which is based on the high repetitive behavior of the applications. The PAS2P methodology is composed by two-step. The first step is to analyze the application to build an application model and extract its phases and weights. Finally, it uses that information to build an executable signature. The second step is to execute the application signature in a target system in order to predict the total execution time of the application.

To extract the three characteristics for the I/O model of parallel application, we have focused on the first step of PAS2P methodology: data collection, pattern identification, parallel application model, and extraction of phases and weights.

**Data collection** The aim of this step is to generate a trace log with the behavior of computation and communication of a parallel application. In order to intercept and collect communication events (action of sending or receiving a message) interposition functions are used. To collect the computational time, PAS2P has extended the concept of Basic Block (BB)[8] for parallel applications. We have defined a new concept named Extended Basic Block (EBB) as a segment of a process, whose beginning and end are defined by occurrences of MPI events, either sent or received. Also it may say that it is a "computational time" segment bounded by communication events, as shown in Figure 1 that also illustrates the event information recollected, which will be used to generate the trace log.

An event contains information of each application process, the MPI event, the source-destination the process is involved with, the count events, the communication volume, the wall clock time, the computational time of the event occurred as well as its type (send: 1, recv: 0, collective events: -1). This log contains the whole application trace. It can be used in order to analyze the application behavior and the trade-off between compute-communication.



**Fig. 1. Data Collection**

In order to consider the I/O data of the application, we use the concept of the I/O event, which is a segment of process where the I/O operation is called. The I/O event is composed by ID process, ID file, offset, displacement, request size, type of I/O operation, event number, and logical time. We have incorporated the I/O events to PAS2P library. This allows identifying the relationship between the events of communication, computation and I/O. Fig.1(b) shows an example of physical traces with EBB and the I/O events.

**Parallel application model** Synchronization between processes is necessary in parallel applications. PAS2P has developed a Logical ordering algorithm [7] inspired by Lamport's[9]. Through this algorithm, PAS2P defines a new logical ordering, in which, if one process sends a message in a Logical Time (LT), it's receive will be modeled to arrive at  $LT + 1$ .

Once all events have been assigned a LT, the logical trace is created from a physical trace, where Logical Times will be given by LT for the Send events (LTSend) and LT for the reception events (LTRcv). Finally, once each event has been located in the respective Logical Time, the logical trace is divided into more logical times, that is, there can only be one event for each process in a Logical Time. Once we know this, we are able to introduce two new concepts. The first new concept is a *tick*, tick is defined as a logical unit time, and it is incremented by each communication event. Another new concept is *Parallel Basic Block (PBB)*, which is defined as a set of Extended Basic Blocks delimited by two ticks. The first tick defined as entry point has at least one event, and the second tick defined as exit point also has at least one event.

Synchronization of the I/O operations depend on the data consistency managed by the filesystem. Therefore, for the I/O operations the logical trace is the same as the physical trace and a *tick* is equal to a logical time unit. A PBB in I/O is equal to an I/O event where the entry/exit point is the call to I/O operation. In order to obtain the I/O abstract model of the application, we have to analyze the application I/O. Therefore, the response time of I/O operations is not needed, because it depends on performance of the target I/O subsystem.

Once this step has been done, the I/O model of application is obtained. This model is made up by three events: computing, communication and I/O events.

**Pattern identification** In order to find a representative behavior of a message-passing application, once we have obtained the logical trace, as shown in Fig.2(a), we search the application phases.

With the objective of finding these phases, PAS2P has an algorithm [7], which is based on three criteria (type of communication, volume of communication and computational time by process). Finally, phases are created as sub-chains of grouped PBB's that repeat along the execution, as shown in figure Fig.2(b).

The I/O phases depend on the local and global pattern, as well as the spatial and temporal occurrence of these patterns. To identify the I/O phases, we analyze the similarity of I/O events. I/O events are named Access Pattern (AP). An I/O phase is composed by AP with similar order and without events communication between the I/O events. From the logical trace (Fig.3(a)) the I/O phases of the application are obtained (Fig. 3(b)). We show the access patterns denoted by AP and I/O phases identified in the logical trace.

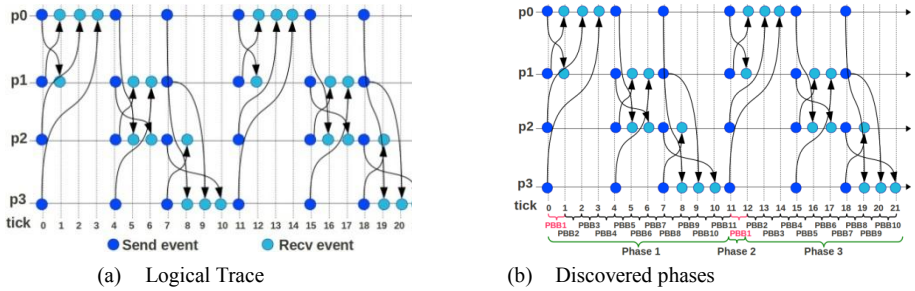


Fig. 2. Extracting phases from the Logical trace

**Extract relevant phases and weights** Once we have identified the phases of a parallel application, we define the weights and the relevant phases. The weight will be given by the frequency in which each phase repeats. A relevant phase is when the weight multiplied by the phase runtime is representative of the total application runtime. We have considered that this "representativeness" will be given if the phase is 1% or more of the total execution time of the whole application. Each phase contains also information about its patterns, such as the communication pattern, the volume of communication and the computational time by process.

PAS2P definitions of phase and weight predict the performance of an application in target systems with a minimum error. However, for the I/O analysis, we need

to define concepts of the phase and the weight to create the I/O abstract model in which all I/O phases are considered.

The weight depends on the number of processes, request size and repetitions of each  $AP$  that is part of a phase. The weight is expressed in Megabytes and it is necessary to determine transferred data in each I/O phase. The I/O abstract model depends on I/O phases and weight, allowing us to know "when" and "how" the I/O subsystem will be used. The I/O abstract model of application can be used to mimic the application I/O and determine the I/O subsystem where the application will be less penalized by its I/O behavior.

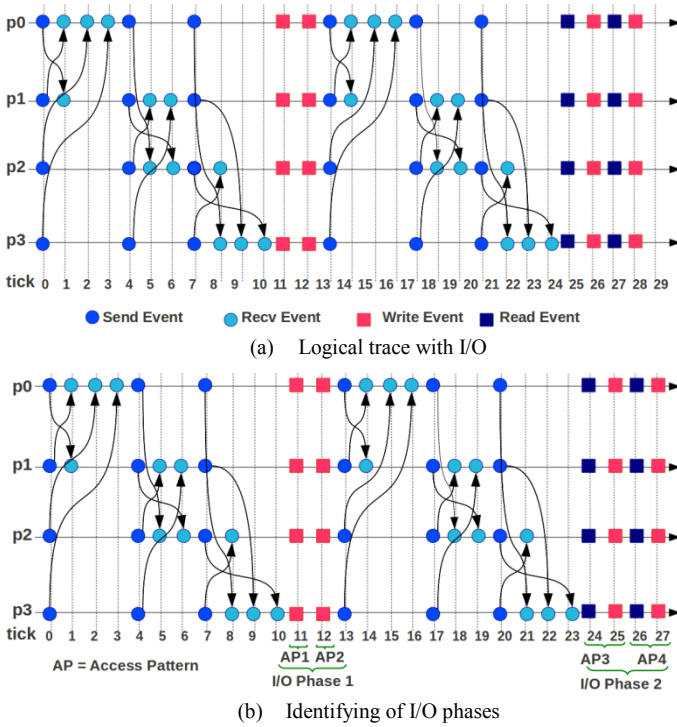


Fig. 3. Identifying the I/O phases from the Logical trace

### 3.2 Using the Application Input/Output Model

In previous work [1], we have presented the I/O system performance evaluation through exhaustive characterization with the benchmark IOR [10] for I/O library (MPI-IO) and benchmark IOzone[11] for I/O devices.

In the present paper, we use the I/O abstract model to estimate the application time in different subsystems in order to select the configuration with less time. In this way, we achieve a characterization of I/O subsystem for the different types of access patterns of the application. Furthermore, we can obtain a performance characterization of a specific application in less time. The characterization time represents from 10% to 20% of running time of full application.

We use the I/O abstract model to set up the input parameters of the benchmark IOR. We only execute the benchmark for the phases of the I/O model.

The following setting of input parameters is applied on IOR for each I/O phase:  $s = 1$ ;  $b = weight_{(phase[i])}$ ;  $t = rS_{(phase[i])}$ ;  $NP = np_{(phase[i])}$ ;  $-F$  if there is 1 file per process;  $-c$  if there is collective I/O.

The selected metric for IOR is the transfer rate ( $MB/sec$ ), named  $BW_{CH}$ . The estimated I/O time for each I/O phase is calculated by expression (1).

$$Time_{io} = \sum_{i=1}^n Time_{io}(phase[i]) \quad (1)$$

Where the  $Time_{io}(phase[i])$  is calculated by expression (2)

$$Time_{io}(phase[i]) = \frac{weight(phase[i])}{BW_{CH}(phase[i])} \quad (2)$$

Where  $BW_{CH}(phase[i])$  is the characterized transfer rate at I/O library level for a similar access pattern. The I/O model is used to determine what system can provide the best performance for the application at I/O library level.

### 3.3 Validation of Application I/O Model Applicability

It is necessary to evaluate the estimation's accuracy for the selected configuration. In order to do that, we evaluate the relative error produced by the I/O time estimation.

Relative error is calculated by expression (3).

$$error_{rel} = 100 * \left( \frac{error_{abs}}{BW_{MD}(phase[i])} \right) \quad (3)$$

Where absolute error is calculated by the expression (4)

$$error_{abs} = |BW_{CH}(phase[i]) - BW_{MD}(phase[i])| \quad (4)$$

Where  $BW_{CH}(phase[i])$  is *transferRate* characterized at I/O library level. When a phase has two or more I/O operations, the  $BW_{CH}(phase[i])$  is defined by the average of the  $BW_{CH}$  of each I/O operation that composes the I/O phase.

## 4. Experimental validation

We have applied the proposed methodology to select I/O configuration of two I/O subsystems. Table 1 shows both the configuration A and the configuration of cluster Finisterrae[12].

The I/O phases identification is applied to Block Tridiagonal(BT) application of NAS Parallel Benchmark suite (NPB) [13]. The BTIO benchmark performs large collective MPI-IO writes and reads of a nested strided datatype, and it is an important test of the performance that a system can provide for non-contiguous workloads. After every five time steps the entire solution field, consisting of five double-precision words per mesh point, must be written to

one or more files. After all time steps are finished, all data belonging to a single time step must be stored in the same file, and must be sorted by vector component, x-coordinate, y-coordinate, and z-coordinate, respectively.

**Table 1.** Description of configuration A and Finisterrae

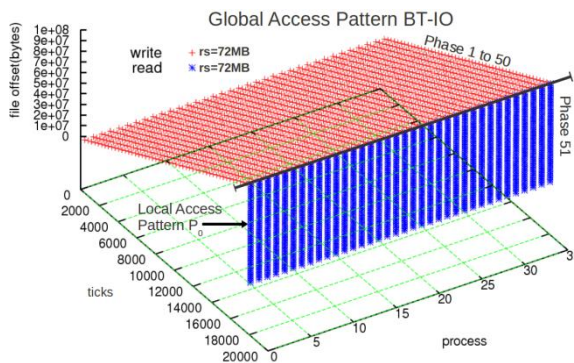
I/O Element	Configuration A	Finisterrae
I/O library	OpenMPI	mpich2,HDF5
Communication	1GbpsEthernet	1Infinibad20Gbps
Storage Network	1GbpsEthernet	1Infinibad20Gbps
Filesystem Global	NFSVer3	Lustre(HPSFS)
I/O nodes	8DASand1NAS	180S
Metadata Server	1	2with72cabinsSFS20
Filesystem Local	Linuxext4	Linuxext3
Level Redundancy	RAID 5	RAID
Number of I/O Devices	5disks	866disks
Capacity of I/O Devices	1.8TBhot-swapSAS	866*250GB
Mounting Point	/home	\$HOMESFS

Since NAS BTIO has an access mode strided and the IOR is not working in this mode, we have selected the sequential access mode to replicate the I/O phases.

We have obtained the following meta-data of NAS BT-IO in the FULL subtype with our tool:

- Explicit offset, Blocking I/O operations, Collective operations.
- Strided Access mode, Shared Access type.
- MPI-IO routine `MPI_File_set_view` with type of 40.

We have obtained the I/O model of NAS BT-IO for 36, 64 and 121 processes for the Class D and this model has been applied in the configuration A and Finisterrae. Fig.3 shows the I/O abstract model for 36 processes.



**Fig. 3.** I/O model to NAS BT-IO, class D, 36 processes, and subtype FULL on Configuration A and Finisterrae

Table 2 shows description of the I/O phases for NAS BT-IO where  $ph$  is the number of phase,  $rs$  is request size,  $np$  the number of processes of  $ph$ ,  $idP$  is the rank of MPI process,  $iter$  is the number of iteration into  $ph$  and  $rep$  is the number of repetitions of  $ph$ . Initial offset depends on  $rs$  and the number of phases. Also the request size depends on number of processes of the application. NAS BT-IO has the following values for the class D:  $rs = 72MB$  for 36p,  $rs = 40MB$  for 64p and  $rs = 24MB$  for 121p.

**Table2.** I/O phases description of NASBT-IO subtype FULL, class D using  $np$  processes

Phase	#Operation	InitialOffset	rep	weight
1-50	$np * W$ in each phase	$rs * idP + (rs * (ph - 1) + (rs * (np - 1))) * xphase$	1	$np * rep * rs$
51	$np * R$	$rs * idP + (rs * (iter - 1) + (rs * (np - 1)) * (iter - 1))$	50	$np * rep * rs$

**Table 3.** I/O time estimation ( $Time_{io}(CH)$ ) on configuration A and Finisterrae using 64 processes

Phase	$Time_{io}(CH)$ on conf.A	$Time_{io}(CH)$ on Finisterrae
Phase1-50	1167.40	932.36
Phase51	2868.51	844.42

We have calculated the I/O time using our proposed methodology. Table 3 shows the  $Time_{io}(CH)$ , where we can observe that the configuration with less I/O time for NAS BT-IO with 64 processes is Finisterrae.

Table 4 shows the  $error_{rel}$  in the estimation for 64 processes on configuration A and Finisterrae.

**Table 4.** Error of I/O time estimation on configuration A and Finisterrae for 64 processes

Phase	$Time_{io}(CH)$	$Time_{io}(MD)$	$error_{rel}$
<b>Configuration A</b>			
Phase1-50	1167.40	1153.05	1%
Phase51	2868.51	2984.75	4%
<b>Configuration Finisterrae</b>			
Phase1-50	932.36	924.85	1%
Phase51	844.42	909.43	7%

We evaluated these errors by executing several times NAS BT-IO and the error was similar in all the different tests. Furthermore, the I/O model has been obtained at a different time to discard the influence of the tracing tool.

The same I/O model can be applied to estimate the I/O time in other systems, where  $Time_{io}(CH)$  will be obtained by the expression (2), the  $BW_{CH}$  will be obtained by executing IOR with the input parameters explained in the section 3.2.



As we can see, the estimation improves when increasing the number of processes. The error rate on both configurations is less than 10% and is reduced when workload and number of processes is increased.

## 5. Conclusion

A methodology to obtain an I/O abstract model of a parallel application has been proposed and tested. The application I/O model is defined by three characteristics: metadata, spatial and temporal global pattern. We instrument the application to obtain the access pattern and we analyze it to identify the I/O phases. This instrumentation is done at MPI-IO level, which does not require the source code. We have obtained the I/O model of NAS BT-IO and we have evaluated two configurations taking into account the I/O phase behavior. We have used the I/O model to estimate the I/O time. Relative errors are acceptable and we have observed that the error rate decreases when the number of processes is increased. The error was about the 10%.

As future work, we are designing a benchmark to replicate the I/O when there are two or more operations in a phase to improve the characterization and reduce estimation error. We are extending the I/O phase's identification to different applications that show different I/O behaviors. We are analyzing *upwelling* of ROMs framework, which is an application that opens different files in execution time. Our model is applicable for each file, but still, it is necessary to refine the methodology for the I/O phases with complex access patterns.

## References

1. Mendez, S., Rexachs, D., Luque, E.: Methodology for performance evaluation of the input/output system on computer clusters. In Workshop IASDS on Cluster Computing, 2011 IEEE International Conference on. (sept. 2011) 474–483.
2. Carns, P. H., Latham, R., Ross, R. B., Iskra, K., Lang, S., Riley, K.: 24/7 Characterization of Petascale I/O Workloads. In: CLUSTER, IEEE (2009) 1–10.
3. Konwinski, A., Bent, J., Nunez, J., Quist, M.: Towards an I/O Tracing Framework Taxonomy. In: Proceedings of the 2nd Int. Workshop on Petascale data storage: in SC'07, PDSW'07, New York, USA, ACM (2007) 56–62.
4. Zaki, O., Lusk, E., Swider, D.: Toward Scalable Performance Visualization with Jumpshot. High Performance Computing Applications **13** (1999) 277–288.
5. Shende, S. S., Malony, A. D.: The TAU Parallel Performance System. Int. J. High Perform. Comput. Appl. **20**(2) (May 2006) 287–311.

6. Arnold, D., Ahn, D., deSupinski, B., Lee, G., Miller, B., Schulz, M.: Stacktrace analysis for large scale debugging. In: IPDPS2007. IEEE International. (2007)
7. Wong, A., Rexachs, D., Luque, E.: Extraction of Parallel Application Signatures for Performance Prediction. HPCC, 10th IEEE Int. Conference (2010) 223–230.
8. Lau, J., Schoemackers, S., Calder, B.: Structures for Phase Classification IS-SN- Performance Analysis of Systems and Software, ISPASS (2004) 57–67.
9. Lamport, L., Time, C.: The Ordering of Events in a Distributed System. Communication of the ACM **21**(7) (1978) 558–565.
10. William Loewe, T.M., Morrone, C.: IOR Benchmark (2012).
11. Norcott, W.D.: IOzone filesystem benchmark (2006).
12. Finisterrae, C.: Centre of Supercomputing of Galicia (CESGA). Technical report, Science and Technology Infrastructures (in Spanish ICTS) (2012).
13. Wong, P., Wijngaart, R.F.V.D.: NAS parallel benchmarks I/O 2.4. Technical report, Computer Sciences Corp., NASA Advanced Supercomputing Division (2003).

# Analysis of Pure and Hybrid Parallel Solutions in a Simulation Problem.

SILVANA LIS GALLO<sup>1,2</sup>, FRANCO CHICHIZOLA<sup>1</sup>, LAURA DE GIUSTI<sup>1</sup>,  
MARCELO NAIOUF<sup>1</sup>

<sup>1</sup> Instituto de Investigación en Informática LIDI (III-LIDI),  
Facultad de Informática,  
Universidad Nacional de La Plata,  
50 y 120 2<sup>do</sup> piso, La Plata, Argentina.  
<sup>2</sup> Fellow, CONICET.

{sgallo, francoch, ldgiusti, mnaiouf}@lidi.info.unlp.edu.ar

***Summary.** Beyond the continuous improvements in physical architectures, one of the greatest challenges is how to exploit their power to the maximum. In recent years, one of the changes with greatest impact has been the massive use of processors that have more than one core (multicore). The combination of several of these processors has produced hybrid platforms (with shared and distributed memory), which created the need to develop operating systems, languages and algorithms that properly used them. In this article, our purpose is to carry out a comparative analysis of the different existing parallel programming models when applied to discrete simulation problems. The solution to the Sharks and Fishes problem is presented as study case, analyzing the efficiency and scalability of the implemented algorithms.*

***Key words:** Parallel and distributed algorithms. Multicore. Multicore cluster. Performance assessment. Simulation.*

## 1. Introduction

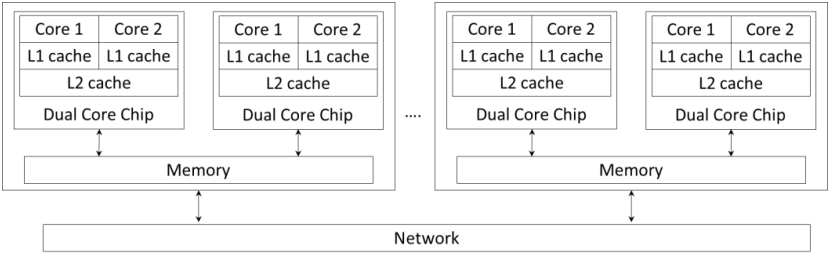
The main purpose of parallel processing is reducing execution time for applications with large processing requirements. This is due to the continuous increase in processing volume and limitations imposed by sequential computing in relation to response times, access to distributed data, and handling the implicit concurrency in real-world problems [1][2].

Parallel processing implies the existence of multiple processors, and their type and organization within support architectures affects the design of the applications required to meet the proposed objective.

Due to the increase in heat generation and energy consumption limits reached by processors, it is possible to combine two or more computing cores within the same chip, which is known as *multicore* processor. Multicore processors can improve application performance if the work to be done is distributed among the available cores [3].

The appearance of the multicore technology has had an impact on clusters (collection of interconnected individual computers working together as a single, integrated computing resource), and led them onto a new stage. A multicore cluster is similar to a traditional cluster, but instead of being formed by monoproductors, it is formed by multicore processors, adding one additional level of communications [4] [5]. In addition to the cache memory shared between pairs of cores and the memory shared among all cores within the same processor, there is the distributed memory that is accessed through the network (Fig. 1).

When writing a parallel algorithm in this architecture, memory hierarchy has to be considered, since it affects the performance that the algorithm can reach. On the other hand, new parallel algorithm programming techniques should be considered that allow efficiently exploiting the power offered by these architectures [6].



*Fig. 1. Multi-core cluster*

Several programming languages and libraries have been developed for explicit parallel programming. They differ mainly in the way in which users see the address space. Models are basically divided into those that provide a shared address space or a distributed one, although there are also hybrid models that combine the features of the two. The different models are detailed below [7].

**Shared memory:** all data accessed by the application are in a global memory that is accessible to all parallel processors. This means that processors can independently retrieve and store data from any memory position. It is characterized by the need of synchronization in order to preserve the integrity of shared data structures [8].

**Message passing:** data are seen as being associated to a specific processor. Thus, communication among processors is required to access remote data. In general terms, when data stored in remote memory are to be accessed, the processor that owns those data must send them to the processor that requires them, which must in turn receive them, and the data are transferred through a channel. In this model, sending and receiving primitives are responsible for handling synchronization [8]. The different options for sending and receiving the data (synchronous or asynchronous), as well as the features of

communication channels (one- or two-way), are reflected in the various communication mechanisms.

Due to the advance of parallel architectures, and especially the appearance of multicore architectures, the **hybrid** programming model is conceived, combining the strategies described above. For instance, a multicore cluster can use the shared memory model for the logical processors within each processor, and the message passing model for communication among physical processors.

This paper presents various parallel solutions (pure and hybrid) to a simulation problem, analyzing solution behavior in terms of efficiency and scalability.

## **2. Description of the Application Problem**

The problem we selected to carry out experiments is the Sharks and Fishes problem. It consists of a simulation of the evolution of sharks and fishes populations in the ocean, following various rules. This problem is representative of a class of biological systems that allow studying the behavior of various populations evolving in time.

In this problem, the ocean is divided in an  $N \times N$  grid, with the right end connected to the left end, and the upper border connected to the lower one (toroidal space). The various squares that form the grid are either empty or occupied by a fish, a shark, or plankton. The population evolves in discrete time steps according to the rules defined for the individuals.

### **2.1 Rules for the Fish Population**

For each time step, a fish moves in one of the adjacent squares that contains plankton (to feed); if found, it occupies this new square and its energy increases according to a simulation parameter. If the neighboring squares are occupied (by other fish or sharks), it remains in its previous position.

One fish cannot reproduce if it cannot move. If a fish reaches its reproduction age with a new time step, it reproduces by leaving a new fish in the square it was occupying before moving; the age of this new fish is 0 and it has half the energy the initial fish had. The fish die when they do not have enough energy (energy decreases with time).

### **2.2 Rules for the Shark Population**

For each step, if a shark finds a fish in any of its adjoining squares, it moves to that square to eat the fish, so after moving it will have gained a certain amount of energy and it will be occupying the square where the fish was. If there are no fish around it, but it finds an empty square (water), the shark moves to that square. Otherwise, it does not move. Sharks eat fish only.

When they do not have enough energy, they die (energy decreases as time goes by).

### **3. Description of the Implemented Sequential Solution**

A cellular automaton was used to solve the problem in a traditional or sequential way. The ocean was represented by means of a matrix of structures or records that keeps its status at each of its positions.

The algorithm follows some simple steps. First, the matrix is initialized with the necessary numbers of sharks and fish in accordance with the constants defined for simulation. Then, the matrix is run through each time step, invoking a function that analyzes and randomly chooses the behavior to be followed by the fish or shark, depending on the status of the adjoining squares and the action it wishes to carry out. This function verifies the eight squares that surround the current position, and chooses one random movement among all possible movements for the action to be performed.

### **4. Description of the Implemented Parallel Solutions**

#### **4.1 Shared Memory Solution**

In this section, the mechanism used for the implementation of the shared memory model with the OpenMP tool is detailed.

To implement this solution, the matrix was divided into horizontal strips (or groups or rows) of equal size for each thread. As a consequence of this type of divisions, there can be conflicts in the boundaries of fragments, since there would be two threads working in adjoining squares and the original operation of the algorithm in each particular square needs to interact with the eight adjoining squares.

To prevent such conflicts, the runs on the various matrix fragments were parallelized as follows; each fragment of the matrix has three regions.

First, the upper part, formed by the  $N$  elements that are “shared” with the thread preceding the current one. This means that, if the identifier of the current thread is  $i$ , the thread with which this region is shared will be the thread  $i-1$  or, if the number of threads is  $th$  and this is the first thread (whose identifier is 0), the data will be shared with the thread whose identifier is  $th-1$ . Then, there is a central part that includes all squares or elements that belong to the thread. These will be modified only by the thread to which they were assigned. Lastly, the lower part of the fragment contains, the same as the upper part, squares that are shared but in this case with the thread that comes after the current one (the thread whose identifier is  $i+1$  or 0 in the case of the thread whose identifier is  $th-1$ ).

Therefore, there will be as many fragments as threads are created, and they will take turns to work on the various regions. For any given time step in the simulation, there will be three processing stages. Initially, each thread works on the upper part of its region, making any necessary changes in the lower part of the previous thread. Later, during a second phase, the central part of each submatrix is processed by the corresponding thread. Finally, the lower part is processed, with the possibility of introducing changes to the upper part of the following thread. This mechanism is repeated for each time step of the simulation, allowing the execution throughout the matrix, so that, for any grid fragment and at any given point in time, only one thread can be carrying out the necessary modifications.

## **4.2 Message Passing Solution**

This section presents an alternative implementation that uses message passing to solve the problem. In this case, the implemented solution divides the matrix into horizontal strips (the same as the shared memory solutions), taking into account the storage mode used. Adjoining horizontal storage space is exploited so as not to have any delays when building the data structures that will be sent.

As regards the assignment of fragments to processes, it was decided that matching one fragment to each process would be a good alternative since, if there is only one fragment throughout the simulation, only boundary update messages would have to be exchanged.

As already mentioned, there might be conflicts at matrix segment boundaries between processes. To solve this, processes were synchronized by sending information through message passing (MPI library). The operation of the algorithm uses a process circular communication scheme that uses a two-way message communication method. Based on the number of available processors, the matrix is divided evenly among them.

There are 2 types of processes – the first one, or process 0, distributes the initial matrix, executes the simulation in its matrix fragment, and compiles the end result (matrix obtained when the algorithm finishes its operation). All other processes receive their portion of the matrix and communicate with their neighbors through messages, sending the necessary fragments for the operations that are carried out at the boundaries.

At a first stage, the upper fragment and the independent segment of the submatrix corresponding to each process are processed. Then, shared fragments are sent and received, and processing continues until the last shared portion has been processed. At this point, there is a new exchange of boundaries and the following steps in the simulation are then executed.

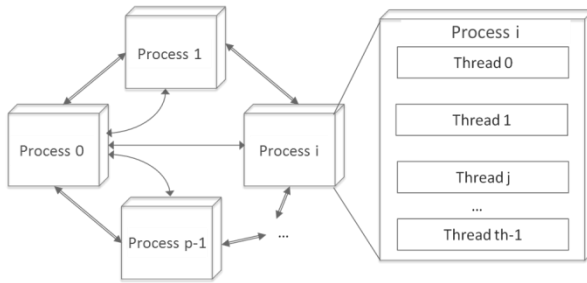
### 4.3 Hybrid Solution

In order to better exploit the architecture and improve services, a solution combining the shared memory and message passing models is proposed. This solution combines MPI with OpenMP.

For its implementation, the data distribution mode and the communications required among processes and threads were assessed. A decision was made to partition the data as with the MPI algorithm, so each process is assigned a block of rows from the original matrix. Also, to enable cooperation among the threads within each of the processes, the data division schema used by the shared memory solution was applied, each thread being assigned a submatrix, with the difference that this time they will receive a fragment corresponding to a portion of the process.

When working with hybrid algorithms, communication occurs among the threads of the various nodes in the architecture. As seen in Figure 2, there is a group of processes where each process communicates with its predecessor and successor, and where process 0 communicates the initial data by means of collective communication statements, then carries out its processing, and receives the end result. The purpose of communication between processes is exchanging the corresponding boundary rows, as with the MPI algorithm.

Within each process there is a set of threads that solve the time steps of the simulation for the process submatrix they belong to (see Figure 2). Due to the hybrid nature of this communication model, two types of threads are required for this solution.



*Fig. 2. Hybrid solution structure*

On the one hand, there are “frontier threads” that can communicate with their adjoining processes. These include the initial threads, which are in charge of synchronizing computation with the other threads within their process but also use message passing to communicate with the last thread of the previous process (in the case of process 0, it communicates with process p-1). Frontier threads also include the end threads, which interact with the initial thread of the following process to send and receive matrix boundaries (in the case of process p-1, this interaction is with process 0) and which also communicate with the other threads within the same process.



There is a second type of threads, the “intermediate threads,” which interact only with other threads within their process (in Figure 2, threads 1 and j). Similar to the shared memory solution, they carry out the simulation by alternating processing in their corresponding regions.

During a first stage, process 0 distributes the initial data and the other processes receive these data. Then, each process creates and initializes the necessary structures for the threads that will carry out the simulation. In a new phase, the threads work to complete the time steps by communicating and synchronizing with one another. Finally, the threads are destroyed and the results of the simulation are compiled.

## 5. Tests Carried Out

For the tests, we used a blade with 8 blades, each with 2 quad core Intel Xeon e5405 2.0 GHz processors. Each blade has 2 Gb RAM memory (shared between both processors) and 2 x 6Mb L2 cache for each pair of cores.

To analyze the behavior of each of the parallel solutions that were implemented for this work, their execution time, speedup, and efficiency were measured. Scalability is also analyzed from three points of view: by varying the size of the ocean (1024x1024; 2048x2048; 4096x4096; 8192x8192), modifying time elapsed during the simulation (1024; 2048; 4096; 8192 time steps), and by using various numbers of processes and/or threads (4, 8, 16 and 32 total cores). These combinations are grouped in 16 test scenarios, which are detailed in Table 1.

**Table 1.** Test scenarios.

Scenario	Time step	Ocean size
1	1024	1024
2	1024	2048
3	1024	4096
4	1024	8192
5	2048	1024
6	2048	2048
7	2048	4096
8	2048	8192
9	4096	1024
10	4096	2048
11	4096	4096
12	4096	8192
13	8192	1024
14	8192	2048
15	8192	4096
16	8192	8192

### 5.1 Pure Solutions

In this section, the efficiencies obtained with the solutions detailed in sections 4.1 (OpenMP) and 4.2 (MPI) are compared. Figure 3 shows the differences in

efficiency for the 16 test scenarios using 4 threads/processes, depending on the case. Similarly, Figure 4 shows the same comparison for 8 threads/processes.

As the chart on Figure 3 shows, when the number of threads/processes is not large, the efficiency achieved by both algorithms is generally similar, with OpenMP being slightly better (in most of the cases).

Also, as problem size increases (ocean size and simulation time), the results obtained with MPI approach those of OpenMP, and in some cases they even improve them.

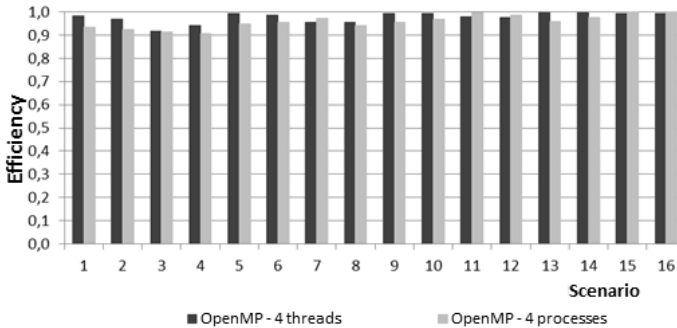


Fig. 3. Efficiency obtained with the different pure solutions for 4 threads/processes.

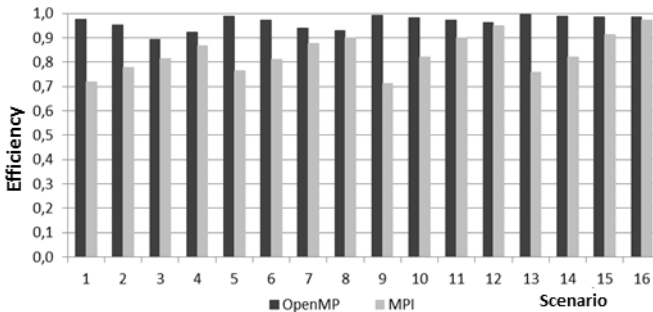


Fig. 4. Efficiency obtained with the different pure solutions for 8 threads/processes.

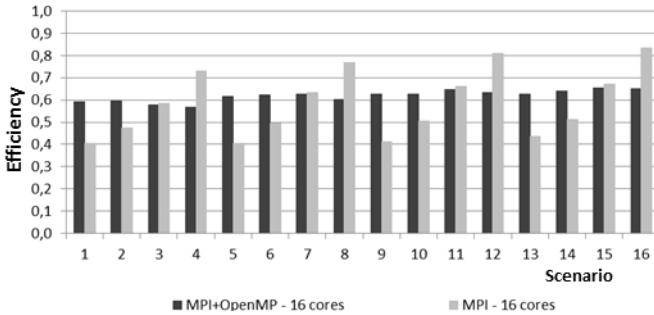
As the number of threads/processes increases, the difference in the efficiency achieved by the algorithms becomes more apparent in favor of OpenMP, as seen in Figure 4. In all scenarios, this algorithm behaves better than MPI.

Taking these results into account (especially when using a larger number of threads/processes) where the OpenMP solution is more efficient, together with the inability of running this solution in more than one blade of the architecture, the possibility of improving performance by means of a hybrid algorithm becomes more relevant. Thus, the enhanced efficiency offered by OpenMP could be exploited in each multicore, accompanied by the communication possibilities offered by MPI.

## 5.2 Solutions for a Hybrid Architecture

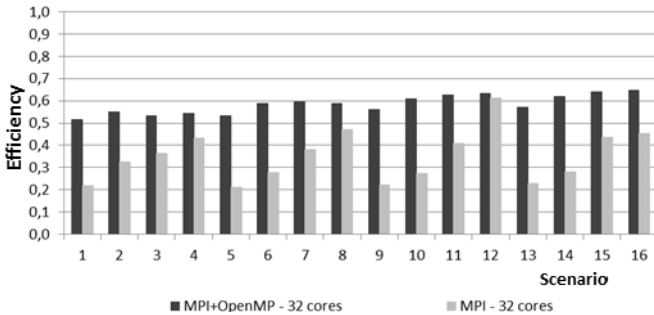
In this section, the efficiencies obtained with the solutions that can be used in hybrid architectures (multicore cluster) are compared. In our case, these are the MPI solution (detailed in Section 4.2) and the hybrid solution combining MPI+OpenMP (detailed in Section 4.3).

Figure 5 shows a graphic representation of the efficiencies achieved by both algorithms for the 16 scenarios and using 16 cores. Similarly, Figure 6 shows the same comparison for 32 cores.



*Fig. 5. Efficiency obtained with the hybrid solution and message passing for 16 cores.*

As shown in Figure 5, it can be observed that the hybrid algorithm keeps a more stable efficiency regardless of problem size (both ocean and simulation time), and it is generally above the efficiency achieved by MPI. On the other hand, the Message Passing (MPI) algorithm increases its efficiency as ocean size increases, and eventually overpasses the efficiency obtained with the hybrid solution in tests with larger oceans.



*Fig. 6. Efficiency obtained with the hybrid solution and message passing for 32 cores.*

With larger architectures, the advantage of the hybrid algorithm is more notorious, as shown in Figure 6. In this case, the efficiency achieved with the hybrid solution is always better than that of the MPI solution.

## 6. Conclusions and Future Work

In this paper, research and development activities with parallel algorithms on current multicore support architectures (multicore and multicore cluster) were carried out, applicable to high computational demand simulation problems.

The study case chosen was a known problem, namely, the Sharks and Fishes problem, whose algorithmic pattern is representative of other scientific computation problems.

Various parallel solutions that used various communication models were proposed: shared memory (OpenMP), message passing (MPI), and hybrid combining both (MPI+OpenMP). Also, the performance obtained with each of these solutions was assessed, comparing the “pure” solutions versus those that can work on hybrid architectures.

When comparing the “pure” solutions, the efficiency achieved with the shared memory algorithm is (in general) better than that obtained with message passing, and this difference increases as the number of threads/processes increases. This is partly because the shared memory levels of the architecture can be efficiently exploited, taking into account that this type of problem requires extensive memory use, thus avoiding the explicit communications and implicit synchronizations required with message passing.

If the hybrid solution is compared to those that use message passing, it can be seen that, as the number of cores used increases, the efficiency decreases with both algorithms. However, this decrease is greater with MPI than with the hybrid algorithm. These results indicate that the hybrid solution behaves better than the message passing solution.

As future work, the scalability study will be expanded to larger problems and heterogeneous architectures in order to analyze the impact of process and data mapping to processors. Finally, the strategies and results obtained will be extended to other scientific computation problems with similar characteristics.

## References

1. Barney B, “Introduction to parallel computing”, Lawrence Livermore National Laboratory, 2010.
2. Wilkinson B, Allen M, “Parallel Programming. Techniques and Applications Using Networked Workstations and Parallel Computers”, 2<sup>nd</sup> Edition, Pearson Prentice Hall, 2005.
3. Mc Cool M, “Programming models for scalable multicore programming”, 2007, <http://www.hpcwire.com/features/17902939.html>.
4. Chai L., Gao Q., Panda D. K., “Understanding the impact of multi-core architecture in cluster computing: A case study with Intel Dual-Core

- System”. IEEE International Symposium on Cluster Computing and the Grid 2007 (CCGRID 2007), pp. 471-478. 2007.
5. Chapman B., “The Multicore Programming Challenge, Advanced Parallel Processing Technologies”; 7th International Symposium, (7th APPT'07), Lecture Notes in Computer Science (LNCS), Vol. 4847, p. 3, Springer-Verlag (New York), November 2007.
  6. Rauber T, Rünger G, “Parallel programming for Multicore and Cluster Systems”, Springer, 2010.
  7. Grama A, Gupta A, Karypis G, Kumar V, “Introduction Parallel Computing”, Pearson Addison Wesley, 2<sup>nd</sup> Edition, 2003.
  8. Dongarra J. , Foster I., Fox G., Gropp W., Kennedy K., Torzcon L., White A. “Sourcebook of Parallel computing”. Morgan Kaufmann Publishers - ISBN 155860 871 0 (Chapter 3).



# PH-Helper: a Syntax-Directed Editor for Hoshimi Programming Language

MARIANO LUZZA, MARIO BERÓN, MARIO PERALTA, CARLOS SALGADO

Departamento de Informática - Facultad de Ciencias Físico-Matemáticas y Naturales  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - San Luis - Argentina  
{mluzza, mberon, mperalta, csalgado}@unsl.edu.ar

***Abstract.** Teaching programming is not a simple task. This is mainly because of two problems: i) The teacher must address the traditional problems of teaching-learning process; and ii) Overcome the disadvantages provided by the application domain and the programming language.*

*It is clear that if the application domain is not appealing, the student will not be motivated to solve the problems posed by the teacher. Furthermore, programming languages generally use syntax and an idiom that students have difficulty understanding. For these reasons, the development of tools to make easier the teaching of programming represents a significant challenge for educational computer professionals.*

*In order to overcome the drawbacks previously exposed, PH-Helper, a tool aimed at facilitating the teaching of programming, was developed. PH-Helper tries to reach this goal through: i) An attractive domain for teaching programming: the Project Hoshimi; ii) The extension of Hoshimi language with actions that simplify programming; and iii) The definition and implementation of a visual, proactive and syntax-directed editor. This editor allows the student to concentrate only in the problem to solve by removing obstacles sparsely related with the problem solving process.*

***Keywords:** DSL, syntax-directed editor, Project Hoshimi.*

## 1. Introduction

For a long time, computer scientists have searched an easier way to teach programming [14]. On the one hand, the causes that makes so difficult for the student to learn the basics of programming were investigated. In this context, one of the main sources of information was the previous courses that the student has had taken. On the other hand, more languages for teaching were designed, whose main objective is to facilitate the learning of programming concepts. Examples of such languages are *Logo* [5, 6, 7], *Scratch* [8, 9],

Design Languages, among many others. However, despite the efforts of the scientific community, the problem is not yet resolved; students cannot assimilate in a clear and simple way, programming concepts. In addition to the aforementioned drawback are other problems that have to deal with students, like how to understand the problem statement and write the program to solve it. It is clear that both problems can be addressed with the guidance and teaching of the teacher. However, other drawbacks appear closely linked to programming languages as the use of keywords in English and syntax (such as punctuation symbols corresponding to the structures of sentences). The difficulty related to the idiom of the language can be avoided through the use of visual languages. The obstacle presented by the syntax can be overcome using the Syntax-Directed Editors [10]. These editors are aware of the syntax or language constructs and therefore provide significant help to the beginner when expressing an algorithm in a programming language.

Another problem faced by teachers of computer science, is the problem statement itself, this includes the context, the application area and objectives. Many times the proposed problems are not sufficiently motivating for students and / or results are not "tangible". That's why this issue (the choice of interesting and effective problems) must also be taken into account to achieve a successful educational process.

The Hoshimi Project [11], which will be introduced in Section 4, is another attempt to mitigate the problems mentioned in the preceding paragraphs.

The paper is organized as follows. Section 2 and 3 introduce the concepts of Domain Specific Language and Syntax-Directed Editor respectively. Section 4 presents the Project Hoshimi, an attractive game for teaching programming. Section 5 describes PH-Helper a front-end for the Project Hoshimi. Section 6 presents a study case where it can be appreciated the use of PH-Helper solving typical programming problems. Finally, Section 7 presents the conclusions of this article.

## **2. Domain Specific Language**

A Domain Specific Language (DSL) [1] is a reduced set of structures and operations that provide greater expressiveness and optimization for a particular domain. According Hudak [2] a DSL is the "ultimate abstraction", which precisely captures the semantics of an application domain.

Some well-known DSL include SQL and regular expressions among others. Clearly, each one is better than a general-purpose language to represent database operations and chains respectively. But they aren't when trying to describe solutions outside their domain. Certain industries also have their own DSL. For example, in telecommunications, languages of call description are widely used to specify the sequence of states in a phone call. In the travel industry, it's usually needed a language to describe flight bookings. Other



areas where DSL are also used include route planification, navigation of a website, connection diagrams for electronic components, a family tree, etc.

When designing a DSL, the concepts and relationships relevant to the domain [4, 12] to which such language will be designed, must be clearly specified. For example, to build a family tree is necessary to represent family relationships between different people. From here you can extract "Person" and "Relationship" as important concepts.

DSL users create models which are instances of the DSL [13, 17, 18]. Following the example of the family tree, a model could describe several generations of a particular person or just the actual family relations.

The advantage of DSL is that by focusing on the domain, programs are described in terms that domain users handle [3]. In this way, end users are those who build the programs, without being necessarily programmers. In this paper we use and extend the Project Hoshimi DSL, an interesting game for teaching programming.

### 3. Syntax-Directed Editor

There are two types of syntax-directed editors [10, 19]: reactive and proactive. The *reactive* editors know the syntax but allow the developer to write what he wants. While the programmer is writing or after performing this task, the editor somehow highlights keywords and code structures. *WinEdt*, a powerful [La] *Tex* document editor for Windows, is an example of this kind of editors. WinEdt has highlighting schemes that can be customized for different modes and a spell checker for several languages. Another feature that reactive editors often provide is *autocomplete*. *Autocomplete* is the ability to complete words as they are typed, according to the context. This feature greatly improves productivity while reducing errors. Today many editors bring *autocomplete* implemented; one of the best known is Microsoft Intellisense. Intellisense not only knows the syntax of the language, but it is also sensitive to the semantics and recognizes all variables declared until the moment of its use.

*Proactive* editors are more interesting because they know the syntax and provide guidance for the developer when they type. At each time of writing a program, the editor knows what constructs are applicable and can be written. In this way, the editor helps the developer suggesting the variable or the value that he is looking for, among many other features. Clearly, the editor is very helpful, especially for those who do not know the whole language syntax. Importantly, these tools can be automatically generated from the grammars.

It is noteworthy that the tool described in Section 5 is a proactive editor.

## 4. Project Hoshimi

Project Hoshimi was created in 2005 by Richard Clark for the Imagine Cup, a Microsoft Competency. Project Hoshimi is a .Net technology-based game, particularly focused in object-oriented language C#. The basic idea is to program tiny robots (Nanobots) to cure diseases that affect human tissue (game scenario). The students' task is to program the strategies for the different Nanobots, through a set of simple and intuitive actions. Clearly, this game is attractive yet challenging enough.

Project Hoshimi (PH) is a game that aims to promote the creative use of programming languages and other tools. PH is useful for: i) Bringing young people to information technology through a pleasant game; ii) Conceptualize programming as a creative task through a strategic simulation; and iii) Teaching both the very basic concepts of programming and logic, and object-oriented paradigm with .Net technology.

Overall, PH is a game whose main objective is to cure human diseases, although there are other secondary objectives such as moving to different points or keeping alive some nanobots. These objectives are accomplished with different nanobots, who receive their orders from a special nanobot named *NanoAI* (Artificial Intelligence). The behavior of each nanobot is described by a strategy developed by the student with .Net technology. The reader interested in learning more about the game can study the work made by Luzza et al. [16] or analyze the material available at [15].

## 5. PH-Helper

Currently, the Project Hoshimi has two forms for developing strategies:

1. **Using the built-in visual editor:** This is a dedicated editor that presents the problem of providing low expressiveness, for example:
  - i) It has no actions to declare and use variables;
  - ii) Although it allows the use of a conditional sentence, it has very few options for developing the predicates;
  - and iii) The editor it is not extensible.
2. **Using an editor and a .Net compiler:** This way allows overcoming the limitations mentioned in the previous item, but requires the student to master the concepts of object-oriented programming. In addition, the student must deal with common errors related to programming languages, such as those explained in the previous sections, which hinder learning.

Thus PH-Helper arises, a visual editor that has the simplicity of the first approach, and has the potential and flexibility of the second mode. The application language is the Hoshimi language, a DSL whose domain is used

in the game. All the basic commands that already existed in the Hoshimi language are kept in PH-Helper. However, some of them have been improved and / or extended. First, the commands related to flow control were eased. For example, iteration now allows logic expressions as a condition. In the previous version an integer must be specified. This number was employed to indicate the number of iterations. The decision statement was also improved (command similar to an if-then-else) following a scheme similar to the iterations. This feature increases the level of expressiveness of that sentence because more complex expressions can be incorporated. In addition, new features were added, the most important are related to working with variables. In PH-Assisted is possible to define variables, change their value using an assignment action and use them as parameters for other commands. The available types are those involved with the domain, such as "Point" (pair  $\langle x, y \rangle$  for symbolizing a location) and "Integer" (commonly used to quantify) among others. It is important to remark that the assignation action has a visual mapping that assists in building the operation. On the one hand, it is specified the left side of the assignment. First the type of the variable is selected from a list and then the variable is selected from another list filtered by the selected type. This particularity avoids syntax errors for both the type and for the name. On the other hand, it is specified the right side of the assignment with a tool that allows the construction of expressions. This tool has three tabs: i) *Variables*, it allows to select from a list of variables filtered by taken into consideration the type expected by the expression; ii) *Functions*, it allows to select a function from a list of functions filtered by considering the return type expected by the expression; and iii) *Constants*, it allows to specify literal constant. In the case that a function is selected, the process is repeated until the expressions for the parameters are build. Otherwise the process is ended. It is important to remark that the characteristics described above have not only the effect to avoid syntax errors but also other semantic errors as type errors, etc.

The Assisted Expressions Construction Process (AECPP) mentioned in the preceding paragraph, is reused in all parameterization processes of all the commands used in the strategy. This ensures that the all expressions are valid at construction time. While the extensions previously described are sufficient to improve the development, the user may need some more, which are in the API of the game but not in the editor. For this reason, it is possible to extend the functions and variables / properties that can be used in the AECPP. This task can be carried out by editing the corresponding xml file (Functions.xml and Variables.xml) and adding the elements needed. The process involved in the tool extension and some examples can be studied in the work of Luzzza et al. [16].

## 5.1 Strategies

In PH-Helper, a strategy is represented by a set of tabs. Each tab represents a nanobot and has an actions panel to specify the nanobot's behavior. The behavior is represented by a tree, where the root is the nanobot itself. Each action can be parameterized accordingly. The initial action allows parameterization of the nanobot's name and other characteristics. Also it is possible to declare variables to be used later in the parameterization of other actions.

## 5.2 Compilation

Once the strategy is created in the editor, only remains to compile it to obtain the final assembly (which will be used by the game). The compilation process scans twice all root nodes of the strategy. The first time searches for errors. If errors are discovered the process ends. If not, action trees are once again scanned, this time for each action the method "GetCode" is invoked. In both cases, the invocation is made recursively on the child nodes and so on until the leaf nodes are reached. The process ends with all the error-free code collected from the strategy. This code is then used for two purposes: 1) Compile it into a dll assembly and 2) Save it as source code to facilitate the passage from Hoshimi language to C #.












## 6. Study Case

A regular exercise in the game is calculating the closest entity to a nanobot. This is similar to basic programming exercises which seeks the greater or the lesser number of an array. In this domain, the calculation is made based on the distance between an entity and a nanobot. Another difference is that a dynamic list is used instead an array. These are very common in the game and are easy to use due to the encapsulation. The procedure involves the following steps:

1. Establish a minimum reference distance, in this case it is valued in 400 because the map has 200 x 200 points
2. For each point in the list of desired points
  - a. Calculate the distance between the point and the nanobot
    - i. If the distance is less than the minimum then: i) Update the minimum distance; and ii) Save the point
3. Show the result

The algorithm can be solved using the nanobot AI. This nanobot is added by default (and is mandatory) when a new strategy is created with PH-Helper

(File → New). All commands used are in the toolbar on the left of the actions panel. The translation of the above algorithm expressed in pseudo-code to Hoshimi language can be done with the following steps:

1. On the root action , the parameterizations panel (right panel) is used to add two integer variables (“dist” for the minimum distance and “aux” to storage intermediate results) and two point variables (“ptoCercano” to storage the nearest location and “pto” for auxiliary work).
2. Add a new state , where the nearest point calculation will be done.
  - a. Set “dist” value to 400 for minimum reference distance using an assignment action .
  - b. Iterate the list of desired points (for example “ListaHP”, which is provided by the system) using the variable “pto”. Both specifications can be set in the parameterization panel for the “Iteration” action  which is similar to a “foreach” statement.
    - i. Assign  to “aux” the distance between “pto” and the nanobot’s location (“Ubicación\_AI” which means AI’s location in English). This operation can be done using the “Distance” function under the “Functions” tab in the AACP.
    - ii. Evaluate if the calculated distance (aux) is less than the reference (dist). This is done with a decision action  and by specifying in its parameterization the function “Menor” (“Less than” in English) with “aux” and “dist” as parameters.
    - iii. If the proposition turns to be true , then the reference distance and the nearest point must be updated using assignment actions.
  - c. Once the iteration is finished, the control is transferred to the next state to inform the result. This task is carried out through the execution of a “Change State” action . In this action the target state can be specified.
3. Add a second state  to report the result.
  - a. Write a message to announce the result. This is implemented with a “Write” action . This action allows strings creation with AACP help. In this case, the “Constant” tab is used to write “El punto más cercano es:” (“The nearest point is:” in English).
  - b. Write the nearest point location. This value is stored in the variable named “ptoCercano”. So to write its value a “Write” action  is used once again. But this time a “Point

to Text” function (shows the point in string notation) is used with the variable as parameter.

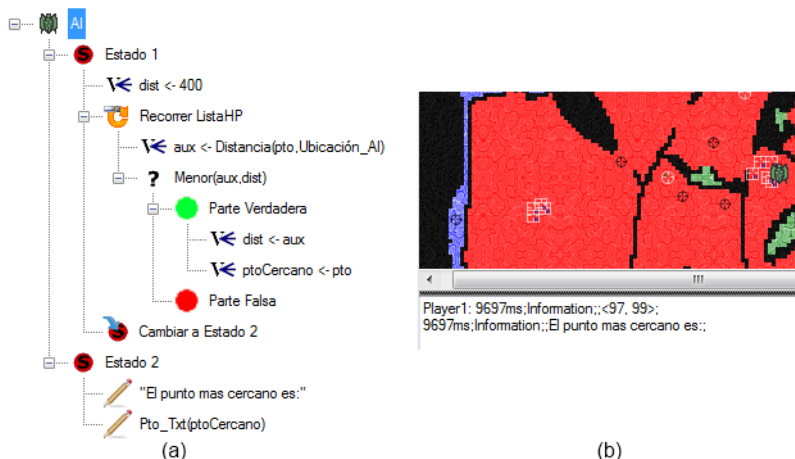


Fig. 1. (a) Program in Hoshimi language and (b) its corresponding result

Figure 1a shows the program developed in language Hoshimi using PH-Helper to solve the problem.

The compilation process generates: i) The C# source code for the algorithm (see Figure 2); and ii) A dll assembly file that can be used in the game to run the strategy and verify the solution. In Figure 1b the result of executing the strategy in the game can be seen.

```
public partial class MiEquipo : Player, IAccion{
    public void Accion(){
        switch ( __ESTADO){
            case "Estado 1":
                dist = 400;
                foreach (Punto __Item in ListaHP){
                    pto = __Item;
                    aux = Distancia(pto, Ubicación_AI);
                    if (Menor(aux, dist)){
                        dist = aux;
                        ptoCercano = pto;
                    }
                    else{}
                }
                __ESTADO = "Estado 2";
                break;
        }
    }
}
```

```

    case "Estado 2":
        MiUtils.Escribir("El punto mas cercano es:");
        MiUtils.Escribir(Pto_Txt(ptoCercano));
        break;
    }//fin switch
} //fin acción
string ____ESTADO = "Estado 1";
public Entero aux = 0, dist = 0;
public Lista<Punto> miLista = new Lista<Punto>();
public Punto pto = default(Punto), ptoCercano = default(P
unto);}

```

*Fig. 2. C# source code generated by PH-Helper*

## 7. Conclusion

In this article PH-Helper was presented, a tool that combines syntax-directed editor with domain-specific language: Project Hoshimi language, a game useful for teaching programming. PH-Helper tries to solve two major difficulties that arise at the time of teaching programming to students, such as: i) Find an entertaining and challenging topic; and ii) Overcome the problems presented by the syntax and idiom of traditional programming languages. To achieve this goal, PH-Helper implements a visual editor that assists the user in constructing syntax error-free sentences. Furthermore, it also uses and improves Hoshimi language. The enhancements to the language are related to the introduction of new commands. These commands allow working with variables, selection, iterations, etc., thus facilitating the task of programming. The previously mentioned features allow the student to: i) Focus on learning concepts of logic and programming; and ii) To avoid the distraction caused by common errors such as lack of punctuation symbols or the use of keywords that are not in their natural idiom. PH-Helper also creates a bridge between the game and learning OOP, since for each program written in the Hoshimi language, the equivalent C# source code is generated. This peculiarity allows the advanced student see the relationship between: i) The program code written in the language of the game; and ii) The object-oriented code that actually runs. In this way the student can go internalizing basic constructions provided by object-oriented languages.

The future work is oriented in two main directions. The first is the development of different tests to measure the teaching improvements when using PH-Helper for this task. The second focuses on applying the strategy used in the creation of PH-Helper in new editors. Some of these editors will be for programming languages seen in introductory programming subjects.

These languages are often seen as pseudo-code and therefore lack an execution scope environment to verify the results.

## References

1. A. van Deursen, P. Klint, J. Visser. Domain-Specific Languages: An Annotated Bibliography. ACM Sigplan Notices, Vol. 35, No. 6, 2000.
2. P. Hudak. Building domain-specific embedded languages. ACM Computing Surveys, 28(4es), December 1996.
3. D. Wile. Lessons learned from real DSL experiments. Science of Computer Programming, Vol 51, Issue 3, pages 265-290, 2004.
4. Krzysztof Czarnecki and Ulrich W. Eisenecker, Generative Programming, Addison-Wesley, 2000, ISBN 0-201-30977-7.
5. Seymour Papert. Mindstorms: children, computers, and powerful ideas. Basic Books, Inc., New York, NY, USA, 1980.
6. Daniel Watt. Learning With Logo. McGraw Hill, 1983.
7. MollyWatt and DanielWatt. Teaching with Logo: Building Blocks for Learning. Addison-Wesley Pub, 1986.
8. J. Maloney, M. Resnick, N. Rusk, B. Silverman, and E. Eastmond. The scratch programming language and environment. ACM Transactions on Computing Education, 10(4):1–15, 2010.
9. M. Resnick, Y. Kafai, and J. Maeda. A networked, media-rich programming environment to enhance technological fluency at after-school centers in economically-disadvantaged communities. MIT Media Laboratory, Proposal to National Science Foundation (Information Technology Research), 2003.
10. Raul Medina-Mora. Syntax-Directed Editing: Towards Integrated Programming Environments. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1982.
11. Javier Gonzalez Sanchez, Ramiro A. Berrelleza Perez, and Maria Elena Chavez Echeagaray. Introducing computer science with project hoshimi. In Companion to the 22nd ACM SIGPLAN conference on Object-oriented programming systems and applications companion, OOPSLA '07, pages 908–914, New York, NY, USA, 2007. ACM.
12. M. Mernik, J. Heering, T. Sloane. When and How to Develop Domain-Specific Languages. ACM Computing Surveys, 37(4es), December 2005.
13. T. Kosar, P. Martinez, P. Barrientos, M. Mernik. A preliminary study on various implementation approaches of domain-specific languages. Inf. Softw. Technol., Vol. 50, No. 5, 2008.
14. S. Georgantaki, S. Retalis. Using educational tools for teaching object oriented design and programming. Journal of Information Technology Impact (Jiti), 7(2):111–130, 2007.
15. Gaming.Net. Available online on <http://www.pilp.edu.ar/PILPAsp/Paginas/Pagina.asp?PaginaPilpId=22>. Seen in May 2012.



16. M. Luzzi, M. Berón, P. Henriques. PH-Helper - a Syntax-Directed Editor for Hoshimi Programming Language, HL. SLATe'12, pages 71-89 Symposium on Languages, Applications and Technologies, 2012.
17. S. Goldrei. The Design, Implementation and Use of Domain Specific Languages. 2007.
18. J.C. Cleaveland. Building application generators. IEEE Software, 1988.
19. T. Teitelbaum. The cornell program synthesizer: a syntax-directed programming environment. SIGPLAN Not., 14(10):75–75, Oct. 1979.



# ParasitePics: An Animal Parasitology Image Repository Prototype for Teaching and Learning

SABRINA L. MARTORELLI<sup>1</sup>, CECILIA V. SANZ<sup>1</sup>, JAVIER GIACOMANTONE<sup>1</sup>,  
SERGIO R. MARTORELLI<sup>2</sup>

<sup>1</sup>III LIDI, School of Computer Science. National University of La Plata, La Plata, Buenos Aires, Argentina. {smartorelli,csanz,jog}@lidi.info.unlp.edu.ar.

<sup>2</sup>CEPAVE (CONICET). National University of La Plata, La Plata, Buenos Aires, Argentina. sergio@cepave.edu.ar.

**Summary.** *This article presents the design of an animal parasitology image repository prototype (ParasitePics) that offers educators and researchers a space to share cases and experiences from ad-hoc material, so as to enrich the learning and development processes in this area. This development is part of a research process on discipline-specific microscopic image storage, search, and retrieval possibilities from relevant metadata. Thus, work has been a joint collaboration with experts and professionals, so as to define a metadata schema and then incorporate the proposed prototype. The schema is based on existing standards, but taking into account the specific aspects of the case. Field work is detailed, as well as the results obtained, and the first version of ParasitePics is presented. Finally, the conclusions and future lines of work are outlined.*

**Keywords:** *Microscopic Image Repositories, Animal Parasitology, Metadata Schemes, Storage, search and retrieval of microscopic images.*

## 1. Introduction

The constant development of Information and Communication Technologies (ICTs), and their use in the educational process, is part of the current learning scenarios. So much so, that there are increasingly more educational options that use ICTs as mediators for the teaching and learning process. This is why we need computer tools and applications that accompany and build on these processes, and offer new possibilities for distance-work scenarios [1].

In recent years, in the field of Human and Animal Pathology and Parasitology, there has been an increasing interest in working with digital images obtained with robot microscopes and digital cameras, as well as with specific software, called virtual microscopes, that are useful for viewing the images obtained. This combination is widely used in hybrid (b-learning) or extended classroom courses [2].

In Animal Parasitology research, the use of digital images obtained from histological sections or full animals, and the subsequent use of virtual

microscopes to view them, is starting to widely spread, since several researchers can review and share cases through images.

For these digital images to be accessed by students and educators or shared among colleague researchers, they need to be stored and then searched for and retrieved. It is at this point that metadata schemas and controlled vocabularies become important, since they are great aides in these tasks. Even though there are metadata standards that allow characterizing different web resources, the particular labels that are required in each discipline need to be analyzed in order to tailor the standard or redefine some elements so that resources can be searched for and retrieved in a manner that is appropriate for the filed in question.

These are the reasons behind the analysis of the metadata that are required to characterize virtual slides for Animal Parasitology, as well as for the implementation of a specific image repository prototype, called ParasitePics, that can be used by any educator and/or researcher working in these topics.

## **2. Field Work: Analyzing the Needs of the Target Group.**

Since the main topic covered by this article is intended for professionals from a specific field of Biology, namely, Parasitology, a methodology that allowed interacting with these experts was required. To this end, the steps carried out and the tools involved in this field work are described below.

As a first step, a survey was prepared and sent to the students attending a post-graduate course at the Faculty of Natural Sciences and Museum of the UNLP<sup>1</sup>. Based on the results obtained, a first review of expectations, interests, and usefulness of virtual microscopy for this discipline was made. All surveyed students expressed interest in working with virtual microscopy and, therefore, the use of repositories that store the images used by these microscopes. However, they knew only a few free image repositories and virtual microscopes.

Then, related theoretical concepts linked to metadata[3][4][5][6], controlled vocabularies[7][8][9], and languages for metanotes[10][11][12], among others, were analyzed. Similarly, some of the existing metadata standards were reviewed, such as Dublin Core [13][14][15][16] [17][18], LOM [19][20], and others [21][22][23]. This was followed by an analysis of currently existing and accessible virtual microscopy systems. A series of criteria were selected to review the various systems that were studied, such as type of license, functional features, metadata used for storage, search and retrieval, specificity for Animal Parasitology, and so forth. This stage and its results, however, will not be described in detail in this paper. It will only be mentioned that it showed that there is a lack of specific repositories for

---

<sup>1</sup> "Parasites and Pathogens Decapod Crustaceans with Commercial and Ecological Significance," implemented in the second semester of 2009.

Animal Parasitology, and that the types of searches that are offered by these systems do not cover criteria that are specific to this discipline.

This was the starting platform for the third stage, which was focused on obtaining information about the metadata that are considered to be of greatest interest to characterize target images.

Among the various techniques that were considered to obtain this information, the survey was initially selected due to the type of information involved and the availability of the group [24][25][26]. After this, individual interviews were carried out with the most experienced experts in order to go deeper into specific issues.

The group was formed by fifteen people that are either directly or indirectly related to the field. Even though this is not a sufficiently representative sample, it has allowed obtaining the required information from a group of Parasitology-related experts, and it has opened the door to a subsequent assessment of the prototype that was developed.

As part of the survey mentioned above, a table with label categories was presented for the experts to review based on their image characterization and identification needs. These categories are described below:

***The image is a histological section:*** data related to the content of an image. Specific for images that contain histological sections of hosts or parasites.

***The image contains an “in toto” parasite or organism:*** data related to the content of an image. Specific for images that contain full parasites or sections of parasites.

***General description of the image:*** data related to aspects of the image that are not content-related, such as size, author, compression, etc.

***About the microscope used to generate the image:*** technical data of the microscope and functionality when the image was taken.

***About the digital camera used to generate the image:*** technical data of the camera used to take the image.

Also, the experts could add other data that were not included in the table, as well as include general comments.

### 3. Decisions Made Based on the Results Obtained

The field work was followed by the process of designing and implementing an image repository prototype specific for Animal Parasitology. The information obtained while reviewing available microscopy systems and repositories was taken into account, as well as the results of the surveys and interviews. At this point, a decision was made to include all elements that had at least 60% of affirmative answers in the main survey as appropriate for the characterization of images (metadata) in this discipline. As a result, of the 32 initial potential data included in the survey, 22 were preselected.

Then, after pre-selecting these data, an in-depth review of each was made through the interviews carried out with the most experienced experts. The key reasons for this review were finding similarities and differences between the terms chosen, and establishing related vocabulary.

Finally, changes and modifications to the list of pre-selected data were introduced on light of these expert interviews, and a final list of 19 selected data was obtained, as well as a list of relevant vocabulary.

#### **4. Metadata Proposal for Representing Digital Microscopic Images Used in Animal Parasitology.**

The schemas and metadata that will be used to create the proposal of metadata to represent digital microscopic images used in Animal Parasitology are based on the data selected.

In order to implement the schemas and their metadata, the recommendations of the W3C “Describing and retrieving photos using RDF and HTTP [27]” have been used as starting point. In these recommendations, metadata are divided into three different schemas: 1. Dublin Core schema, which uses the Dublin Core schema in RDF format. 2. Technical schema, which captures the technical data of the photo and camera; and 3. Content schema, which is used to categorize the subject of the photo by means of controlled vocabulary.

Also, a schema was defined to represent each of the same classes into which the data in the main survey had been divided and presented. Thus, the schemas defined are: Image Schema, Histological Section Schema, Parasite Schema, and Microscope/Camera Schema.

Both the data used to describe the image in general, and those used to describe the microscope or camera can be used regardless of the content of the image.

In the case of images of histological sections, the data corresponding to the “Histological Section Schema” will be highly significant. In the case of images showing parasites, they will be represented by data corresponding to the “Parasite Schema”. In the case of images that represent a section of a parasite, both the “Histological Section Schema” and the “Parasite Schema” can be used for identification and/or characterization purposes. For the representation proposed here, RDF notation and RDFS schema were used.

It should be noted that when reviewing Dublin Core data, as done in the following recommendation, a decision was made to include some additional data that are useful for the general identification of an image. These data are: Content (represented through dc.subject), describing the main content of the photo. Possible values for this datum are: Histological Section, Parasite, or Parasite Section, all images belonging to one of these. The other data included are Type (dc.type), which will always be labeled as "image" as per standard specifications; Title (dc.title), which provides a brief description of the content of the photo; and Rights (dc.rights), which contains information about legal rights affecting the use of the resource. With these last additions, the final data that are within the four schemas mentioned above are those shown in Figure 1.

<b>Image Schema</b>	<b>Image Format</b>	Format of the image.
	<b>Image Size</b>	image size in pixel
	<b>Author of the image</b>	Details of the author of the image
	<b>Main content</b>	Main content of the image
	<b>Resource Type</b>	It will always be "Image"
	<b>Title</b>	Title that serves to identify or describe the image
	<b>Right</b>	Information on legal rights on resource use
<b>Histological Section Schema</b>	<b>Source</b>	Anatomical part of the body from which is obtained the histological section
	<b>Organism Type</b>	<b>Organism</b> type from which is obtained the histological section
	<b>Section type</b>	Orientation of the histological section performed on the sample.
	<b>Staining used</b>	Dying or staining used in the section.
	<b>Development Stage</b>	Stage of development of the organism from which is obtained the histological section
	<b>Diagnostic</b>	Description of injury or parasite identification.
<b>Parasite Schema</b>	<b>Parasite name</b>	Scientific or common name of the parasite.
	<b>Host name</b>	Scientific or common name of the host.
	<b>Place host capture</b>	Geographical coordinates where the host was obtained
	<b>Host Capture Date</b>	Date where the host was obtained
<b>Microscope /Camera Schema.</b>	<b>Brand and Model of the microscope</b>	Brand and model of microscope used for image acquisition.
	<b>Type of microscope</b>	Type of microscope used for image acquisition.
	<b>Lighting type</b>	Type of lighting used for image acquisition
	<b>Objective magnification</b>	Objective magnification used for image acquisition
	<b>Brand and model of the camera</b>	Brand and model of camera used for image acquisition.

*Figure 1. Final data table.*

Based on the recommendations used as starting point, it is proposed that each element in the DC standard corresponding to the Image Scheme is represented by means of an RDF property using the labels `rdfs:label` and `rdf:comment` to describe title and content, respectively, in Spanish. The label `rdfs:subPropertyOf` is used to indicate which type of element from the standard is referred to by each datum. In the case of the Image Schema, all elements in the original DC schema are defined, regardless of their being used. This is in accordance to the recommendations used as starting point.

To represent the Histological Section, Parasite, and Microscope/Camera schemes, an RDF is used for each of them. Then, each datum that is part of each schema is represented as a property, similar to what is done with the Image Schema. The label `rdfs:domain` is used to refer to the schema to which the datum (property) belongs.

Figure 2 shows an example of metadata created from the schema defined.

```

<?xml version='1.0' encoding='ISO-8859-1'?><rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:p=" www.parasitepics.com.ar/EsquemaParasitePics#"
<rdf:Description rdf:about="">
<p:title> Sagittal section of Palaemon
macrodactylus </p:title>
<p:creator> Dr. Sergio R. Martorelli</p:creator>
<p:subject>Histological Section</p:subject>
<p:format> image</p:format><p:format.extent> jpg
</p:format.extent>
<p:rights>There are no rights defined on this
image </p:rights>
<p:origin>Cephalothorax</p:origin>
<p:organism>Decapod Crustaceous </p:organism>
<p:section>Sagittal</p:section>
<p:stain>Hematoxylyn and Eosin </p:stain>
<p:development>Adult </p:development>
<p:diagnosis>Normal cephalothorax tissues.
</p:diagnosis>
<p:microscope> Olympus BX41.</p:microscope>
<p:microscope-type> Compound Optical
</p:microscope-type>
<p:magnification> 20 X </p:magnification>

```

*Figure 2. Metadata created from the schema.*

## 5. Image Repository Prototype: ParasitePics

ParasitePics (“Parasite” + “Pics” = Images) is presented as the initial version of an Animal Parasitology image repository prototype whose main purpose is facilitating the search, storage, and subsequent retrieval of relevant images by means of the metadata schema created based on the field work and research carried out.

This repository is developed as a free access website available at <http://www.parasitepics.com.ar>.

The architecture used for developing the repository is of the Client-Server type. For the development on the client side of the repository, HTML 4, CSS, and JavaScript languages were used. On the side of the server, Web Apache server was used, with PHP as language to process requirements and format the results obtained from the database, and MySql as database engine and database query language.

For this initial version, the repository is hosted on a private server with Linux operating system, 1-GB memory, and 250-GB hard drive. The system can be installed in Apache server version 2.4.1, PHP 5, and MySQL 5.0.

To access and use the repository, a PC with Internet connection and a web browser installed is required.

The system supports two types of users: anonymous user and registered user. An anonymous user is any person accessing the site through a web browser in a PC connected to the Internet. A registered user is any person who, as



anonymous user, registers in the system to access the functionalities for sharing and managing images within the repository.

## 5.1 Sharing Images

To share an image, after logging into the system, a form must be filled in to identify the image based on the data that describe it. It should be noted that these data are directly related to the set of metadata that has been presented. The data that are entered will be the input for the metadata schema that will be created and associated to the image in the repository. Once the image has been stored, it can be retrieved through download (in its original format) and the data that are included in the metadata schema can be checked. Image metadata can be edited, or the image can be removed. Currently, the images that can be shared must be 50Mb at most and the supported image formats are: jpeg, jp2, tiff, and bigtiff.

Figure 3 shows the page that allows sharing images through the form and the other functionalities corresponding to image management.

Título	Miniatura	Detos	Acciones
Metacercaria recién ingerida de Microphallus		Ver detalle	Modificar Eliminar
Cinta sagital de Palaemon macrodactylus		Ver detalle	Modificar Eliminar
Macrovalvirena argentina		Ver detalle	Modificar Eliminar

*Figure 3. Image management.*

## 5.2 Searching for images

One of the main functionalities offered by the repository, and one that is directly related to the image sharing functionality, is the search option. This can be done following various criteria to help image retrieval. These criteria are based on the schema, which is extremely useful for field experts who

need good image characterization in order to be able to search for images based on their specificity. Any user, registered or not, can access this functionality.

Search categories are the same categories used to characterize images: General Data, Histological Section Data, Parasite Data, and Microscope/Camera Data. Depending on the option selected from the drop-down list "Main Content of the Image," the other categories will be enabled. Once a category is enabled, the data that are included in it can be used to refine the search being carried out.

The option "Show All Repository Images" can be selected to include all images in the results.

Text fields in the form allow special characters, such as double quotes (exact phrases), simple quotes (search for different terms within a datum), and so forth.

All the results obtained from a search are presented as a grid. The images or virtual slides can be downloaded, as well as their metadata.

The image shows two side-by-side screenshots of the ParasitePics website. The left screenshot is the search interface, titled "ParasitePics Repositorio de Imágenes de Parasitología Animal". It features a search bar, a "Buscar imágenes" button, and several filter categories: "Datos Generales de la imagen", "Datos del Parasito", "Datos del Corte Histológico", "Datos de Microscopio y Cámara", and "Mostrar todas las imágenes del Repositorio". The right screenshot shows the search results, titled "ParasitePics Repositorio de Imágenes de Parasitología Animal Resultados de la Búsqueda". It displays a table with columns: "Título de la imagen", "Contenido", "Formato de imagen", "Autor de la imagen", "Tamaño de la imagen en píxeles", "Miniatura", and "Opciones". Three results are visible in the table, each with a thumbnail and a "Ver detalle" link.

Título de la imagen	Contenido	Formato de imagen	Autor de la imagen	Tamaño de la imagen en píxeles	Miniatura	Opciones
Micre. Integres	carina recién de Microphala	PDF	Dr. Sergio R. Martorell	800 x 600 píxeles		Ver detalle
Corte de Falax	señal de primacrostictical	PDF	Dr. Sergio R. Martorell	2412 x 1468 píxeles		Ver detalle
Micre. Integres	Parasito	PDF	Dr. Paula Haronczyk	4000 x 3072 píxeles		Ver detalle

Figure 4. Image search and results obtained.

There is also a feature that allows managing site users.

## Conclusions

There is an increasing number of educational offers that use ICTs as mediators in the teaching and learning process, so it is important that computer tools are available to assist in and improve those processes.

ParasitePics is then presented as the initial version of an Animal Parasitology image repository prototype aimed at covering these needs. It should be noted that the functionalities offered by ParasitePics focus on the use of a metadata schema and specific vocabularies created after a review and constant collaboration with field experts.

Currently, ParasitePics is being assessed by experts in Animal Parasitology. In parallel with this assessment, and in relation to the evolution of the prototype, we are working on new functionalities to be added, such as improving the registration process of expert users in order to filter users based on their expertise in the area. Another improvement that is being developed is related to the possibility of reporting images that do not match their metadata. Also, the possibility of supporting more image sizes and formats is being considered, as well as having an English version of everything. Finally, the possibility of linking the animal shown in the images with specific field databases is being considered.

## References

1. Barberá, Elena (2004) *La educación en la red: actividades virtuales de enseñanza y aprendizaje*. Barcelona: Paidós.
2. Bartolomé, A. (2000) "Innovaciones tecnológicas en la docencia universitaria". Universidad de Barcelona.
3. Swick, Ralph. (2002) *Metadata Activity Statement*. W3C <http://www.w3.org/Metadata/Activity.html>.
4. Durrell, William R. (1985) *Data Administration. A Practical Guide to Data Administration*. McGraw-Hill.
5. Lamarca Lapuente, María Jesús .El nuevo concepto de documento en la cultura de la imagen. <http://www.hipertexto.info/documentos/metadatos.htm>
6. Daudinot Founier, I. (2006) *Organización y recuperación de información en Internet: teoría de los metadatos*. ACIMED.
7. Méndez, Eva. (2006) *Dublin Core, metadatos y vocabularios*. El profesional de la información, v. 15, n. 2.
8. Fast, K., Leise, F., Steckel, M. (2002) *What Is A Controlled Vocabulary? Boxes & Arrows*.
9. Urdiciain, Gil. (2004) *Manual de lenguajes documentales*. Gijon: Trea.
10. Rodil Garrido, Antonio. (2006) *Estudio de los lenguajes de consulta para documentos RDF*. Universitat oberta de Catalunya. Ingeniería en Informática.
11. Dinos Rojas, Juan Larry. (2004) *Arquitectura de un Sistema basado en Agentes para la recuperación de metadatos RDF en base a una ontología de documentos*. Theses submitted in partial compliance of the requirements for the Science Master in Computer Engineering. University of Puerto Rico, Mayaguez Campus.
12. Barrón Cedeño, Alberto. (2005) *Web semántica. Ontologías, una introducción*.
13. Olivé, Enric Peig. (2003) *Interoperabilidad de Metadatos en Sistemas Distribuidos*. Doctoral Thesis. Universitat Pompeu Fabra.
14. Senso, José A; Antonio de la Rosa Piñero. (2004) *Evolución del Dublin Core Metadata Initiative*. Universidad de Granada. Facultad de

- Biblioteconomía y Documentación. Wisdom Information Consultants B.V. Amsterdam. Holland.
15. Hillmann Diane. (2005) Guía de uso del Dublin Core. <http://dublincore.org/documents/usageguide/>
  16. Comité de Metadatos de la Biblioteca Nacional de Chile. (2009) Guía para la creación de metadatos usando Dublin Core.
  17. AABA .Sistema de Información Geoespacial. (2006) Sistema de Información Geoespacial Set de Elementos de Metadatos Dublin Core 1.1.
  18. Calderón Corail Marcela. (2006) Spanish Translation of "Dublin Core Qualifiers" <http://www.ciw.cl/proyectos/calificadoresDC.html>
  19. IMS Meta-data Best Practice Guide for IEEE 1484.12.1-2002 Standard for Learning Object Metadata Revision 2004. [http://www.imsglobal.org/metadata/mdv1p3pd/imsmd\\_bestv1p3pd.html](http://www.imsglobal.org/metadata/mdv1p3pd/imsmd_bestv1p3pd.html).
  20. Comité de Estandarización de Tecnologías Educativas de IEEE (2002) Estándar para Metadatos de Objetos Educativos IEEE Standards Department.
  21. Martínez, José M. (2004) MPEG-7 Overview. Palma de Mallorca <https://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>
  22. Delcor Ballesteors, Jordi; Pérez Noriega, Verónica. (2006) Descripción, indexación, búsqueda y adquisición de secuencias de vídeo mediante descriptores MPEG-7. Universidad UPC.
  23. Vinancos Vicente, Pedro José. (2005) El estándar MPEG-7 .Revista de Ingeniería Informática del CIIRM. [http://www.cii-murcia.es/informas/jul05/articulos/El\\_estandar\\_MPEG-7.php](http://www.cii-murcia.es/informas/jul05/articulos/El_estandar_MPEG-7.php).
  24. García Ferrando. (1992) El análisis de la realidad social. Métodos y Técnicas de investigación. Compilador: Manuel, Madrid, Alianza Universidad.
  25. Sierra Bravo R. (2003) Técnicas de investigación social. Teoría y ejercicios. Madrid: Thomson.
  26. Kerlinger, F. (1997) Investigación del comportamiento. México, D.F.: McGraw-Hill.
  27. Lafon, Yves; Bos, Bert. (2002) Describing and retrieving photos using RDF and HTTP. <http://www.w3.org/TR/photo-rdf>.

# Discovering Group Roles by Analyzing Collaborative Skills

VÍCTOR OZÁN, ROSANNA COSTAGUTA AND DANIELA MISSIO

Departamento de Informática, Facultad de Ciencias Exactas y Tecnologías  
Universidad Nacional de Santiago del Estero (UNSE)  
Avda. Belgrano (S) 1912, Santiago del Estero, 4200, Argentina  
victoraozan@gmail.com; rosanna@unse.edu.ar; daniela\_mmi@yahoo.com.ar

***Abstract.** One of the more effective techniques for student learning is group learning. There is huge research carried out to define the best way of constituting a group. Some of the research done is focused to find out which collaborative skills should be present in students in order to fulfill an efficient group work, while some other is centered in the roles which students should adopt to constitute a group. Both positions aim at characterizing a successful group. The present work integrates both approaches by identifying the collaborative skills which are associated with each of the possible roles that can come along in a learning group. In order to reach this goal a sample of the substantive interactions was taken which were manifested by students in a Computer-Supported Collaborative Learning group when the process of Knowledge Discovery in Databases was applied. The discovered knowledge related to the connections between skills and roles will introduce to study a new way of defining the constitution of learning groups.*

***Keywords:** Computer-mediated Communication, Cooperative/Collaborative Learning, Group Roles, Collaborative Skills, Datamining.*

## 1. Introduction

The quick development of information technology allowed informatics to expand their applications to a wide range of areas. In particular, people at colleges are taking advantage of the opportunities of informatics to enhance teaching and learning processes. As an example, the area of Computer-Supported Collaborative Learning (CSCL) can be brought up, where the student can use distant learning, without geographical or temporal limitations.

Learning groups are defined as a structure consisting of a group of interacting people trying to attain certain learning goals through their participation in the group (Souto, 1990). The mere fact of belonging to a group does not assure successful collaborative learning, neither is granted that the group will work

in a coordinated and efficient way. Success or failure depends, among other things, on the collaborative skills of the participants. Without these skills, group members are incapable of providing the different functions or roles which will allow students to accomplish coordinate work and achieve successful collaborative learning. In order to obtain a coordinated work in a group it is necessary to have a sound balance of roles (Belbin, 2001). Such a balance is only present when the group can count on as high a quantity of roles as possible, and when these roles are not repeated among participants (Belbin, 2001).

Very seldom do teachers analyze the student features at the moment of defining a constitution of learning groups. In general, this fact does not change when the students themselves decide how to constitute the group where they are going to work. Therefore, groups are often constituted with lack of balance, and this fact, in the long run, causes strain and uneasiness among members, and often groups end up in dissolution, or in lack of proficiency. However, if the features of each of the students were taken into account, it would be possible to create groups where participants would use the necessary collaborative skills to fulfill the required roles, thus attaining a sound development of group dynamics.

Considering that roles are a source of behaviors, and these in turn are materialized through the use of collaborative skills, it turns out to be necessary to define an explicit connection between collaboration skills and group roles. This link will allow members to be properly selected and be able to interact and produce a balanced, collaborative, and successful group.

The aim of the present research is to identify the collaborative skills which are associated with each of the possible roles to be played in a learning group. To achieve this goal, skills were adopted according to Soller (2001) proposal, and the group role classification as proposed by Belbin (1981). The present research was the subject of the final work for the graduation of one of the authors, to obtain the certificate as Licensed in Information Systems.

This paper is structured as follows: sections 2 and 3 deal with the collaboration skills taxonomy and the role theory; section 4 describes the Knowledge Discovering Process; section 5 presents relevant antecedents; and in section 6 the conclusions of the work are stated.

## **2. Collaboration Skills**

Soller *et al.* (1996) have formulated the conversation skill taxonomy for Collaborative Learning adapted from McManus (1995) and Aiken's work on Collaborative Skill Network (1995). The taxonomy considers three types of skills (Active Learning, Conversation Skill and Creative Conflict), and the respective sub skills, as well as the corresponding attributes to each sub skill (Table 1).

Skill	Subskill	Attribute	Sentence Opener
Creative Conflict	Mediation	Teacher mediation	<i>"Let's ask the Teacher"</i>
		Conciliate	<i>"Both are right in that"</i>
	Argumentation	Agree	<i>"I agree, because..."</i>
		Disagree	<i>"I do not agree, because..."</i>
		Offering alternative	<i>"Alternatively..."</i>
		Infer	<i>"Then...", "Therefore..."</i>
		Suppose	<i>"If, ...then..."</i>
Doubt	<i>"I am not sure because..."</i>		

**Table 1.** Taxonomy for Collaborative Learning Skills (Soller, 2001)

Skill	Subskill	Attribute	Sentence Opener
Active Learning	Motivation	Encourage	<i>"Very well"</i>
		Reinforce	<i>"That's correct"</i>
	Information	Rephrase	<i>"In other words..."</i>
		Lead	<i>"I think we should ..."</i>
		Suggest	<i>"I think..."</i>
		Elaborate	<i>"To elaborate.. " "Besides..."</i>
		Explain	<i>"Let me explain in this way ..."</i>
		Justify	<i>"To justify..."</i>
		Assert	<i>"I am reasonable sure..."</i>
	Request	Information	<i>"Do you know...?"</i>
		Elaboration	<i>"Can you tell me more?"</i>
		Clarification	<i>"Can you explain how/why?"</i>
		Justification	<i>"Why do you think that"</i>
		Opinion	<i>"¿Do you think...?"</i>
Conversation	Acknowledgment	Illustration	<i>"¿Please show me..?"</i>
		Appreciation	<i>"Thank you"</i>
		Accept/Confirm	<i>"That's right" "Yes"</i>
	Maintenance	Reject	<i>"No"</i>
		Request Attention	<i>"Listen to me..."</i>
		Suggest Action	<i>"¿Would you please...?"</i>
		Request	<i>"¿Is that right?"</i>
		Confirmation	
	Task	Listening	<i>"I see what you're saying"</i>
		Apologize	<i>"Sorry"</i>
Coordinate group process		<i>"Let's go on", "Are you ready?"</i>	
Request focus change		<i>"Let me show you..."</i>	
Summarize information		<i>"In order to summarize..."</i>	
Ending the participation	<i>"Goodby"</i>		

**Table 1.** Taxonomy for Collaborative Learning Skills (Soller, 2001) (Cont.)

### 3. Team Role Theory

In learning groups, participants are engaged based on acquired knowledge and work to attain a common goal, namely, the construction of knowledge. In these groups, peers are related to each other and exhibit behaviors, so it is possible to determine roles by identifying their behaviors. Such roles are influenced by individual personality as much as by learned behavior, experience, values and motivations, external factors and mental skills. Belbin (1993) defines the role as the trend of the individual to behave, contribute and relate with peers in a certain way within the group.

In 1981, Belbin proposed the Team Role Theory, adding flexibility to the group structure with more individual responsibility in the definition and evaluation of roles. Belbin established a nine-group roles theory (Table 2)

Role	Description
Plant	Creative, imaginative, unorthodox. Solves difficult problems.
Coordinator	Mature, confident. A good chairperson. Promotes decision-making. Delegates well.
Monitor- Evaluator	Sober, discerning and strategist. Sees all options. Judges accurately.
Implementator	Disciplined, reliable, conservative and efficient. Turns ideas into practical actions.
Completer Finisher	Painstaking, conscientious, anxious. Searches out errors and omissions. Delivers on time.
Resource investigator	Extrovert, enthusiastic, communicative. Explores opportunities. Develops contacts.
Shaper	Challenger, dynamic person, thrives on pressure. Has the drive and courage to overcome obstacles.
Team worker	Co-operative, mild, perceptive and diplomatic. Listens, builds and averts friction.
Specialist	Single-minded, self-starting, dedicated. Provides knowledge and skills in rare supply.

**Table 2.** Belbin Roles

### 4. Knowledge Discovery Process Application

Knowledge Discovery in Databases (KDD) implies the performance of three steps: pre-processing, data mining and post-processing (Hernández Orallo *et al.*, 2004), which are briefly explained below.

The pre-processing stage aims at preparing data to be tackled in the next stage of the process. Among the pre-processing techniques Han & Kamber's work (2001) is worth mentioning: data cleaning, data integration, data transformation, and data reduction. Data mining stage can be defined based on a set of primitives specially designed to facilitate an efficient and fruitful discovery of knowledge. Such primitives (Han & Kamber, 2001) include: the specification of database portions or the set of data with which it is intended to work; the kind of knowledge to be discovered; the existing knowledge



which could be useful to guide the KDD process; the interesting metrics to perform pattern evaluation on the analyzed data; and in the end, the shapes in which knowledge could be visualized. The post-processing stage implies the execution of some type of reformulation of the obtained results by means of the performed data mining.

In particular, for this research, the KDD application consisted in the performance of the activities which will be described in the following section.

#### 4.1 Pre-processing

In this stage, interactions between 15 college students in the area of informatics were collected. The students worked on programmed activities in a CSCL environment, trying to solve problems in a collaborative way. The students were randomly assigned to the groups. The content of the interaction segments were analyzed, and each student's intervention was re-codified according to the collaboration skills proposed by Soller (2001). Afterwards, the different roles played by the participants were recognized and each dialogue segment was categorized as is shown in the list of roles proposed by Belbin (1981). Since the collected material during the analysis had been stored in different formats, three types of files were obtained where the CSCL work sessions were stored (Figure 1). What follows is a brief description of the fields and attributes present in the various tables of Figure 1.

- *Contribution*: Set of words or expressions used by a peer of the group to refer to his/her point of view at a certain moment.
- *Sub skill*: It specifies the sub skill of the collaboration skills as proposed by Soller, expressed by a peer and reflected in the performed contribution.
- *Attribute*: It expresses the attribute connected to the sub skill, according to Soller's categorization (2001).
- *Sentence Opener*: Expression used in a contribution which allows an attribute to be assuredly identified.
- *Role*: It specifies the played role of a peer at a certain moment. This role is directly related to the performed contribution and belongs to Belbin's categorization (1981).
- *Sub skill and attribute*: It is a combination of a sub skill and an attribute in the same field.

*Type 1 File (Text file)*

Contribution	Subskill and Attribute	Role
Contribution 1	Sub skill-Attribute 1	Role 1
...	...	...
Contribution <i>n</i>	Sub skill-Attribute <i>n</i>	Role <i>n</i>

*Type 2 File (Spreadsheet)*

Subskill	Attribute	Sentence Opener	Contribution	Role
Subskill 1	Attribute 1	Sentence Opener 1	Contribution 1	Role 1
...	...	...	...	...
Subskill <i>n</i>	Attribute <i>n</i>	Sentence Opener <i>n</i>	Contribution <i>n</i>	Role <i>n</i>

*Type 3 File (Text file)*

Participant	Contribution	Sub skill and Attribute	Role
Participant 1	Contribution 1	Sub skill- Attribute 1	Role 1
...	...	...	...
Participant <i>n</i>	Contribution <i>n</i>	Sub skill- Attribute <i>n</i>	Role <i>n</i>

**Fig. 1.** Different types of files for interaction sentences storage

Since interaction sentences were stored in files with different formats (spreadsheets and text files) it was necessary to generate storage with a unique data format, integrating all the available information in a spreadsheet file, keeping all the existing attributes. Then, a content analysis was done to evaluate the possible need of introducing changes in the structure. First, all inputs with errors or inconsistencies were eliminated. Then, the field ‘Sub skill and Attribute’ was split into two, one named ‘Sub skill’ and the other ‘Attribute’. This change affected all the previous entries in type 1 and type 3 files. And finally, the ‘Contribution’, ‘Sub skill’, ‘Sentence Opener’ and ‘Participant’ attributes were eliminated because they were irrelevant for the targeted discovery process. So at the end of the pre-process stage, a minable dataset was obtained, consisting in a single spreadsheet file where all the interaction sentences corresponding to the carried out collaborative sessions were included. In each contribution the expressed collaboration skill attribute was kept and also the role of the corresponding group. Finally, since WEKA software was applied, it was necessary to change the spreadsheet format to an .arff extension file format (Witten & Frank, 2000).

#### 4.2 Processing

This stage was to determine the type of application task that would correspond to the descriptive type, since the target was to identify non-explicit relations between categorical attributes, namely, between collaboration skills and group roles manifested by students.

Considering the targeted type of knowledge to discover and the model or technique to apply for data processing, the Association Rules methodology was selected. For this reason WEKA software was used, applying association rules with the ‘a priori’ algorithm. For this purpose a *support value* and a *confidence value* were established. Support value is the minimum percentage of instances covered by the rule, in reference to the total of instances. Confidence value shows the accuracy of a rule when classifying the register entry to which is applied. For this particular research with WEKA the support value was 0.6

(MinMetric), and the confidence value was 0.001 (LowerBoundMinSupport). The result of the processed file is shown in Table 4.

<b>N°Inst</b>	<b>Attribute</b>	<b>Role</b>	<b>Coverage</b>	<b>Confidence</b>
190	Process Coordination	Coordinator	190	conf:(1)
118	Information	Resource Investigator	118	conf:(1)
60	Appreciation	Team Worker	60	conf:(1)
47	Opinion	Coordinator	47	conf:(1)
45	Request Confirmation	Coordinator	45	conf:(1)
34	Clarification	Resource Investigator	34	conf:(1)
34	Apologize	Team Worker	34	conf:(1)
32	Encourage	Coordinator	32	conf:(1)
30	Elaborate	Implementator	30	conf:(1)
29	End Participation	Coordinator	29	conf:(1)
27	Lead	Coordinator	27	conf:(1)
15	Reject	Completer Finisher	15	conf:(1)
11	Suppose	Coordinator	11	conf:(1)
8	Illustration	Resource Investigator	8	conf:(1)
6	Teacher Mediation	Coordinator	6	conf:(1)
6	Conciliate	Coordinator	6	conf:(1)
3	Disagree	Completer Finisher	3	conf:(1)
3	Request focus change	Plant	3	conf:(1)
176	Acceptance	Monitor-Evaluator	175	conf:(0.99)

**Table 4.** Results obtained with WEKA software

<b>N°Inst</b>	<b>Attribute</b>	<b>Role</b>	<b>Coverage</b>	<b>Confidence</b>
80	Assert	Specialist	79	conf:(0.99)
71	Doubt	Completer Finisher	70	conf:(0.99)
53	Justify	MonitorEvaluador	52	conf:(0.98)
51	Suggest Action	Impulsor	50	conf:(0.98)
50	Elaboration	Coordinator	49	conf:(0.98)
120	Suggest	Monitor-Evaluator	117	conf:(0.98)
32	Justification	Resource Investigator	31	conf:(0.97)
42	Summarize Information	Coordinator	40	conf:(0.95)
18	Offer Alternative	Plant	17	conf:(0.94)
16	Reinforce	Shaper	15	conf:(0.94)
16	Listen	Coordinator	15	conf:(0.94)
84	Explain/Clarify	Specialist	77	conf:(0.92)
18	Request Attention	Resource Investigator	16	conf:(0.89)
53	Deduce	Monitor-Evaluator	43	conf:(0.81)
48	Agree	Coordinator	35	conf:(0.73)
23	Rephrase	Monitor-Evaluator	14	conf:(0.61)

**Table 4.** Results obtained with WEKA software (Cont.)

### 4.3 Post-processing

Considering the result of the previous stage it was possible to discover different knowledge. On the one hand, a certain connection was found

between roles and collaborative skills (Table 5). It was also possible to calculate the percentage with which a specific role was manifested through a specific collaboration skill during the group dynamical performance (Table 6). An example for this issue is the ‘plant’ role, in which a 75 % of the occurrences were identified through the ‘creative conflict’ skill and a 15% through the ‘conversation’ skill. This example shows the case where a skill indicator occurs more frequently when certain role is played.

## 5. Related works

Considering all the reviewed research works, it is possible to assert that some of the works are targeted to analyze collaboration skills and some to study the roles played in learning groups. For instance, Henry & Stevens (1999) introduce a controlled trial demonstrating how useful is the consideration of group constitution team roles, as proposed by Belbin. Soller (2001) points out that a group needs to have participants with a set of collaborative, acquired and naturalized skills, in order to create a successful collaborative learning environment.

Napier & Hasler-Waters (2003) have demonstrated that when paying attention to collaborative skills in online groups, several successful groups can be found where the participants are able to receive necessary support, establish proper communication, increase confidence, and achieve adequate organization.

Role	Skill	Subskill	Attribute	Coverage	Confidence
Plant	Conversation	Task	Request focus change	3	conf:(1)
	Creative Conflict	Argumentation	Offer Alternative	17	conf:(0.94)
Team Worker	Conversation	Acknowledgment	Appreciate	60	conf:(1)
		Maintenance	Apologize	34	conf:(1)
Coordinator	Conversation	Task	Coordinate Process	190	conf:(1)
			End Participation	29	conf:(1)
		Summarize	Information	40	conf:(0.95)
		Confirmation	45	conf:(1)	
	Maintenance	Listen	15	conf:(0.94)	
		Motivate	Encourage	32	conf:(1)
	Active Learning	Inform	Opinion	47	conf:(1)
			Request	Elaboration	49
		Inform	Lead	27	conf:(1)
		Argumentation	Agree	35	conf:(0.73)
Creative Conflict	Mediation	Suppose	11	conf:(1)	
		Conciliate	6	conf:(1)	
Specialist	Active Learning	Inform	Teacher Mediation	6	conf:(1)
		Inform	Assert	79	conf:(0.99)
Completer Finisher	Conversation	Acknowledgment	Explain/Clarify	77	conf:(0.92)
Implementator	Active Learning	Inform	Reject	15	conf:(1)
			Argumentation	Disagree	3
	Conflict		Doubt	70	conf:(0.99)
			Elaborate	30	conf:(1)

Shaper	Conversation	Maintenance	Suggest Action	50	conf:(0.98)
	Active Learning	Motivate	Reinforce	15	conf:(0.94)
Resource Investigator	Conversation	Maintenance	Request Attention	16	conf:(0.89)
			Information	118	conf:(1)
			Clarification	34	conf:(1)
			Illustration	8	conf:(1)
	Active Learning	Request	Justification	31	conf:(0.97)
Monitor Evaluator	Conversation	Acknowledgment	Acceptance	175	conf:(0.99)
			Justification	52	conf:(0.98)
		Inform	Suggest	117	conf:(0.98)
	Active Learning		Rephrase	14	conf:(0.61)
		Creative Conflict	Argumentation	Infer	43

**Table 5.** Collaboration skills which are significant for each role

Winter & McCalla (2003) have successfully identified some of the individual factors which contribute to success of the group, and other factors which affect the group performance. Heckman & Misiolok (2005) introduce an empirical study exploring the leader behavior in online student teams. The authors establish certain relationship between leader patterns and the different results obtained in these groups. Joan Anton Ros Guasch (2006) tried to establish a connection between competences and roles, but was not successful, in spite of the fact that both theories have many points in common. Finally, Costaguta *et al.* (2011) introduced a multi-agent model which performs the personalized training of the collaboration skills used by the students during CSCL sessions. Nevertheless, antecedents were not found in which a study had been made showing that team roles and collaboration skills are connected, which emphasize the quality of this work.

Role	Skill	No. Occurrences	Percentage
Plant	Conversation	3	15%
	Creative Conflict	17	85%
Team Worker	Conversation	94	100%
	Conversation	319	59,96%
Coordinator	Active Learning	155	29,14%
	Creative Conflict	58	10,90%
Specialist	Active Learning	156	100%
	Conversation	15	17,04%
Completer Finisher	Creative Conflict	73	82,96%
	Active Learning	30	100%
Implementator	Conversation	50	76,92%
	Active Learning	15	23,08%
Resource Investigator	Conversation	16	7,73%
	Active Learning	191	92,27%
	Conversation	175	43,64%
Monitor Evaluator	Active Learning	183	45,64%
	Creative Conflict	43	10,72%

**Table 6.** Percentage of occurrences of skill types related to group roles

## 6. Conclusions

The application of KDD processes allowed certain connections between group roles and collaborative skills to be established, which was manifested during the student work dynamics. It should be outlined that even though each collaborative skill and an associated sub skill are found to be connected only with one role, not in every case the relation has a confidence level of 100 %. This is the case in the ‘rephrase’ attribute of the ‘active learning’ skill, which is connected to the ‘Monitor Evaluator’ role with a confidence level of 61 %. Besides, certain connections found in this work have a low number of occurrences and therefore the confidence level of 100 % obtained cannot be firmly accepted. This happens, for instance, with the ‘Request Focus Change’ requirement and the ‘Plant’ role, having only 3 occurrences. This fact is indicating that it is necessary to perform a much deeper analysis of the trial result. Therefore, the obtained result should be considered as an initial mapping between different roles and collaborative attributes which are usually found in the collaborative work dynamics of learning groups. In addition, and with the aim of obtaining conclusions with solid statistical value, it should be convenient to increment the number of the considered interactions in a KDD process, increasing the number of groups and the number of evaluated instances.

Nevertheless, it is considered that the knowledge discovered by means of this work is an important contribution to the group theory and particularly to the area of learning groups. A future application of this outcome could be to optimize the student learning process, offering a new way of constituting learning groups. With this approach, group members should be selected considering the personal skill of each student to play a certain role, paying attention to the student development level of collaboration skills connected to each role. Presently, this is one of the working lines of the authors, under the frame of the research project: “Web Information Systems Based on Agents to Promote Computer-Supported Collaborative Learning” (UNSE Code 23/C115), to be executed in period 2012-2015.

## References

1. Belbin, R.M. (1981) *Management teams: Way they succeed or fail*. Oxford: Butterwoth-Heineman.
2. Belbin, R.M. (1993) “A reply to the Belbin Tems-Role Self\_Perception Inventory By Fumham, Steele and Pendelton”. *Journal of Ocupational and Organizational Psychology*, Vol. 166 (3).
3. Belbin, M. (2001) *Managing without power: gender relationships in the story of human evolution*. Oxford: Butterworth-Heinemann.
4. Costaguta, R., García, P., y Amandi, A. (2011) “Using Agents for Training Students Collaborative Skills”, *IEEE Latin America Transactions*, Vol. 9 (7).

5. Hernandez Orallo, J., Ramirez Quintana, L., & Ramirez, N. (2004) "Introducción a la Minería de Datos." Editorial Pearson Educación, S.A. Madrid.
6. Heckman, R. & Misiolek, N. (2005) "Leaders and Followers In Student Online Project Teams". System Sciences HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on System Sciences.
7. Henry, S. & Stevens, T. (1999) "Using Belbin's Leadership Role to Improve Team Effectiveness: An Empirical Investigation". Journal of Systems and Software, Vol. 44 (3).
8. Napier, W. & Hasler-Waters, L. (2003) "Building Team Collaboration in the Virtual Classroom". Journal of Educational Perspectives. College of Education/University of Hawaii at Manoa, Vol. 35 (2).
9. Soller, A. (2001). "Supporting Social Interaction in an Intelligent Collaborative Learning System". International Journal of Artificial Intelligence in Education, Vol. 12 (1).
10. Guasch, R. & Anton, J. (2006) "Análisis de roles de trabajo en equipo: Un enfoque centrado en comportamientos" Tesis doctoral. Departamento Psicología Social, Universidad Autónoma de Barcelona
11. Souto, M. (1990) Didáctica de lo grupal. Ministerio de Educación y Justicia. INPAD.
12. Winter, M. & McCalla, G. (2003) "An Analysis of Group Performance in Terms of the Functional Knowledge and Teamwork Skills of Group Members". Proceeding of Workshop on User and Group Models for Web-based Collaborative Environments, 9<sup>th</sup> International Conference on User Modeling (UM 2003).





# Voice Recognition as Interaction Paradigm for People with Motor Difficulties

M. LUCRECIA MORALEJO<sup>1,2</sup>, CECILIA V. SANZ<sup>2</sup>, PATRICIA PESADO<sup>2,3</sup>  
{LMORALEJO, CSANZ, PPESADO}@LIDI.INFO.UNLP.EDU.AR

<sup>1</sup>Research Fellow, UNLP (Universidad Nacional de La Plata), Buenos Aires, Argentina.

<sup>2</sup>III-LIDI, Facultad de Informática, UNLP, La Plata, Buenos Aires, Argentina.

<sup>3</sup>CIC (Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires), Argentina.

***Abstract.** This article corresponds to the field of human-computer interaction (HCI), and it analyzes some of its various paradigms applied to the special education scenario. Thus, a background review is presented. Then, the use of the HCI paradigm based on voice commands is described for the development of an adaptation of the educational software JClíc, so that it can be used by users/students with motor disabilities that do not affect—or barely affect— speech development. For this application, various voice recognition (VR) engines were studied, and an in-depth analysis of the VR engine Sphinx-4 was carried out. In this paper, we present part of this work and the details pertaining to the implementation of a prototype (JClícVoice) that is responsible for the adaptation to JClíc. Finally, results are described and conclusions are drawn, after assessing the prototype.*

***Keywords:** Special Education, Human-Computer Interaction, JClíc, Sphinx, Voice Commands.*

## 1. Introduction

Technology development helps people carry out their daily activities, and even more so in the case of people with disabilities, since the technologies and technical tools offered help them lead more autonomous and independent lives. However, it is also true that the elements required for this technology to be useful for the highest possible number of people should be taken into account. To this end, there is a constant need to review existing techniques and resources, and search for new ones, to prevent them from becoming communication, information or accessibility barriers. That is, technology should be developed following universal design patterns, favoring true social inclusion. This is relevant because, as stated by Grau [1], technologies have been traditionally conceived, projected, produced and applied with the average person in mind, disregarding, or considering to a lesser extent, the

differences derived from disabilities, which sometimes forces the introduction of technical adaptations or using additional resources for the technologies to be usable by certain people or under certain situations [2].

## **2. Human-Computer Interaction**

The field of Human-Computer Interaction (HCI) is the discipline that studies the interaction between human beings and computer systems. Its main purpose is improving this interaction by making computer systems more usable, so that the productivity of those using them to work improves. The Association for Computing Machinery (ACM) [3] defines HCI as “the discipline in charge of the design, assessment and implementation of interactive computing systems for human use, and the study of what surrounds these systems.”

Thus, it considers HCI as a science that analyzes both aspects, human and computer, combined. This is one of the main reasons why different approaches are used to consider HCI based on the science. From a human standpoint, HCI is complemented by other disciplines such as psychology, cognitive sciences, communication sciences, graphic and industrial design, and so forth. From the computer and machinery standpoint, it includes computer graphics, operating systems, programming languages, and environment development.

At the Center for the Study of Digital Libraries [4], a conceptual model of HCI with four elements is proposed: (a) people, the users of the system, (b) the task, various steps to be carried out to perform one or more activities, (c) the environment, physical, organizational and social aspects of the environment, and (d) technology, any device with which interaction takes place.

Unlike environment aspects and task, interaction between people and technology occurs through a somewhat implicit component: the interface. This is formed by several components, including hardware interfaces—keyboard, mouse, touchpads, pens, etc— and software interfaces, such as the graphic user interface (GUI).

## **3. HCI Paradigms**

HCI paradigms are aimed at satisfying the need for interfaces that feel as natural as possible for human beings. With time and as technology evolved, an increasing number of interaction paradigms have arisen. On the other hand, certain ways of interaction that were considered as independent paradigms are now unified under a single paradigm. A brief summary of some interaction paradigms [5] follows, reviewed as part of this research work.

*Vision as means for interaction (vision-based HCI).*

Taking into account the scope of the applications and the diversity of open problems and approaches, researchers have tried to tackle different aspects of human responses that may be recognized as a visual signal. Some of the most significant research areas in this section are the following:

- Facial expression analysis.
- Body movement tracking.
- Gesture recognition.
- Gaze detection (tracking the movement of the eyes).

*Audio as means for interaction (audio-based).*

This paradigm is based on the use of sounds as medium to give or receive instructions to and from computer systems.

The research areas in this section can be divided as follows:

- Voice recognition
- Speaker recognition.
- Auditory analysis of emotions.
- Detection of human-produced noise/signals.
- Interaction with music.

*Sensors as means for interaction (sensor-based).*

This is a combination of various areas with a wide range of applications. The common feature of these various areas is that there is at least one sensor between the user and the machine to provide interaction.

- Pen-based interaction (as in the case of mobile devices).
- Mouse and keyboard.
- Joysticks.
- Motion tracking sensors and digitizers.
- Haptic sensors.
- Pressure sensors.
- Taste/smell sensors

*Multimodal Interaction.*

In addition to the types of interactions listed above, there are also the so-called multimodal interactions (MMHCI). The term "multimodal", refers to the combination of multiple modes of interaction.

An interesting aspect of multimodality is the collaboration of the different types of interaction to assist in recognition. For instance, tracking lip movement (vision-based) can help voice recognition methods (audio-based), and voice recognition methods (audio-based) can help acquire gesture recognition commands (vision-based).

*Ubiquitous computing (Ubicomp).*

Environmental intelligence or ubiquitous computing, which is referred to as the Third Wave, is trying to integrate technology to the environment in a way that is more natural and invisible at the same time.

The design and location of these devices should be specially conceived for the intended interaction task. Thus, computing is no longer in one single location, but rather "diluted" in the environment.

*Background of various HCI paradigms focused to the special education context.*

As part of this research work, a background review of different applications of HCI paradigms and special education is being carried out. Some of these will be mentioned as way of examples.

The Pictogram Room [6] project proposes that, with the help of augmented reality, the possibility of using pictograms that are placed on top of real objects can help autistic people see the connection between the real image and the pictogram in real time. In this case, the prevailing interaction paradigm is vision-based.

Project NAVI [7] is another example. This is an application aimed at people with vision difficulties. It collects visual data from the environment, such as shapes, colors, relative speed of objects, etc. It then processes them and provides oral indications to the visually-impaired user. It also uses a vibrating belt to indicate obstacle proximity.

The ABI (Adaptive Brain Interface) project is an example of sensorial substitution that is very useful for people with disabilities [8]. It enables people to transmit orders to the computer by means of electrical impulses emitted by their brains when they think about performing a certain movement. The idea is that a person can write texts by means of a keyboard simulator or handling a robot chair, for example, and the electric impulses generated before the movement to perform the action are then captured.

As part of this research work, the development of a prototype to adapt educational software by means of the audio-based HCI paradigm, following the classification described above, has been proposed. The following section describes the steps carried out to achieve this objective.

#### **4. Voice Recognition as Interaction Paradigm for People with Motor Difficulties**

Technological advances have provided human beings new and greater possibilities of developing a fuller lifestyle, but at the same time, this lifestyle continuously demands new and specific knowledge and skills for individuals to be able to take advantage of the possibilities being offered. In the case of people with some type of disability, the progressive complexity of the social media may however have the opposite effect to the desired social progress

[9]. Thus, voice recognition is an alternative for communicating with the computer, allowing people with motor disabilities that cannot access the standard keyboard and mouse to be able, through speech, to carry out actions that they would not be able to carry out without this technology. In other words, the purpose is converting human speech into actions that can be interpreted by the computer. These systems are not 100% reliable, so in-depth research would be required given that they could potentially improve the autonomy and quality of life of human beings, among numerous other applications.

## **5. JclicVoice: Adapting Jclic Software Through Voice Commands.**

As an application of the type of studies that are being developed in the area, one of the contributions made to the special education field is presented: jClicVoice, which is an adaptation to the educational software Jclic [10], aimed at incorporating a new mode of interaction to this application, namely, voice commands.

This paper is aimed at people with motor problems, but with little or no consequences in language development. This subset of people was selected because there is a wider variety of technical aides for people with motor disability that use various parts of the body, and we considered that the use of voice would be a good alternative if the person affected by the disability had no difficulty in oral expression. Also, this type of adaptation would require less effort from the person to use the computer, which would help prevent injuries caused by a “repetitive strain”

In the following sections, the methodology used for developing JclicVoice will be explained, after detailing the software components in Jclic, which is necessary to understand the adaptation proposed.

Jclic is formed by three applications, one of which is used for activity resolution [10]:

- Jclic Player: This component is used for solving the activities, which can be presented as applets or jClic applications.
- Jclic author: This is the authoring tool that allows creating, editing, and publishing activities.
- Jclic reports: This module is in charge of collecting data (time used for each activity, attempts, correct solutions, etc.), and then present them in statistical reports of various types.

The following subsection describes the methodology used to carry out the adaptation and create JClivVoice.

## **5.1 JClivVoice as Prototype Adapted to Use Voice Commands**

### **5.1.1 Previous Work.**

The adaptation proposed has tackled the modification of JCliv activities so that they can be solved through the use of voice commands.

As a first instance, only simple association activities were taken into account. Previous articles include a detailed description of each of the stages involved in the design of the adapted prototype. In this article, the new activities implemented as part of JClivVoice [11] will be described. However, some previous decisions are also mentioned as way of context for the new developments. One of the first steps was analyzing the use of the voice recognition engine. Different options were reviewed and compared. Finally, a voice recognition software called Sphinx [12] was selected, more precisely, its version 4 developed in Java. Also, the manner in which users would refer to the interaction elements shown during the activities had to be studied. A decision was made to use labels with numbers that could be mentioned in voice commands. After developing the first prototype, it was assessed by a group of experts. A series of surveys were designed, and they were completed by experts from the special education, computer science, and educational technology areas.

At the end of this assessment, it was concluded that the choice of educational software to adapt was good, as well as the use of voice commands as technical aide for the target group. Also, some modifications to the prototype were proposed, and then implemented. For instance, giving the educator the possibility of selecting or not the use of the voice command option for each individual student. Other modifications were related to interface design and label visualization. The most significant result was the interest in the use of the prototype shown by specialists and professionals in the area. Our work was presented during a special education event that had a large attendance of therapists and educators from this educational scenario, and their interest in having JClivVoice available for download was apparent.

Taking into account the results obtained, the prototype has been expanded to include other types of activities that can be created with JCliv, which are complex association activities, swapping puzzles, and sliding puzzles.

### **5.1.2 Description of the New Activities Included in JClivVoice.**

One of the new types of activities added to JClivVoice is the sliding puzzle. It consists of a single window with all puzzle pieces scrambled and one empty square. The content of this square, which is one of the pieces selected randomly by the program, appears to the right. Once the puzzle is solved, the

piece on the right (corresponding to the missing piece or empty square), is auto-filled.

Piece movements are restricted – pieces can only be moved to the empty square if they are right next to it, which makes this type of puzzle the hardest of all, especially in the case of non-graphical contents or with large numbers of pieces.

Figure 1 shows an example of a sliding puzzle activity using the adaptation of JClic.

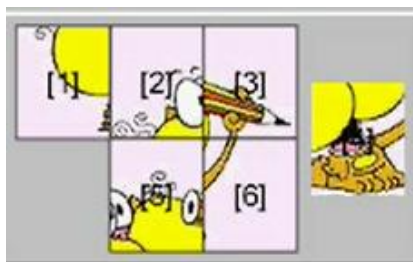


Figure 1: Association activity. Sliding puzzle.

To solve this type of activity in JClicVoice, students must say out loud the label corresponding to the element they wish to move to the empty square, followed by the word "Aceptar" (accept). Thus, only voice commands are required to solve the activity.

In the case of a swapping puzzle, the information is presented in a scrambled state and students have only one panel to rebuild the contents. Thus, they must swap the tiles until all squares are in the correct position.

For instance, in the case shown in Figure 2, students will have to say: "One with Three Accept" to swap the square labeled "1" with the square labeled "3."



Figure 2. Swapping puzzle

When an “Aceptar” (accept) voice input is received, after selecting two squares as described above, the system shows a message with the values that are going to be processed; the user has to confirm the values for the action to be performed.

For the confirmation, the word “Aceptar” has to be uttered again. At this point, the program checks if that is a correct match, i.e., if the selected squares are part of the solution. If so, the elements are removed from the set of possible elements to be chosen, and it moves on to the next correspondence, until getting to the last one. When the last correspondence is checked, the activity finishes.

In a nutshell, both in the case of simple and complex association activities, as well as with swapping puzzles, students must say the label associated to an element from the first dataset, followed by the word "Con" (with), and then the label corresponding to an element from the second dataset. To confirm their decision, they must say the word "Aceptar" (accept).

To carry out this adaptation, the JClic Author component had to be modified as well, so that it allows creating activities that have been adapted in jClicVoice, obtaining a complete and fully functional application.

The adaptation proposed here is available for download<sup>1</sup>. Also, an assessment tool has been designed to send community feedback to the authors and help them improve the prototype. The assessment tool is also available for download<sup>2</sup>.

## 6. Conclusions and future work

This work is but the first step on the road we are traveling. So far, we have worked with an HCI paradigm, focusing its use towards the educational scenario of interest. We are currently reviewing the possibility of using vision-based paradigms for other target groups within the context of special education. As regards jClicVoice, we are working on an assessment process that will allow us get feedback from both students and teachers working in the field, and test the activities that we have developed.

## References

1. Grau. "Tecnología y discapacidad visual. Necesidades tecnológicas y aplicaciones en la vida diaria de las personas con ceguera deficiencia visual". Madrid. Fundación UCM-ONCE, 2004.

---

<sup>1</sup><https://projectes.lafarga.cat/projects/jclicvoice/downloads>

<sup>2</sup> <https://projectes.lafarga.cat/projects/jclicvoice/surveys>



2. Luque Parra, Rodriguez Infante. "Tecnología de la información y comunicación aplicada al alumnado con discapacidad: Un acercamiento docente". Revista Iberoamericana de Educación, ISSN: 1681-2653, 2009.
3. ACM. Association for Computing Machinery (ACM). <http://www.acm.org>. Last access in 2012.
4. Center for the Study of Digital Libraries (CSDL), Texas A&M University, Course on HCI (CPSC 436). <http://www.csdl.tamu.edu/leggett/courses/436/part1/sld015.htm>. Last access in 2008.
5. Karray, Milad, Abou, Arab. Human-Computer Interaction: Overview on State of the Art. <http://www.s2is.org/Issues/v1/n1/papers/paper9.pdf>. 2008. Last access in 2012.
6. Pictogramroom. [http://fundacionorange.es/areas/22\\_proyectos/proy\\_230.asp](http://fundacionorange.es/areas/22_proyectos/proy_230.asp). Last access in 2012.
7. NAVI. [www.webayunate.com/ojos-virtuales-para-personas-ciegas-gracias-a-kinect](http://www.webayunate.com/ojos-virtuales-para-personas-ciegas-gracias-a-kinect). Last access in 2012.
8. Millar, Hausen, Renkens, 2002 - Adaptive Brain Interface - ABI: ABI: Simple Features, Simple Neural Network, Complex Brain. Available at: [www.cs.cmu.edu/~tanja/BCI/ABI2000.pdf](http://www.cs.cmu.edu/~tanja/BCI/ABI2000.pdf). Last access in 2012.
9. Madrid Vivar. <http://www.tecnoneet.org/docs/2002/2-82002.pdf>. 2002. Last access in 2012.
10. JClic, official web site. <http://clic.xtec.cat/es/jclic/index.htm>. Last access in 2012.
11. Moralejo Lucrecia, Ostermann Stefania, Sanz Cecilia, Patricia Pesado. "Adaptación mediante comandos de voz, para ámbitos de la Educación especial." Congreso Argentino de Ciencias de la Computación CACIC 2010.
12. CMU Sphinx, Official web site - <http://cmusphinx.sourceforge.net/sphinx4/>. Last access in 2011.



# Techniques for an Image Space Occlusion Culling Engine

LEANDRO R. BARBAGALLO, MATIAS N. LEONE, MARIANO M. BANQUIERO,  
DIEGO AGROMAYOR, ANDRES BURSZTYN

Proyecto de Investigación “Explotación de GPUs y Gráficos Por Computadora”, GIGC - Grupo de Investigación de Gráficos por Computadora, Departamento de Ingeniería en Sistemas de Información, UTN-FRBA, Argentina

{lbarbagallo, mleone, mbanquero, dagromayor }@frba.utn.edu.ar  
andresb@sistemas.frba.utn.edu.ar

***Abstract.** In this work we present several techniques applied to implement an Image Space Software Occlusion Culling Engine to increase the speed of rendering general dynamic scenes with high depth complexity. This conservative culling method is based on a tiled Occlusion Map that is updated only when needed, deferring and even avoiding the expensive per pixel rasterization process. We show how the tiles become a useful way to increase the speed of visibility tests. Finally we describe how different parts of the engine were parallelized using OpenMP directives and SIMD instructions.*

***Keywords:** Rasterization, Occlusion Culling, Visibility Algorithms, Hierarchical Occlusion Map, Lazy Grid, Tiles, Depth Buffer, SIMD, OpenMP, GPU.*

## 1. Introduction

In Real-Time Computer Graphics, rendering only the objects that will contribute to the final image is of vital importance especially when rendering large and complex 3D scenes. Visibility algorithms are in charge of determining which objects or parts of them will be visible from a given viewpoint, and in order to address this issue many techniques have been developed since 1970s [1]. Among these techniques is Occlusion Culling which aims to avoid rendering objects that are occluded by other objects in the scene by discarding them at an early stage in the rendering pipeline. The most widely used Occlusion Culling technique nowadays is the Z-Buffer algorithm that discards invisible fragments based on the comparison between the depth information provided in the rasterizing stage and the values previously stored in the Z-Buffer. Despite being implemented in hardware, this method is applied at an advanced stage in the rendering pipeline, where

the geometry has already been converted to fragments, potentially becoming a bottleneck in the application.

A conservative and a image space Occlusion Culling technique that best adapts to complex environments with high depth complexity is the Hierarchical Occlusion Map (HOM) proposed by Zhang [2] which chooses a small subset of occluders and rasterizes them in a buffer to store both occluder and depth information in their respective Occlusion Map and Depth Buffer. Then to perform the visibility test it divides the problem into two phases: a two dimensional overlap test where the occludee screen space projected bounding box is tested against the union of all occluders represented by the Occlusion Map and a second test which is the depth test performed using the Depth Buffer. To avoid the overhead of the per pixel depth comparison in every test, Zhang proposes buffers that are averaged and down-sampled forming a hierarchical pyramid, taking advantage of the bilinear texturing sampling capabilities available in modern GPUs.

One improvement to HOM suggested by Hey et al. [3] is to divide the occlusion buffer in a low resolution grid that is updated only when needed in a lazy manner, consequently avoiding unnecessary rasterization work and also avoiding the expensive pixel level depth comparisons in the overlap and depth test. Conversely, Décoret [4] suggested an alternative to the HOM image pyramid that performs the depth comparison in constant time storing  $N$  number of buffers of the same size each one with the maximum depth of all the pixels around a given area.

To speed up the image space occlusion culling process, modern graphics hardware support natively a special rendering mode which the user can query the Z-Buffer to determine whether a simplified version of a complex object such as its bounding volume is visible or not [5] [6] [7]. The GPU hardware rasterizes the query object and compares the fragment depths with the ones stored in the Z-Buffer. If the count of fragments that passed the depth test is zero, then the whole object can be culled, otherwise the whole object is considered visible. However, a direct implementation using this hardware capability does not produce the best results as stated by Bittner and Wimmer [5]: “Although the query itself is processed quickly using the raw power of the graphics processing unit (GPU), its result is not available immediately due to the delay between issuing the query and its actual processing in the graphics pipeline. As a result, a naive application of occlusion queries can even decrease the overall application performance due the associated CPU stalls and GPU starvation.” Techniques that exploit temporal coherence are often applied to mitigate these issues.

Nevertheless, this query delay motivated the development of other different approaches, and one of them is the Software Occlusion Culling Queries

processed in the CPU in an attempt to reduce this latency taking advantage of the increasing number of multicore systems and SIMD vector instructions available in modern systems. These techniques were used to speed up video games especially in game console systems as shown in [8] [9] [10].

In this paper we implement a SIMD Optimized Software Occlusion Map Rasterizer that uses a low resolution grid, that divides the screen into cells or tiles to rasterize the image space projected bounding boxes of complex objects as a series of quads. The occluder fusion is performed at image space by rasterizing the quads into the Depth Buffer only when it is absolutely needed.

## 2. Lazy Occlusion Grid Engine

### 2.1 Overview

In this section we will do an overview of how our Tiled Occlusion Culling Engine works. The whole purpose of this Occlusion Engine is to perform occlusion queries, which is to determine whether a given occludee is visible or not with respect of a given set of occluders. Therefore the Occlusion Engine has to solve two different things; First it has to accept a limited set of carefully chosen simplified occluders [11], and second it has to perform the overlap test for the occludees to verify if they are visible with respect of the selected occluders.

As the chosen occlusion culling approach belongs to the 2D image space category [1], the occluders need to be transformed from 3D object space to 2D image space. Initially the engine receives the occluders in the form of 2D projected quads<sup>1</sup> each one consisting of four points that form a coplanar four-sided convex hull. These points are already projected in 2D screen space with the depth value normalized between 0.0 and 1.0. The engine assumes that the given occluder quads already passed backface cull test, meaning that only the visible faces of the bounding box are sent to the Engine.

For every occluder quad received, the Occlusion Engine performs a classification of the tiles that are overlapped by the quads. With this new tile classification, the engine decides whether the part of the quad that falls into the tile should be rasterized pixel by pixel whether it just simply needs to update the tile properties.

---

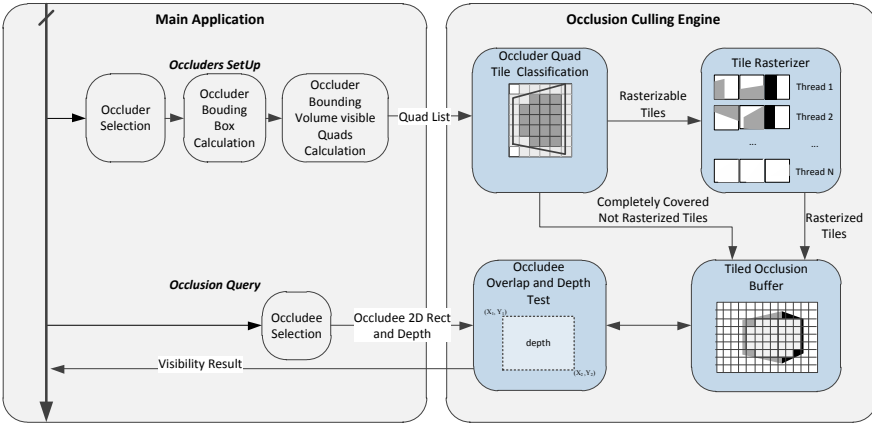
<sup>1</sup> Note: Triangles are accepted primitives as well because quads may require 2D screen clipping.

For the tiles that need to be rasterized, the quad is scan converted using the half-space [12] method and the depth values are linearly interpolated based on the first three vertices values of the quad. On the contrary, for the tiles that are now fully covered by the quad and the depth is less than the tile's previous content, the engine just performs an update to its internal properties updating the depth values at three of its corners.

The result of this occluder rasterization stage is the Occlusion Map and Depth Buffer combined into one single tiled Depth Buffer stored in main memory. The Occlusion Map is represented as all the tiles that have a depth value of 1.0 and unlike Hierarchical Occlusion Map (HOM), our method does not build the full multi resolution Occlusion Map Hierarchy, so it could be considered as a level 0 and a level N (HOM), being the latter the occlusion depth buffer tiled grid.

Finally for visibility determination, the engine receives the query composed of a 2D Axis Aligned Bounding Box that approximate the occludee projected image and a constant depth calculated as the closest point to the camera of all the bounding volume points.

The occlusion test is done first at a coarse level by checking whether any of the tiles inside the occludee region is in *NotInitialized* status, and if none is found then it proceeds to perform the depth comparison, where first it checks for the tile minimum depth and, as a last resource, the engine performs a pixel level depth comparison.



**Fig. 1:** How the Occlusion Culling Engine communicates with the main application. Occluder setup and Occlusion Queries are the most important functions used.

## 2.2 Occlusion Map Tiles

In this approach the Occlusion Map and Depth Buffer are combined into a single Occlusion Buffer which is divided into a grid with tiles of equal size, each with its own status data. This data consists of the Tile Status which can be one of the following:

- *NotInitialized*: There are no occluders or parts of them inside the region of that tile.
- *CompletelyCoveredNotRasterized* : A single occluder completely covers the region of the tile and its depth in all its points is less than any possible preexisting occluder point depths.
- *Rasterized*: The tile is covered by one or more occluders or parts of them, having different depth values.

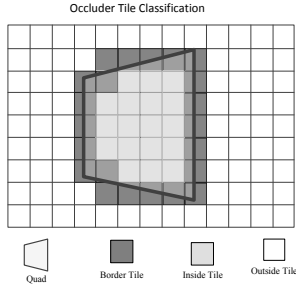
Initially when no occluders are set, the Depth Buffer is composed solely of *NotInitialized* tiles and, as the occluders start to arrive to the engine for classification and processing, the tiles may change their status and their inner properties.

When the user wants to send an occludee to the Engine, the bounding box of the object is calculated and the visible faces of the bounding volume are sent to the Occlusion Culling Engine as separate quads consisting of their four points and their depth values. Thereafter, based on the convex polygon defined by the quad, the engine classifies the screen tiles that intersect it, using the technique proposed by Greene [13] into three kinds: Outside, Border and Inside tiles. To speed up the process of classifying the tiles, we use the Trivial Reject Corner and Trivial Accept Corner suggested by Abrash [14].

The Outside kinds of tiles are quickly rejected and no changes to the screen tile original status or properties are done. However a different treatment must be done for Border and Inside tiles.

For the tiles that have been classified as Border (i.e partially covers a tile), the rasterizer will have to scan-convert the portion of the quad inside the region of the tile pixel by pixel, performing the depth test similar as what the Z-Buffer does.

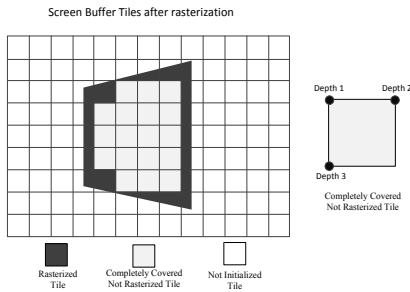
Nevertheless there is a case when no rasterization needs to be done at all, and it is when the screen tile is in *CompletelyCoveredNotRasterized* status and the occluder minimum depth is greater than the minimum depth already stored in the tile. In this case, it results the same as performing a Z-Buffer algorithm, but at a coarser level, that is at a tile grid level, avoiding the expensive rasterization process.



**Fig. 2:** This Occluder Tile Classification example shows how a quad (black polygon) is tested against the occlusion depth buffer tiles into Border, Inside and Outside tiles.

For the Inside kind of tiles there are more options to consider. For example, if the screen tile status was *NotInitialized*, we defer the rasterization in a lazy manner, since we only need to store the quad plane equation or just three depth points at the tile extreme corners. Taking advantage of the quad coplanarity property, the tile can be later rasterized when needed based on the three previously stored depth values.

On the other hand, if the screen Tile status was *CompletelyCoveredButNot Rasterized*, we need to compare the depth values at the extreme corners. If all the occluder depth values are less than the ones previously stored in the screen tile, then the occluder overlaps the whole tile content; If all depths are greater, then the occluder is occluded and the tile is left with the original depth values and properties. The last case occurs when some values are less and others not, in that case the occluder needs to be rasterized in order to calculate which fragments have less depth than the ones stored previously in the screen tile.



**Fig. 3:** This example shows how an empty occlusion depth buffer is updated with the content of an occluder quad. In this case only the border tiles are rasterized while the inside tiles becomes *CompletelyCoveredNotRasterized* and only the tile information (i.e. depth values at the tile extreme corners) is updated.



As shown earlier, the tile is only rasterized when it is absolutely needed, that is when the tile is not fully covered and when we need to count with greater depth detail at pixel level. This way of organizing the screen into a low resolution grid of tiles will later be of great use when doing the occludee overlap and depth test. We can say that the advantage of this method is that we gain more precision for occlusion detection when increasing the occlusion map size, but as we are using the tiled grid we get a similar performance as a much lower resolution occlusion map size.

## 2.3 Tile Rasterization

After the tile classification is done, a set of rasterizable tiles are sent to the rasterizer unit, which depending on the number of available threads in the system, equally distributes the load into a the processing queues [15].

When rasterizing convex quads we take advantage of the fact that we only need to perform a half-space test with four edges in order to check that the point to rasterize is contained inside the primitive. The edge test can be done four at a time utilizing the SIMD instructions available in modern multicore processors.

Another property that we exploit is that the quad is contained inside a plane. Once calculated these plane coefficients, every depth value of every point inside the hull can be obtained by replacing the point position in the plane equation. An optimization to avoid evaluating the plane equation in every pixel is to have accumulator variables so we only have to do additions of the plane's A and B coefficients every time the column or row changes in the X and Y nested rasterization loops [16].

Finally there is a special case in which the occlusion buffer tile is in *Completely CoveredButNotRasterized* status, so the tile doesn't contain pixel level information, only contains the extreme depth values at its corners. In case quad falls into this tile and needs to be rasterized, the previous content has to be rasterized first. In this case we just make a full tile cover taking advantage that no half-space check has to be done, thus speeding up the fill process significantly, and then as soon as the tile is filled, the new quad portion can be rasterized over it.

## 2.4 Occludee Visibility Test

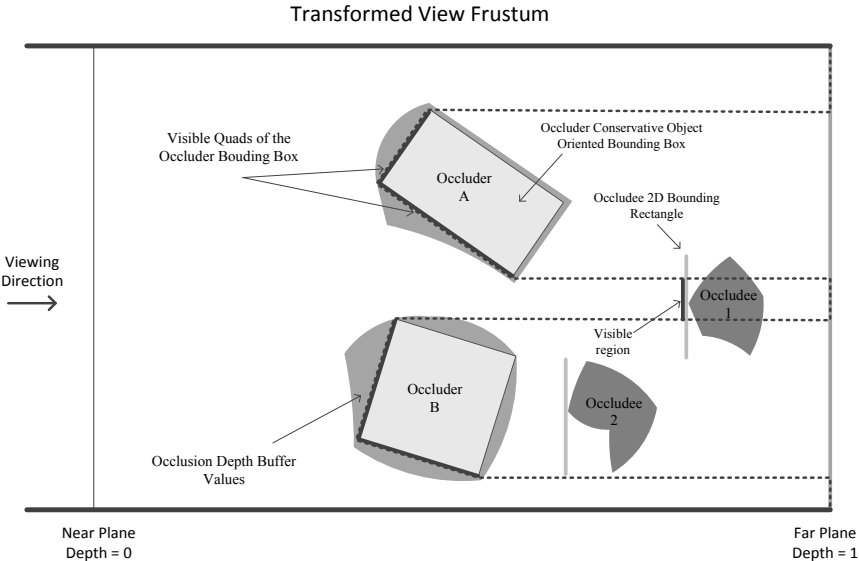
After all the occluders have been sent to the engine and the occlusion depth buffer is processed, the application may want to test if a given set of occludees are visible. To perform this, the first step is to get every occludee object's screen aligned bounding box to overestimate its coverage area. The

second step is to get the depth by getting the occludee closest point with respect of the view point. Given these two elements, the 2D bounding box and the depth, the occludee is sent to the engine.

The trivial solution to solve the visibility problem would be to compare each point of depth buffer inside the region of the occludee bounding box and check if there is at least one occludee point that is closer to the view point. The downside of this approach is that it involves accessing the depth buffer pixel per pixel which may not be the best option performance wise.

However, since the depth buffer proposed is divided into tiles, we can take advantage of several facts: The first one is that if we find that the occludee bounding box contains at least a tile in *NotInitialized* status, then the test query trivially returns a positive visibility result in an early manner. The second one is that we can utilize the minimum depth tile property to avoid per pixel depth comparisons when possible.

Finally, when all the previous options have been discarded and no positive visibility result is returned, the traditional pixel level test is performed.



**Fig. 4:** This example shows a transformed view frustum with a dotted line representing the depth values of the depth buffer when two occluders are rasterized in the Engine. Occludee 1 will result visible since there is a sub region in its rectangle in which the depth of the occludee is less than the value stored in the depth buffer. The Occludee 2 is completely culled by Occluder B. Only the visible faces of the Occluders conservative bounding boxes are sent to the engine as quads.

### 3. Implementation and Results

We have implemented our Software Occlusion Culling Engine in C++, using Visual Studio 2010 to compile it as a DLL module, and tested different scenarios using a PC with Intel Core 2 Duo 2.40 Ghz with 4GB RAM. The engine was integrated as a module in a 3D interactive application, where it was tested in a city scene with a high number of objects giving satisfactory results.

The occlusion engine consists mainly of the functions that perform the following tasks:

- Initialize the engine: Performs the initial sets up of the occlusion buffer and creates the tiles.
- Set Occluders: Receives and processes a list of occluders.
- Test Occludee Visibility: Determines if a given occludee is visible.

The rasterization of the tiles is parallelized using OpenMP 2.0 [17] by specifying simple directives in the “for” loop construct that iterates the rasterization tile queue. The load balancing between threads is done automatically using the dynamic scheduling clause available in OpenMP, however we found that this mode made the rasterization time more irregular and did not perform better than static scheduling mode.

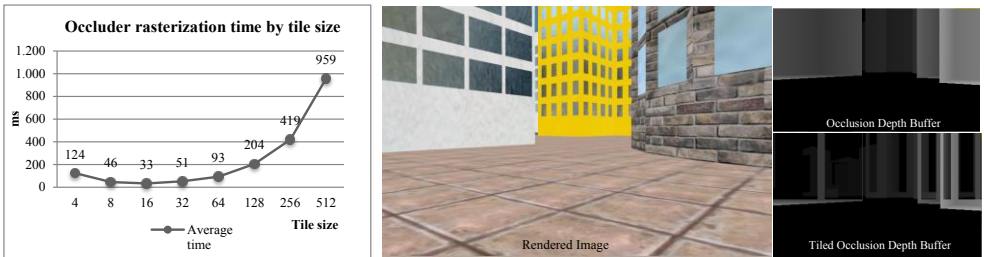
At a more fine grained data parallelism, we optimized the intra tile rasterization using the SSE3 and SSE4 SIMD instructions focusing on two main areas:

- Perform the half space test for the four edges of the quad using four 32 bit integer accumulators packed into a single 128bit XMM register. Only additions to the accumulator are done in the rasterization inner loops, reducing significantly the time taken to generate the coverage mask [18] [16].
- SSE4 Blend instruction was used to avoid conditional branching [19] when comparing depth values between occlusion depth buffer and occludee fragment value. This attempts to remediate the possible branch mispredictions caused by the depth comparison conditional.

The occlusion depth buffer grid is composed of an array of equally sized tiles, whose size was defined arbitrarily depending on several factors, such as resolution, system configuration and characteristics of the scene [3]. Since each thread works with one tile at a time, we found wise to turn a linear depth buffer into a nonlinear tiled depth buffer in which every row of a tile is placed contiguously in memory, one row next to the other. This loop blocking technique [19] has two advantages, first it tends to reduce the cache misses since all the pixels in the tile are closer together and the other advantage is that each point in the inner loop can be accessed by increasing a

simple pointer variable, avoiding the need to convert from 2D framebuffer coordinates into a memory linear address.

Regarding the performance results, we measured the engine response in different scenarios. We were interested in how the engine responds to different tile sizes and in different scene environments and we found that a tile size of 16 or 32 pixels gave the best performance results in the majority of the cases.



*Fig. 5: Left: This line chart shows the average rasterization time of 1000 random occluder quads in a 512x512 pixels occlusion depth buffer when different tile sizes are used. Right: A city rendered image and two versions of the Occlusion Depth Buffer; non tiled and using 16 pixel size tiles.*

## 4. Future work

Even though the average performance of our Software Occlusion Culling Engine is within an acceptable range, there are still pending optimization techniques: Doing some depth sorting method of the quads to avoid quick tile rejection, applying parallelism in the tile classification stage and reducing the memory footprint required to store the occlusion depth buffer tile grid.

We also found that the rasterization stage could benefit from using the x86 Advanced Vector Extensions (AVX) instructions that could potentially result twice as fast by processing eight pixels at a time by storing its coverage masks and edge accumulators in 256 bit YMM registers [19].

## 5. Conclusions

We have presented our Software Occlusion Culling Engine and introduced the techniques used to improve its overall effectiveness and efficiency. The

lazy update of the low resolution grid helps the occlusion culling process to delay or even avoid expensive operations. Moreover, applying SSE instructions to selected spots of the code enhanced the engine general performance by parallelizing tile rasterization.

The GPU accelerated occlusion queries will still continue to rasterize orders of magnitude faster than any CPU approach, however, as long as the GPU and CPU remain as separate cores, the latency required to transfer the query results will continue being an issue in performance.

When hardware accelerated occlusion queries are not available, software approaches have proven to be very effective and many production level applications were shipped using this technology. As result of this we estimate that as the multicore CPUs keep increasing their number of cores and expanding their SIMD instruction sets, software image-based occlusion culling is going to become a more utilized technique in the following years.

## References

1. Cohen-Or, D., Chrysanthou, Y. L., Silva, C. T., Durand, F.: A survey of visibility for walkthrough applications. *Visualization and Computer Graphics*, IEEE Transactions on Visualization and Computer Graphics 9(3), 412-431 (2003)
2. Zhang, H., Manocha, D., Hudson, T., Hoff, K.: Visibility Culling Using Hierarchical Occlusion Maps. In : In *Computer Graphics (Proceedings of SIGGRAPH '97)*, Los Angeles, CA, pp.77-88 (August 1997)
3. Hey, H., Tobler, R., Purgathofer, W.: Real-Time Occlusion Culling with a Lazy Occlusion Grid. In : *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, London, UK, UK, pp.217-222 (2001)
4. Decoret, X.: N-Buffers for efficient depth map query. *Computer Graphics Forum* 24(3), 393-400 (2005)
5. Bittner, J., Wimmer, M., Piringer, H., Purgathofer, W.: Coherent Hierarchical Culling: Hardware Occlusion Queries Made Useful. *Computer Graphics Forum* 23(3), 615-624 (2004)
6. Akenine-Moller, T., Haines, E., Hoffman, N.: *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA (2008)
7. NVIDIA Corporation: NV\_occlusion\_query. (Accessed February 6, 2002) Available at: [http://www.opengl.org/registry/specs/NV/occlusion\\_query.txt](http://www.opengl.org/registry/specs/NV/occlusion_query.txt)
8. Vale, W.: Practical occlusion culling in KILLZONE 3. In : *SIGGRAPH Talks*, p.49 (2011)
9. Andersson, J.: *Parallel Graphics in Frostbite-Current & Future*.

- SIGGRAPH Course: Beyond Programmable Shading (2009)
10. Narkowicz, K.: Software occlusion culling. (Accessed April 2012)  
Available at: <http://kriscg.blogspot.com/2010/09/software-occlusion-culling.html>
  11. Germs, R., Jansen, F.: Geometric Simplification For Efficient Occlusion Culling In Urban Scenes. In : Proc. of WSCG 2001, pp.291-298 (2001)
  12. Olano, M., Greer, T.: Triangle scan conversion using 2D homogeneous coordinates. In : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware, New York, NY, USA, pp.89-95 (1997)
  13. Greene, N.: Hierarchical polygon tiling with coverage masks. In : Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, New York, NY, USA, pp.65-74 (1996)
  14. Abrash, M.: Rasterization on Larrabee. (Accessed June 2012) Available at: <http://www.drdoobs.com/parallel/217200602>
  15. Bethel, Z.: A Modern Approach to Software Rasterization. (Accessed 2011) Available at:  
<http://cse.taylor.edu/~zbethel/MSR/ModernApproachToSR.pdf>
  16. Capens, N.: Advanced Rasterization. In: Advanced Rasterization. (Accessed November 2004) Available at:  
<http://devmaster.net/forums/topic/1145-advanced-rasterization/>
  17. The OpenMP® API specification for parallel programming. (Accessed April 2012) Available at: <http://openmp.org/>
  18. McCormack, J., McNamara, R.: Tiled polygon traversal using half-plane edge functions. In : Proceedings of the ACM SIGGRAPH/EUROGRAPHICS workshop on Graphics hardware, New York, NY, USA, pp.15-21 (2000)
  19. Intel: Intel 64 and IA-32 Architectures Optimization Reference Manual. (Accessed April 2012) Available at:  
<http://www.intel.com/content/www/us/en/architecture-and-technology/64-ia-32-architectures-optimization-manual.html>

# A Virtual Reality Computing Platform for Real Time 3D Visualization

YOSELIE ALVARADO<sup>1</sup>, MANQUEL MOYANO<sup>1</sup>, DIEGO QUIROGA<sup>1</sup>,  
JACQUELINE FERNÁNDEZ<sup>1</sup> AND ROBERTO GUERRERO<sup>1</sup>,

<sup>1</sup> LIDIC- Universidad Nacional de San Luis,  
Ejército de Los Andes 950,  
Tel: 02664 420823, San Luis, Argentina  
{yalvarado, manquelm, dquiroga, jmfer, rag}@unsl.edu.ar

***Abstract.** In the last decade, virtual reality (VR) systems have been used to enhance the visualization of design projects. The VR techniques allow to the designer interacting and modelling in a more intuitive and efficient way. Current 3D and animated simulation tools are a new challenge for 3D visualization. In this paper we propose a general VR computing platform that enables real time visualization of 3D scenarios for manufacturing and forensic simulations. The platform is able to treat static and dynamic 3D environments, allowing sharing the experience of navigation in the scene among the users, even geographically distributed. The platform proposed is validated through a case study using real time 3D models manipulations and interaction in a simulated car crash.*

***Keywords:** Virtual Reality (VR), CAVE, Real Time programming.*

## 1. Introduction

There is a popular quote: "Before anything can happen in the real world, first it must happen in the imagination; the world of dreams and possibilities". Sometimes, it is mandatory to be aware from special situations before they happen in the real world. This is the case of abrupt and unpredicted situations like environmental changes, illness problems and, auto, aerospace and naval accidents.

By creating an environment which simulates a potentially harmful real-life situation could help with these issues. The interactive scenario removes these concerns and helps the user gain knowledge and understanding of the subject matter without being put into a costly or harmful environment.

The design of robust advanced systems for unpredicted control situations is the corner stone of modern simulation theory and systems. Several challenges and issues are involved, for instance:

- In such processes a large number of concurrently unpredicted facts are involved. For instance, in the case of an airplane crash, the data

fields of the accident database must cover a multitude of parameters including aircraft, weather conditions, and flight as well as airport characteristics.

- Eminent problems in large-scale simulation arise from the difficulty in properly fitting all individual components together in a final product.

If in former times digital simulation systems were focused in the static representation of past situations, nowadays it is state of the art to focus on modelling and simulation of complex dynamic systems that characterized by information uncertainty of model structures and control goals, a high degree of freedom and essential nonlinearities, instability, distributed sensors and actuators, high level of noise, abrupt jump changes in structure and dynamics.

Virtual reality is used to create interactive scenarios which reflect real-life situations, simulating the way equipment responds; emulating the way machinery works or replicating soft skills such as human actions and behavior. Complicated pieces of equipment, processes or systems can be recreated using a number of techniques. This form of e-learning allows users to learn about mechanisms and processes that would be physically or logistically difficult to do so in other conditions.

In practice, while VR systems can be regarded as human interaction with dynamic computer-generated virtual environments in real time, a VR system design is not as simple as it might first seem. Research on virtual reality started from input devices, hardware interfaces, simple applications, to system infrastructure, more input and output channels, and sophisticated applications. The researches on virtual reality have become very popular. With different degrees of reality, many systems were developed. Desktop VR system achieves a little VR effect with a fixed screen. Head Mounted Display (HMD) and BOOM achieve VR with a small display, which is close to viewer's eyes and moves with him. Immersive VR uses projection-based screens, which are fixed and distant from the viewer to form virtual environments [24, 2]. The degree to which a system deals successfully with above difficulties depends on the computing platform of such advanced system.

Thus, due to the complex and dynamic nature of these real-life situation projects, the possibility of virtual, interactive and collaborative immersive visualization of whole simulation process and 3D analysis have become an important issue which can determine a project success [14, 9 22].

In this work an approach on a computing platform for immersive collaborative visualization of 3D and dynamic system is proposed. It allows the use of geographically distributed VR media, called a multi-VRmedia. Remote players can navigate and interact through 3D dynamic scenarios in a multi-VRmedia. During the navigation, the players can exchange information in order to cooperatively solve the observed problems.

It is suggested an integration of different VR techniques to get a final distributed collaboration among several players. Such approach was validated through a case study associated with traffic accidents at real scenarios.



The paper is structured in five sections. The next section addresses the main challenges associated with the simulation of unpredicted control situations and enumerates a group of existing solutions to deal with specific subproblems. Section III details the proposed computing platform for the 3D/4D manufacturing/forensic visualization, through aspects of modeling, simulation, visualization and interactivity stages. Section IV describes a group of experiments that were accomplished to validate the proposal. Finally, section V presents the conclusions drawn from the entire project.

## 2. Real-life Situations Simulation

The simulation of unpredicted control situations can reduce cost, complexity and time associated with the understanding and knowledge acquisition process needed for the real-life problem solution. Moreover, virtual reality enables to view the competency of users, see the decisions they make and how they then react to the consequences. For instance, large industrial conglomerates such as automobile (Volkswagen, Ford and General Motors), aerospace (Airbus, Embraer) and the shipbuilding/offshore industries are integrating VR concepts in their manufacturing processes [15, 6, 5, 4]. However, the collaborative multi-visualization process of 3D and dynamic systems using VR resources introduces a new group of challenges and issues. From the conception of the virtual model and its simulation, to its actual use in collaborative immersive visualization, the following main stages are enumerated: i. Modeling & Simulation ii. Visualization; and iii. Interactivity and Collaboration.

- **Modeling and Simulation.** The modeling stage refers to the 3D situation digital model creation, through modeling all situational time process dynamics (workshops, workers, tools, equipments and their interaction). Study and development of modeling techniques and simulations of situational processes is subject of study in different engineering areas. Some models can be implemented through the use of commercial Digital Mockup Systems (DMU) like QUEST and DELMIA, PROMODEL, ARENA [5, 4, 11] that enable different visual quality levels. For instance, the developed models can be visualized as simple 2D structures or like complex dynamic temporal scenarios with 3D features and interactivity. They allow one to design the 3D scenarios but they cannot cope with the collaborative issues and complexity of dynamic immersive visualization of the components operations.

The modelling phase implies the definition of the virtual model attributes and the entities to be supplied to visualization API. To have a virtual 3D scenario the geometric design must be completed with information describing the appearance of objects (color, reflection characteristics, and textures), the lighting environment, interactions, sound, as well as behavior and functionality. Such

definition can be done off-line or applied during real time visualization.

The generated simulations must show 3D dynamics of the different process components, as well as, 4D temporal simulations. Simulation must be consistent with the real scene matching physical conditions and assumptions for the accuracy of resulting simulation.

- Visualization. There are related works that deal with several issues associated with 3D visualization [12, 20, 19, 10]. However aiming at a more general immersive visualization, it is a good choice to adopt an engaging  $n$  multi-virtual reality projection. Traditional techniques for multi-projection handling make use of specialized homogeneous systems that need complex hardware architectures [6, 23]. Some specific questions in multi-visualization should be studied:
  - Maintaining consistence of the projected entities states, their attributes and dynamic behaviors among the projections on the  $n$  nodes;
  - Synchronizing virtual clocks (logical timers) of different physical system components, aiming at the coherence of the dynamic scenarios visualization and its relationship with the displayed frames rates;
  - Increasing the system performance, considering the number of entities to be projected versus the number of projection nodes.
- Interactivity and Collaboration. The visualization process should support different interactivity degrees and collaboration. For instance:
  - In visualization of unpredicted control situations scenarios, the use of different devices for sensory perception of the environment features (mice, keyboards, joysticks, glasses, gloves, trackers, etc) are important factors that should be foreseen;
  - A method that enables users to collaboratively view and interact has to be offered. They must consider the use of geographically distributed multi-virtual reality devices.

Nowadays, no DMU tools are able to deal with all the stages and aspects mentioned above. Similarly, no methodology/system that implements the modeling, simulation, etc., is able to provide the necessary characteristics to manufacturing/forensic processes through a collaborative visual system. There are, however, many studies and techniques regarding each individual subject (modeling, simulation, etc).

Several tools make the design and exportation of CAD models possible allowing specialized systems to convert these CAD projects to VR solutions. WalkInside [22] and Division Reality [17] are some of these systems. There are also visual immersive systems developed for CAD model visualization,

like VRJuggler [1], Viral [21] and ENVIRON [7], which, when associated with VR tools, allow CAD models visualization in VR environments.

Another possibility related to virtual environments is the use of game engines. Some examples of these systems are Unreal Tournament [8] and Quake [18] which provide 3D graphics with high detail levels. Both have high performance and robustness, supporting distributed geographic visualization.

The implementation used in this work is based on the OGRE [16] engine code with the BULLET [3] physics library, both under open-source license.

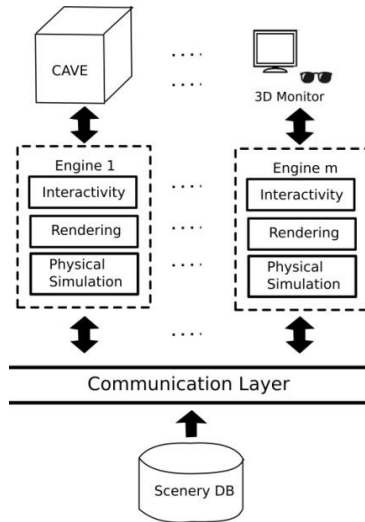
### **3. A Computing Platform for Immersive Visualization**

A computing platform includes a hardware architecture and a software framework (including application frameworks), where the combination allows software, particularly application software, to run. Typical platforms include a computer's architecture, operating system, programming languages and related user interface (run-time system libraries or graphical user interface).

Figure 1 shows an overview of the work's proposal. A system to visualize scenarios in a multi-virtual reality media environment has been defined. Such system will provide the necessary structure for attributes definition, rendering and collaborative multi-visualizations, as well as the needed interactive resources.

Based on the aspects mentioned, an automated platform that enables a collaborative visualization of unpredicted situation processes should present the characteristics described below.

- Heterogeneity: the quality and consistency of the visualization process should be guaranteed, regardless of the different hardware platforms used;
- Scalability: the performance of the system should be independent from the number of VR resources used in the virtual reality nodes;
- Portability: the methodology can be applied to different projection hardware and software types.
- Collaboration: the system must provide a geographically distributed visualization, navigation and collaboration.



*Fig.1. Platform for Collaborative Visualization*

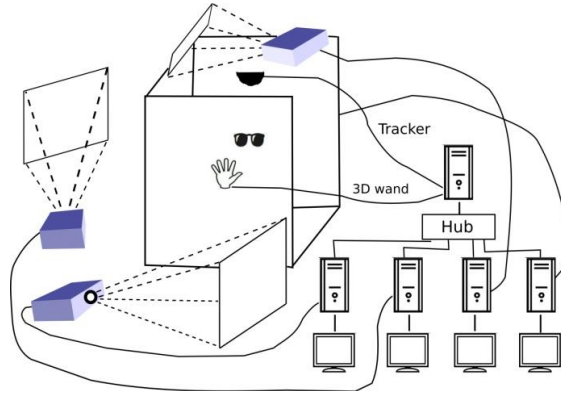
The distributed multiple-display virtual reality hardware components used in this work are Desktop, Head Mounted Display (HMD) and CAVE virtual reality resources. Fig. 2 depicts the projection structure of the implemented PC-based CAVE system, which is a  $3 \times 3 \times 2.5 \text{ m}^3$  surround screen projection system. The hardware components of the PC-based CAVE system are described as follows:

- Personal computers (PCs) - five PCs for driving the system. One is server for receiving input signals; the others are clients for screen display.
- Projectors - four high bandwidth stereo projectors.
- Signal sync devices - the signal sync devices are used to synchronize the RGB signals from client PCs.
- Stereo vision - shutter glasses and other peripherals (emitters, wiring, etc.) needed to operate them properly.
- Viewing point tracking device - a viewing point tracking device (one transmitter, two receivers) with an extended range controller (ERC).
- 3D Wand - 6-DOF hand-held input device.
- Screen/projection surface, supporting structure, cable, EPC2, ELR, etc.

PC connectivity of the PC-based CAVE system is depicted in Fig. 2 and established according to [13]. A software package was developed to drive the PC-based CAVE system.

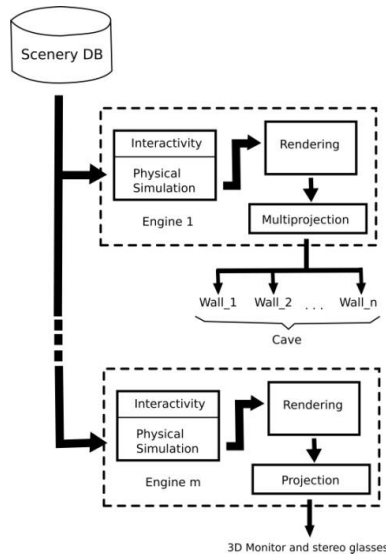
In graphics-based virtual reality applications, the complexity of the modeled virtual worlds dominates the rendering performance of systems. A virtual reality system can be separated into three parts - input, computation, and output. The input part consists of reading tracking devices and wands; the

computation part includes any computation of the virtual world and management of the world database; the output part consists of immersive screen displays.



*Fig. 2. CAVE Structure.*

The details of each one of the virtual reality nodes shown on figure 1 is described as follows. From a XML description, a visual system for rendering and multi-visualization was developed (see figure 3). The system code is based on OGRE [16], an open-source rendering engine. It has been designed to work with a wide variety of input and output hardware, with many device interfaces such as gloves and head-mounted display (HMD). The system allows  $n$ -walls multi-projection and the adjustment of each projection wall, in accordance with CAVE morphology.



*Fig. 3. Modular structure.*

The system could be used by project designers without computer knowledge. Those steps that involve some interaction with the user must be customized and the details documented, in order to be also operated by people without expertise in computing.

Currently, there are several languages and tools that enable the creation of virtual scenarios. However, existing approaches are not intuitive and require a thorough knowledge of the user. The proposed framework relies on the utilization of a generic metadata for the description of virtual scenarios which can be applied by different tools for authoring, and which can facilitate the subsequent automatic generation of OGRE code. With this approach OGRE developers can focus on the codification of the dynamics and strategies of the application being developed which helps reducing considerably the development time of these applications.

#### 4. Case Study

The proposed platform was validated in a case study on car crash simulation. This tool allows optimization of manufacturing layout problems; to determine and validate assembly sequences and ergonomics model aspects; and to make possible global analysis and 3D temporal simulation.

Adopting the stages presented in section 2, the details used in the case study are pointed out below. In the modeling & simulation stage, different 3D models were integrated in a 3D virtual scenery. The system displays the virtual XML scenario using binary space partitioning (BSP) structure. Collision detection and physics simulation was done by BULLET as physics library [3] and *bounding box* technique.

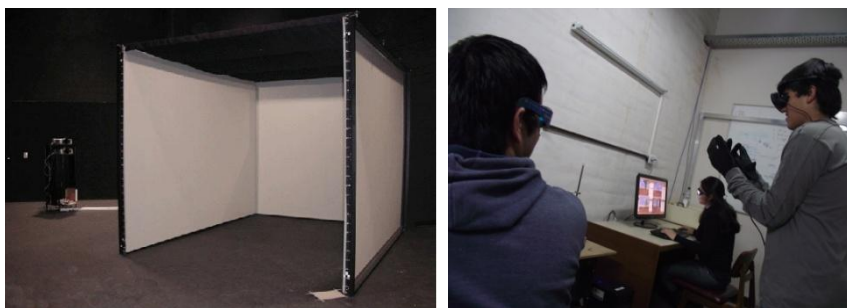
After modeling and simulation programming, the obtained animated scenario could be visualized in a multidisplay way. The platform allows a multi-VR media projection, each node using their own resources. Our implementation is based on the OGRE [16] adapted game engine. Initially, it was used a special monitor with 3D stereo support to validate the proposal (see figure 4(b)). In the test with the monitor, we used active stereoscopic glasses to allow the use of stereoscopic feature. This greatly increased the sense of realism of the scene. In order to prove the portability between different kinds of hardware, the visualization process was also realized in an experimental multiprojection system with three planes. Both multi-projection and monitor implemented test was portrayed as efficient and simple.

Rendering rates were measured with three scenarios, 70 fps for the first, 30 fps for the second and 10 fps for the third. In order to verify scalability of our proposal, the rendering rates were tested with geographically distributed nodes.

For this, we used one engine for each of the  $m$  geographically distributed hosts. In the example, three users at different locations (anywhere in the world) meet in the same virtual world by using stereoscopic glasses (active

and passive), and a Head-Mounted Display, respectively. All users see the same virtual environment from their respective points of view. Each user is presented as a virtual human (avatar) to the other participants. The users can see each other, communicated with each other, and interact with the virtual world.

Some other issues were tested, for example, the collision treatment that was portrayed as adequate, too. An illustration of the experimental CAVE and monitor tests can be observed in the figure 4.



*Fig. 4. Different test geometries.*

## 5. Conclusions

The word “simulation” comes up more and more in the field of industrial/forensic investigation and reconstruction. This tool allows detecting manufacturing layout problems;

to determine and validate assembly sequences and ergonomics model aspects; and to make possible critical analysis to show incorrect assessments.

Although the visualization of projects is already a practice in these contexts, the use of advanced 3D/4D immersive interfaces using VR resources, capable of integrating a range of design tools, is still a challenge. In this work we presented a computing platform to support collaborative multi-VR visualization of unpredicted control situations.

It is proposed to start with dynamic animated scenarios models. After identifying the limitations, restrictions and needs associated with the target problem, a group of procedures that enable the multi-visualizations, with immersive features and VR resources, integrating different existing tools, was proposed. The platform was validated in an actual application associated with a car crash simulation. Scenarios related were modeled, simulated and visualized in centralized and in collaborative multi-VR environments. Up to date, the accomplished visualization and interactive system were proved efficient, concerning the scalability, heterogeneity, portability, rendering, multi-VR media issues and cost. The proposed methodology has been thought to be used in real time visualization and applied directly in the

manufacturing/forensic process, by people with no need for specialized programming knowledge.

Future efforts will be directed at obtaining solutions for supporting a better interoperability between platform resources.

Finally, some interactivity aspects such as the inclusion of mixed reality resources to the system, enabling the visualization of either people or machines, in real or virtual way, and their interaction must be improved.

### **Acknowledgments**

This work is supported by the European Community, Alfa III - GAVIOTA, Contract N°: EuropeAid/129-877/C/ACT/RAL-1.

### **References**

1. Bierbaum A. Vr juggler: A virtual platform for virtual reality application development. Master's thesis, Iowa State University, (2000).
2. Craig Alan B., Sherman William R., and Will Jeffrey D. Developing Virtual Reality Applications: Foundations of Effective Design. Morgan Kaufmann; 1 edition, (2009).
3. Bullet. Collision detection & physics library. [Online], (2008).
4. Benson D. Simulation modeling and optimization using promodel. In Proceedings of the 1997 Winter Simulation Conference, pages 587--593, (1997).
5. Systemes D. Product lifecycle management solutions. [Online] (2002).
6. A. Dietrich, I. Wald, and P. Slusallek. Large-scale cad model visualization on a scalable shared-memory architecture. In Proceedings of Vision, Modelling and Visualization, pages 303--310, (2005).
7. Corseuil E. and Raposo A. Environ - visualization of cad models in a virtual reality environment. In Eurographics Symposium on Virtual Environments (EG-VE), pages 79--82, Sao Paulo, (2004).
8. EpicGames. Unreal tournament. [Online], (2004).
9. Nelson Duarte Filho, Silvia Costa Botelho, Jonata Tyska Carvalho, Pedro De Botelho Marcos, Renan De Queiroz Maffei, Rodrigo Ruas Oliveira, and Vinicius Alves Hax. A General Purpose Cave-Like System for Visualization of Animated and 4D CAD Modeling. 2009 3rd Southern Conference on Computational Modeling, pages 82--87, (2009).
10. Jorge Gascon, Jose M. Bayona, Jose Miguel Espadero, and Miguel A. Otaduy. Blendercave: Easy vr authoring for multi-screen displays. SIACG 2011: V Ibero-American Symposium in Computer Graphics, (2011).
11. Hammann J. E. and Markovitch N. A. Introduction to arena. In Proceedings of the 1995 Winter Simulation Conference, pages 519--523, (1995).



12. Bahram Javidi, Fumio Okano, and Jung-Young Son. Three-Dimensional Imaging, Visualization, and Display (Signals and Communication Technology). Springer; 1st Edition, (2008).
13. Wu-Jeng Li, Chin-Chen Chang, Ken-Yuan Hsu, Ming-Dar Kuo, and Der-Lor Way. A PC-based distributed multiple display virtual reality system. *Displays*, 22(5):177--181, (2001).
14. Konstantinos Loupos, Panagiotis Psonis, Virtual Environment, and Industrial Safety. Virtual reality: the way ahead in industrial safety. In 22nd EUROPEAN Conference on Modelling and Simulation, pages 105-111, Nicosia, Cyprus, (2008).
15. P. Maropoulos. Digital enterprise technology: Defining perspectives and research priorities. In *International Journal of Computer Integrated Manufacturing*, (2003).
16. Ogre. Open source 3D graphics engine. [Online], (2001).
17. PTC. Parametric technology corporation - division reality. [Online], (2003).
18. Quake. Id software. [Online], (1997).
19. J. R. Chardonnet, A. De Carvalho Amaro, J. C. Lon, and M. P. Cani. Hand navigator: Experimenting hand navigation in desktop virtual reality, (2009).
20. Paulo N.M. Sampaio, Roberto Ivo C. Freitas, and Gonalo Nuno P. Cardoso. Ogre-multimedia: An api for the design of multimedia and virtual reality applications. In Ignac Lovrek, Robert J. Howlett, and Lakhmi C. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5179 of *Lecture Notes in Computer Science*, pages 465-472. Springer Berlin Heidelberg, (2008).
21. Bastos T., Silva R., Raposo A., and Gattass M. Viral:um framework para o desenvolvimento de aplicacoes de realidade virtual. In VII Symposium on Virtual Reality, pages 52--62, Sao Paulo, (2004).
22. VRContext. Walkinside. [Online], (2003).
23. I. Wald, C. Benthin, A. Efremov, T. Dahmen, J. Gunther, A. Dietrich, V. Havran, H. Seidel, and P. Slusallek. A ray tracing based virtual reality framework for industrial design. Technical Report UUSCI-2005-009, SCI Institute - University of Utah, Salt Lake City - USA, (2005).
24. Sherman William R. and Craig Alan B. *Understanding Virtual Reality: Interface, Application, and Design*. Morgan Kaufmann; 1 edition, (2002).



# ISO 9001:2008 Certification: Impact on the Requirements Engineering Process

MARIANA CECILIA ARRUZAZABALA<sup>1</sup>, GLADYS DAPOZO<sup>1</sup>, PABLO THOMAS<sup>2</sup>

<sup>1</sup> DEPARTAMENTO DE INFORMÁTICA. FACULTAD DE CIENCIAS EXACTAS Y NATURALES Y AGRIMENSURA

Universidad Nacional del Nordeste, Av. Libertad 5450, 3400, Corrientes, Argentina  
arruzazabala@gmail.com, gndapozo@exa.unne.edu.ar

<sup>2</sup> Instituto de Investigación en Informática LIDI - Facultad de Informática. Universidad Nacional de La Plata, Calles 50 y 120, La Plata, Buenos Aires, Argentina  
pthomas@lidi.info.unlp.edu.ar

***Summary.** In the context of the software industry in Argentina, software quality certification contributes to the competitiveness of the productive sector. Requirements Engineering (RE) is a fundamental part of the software development process, since poor practices during this stage significantly affect the success of a project. In this context, an analysis of the extent to which the ISO 9001:2008 standard considers good RE practices poses interest. In this article, the RE process is assessed before and after the aforementioned certification with the quality standard in companies at Polo IT Corrientes. It is shown that the standard improves the RE process in certain areas, but not all of them. The results obtained will contribute to the definition of RE processes that meet the requirements of the ISO 9001:2008 standard and ensure high maturity levels in RE.*

***Key words:** Requirements Engineering. Quality Certification. Maturity in Requirements Engineering. ISO 9001:2008.*

## 1. Introduction

In the context of the young and dynamic software industry in Argentina, software development quality process certification is promoted as a way of contributing to a greater competitiveness in the productive sector [1]. Act 25,922 for the Promotion of the Software Industry, grants tax benefits to those who comply with it and, among its requirements, it includes quality certification [2]. Among the standards that are applicable by this Act are those certifying CMM processes, CMMI, ISO 9001, ISO/IEC 90003, IRAM 17601 (CMMI - SEI), ISO/IEC 15504, and the one certifying products ISO/IEC 9126 [3]□.

In particular, in the city of Corrientes, Argentina, there is an industry development area called Polo IT Corrientes, whose companies carried out a 12-month process to implement the ISO 9001:2008 Standard [4].

Among software development processes, Requirements Engineering (RE) is particularly critical because the errors that appear during this stage inevitably result in later problems that affect the entire software life cycle [5] [6]□.

In the last years, product development rates have increased dramatically, the changes in technology and staff turnover have impacted experience levels of involved professionals, and outsourcing and offshoring have changed product life cycle. Because of this, requirement specifications must be accurate and detailed [7]. Also, it is known that deficient requirements are the main cause for software project failure [8], and the later an error introduced during the requirements stage is discovered, the greater the cost to correct it [9]□.

For all these reasons, we consider it relevant to analyze the impact that the certification on the ISO 9001:2008 standard had on RE practices, using as study case the companies at Polo IT Corrientes that obtained this certification.

The task plan implemented to meet the expectations proposed can be summarized with the following actions:

- ▲ comparative analysis of RE process assessment models (chapter 2);
- ▲ summary of the features offered by the Uni-REPM model (chapter 3);
- ▲ creation and execution of company characterization surveys as context for research activities (chapter 4);
- ▲ application of the Uni-REPM model (chapter 5);
- ▲ results analysis (chapter 6);
- ▲ conclusions and future work (chapter 7).

## 2. Comparative analysis of RE process assessment models

In order to find the most appropriate model for assessing the maturity level of the RE process, an exploratory documental review was carried out in relation to RE concepts. Search criteria favored terms such as “Requirements Engineering Process Maturity Models,” “Requirements Engineering Process Maturity”, and other related terms. The selection of articles to review was oriented to specific RE models that also offered some methodology to assess the RE process.

The models analyzed were: REGPG, R-CMM and R-CMMi, REPM, MDREPM, Uni-REPM. The comparative analysis of these models is detailed in [10].

The most significant result obtained from this review was the selection of Uni-REPM as reference model, based on the following reasons:

- ▲ *It assesses projects*, which is an advantage for the work to be done, since reference projects will be used to assess their RE processes before and after obtaining the quality certification.
- ▲ *It unifies RE practices* for customized developments and market-oriented developments, both present in the companies to be assessed.

- ▲ *It simplifies the assessment and qualification process.* The assessment process is easily done through a checklist provided by the model, and the qualification process is simple.

### **3. Uni-REPM (Unified Requirements Engineering Process Maturity Model)**

Uni-REPM is a model that assesses the maturity of a project's RE process through a set of good practices [11]. The assessment consists in mapping the ideal activities with the real work of an organization.

The activities of the model are divided in 7 areas: Organizational Support, Requirements Process Management, Requirements Elicitation, Requirements Analysis, Delivery Planning, Requirements Specification and Documentation, and Requirements Validation.

Each area is in turn formed by subareas, and each subarea has actions, which are the smallest assessment units. Upon assessment, an action may have three possible states: complete, incomplete, or inapplicable. If an action is complete, it means that the organization has implemented it, if an action is incomplete, it means that the organization has to implement it but it does not, and if an action is inapplicable, it means that it should not be assessed for a specific organization because it is not essential for its business. As an example of inapplicable actions, if the company only develops customized products, actions related to market-oriented products should not be assessed. The classification of actions as 'inapplicable' must be careful, since its inappropriate use could bias the end result of the assessment.

The model proposes three maturity levels as the result of the assessment: 1-Basic, 2-Intermediate, and 3-Advanced. The objective of the first level is ensuring that processes are defined and that they are applied by the organization. Relevant stakeholders are involved in the elicitation process, requirements are analyzed in detail, and standard documents are predefined. The Intermediate level tries various perspectives and is directed by product strategies/objectives. Roles and responsibilities are well defined and documented, change requests are consistently managed throughout the project, and requirements selection is done systematically. The Advanced level represents the most mature RE process. Process improvement is seen in the advanced requirement collection method, requirement quality assurance, constant communication and common understanding among various stakeholders, and the proactive assessment of the decision-making process.

To achieve any given maturity level, all the actions associated to that level must be fulfilled, in addition to the actions of the lower levels. For a more detailed analysis, maturity levels can be considered by area and subarea, with an area or subarea reaching a given level if all actions corresponding to that level plus all actions from the immediately previous level are fulfilled. Not all subareas have actions from all three levels – there are subareas whose actions have a minimum level of two, and subareas whose actions have a maximum level of two, so any given area can reach level three but have one

of its subareas at level two only, since this is the maximum possible level for that subarea.

The creation of this model was based on a review of literature on models REPM, CMMI and ISO 9000-TickIT, as well as a systematic review of research on market-oriented RE practices. The Uni-REPM model was validated in both the academic and industrial contexts [12][13].

#### **4. Research Context**

Polo IT Corrientes is formed by nine companies, eight of which were part of the implementation process of standard ISO 9001:2008. From these eight companies, six certified RE-related processes, which can be verified at IRAM's website, section 'Certificación de Sistemas de Gestión' (Management Systems Certification), subsection 'Certificación IRAM-ISO 9001' (IRAM-ISO 9001 Certification), searching for company name [14]; one of these companies did not maintain the quality management system. Our research considered the five companies that still maintain their quality management systems.

Interviews with these organizations were carried out between May and June, 2012. In most of the cases, the person who attended the interview on the side of the company was the company manager or director. In one case only, the person attending the interview was the person in charge of analysis and design, and in another case it was the person in charge of quality, but both are partners in their respective companies, meaning that, in addition to their roles, they also have managerial information.

Each interview had two parts: the first portion of the interview was aimed at characterizing the companies and selecting the projects to be assessed, while the second portion consisted in using the checklist proposed by Uni-REPM to assess the maturity of the RE process in those projects.

All companies are from the city of Corrientes, province of Corrientes, Argentina, and started their operations between 2000 and 2008.

The main service offered by these companies is customized software development, followed by implementation and tune-up of software products and technical support for software products. They also offer software product updates and other related services, including training, website hosting, and audio and video streaming. These are followed by (canned) software product development, added value IT services (consulting), and, to a lesser extent, services provided by the use of software and development of software embedded in electronic equipment.

As regards products, 80% of the companies is business-oriented, 60% of the companies offers solutions for the public sector, 40% offers solutions for the primary sectors of construction, finances and insurance, transportation, communications and real estate; and 20% of the companies covers the software needs of the manufacturing, hospitality and restaurant sectors. None of the companies mentioned education as one of their target markets. All of them cover other sectors that were not mentioned in the options of the survey,

including communication media, healthcare, and other service companies, such as security and nautical.

The average number of employees per company is ten, 92% working at the company's physical location and the remaining 8% working as distance workers. Forty-eight percent are full-time employees, and 52% are part-time employees; full-time employees are mostly from two of the companies, and part-time employees are mostly from three of the companies.

As regards education, human resources are mostly university graduates from the Information Systems area – 30.77% are Information Systems graduates, 21.15% are Systems Engineers, followed by 19.23% of Systems Analysts or Application Programmers, 11.54% Technicians, 5.77% Graphics Designers, 5.77% students, and 5.77% corresponding to other education levels related to administration, marketing or business.

## **5. Applying the Uni-REPM model**

For the application of the Uni-REPM model, the companies were requested to select two software projects, one that had been carried out before their certification on the ISO 9001:2008 standard, and another one carried out after the standard had been applied.

Sixty percent of the companies selected a customized development program, and 40% chose a project for the adaptation of an existing product they had already developed; all companies chose the same type of project pre- and post-certification. As regards the starting point for obtaining project requirements, in the case of pre-certification projects these were 100% client-defined, whereas in post-certification projects, requirements were 60% client-defined and 40% a combination of client requirements and market possibilities.

Regarding the lifecycle model applied to the projects, 60% of pre-certification projects used the incremental model, with the remaining 40% following a cascade model. Post-certification projects were carried out following an incremental lifecycle model (40%), iterative model (40%), and cascade model (20%).

Project duration ranged between 1 and 18 months, and the difference between pre- and post-certification projects within each company was between 0 and 6 months. The number of people involved in the projects varied between 2 to 6 people per project.

As for the roles in the development team, all pre-certification projects had developers and a project leader, with 60% of the projects including the role of systems or requirements analyst, 40% including the role of user interface manager and tester, and 20% with the role of documenter and other roles. All post-certification projects had a project leader, developer, systems or requirements analyst, tester, and documenter; 80% of the projects included the role of user interface manager, and 20% mentioned other roles, such as implementer and trainer.

## 6. Results Analysis

The projects that were assessed do not reach level one, which is the Basic level, meaning that the processes defined by the organization do not cover all the actions from the RE areas proposed by Uni-REPM. However, after analyzing the results by area, significant improvements are observed, such as the fact that in all cases, the areas 'Organizational Support' and 'Requirements Specification and Documentation' increased their maturity level after obtaining the certification on standard ISO 9001:2008. This indicates that the actions corresponding to these RE areas are considered in the processes defined and applied by the organization. For the 'Organizational Support' area, three companies reached the Basic level, and two obtained the Advanced level. In the 'Requirements Specification and Documentation' area, four companies reached the Basic level and one obtained the Advanced level. Two companies improved their maturity level in the 'Requirements Process Management' area; they are the same companies that reached the Advanced level in the 'Organizational Support' area.

One of the companies increased its maturity level in the 'Release Planning' area; although it should be noted that only two of the companies selected projects that allowed assessing this area.

Only one company reported an improvement in maturity level for the 'Requirements Elicitation' area.

The 'Requirements Analysis' and 'Requirements Validation' areas did not present any maturity level improvements, which would indicate that the ISO 9001:2008 standard does not have any direct requirements in relation to these areas, i.e., these actions are not a relevant factor when applying the ISO 9001:2008 standard.

Table 1 shows the levels reached by each company for each area. The first row shows an ID for the companies, the second row includes PRE as ID for projects carried out before obtaining the certification on the ISO 9001:2008 standard or POST to indicate projects that were carried out after obtaining this certification. All other rows show the areas proposed by the Uni-REPM model. Level changes between projects within a company are highlighted with a gray background, showing the areas involved.

**Table 1.** Levels by area.

Companies	A		B		C		D		E	
Areas	PRE	POST	PRE	POST	PRE	POST	PRE	POST	PRE	POST
Requirements Analysis	0	0	0	0	0	0	0	0	0	0
Organizational Support	0	1	0	1	0	1	0	3	0	3
Requirements Specification and Documentation	0	1	0	1	0	1	0	3	0	1
Requirements Elicitation	0	0	0	0	0	0	0	0	0	1



Companies	A		B		C		D		E	
Areas	PRE	POST	PRE	POST	PRE	POST	PRE	POST	PRE	POST
Requirements	0	0	0	0	0	0	0	1	0	1
Process Management										
Release Planning	N/A	N/A	N/A	N/A	0	1	N/A	N/A	0	0
Requirements Validation	0	0	0	0	0	0	0	0	0	0

When analyzing the results by subarea, it can be seen that within the 'Organizational Support' area, four out of the five companies improved their 'General Actions;' the only company that did not show an improvement ranked at level two in both projects, which is the highest possible level for the subarea, meaning that the only company that did not improve was already at the highest possible level before obtaining the certification. All companies improved the 'Roles and Responsibilities' subarea. The 'Strategic' subarea was not applicable for three of the companies; out of the other two, one did not reach the Basic level in neither of the projects, and the other one reached the Advanced level after the certification.

As regards the 'Requirements Specification and Documentation' area, all companies improved their 'General Actions,' two companies improved their 'Documentation Deliverables,' one of the companies that did not improve ranked at the highest level in both projects, meaning that it had already reached the highest level before the certification, and the other two companies did not reach level two or Intermediate in their projects, which is the minimum level for this subarea.

In the case of the 'Requirements Elicitation' area, two of the companies improved their maturity level for 'General Actions,' and one company improved its level for 'Stakeholder and Requirements Source Identification;' the subarea 'Domain Consideration and Knowledge' did not show any level changes.

In the 'Requirements Process Management' area, all companies improved their 'General Actions,' four out of the five companies improved their 'Requirements Communications and Negotiation,' and the only company that did not improve ranked at level one for both projects.

**Table 2.** Levels by subarea.

Companies		A		B		C		D		E	
Area	Subarea	PRE	POST	PRE	POST	PRE	POST	PRE	POST	PRE	POST
Requirements											
Analysis	General Actions	0	0	0	0	0	0	1	1	0	0
Requirements	Problems and										
Analysis	Solution Analysis	0	0	0	0	1	1	0	0	0	0
Requirements	Quality Attribute										
Analysis	Analysis	0	0	0	0	0	0	0	0	0	0

Companies		A		B		C		D		E	
Area	Subarea	PR	POS	PR	POST	PR	POST	PR	POST	PR	POST
		E	T	E	POST	E	POST	E	POST	E	POST
Organizational Support	General Actions	0	2	0	2	2	2	1	3	0	2
Organizational Support	Strategic	N/A	N/A	N/A	N/A	0	0	N/A	N/A	0	3
Organizational Support	Roles and Responsibilities	0	1	0	1	0	3	0	3	0	3
Requirements Specification and Documentation	General Actions	0	1	0	1	0	1	0	3	0	3
Requirements Specification and Documentation	Documentation Deliverables	0	2	0	0	0	3	3	3	0	0
Requirements Elicitation	General Actions	0	0	0	0	0	1	2	2	0	1
Requirements Elicitation	Domain Consideration and Knowledge Stakeholder and Requirements Source Identification	0	0	2	2	0	0	0	0	1	1
Requirements Elicitation	Identification	0	0	1	1	0	1	0	0	1	1
Requirements Process Management	General Actions	0	2	0	2	0	2	0	1	0	2
Requirements Process Management	Requirements Communications and Negotiation	1	3	1	3	1	1	1	3	1	3
Requirements Process Management	Configuration Management Requirements	0	0	0	0	0	0	0	2	0	2
Requirements Process Management	Traceability Policies	0	0	0	0	0	0	0	1	0	1
Release Planning	General Actions	N/A	N/A	N/A	N/A	0	0	N/A	N/A	0	0
Release Planning	Requirements Selection	N/A	N/A	N/A	N/A	0	1	N/A	N/A	0	0
Requirements Validation	General Actions	0	0	0	0	0	0	0	0	0	0

Two companies improved their 'Configuration Management' and 'Requirements Traceability Policies' – these are the two companies that improved their maturity level for this area. This would indicate that the ISO 9001:2008 standard includes certain 'Requirements Process Management'

actions, especially those included in the 'General Actions' and 'Requirements Communications and Negotiation' subareas.

As regards the 'Release Planning' area, only one company out of the two whose projects qualified for assessment in the area improved its level in the 'Requirements Selection' subarea; the 'General Actions' subarea did not present any level improvements.

The subareas corresponding to the 'Requirements Analysis' and 'Requirements Validation' areas did not present any changes.

Table 2 shows the levels reached by subarea of the Uni-REPM model. Level changes between projects within a company are highlighted with a gray background, showing the subareas involved.

## 7. Conclusions and future work

The ISO 9001:2008 standard implemented by the software companies of Polo IT Corrientes improved the maturity level of the RE process in the 'Organizational Support' and 'Requirements Specification and Documentation' areas and the 'General Actions' and 'Requirements Communications and Negotiation' subareas, both corresponding to the 'Requirements Process Management' area.

Some of the companies improved the 'Release Planning' and 'Requirements Elicitation' areas and the 'Configuration Management' and 'Requirements Traceability Policies' subareas from the 'Requirements Process Management' area.

No improvements were observed in the 'Requirements Analysis' and 'Requirements Validation' areas.

Since the ISO 9001:2008 standard is a generic standard that is applicable to any type of organization, and each organization defines its own processes, it would be helpful to determine which of the improvements are a consequence of meeting standard requirements, and which are attributable to the standard for being a requirement of the process defined by the organization.

As a result of the previous analysis, the following hypotheses are proposed:

- ⤴ The standard favors improvement in the 'Organizational Support' and 'Requirements Specification and Documentation' areas and the 'General Actions' and 'Requirements Communications and Negotiation' subareas from the 'Requirements Process Management' area.
- ⤴ The standard does not favor (at least in the cases analyzed for this paper) process improvements in the 'Release Planning', 'Requirements Elicitation', 'Requirements Analysis' or 'Requirements Validation', nor in the 'Configuration Management' and 'Requirements Traceability Policies' subareas from the 'Requirements Process Management' area, unless these practices are specifically defined in the processes that are certified by the standard.

Finally, it should be noted that, even though the ISO 9001:2008 standard is of a generic nature, it positively contributes to the software development process, and specifically and at least partially to RE, in the study cases.

As future work, a comparative analysis of the Uni-REPM model and the ISO 9001:2008 standard will be carried out, and then the RE processes that meet standard requirements and reach the maturity levels proposed by Uni-REPM will be defined.

**Acknowledgments.** To Tony Gorschek and Mikael Svahnberg for their trust and collaboration with all our queries about the Uni-REPM model.

## References

1. Cámara de Software y Servicios Informáticos - CESSI: Propuestas para el Plan de acción 2008-2011, [http://www.cessi.org.ar/documentacion/PLAN-2008-2011\\_Documento\\_Principal\\_11.pdf](http://www.cessi.org.ar/documentacion/PLAN-2008-2011_Documento_Principal_11.pdf), (2007).
2. Ley 25.922 - Ley de Promoción de la Industria del Software, [www.mincyt.gob.ar/multimedia/archivo/archivos/ley\\_25922\\_3.pdf](http://www.mincyt.gob.ar/multimedia/archivo/archivos/ley_25922_3.pdf).
3. Department of Industry, Commerce, and Small and Medium Businesses: Resolution 61/2005: Software Producers and Computer Technology Services, <http://inforeg.mecon.gov.ar/inforegInternet/anexos/105000-109999/106061/norma.htm>, (2005).
4. Polo IT Corrientes: Software and computer technology services SMBs from Corrientes will certify their quality, <http://poloitcorrientes.com/noticia.php?i=05229>, (2009).
5. Sommerville, I.: *Software Engineering*. Addison-Wesley (2005).
6. Loucopoulos, P., Karakostas, V.: *System Requirements Engineering*. McGraw-Hill (1995).
7. Berenbach, B., Paulish, D.J., Kazmeier, J., Rudorfer, A.: *Software & Systems Requirements Engineering: In Practice*. McGraw-Hill Osborne Media (2009).
8. Hofmann, H.F., Lehner, F.: Requirements engineering as a success factor in software projects. *IEEE Software*. 18, 58-66 (2001).
9. Mizuno, Y.: Software Quality Improvement. *Computer*. 16, 66-72 (1983).
10. Arruzazabala, M., Dapozo, G., Thomas, P.: Análisis comparativo de modelos de evaluación de procesos de Ingeniería de Requerimientos, *II Jornadas de Investigación en Ingeniería del NEA y Países Limítrofes*, <http://www.frre.utn.edu.ar/IJCyT/clean/files/get/item/2186>, (2012).
11. Nguyen, T.T.L.: The creation of Uni-REPM A universal model for assessing requirements engineering process maturity, Blekinge Institute of Technology, (2010).
12. Svahnberg, M., Gorschek, T., Nguyen, T.T.L., Nguyen, M.: Uni-REPM: validated and improved. *Requirements Engineering - Springer London*. 17, 19 (2012).
13. Nguyen, M.: Empirical Evaluation of a Universal Requirements Engineering Process Maturity Model, Blekinge Institute of Technology, (2010).
14. IRAM: IRAM | Instituto Argentino de Normalización y Certificación, <http://www.iram.org.ar/seccion.php?ID=3&IDS=42>. Accessed on May 10, 2012.

# Using of knowledge representation techniques as a way to support requirements engineering

GILBERTO DESSBESELL JR<sup>1</sup>, EDUARDO KROTH<sup>1</sup>

<sup>1</sup>Departamento de informática – Universidade de Santa Cruz do Sul (UNISC)  
Av. Independência, 2293 – 96.815-900 – Santa Cruz do Sul – RS – Brasil  
gjunior@mx2.unisc.br, kroth@unisc.br

***Abstract.** The Requirements Engineering is a fundamental part of the process of software development. Even if a program uses the most advanced technologies and has the best source code, it is not useful if it does not meet its requirements. The large number of failures occurring in software projects, attributed to deficiency of capturing and analyzing their requirements emphasize the importance of this area. In this work, we used knowledge representation in order to store the knowledge about the requirements of a project and then use it as a support to the capture and analysis of requirements. In addition, techniques of similarity were applied on the requirements, to ensure they are contextualized and mutually consistent.*

## 1. Introduction

Most part of the existing problems in the software development process is related to poor quality in the capture and analysis of the requirements [20]. This deficiency can be verified through products that only partially meet the needs of stakeholders or, even worse, that are different from what was requested by them.

The Chaos Report 2009 publication [20] asserts that in 2009, in the United States, 24% of software projects were canceled or never used. Other 44% were delayed, went over the budget or had its scope reduced. The remainder, only 32%, has been completed successfully. In failed projects, the main cause pointed (13%) were incomplete requirements. In projects with partial success, 24% of the problems were attributed to incomplete requirements and specifications. The study further discloses that the statement of clear and clean requirements was a factor (13%) that most influenced the projects that were successful.

Several techniques have been developed to improve the quality of the capture and analysis of requirements. However, most of them only address the ways in which requirements are captured and analyzed, failing to consider the fact that the interpretation and management of the requirements, by the people involved in these tasks, it's a very important issue in this process and that depends on the ability and knowledge of human beings.

The Knowledge Representation is a subfield of artificial intelligence (AI), which seeks ways to represent, in a computational way, the knowledge

created by humans, performing the classification and storage of the knowledge in a structured manner in knowledge bases. Through structures such as ontologies, it's possible to organize the knowledge acquired by a group of people and then make it available for computational queries. The goal of this approach is to create a knowledge base in digital and non-volatile format, preserving and accumulating more knowledge about a particular subject.

To refer to this stored knowledge, the Information Retrieval offers activities that involve the intellectual aspects of information description and its specification for searching. In the computing field, this activity is best described as the retrieval of information from databases wherein information in textual form, not tabulated, predominates.

Based on these concepts, this paper conducted queries and indications on knowledge bases of software projects, reaching a close result, or better, that a human would reach, having the same information.

## 2. Requirements engineering

[10] defines a software requirement as a condition or capability that the software must perform or have to meet the need of a stakeholder. [6] enlarges these definitions, stating that requirements are properties that a software must have to meet the needs of the stakeholders in the environment where it will be used. Covering these settings, the goal of a software requirement is to set the contracts, standards, specifications and other documents, in other words, the conditions and capabilities externally imposed to the context of the software development, which must be met by a software [8].

[19] recommends the characteristics that a Software Requirement Specification (SRS) must have. It should be, at the same time: correct, unambiguous, complete, consistent, ranked by importance and/or stability, verifiable, flexible and traceable. [8] contributes to this definition reinforcing the idea that what is important, especially when using natural language, is to ensure that the contents of the SRS is clear, precise and unambiguous.

At the same time that the use of the natural language facilitates the understanding of the requirements, by being simple and easy to understand, it also hampers some other aspects. In cases where it is necessary to be very specific in describing a requirement, the document ends up becoming very detailed and, therefore, difficult to read. The use of the natural language, in some cases, also results in confused specifications, because functional and non-functional requirements are eventually mixed in the same specification. [16]

[18] defines that Requirements Engineering (RE) provides a mechanism for understanding what the stakeholders want, analyzing the necessity and the feasibility of the requests, negotiating a reasonable solution, specifying the solution unambiguously, validating this specification and managing the requirements for them to be transformed into a functional software.

[14] highlights that there is no incontestable way to ensure that the specification of a system is properly in accordance with the needs of the stakeholders. This is a complex challenge faced by requirements engineers, and the best way to face it is through a consistent process of RE.

### **3. Information retrieval**

[12] defines Information Retrieval (IR) as the activity that involves the intellectual aspects of information description (indexing, standardization) and its specification for search, beyond any technique, system or machine used to perform or assist these relevant tasks to the user. In order to differentiate between the terms "Data Recovery" (DR) and IR [1] states that DR consists in determining which documents in a collection contain the keywords contained in the user's query. The main difference is that the texts in IR are in natural language and, in most cases, are not well structured, with the possibility of being semantically ambiguous.

According to [9] an Information Retrieval System (IRS) is able to store, retrieve and manage information. Although the shape of an object in an IRS being diverse, the textual aspect has been the only data type that allows full functional processing. The goal of an IRS is to minimize the cost for an user to locate the desired information, assuming that the user could locate the information by analyzing all the documents in a collection, manually. The cost can be understood as the time spent by the user to complete the task of recovering the relevant information.

According to [7], one of the most used methods for storing and subsequent information retrieval of free text is indexing. The idea of indexing is to produce a smaller, but more efficient, set of data to represent the content and facilitate recovery of the original document. On the other hand, the technique called controlled vocabulary [11] makes it much easier to find the information because the users are accustomed to using the terms commonly used in their area of interest.

The automatic indexing process has the objective of identifying relevant terms in documents from a collection and stores them in an index structure. The following stages typically are found in the process: identification of terms, removing stopwords, normalization and morphological identification of compound terms. For each of these stages, several techniques can be adopted. The order of application of each stage can vary, or even some of them may not be used [15].

### **4. Ontologies**

The term "ontology" began to be used in Computer Science in the early 90s, in projects of organization of large knowledge bases. Since the 70s, the

subarea of AI was already concerned with the organization and manipulation of knowledge bases, but it was in the 90s that the interested in the study of shareable and reusable knowledge bases [13] increased.

To [4], ontology is a formal and explicit specification of a shared conceptualization. A clearly defined (explicit) specification and machine-processable (formal), of a shared abstract model. To [17], the term "ontology" has been adapted and, for professionals in the Computer Science, ontology is a document or file that formally defines the relations between terms and concepts, keeping thus similarities with the thesauri used for the definition of controlled vocabularies.

Also according to [17], the goal of building an ontology is to supply the need for a shared vocabulary for exchanging information between members of a community, being they human beings or intelligent agents. [2] agrees with this definition, complementing that ontologies work as a foundation to secure a communication free of ambiguity. What [3] complements also saying that an ontology provides a common and shared understanding of a domain, that can be communicated across people and systems.

According to [5], ontologies help in the communication process and the collaborative work between different people with different perspectives of a domain. It creates a "knowledge's unified skeleton". Still according to these authors, ontologies can work as source of learning, publishing and reference. This occurs due to the fact that ontologies are the result of a broad consensus about the structure of a domain and are able to provide reliable and objective information about this domain. Likewise, domain experts can use ontologies to share their understanding about the conceptualization and the structure of that domain.

There are several proposed languages for representing ontologies, each with its own peculiarities, but there is no consensus on the standardization of a language for this purpose. One option that points to this consensus is the Ontology Web Language (OWL), a language for knowledge representation, defined as the standard language for this purpose by [21].

## **5. Developed solution - OntoRE**

This section begins with an overview of "Ontology based Requirements Engineering" (OntoRE), to present the solution as a whole, as well as the process flow and key features developed. Then each feature is detailed, so that one can understand where and how the concepts presented in the development of this work were employed.

The process flow is depicted in Figure 1, where letters A, B, C, D, and E are used to enumerate the distinct elements of the proposed solution.



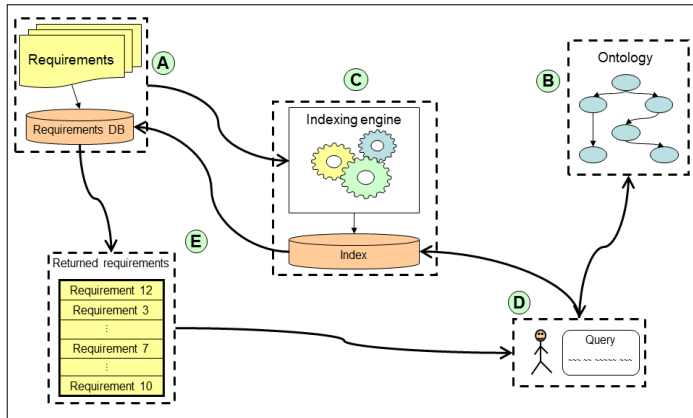


Figure 1 – OntoRe's process flow.

The first developed feature was the management of a project's SRS collection (A). We chose to use textual SRS expressed in natural language and store them in a relational database. The purpose of this feature is to create a database of the SRS collection used in a project.

In order for this system to work, an ontology (B) that represents the knowledge existent in the SRS used is required. This task aims to create a knowledge base with controlled vocabulary and free of ambiguities to the project. The ontology used was built in an external tool, because it was not the objective of this work to create an environment for ontologies maintenance.

After going through a process of data treatment, the SRS are subjected to an indexing engine (C), which results in an indexed requirements data base. At this time, the SRS may already be used by users through the features described below.

The second developed feature was a query environment (D), where a system user can retrieve the SRS of his interest. He can formulate textual queries, without necessarily using the controlled vocabulary of the project, which after being linked to the knowledge base returns the relevant SRS to the performed query. The SRS selected by the search engine are returned to the user sorted by a relevance ranking (E), so that the SRS which are likely to meet the user needs are verified first, avoiding the waste of effort on the task of finding the desired SRS.

The third implemented feature was the possibility of suggesting a new form of writing a SRS, replacing words that are not part of the controlled vocabulary of the project by terms that are. The purpose of this feature is to consolidate and reduce the SRS vocabulary in order to facilitate communication and simplify the language used among the stakeholders of a project to avoid possible communication problems.

The fourth created feature was the possibility to present to the user, SRS that possibly have similar description to another specific SRS. The purpose of this

feature is to present to the user a list of SRS that should be analyzed, because they are related to the SRS that is being analyzed.

### 5.1 Features

The initial screen of OntoRE displays a menu that allows access to features developed. The menu has the following options: Index, SRS maintenance and Search. By accessing the Index option, the SRS's database is accessed to retrieve all the SRS from the project supported by OntoRE and submit them to the indexing process.

Initially we conducted a cleaning procedure that removes unwanted characters from the SRS. Then, each SRS passes through the process of removing stopwords, to then be indexed. At the end of the process, we have an index referring to the project's SRS, to which it is already possible to submit queries and retrieve SRS.

By using the SRS maintenance option (Figure 2), it is possible to add new SRS to the project, as well as change and delete existing SRS. Through the buttons on the top of the screen, you can also navigate among the SRS of the project.

After entering a description of a new SRS or query an existing SRS, you can use the "Search similar" button, visible in Figure 2, to seek existing SRS in the database which, according to the knowledge modeled in the ontology, have some resemblance with the reference SRS.

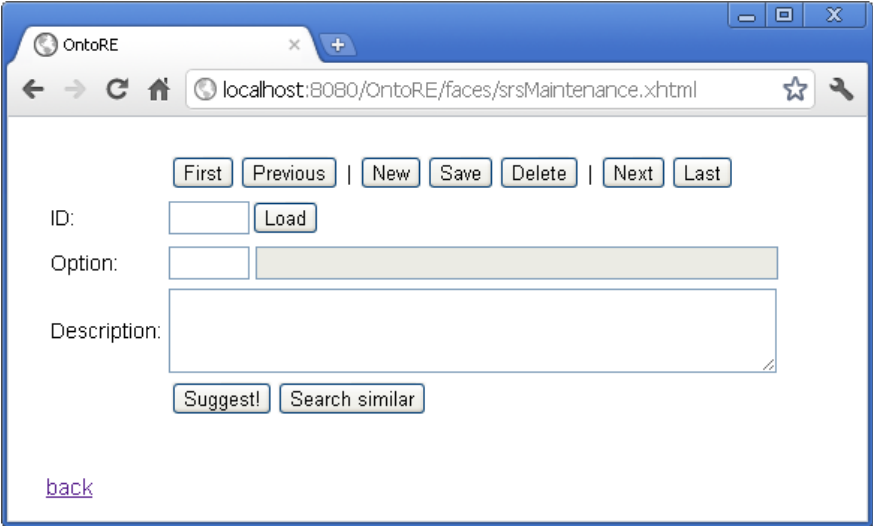


Figure 2 – SRS's maintenance.

The beginning of the process is the same of the previous functionality: the text of the SRS passes through the cleaning process and the words that refer to any term of the ontology are identified. For each identified word, the

existence of synonyms is verified in the ontology. A search is then performed by replacing each word that refers to an ontology term by all its synonyms.

It is also possible to assign a relevance weight to the sets of similar words in the query, because of the fact that if they exist in the ontology then we presuppose that they are most relevant to the context. To the search result is additionally applied a cutoff point, referring to the relevancy score given to each search result by similar SRS. This cutoff aims to display only the relevant SRS, not including those which have little similarity with the analyzed SRS.

Finally, as can be seen in Figure 3, a list of SRS identified as similar is displayed. These SRS do not necessarily describe the same functionality of the analyzed SRS. They should be analyzed, because they are related to the SRS being analyzed. The user should decide whether SRS presented are indeed relevant in the context in which the SRS is being analyzed. They work as a sort of reminder so the user can analyze the situations they describe and thus improve the SRS's analysis quality.



Figure 3 – Query sample for similar SRS.

Likewise the previous functionality, by using the "Suggest!" button, visible in Figure 2, the system suggests a new way of writing the selected SRS using controlled vocabulary terms defined in the ontology.

When the user uses the SRS similar suggestion feature, the SRS description to be analyzed is subjected to the same cleaning process used at indexing time. Then, an algorithm scans which words in the text make reference to some existing term in the ontology. For each word that exists in the ontology, it is verified which controlled vocabulary term should be used instead of that word, if there is a term. Finally, as shown in Figure 4, is presented to the user a suggested SRS description, where the changes suggested by the system are highlighted.

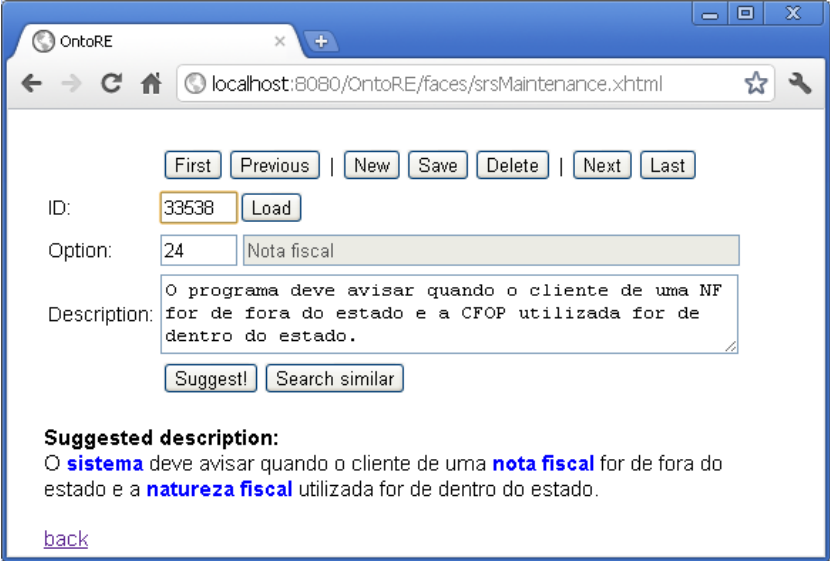


Figure 4 – Changes suggested in the SRS description.

The Search option, shown in Figure 5, has great importance to the system because it is the basis to the other developed features. The text treatment in the searching process is quite similar to the indexing process. Again, the process begins by cleaning the text that will be submitted to the search, so that special characters and punctuation are removed. After identifying the terms that exist in the system ontology, the terms of a level immediately above the search term are retrieved and added to the search. To each group of terms, the search ones and those added by the system, is given a degree of relevance. The idea behind this behavior is to add some kind of context to the search and thus make it more precise.

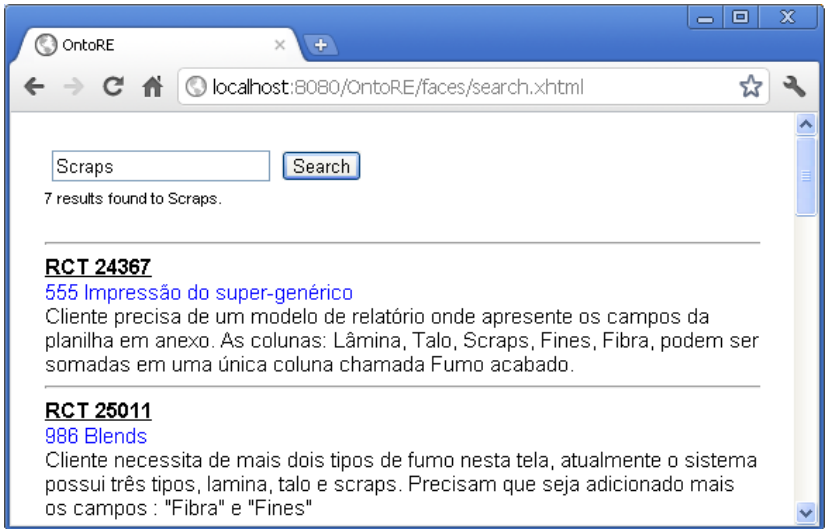


Figure 5 – SRS Query functionality.

## 6. Case study

In order to assess, in a practical way and in a real environment, the results obtained through the OntoRE, a small case study was performed. We used about 12,000 (twelve thousand) SRS from the requirements collection of an ERP project that is being developed for 5 years.

The purpose of the case study was to verify the effectiveness of the similar SRS suggestion functionality. For this, the set of SRS was added to the system and subjected to the functionality in chronological order, simulating the natural growth of the project. The result can be seen in figure 6.

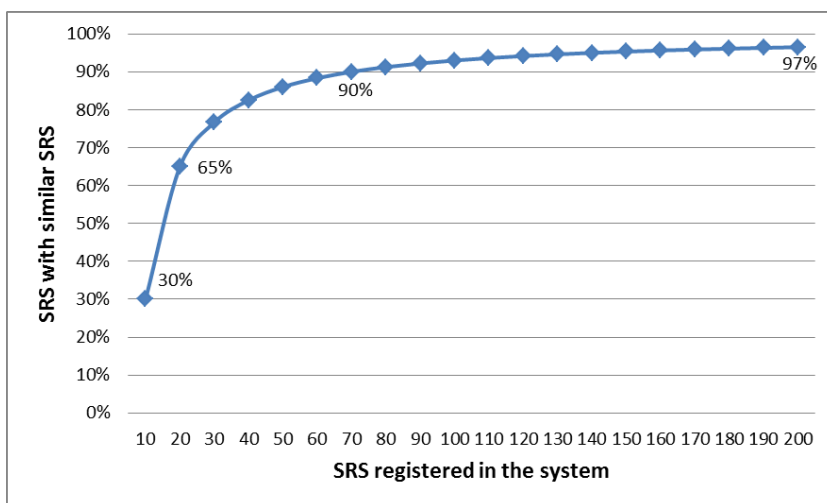


Figure 6 - Percentage of SRS with similar SRS.

For the results presentation were considered only the first 200 SRS, since it is possible to point out that practically all the SRS from the project have some kind of similar SRS. Considering the whole set of SRS used in the study, it was found that over 99% of them had at least one similar SRS.

## 7. Conclusion

The growing demand for information that serves a wide range of users boosts the emergence of large software development projects. The size of such projects is proportional to the level of importance that should be given to the tasks performed by the area of software engineering, in particular in the RE subarea, if they aspire to achieve success in the main spheres of time, cost and quality.

Analyzing the problems responsible for most of software projects problems in recent years, it is clear that the problems faced today are not related to how software should be developed, but with what should be developed. This analysis is necessary so that the end result of the developed product actually adds value to the activities of its members and not just become a digital way of bureaucratize them.

This paper introduced such problems and presented ways to deal with them in order to improve the quality of the tasks of capturing and analyzing software requirements and, thus, the quality of the produced result by the software development process. Features were developed which assist the system analyst to expand its field of view on the project requirements, thus having better information quality that it considers when making decisions about the features that should be developed.

## References

1. Baeza-Yates, R.; Ribeiro-Neto, B. (1999) "Modern Information Retrieval". ACM Press, New York.
2. Breitman, K. (2005) "Web Semântica: a Internet do futuro". LCT, Rio de Janeiro.
3. Cunha, L. M. S. (2002) "Web Semântica: estudo preliminar". Embrapa Informática Agropecuária, Campinas.
4. Fensel, D. (2001) "Ontologies: A Silver Bullet For Knowledge Management And Electronic Commerce". Springer, Berlin.
5. Gasevic, D.; Djuric, D.; Devedzic, V. (2006) "Model driven architecture and ontology development". Springer-Verlag, Heidelberg.
6. Goguen, J. (1996) "Formality and Informality in Requirements Engineering". International Conference on Requirements Engineering, Colorado Springs, EUA: IEEE Computer Society.
7. Hersh, W. R.; Detmer, W. M.; Frisse, M. E. (2001) "Information Retrieval Systems" In: SHORTLIFFE E. H. et al. Medical Informatics: Computer Applications in Health Care and Biomedicine. 2. ed. Springer p. 539-72, New York.
8. Kazmierczak, E. (2003) "Requirements Engineering". Course-work Masters - The University of Melbourne, Melbourne.
9. Kowalski, G. (1997) "Information Retrieval Systems: Theory and Implementation". Kluwer Academic Publishers, Boston.
10. Kruchten, P. (2003) "The Rational Unified Process: An Introduction". 3. ed. Addison-Wesley Professional, Boston.
11. Lancaster, F. W. (1968) "Information Retrieval Systems: Characteristics, Testing and Evaluation". John Wiley & Sons, New York.
12. Moores, C. N. (1951) "Data coding applied to Mechanical Organization of Knowledge". American Documentation, Apud (GUPTA e JAIN, 1997), v. 2, p. 20-32.
13. Moreira, A.; Alvarenga, L.; Oliveira, A de P. (2004) "O nível do conhecimento e os instrumentos de representação: tesauros e ontologias". DataGramZero - Revista de Ciência da Informação, v. 5 n. 6.
14. Pressman, R. (2001) "Software Engineering: A Practitioner's Approach". 5. ed. McGraw-Hill, New York.
15. Riloff, E. (1995) "Little words can make big difference for text classification". In: Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95). Proceedings... ACM Press. p. 130-136, Seattle, USA.
16. Sommerville, I. (2004) "Software Engineering". 7. ed. Addison-Wesley, Boston.
17. Souza, R. R.; Alvarenga, L. (2004) "A Web Semântica e suas contribuições para a ciência da informação". Ci. Inf., v 33, n. 1, p. 132-141, Brasília.
18. Thayer, R. H.; Dorfman, M. (1997) "Software Requirements Engineering". 2. ed. IEEE Computer Society Press, Los Alamitos, CA.

19. The Institute of Electrical and Electronics Engineering. (1998) "IEEE Recommended Practice for Software Requirements Specifications: IEEE Std 830-1998". New York.
20. The Standish Group. (2009) "CHAOS Report 2009". Boston, MA.
21. World Wide Web Consortium, <http://www.w3.org>.



# ISO 9001 in software-developing VSEs assisted by the COMPETISOFT Model

SILVIA ESPONDA<sup>1</sup>, ARIEL PASINI<sup>1</sup>, PATRICIA PESADO<sup>1,2</sup>,  
MARCOS BORACCHIA<sup>1</sup>  
{sesponda,apasini,ppesado,marcosb}@lidi.info.unlp.edu.ar

<sup>1</sup>Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática, UNLP, 50 y 120, La Plata, Buenos Aires, Argentina

<sup>2</sup>CIC (Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires), Argentina

**Abstract.** *A reconciliation between the COMPETISOFT software process improvement model and the international standard ISO 9001 is presented. The purpose of this reconciliation is analyzing the possibility that small software development businesses, assisted by the generation of the documents and the control of the activities defined with the COMPETISOFT Model, can obtain the ISO 9001 standard certification for Project Management and Software Development Processes.*

**Keywords:** *Software Engineering - Quality - VSEs - COMPETISOFT - ISO 9001*

## 1.Introduction

This “certification” helps organizations have an objective assessment, and its advantages [1] can be summarized as follows:

- An increase in software development process productivity achieved by decreasing re-work and “non-quality” costs.
- An increase in competitiveness, since potential customers will choose the organization based on their preference for doing business with certified organizations.

The creation of a legal framework to promote the Software Industry in Argentina, created by Act 25,922, its regulatory decree 1504/04, and Resolution 61/05 of the Secretariat of Industries of the Argentine Republic, has strongly contributed to grow awareness in relation to “Software Quality”. This legal framework allows enrolled organizations to obtain benefits, provided they meet all requirements, among them, having (Quality) improvement projects for their Software production processes [2].

Benefited by this legal framework that promotes the software industry, companies are growing their awareness and need to improve their software

development processes. If this trend grows, it may affect mainly the volume of projects that can be carried out with the desired level of quality [2]. The standards and models whose use is established by the Software Promotion Act are:

- CMM
- CMMi
- IRAM-ISO 9001 - ISO/IEC 90003
- IRAM 17601 (CMMi (SEI))
- ISO/IEC 15504 (IRAM-ISO/IEC 15504)

From these, only CMMi and IRAM-ISO 9001 are currently valid and certifiable.

CMMi (Capability Maturity Model Integration) was created in the US and is widely used in that country, as well as in many other countries, especially information technology development areas, such as India and other emerging countries. It is specifically aimed at information technology organizations. The purpose of this model is the continuous improvement of process and product quality in the organization, and it provides guidance by defining maturity levels. Access to this model is difficult for our organizations because a high percentage of information technology businesses are VSEs and the application of the model involves a significant level of investment and preparation that these businesses cannot afford [1].

ISO 9000 standards are generally applicable to any industry or organization, that is, they are not specific to information technology organizations (they have been adapted to more than 90 countries and implemented in all types of industrial and services organizations). The ISO 9001 standard defines the requirements to obtain the certification [3].

This certification is more accessible for software developing VSEs because the required level of investment to obtain it is considerably reduced in relation to costs and time, which allows the organization to increase its production level and compete in international markets.

As mentioned before, the ISO 9001 standard is generic, so the ISO standards committee published the ISO 90003:2004 standard to provide guidelines for its interpretation for software processes. For the interpretation, the ISO 9001:2000 version is used, which was replaced by a new version in 2008. The application guidelines for software (ISO 90003) have not been updated yet, but the changes introduced are not substantial in this case. The ISO 90003 standard provides guidance to identify evidence, within the software process, to meet ISO 9001 requirements [4].

Based on all this, there are two issues to take into account to correctly apply the ISO 9001 standard to software development processes:

- The ISO 90003 standard is not based on the latest version of the standard.
- It only provides guidance for the organization.

COMPETISOFT proposes solutions for process and product quality assurance, and it offers the possibility of helping software developing VSEs to start their improvement process, in particular by means of COMPETISOFT's Basic Profile. This model carries the organization through an improvement process in a staged manner, generating all the required documentation for a correct management of quality.

The idea of comparing COMPETISOFT's Improvement Model with the ISO 9001 Standard is based on all of the above.

The work group has experience in the application of the ISO 9001 Standard to the process of the pre-entry distance course of the School of Computer Science, which obtained its certification in March 2012. This application was implemented over a period of two years, during which each of the requirements of the Standard was thoroughly followed.

On the other hand, the team has a proven trajectory in the use of the COMPETISOFT Model after being part of its creation project and applying the Model to various organizations with good results.

Currently, the software processes of the Department of Medical Information Technology of Hospital Italiano de Buenos Aires are being diagnosed.

In the following sections, the models proposed by ISO 9001 and COMPETISOFT are described, followed by the reconciliation of both models to conclude, in Section 4, with the results obtained.

## **2. Process-oriented models**

### **2.1 The ISO 9001 Standard**

The first publication of the family of ISO 9000 standards dates back to 1987. They follow the ISO protocol that requires all standards to be reviewed at least every five years. The last review was in 2008 [3].

The family is basically formed by:

ISO 9000: Guidance and terminology.

**ISO 9001: Requirements**, it provides the requirements for obtaining the certification corresponding to quality systems aimed at external clients, including aspects such as design, manufacture, installation and maintenance.

ISO 9004: Guidance for continuous improvement, in general aimed at providing guidelines for the implementation of internal quality for the organization itself.

The ISO 9001 standard "Quality management systems – Requirements," specifies the requirements for quality management systems that can be used internally by organizations, for the certification, or for contractual purposes. It focuses on the efficacy of the management system to comply with customer requirements. Since 2008, it promotes the adoption a process-based approach when developing, implementing and improving the efficacy of a

quality management system in order to increase customer satisfaction by meeting customer requirements.

For an organization to operate efficiently, it has to determine and manage numerous interrelated activities. An activity or a set of activities that uses resources and is managed so that input elements can be transformed into results can be considered as a process. The application of a system of processes within the organization can be referred to as “process-based approach”.

One of the advantages of a process-based approach is the continuous control that it provides over the relations among the individual processes that form the system, such as their combination and interaction.

When this type of approach is used for a quality management system, the significance of the following is emphasized:

- a) Understanding and meeting requirements
- b) Considering processes in terms of the value they contribute
- c) Obtaining process performance and efficacy results
- d) Continuous improvement of processes based on objective metrics

## **2.2 The COMPETISOFT Model**

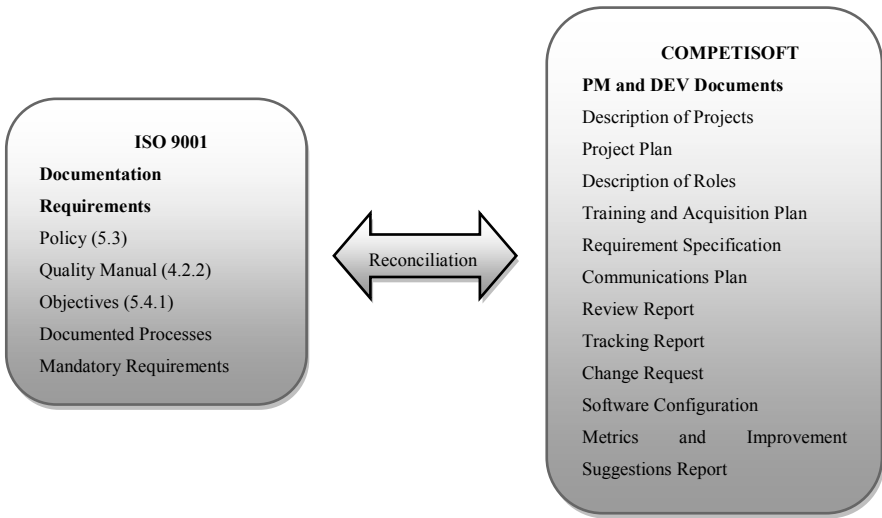
COMPETISOFT is part of various process improvement proposals for small software organizations. Its purpose is increasing the competitiveness of software producing VSEs by creating and promoting a common methodological framework that, adapted to their specific needs, can become the basis upon which an assessment and certification mechanism can be established for the software industry that is accepted throughout Ibero-America. It is based on the model defined by MoProSoft, and it is very similar to the new ISO 29110 standard, not yet certifiable [5] [6].

The model has three process categories: Upper Management, Management and Operation, reflecting the structure of an organization. When the model was applied to software development organizations in the countries involved in the project, it was observed that it still took VSEs a long time to start the improvement process. For this reason, COMPETISOFT's Basic Profile was developed. This model was well accepted by the community researching in the quality area [7] [8].

In this paper, we focus on the Operation category, which contains three defined processes: Project Management (PM), Software Development (DEV) and Maintenance (MAI).

## **3. Comparing ISO 9001 and COMPETISOFT**

Figure 1 shows a brief description comparing the documents from both models.



*Fig. 1. Reconciliation of the models*

The ISO 9001 certification requires that the Quality Manual complies with all the requirements detailed in the Standard and that all definitions included in the Manual are met, which must be demonstrated with recordable evidence. The **Quality Management System (QMS)** is detailed as from section 4 of the Standard.

First, **General Requirements** are established (section 4.1 of the Standard); this section is part of the quality manual and any software company must describe the interaction with its own processes as well as those of the QMS.

The **Documentation Requirements** (section 4.2 of the Standard) section defines QMS documents, the Policy, Objectives, Quality Manual, Documented Procedures and Mandatory Records, in addition to the documentation that the organization considers necessary to provide the service or manufacture the product. Software companies should generate all QMS-related documents and those specific to the organization based on the considerations detailed in *Document Control Procedure*. The records that will be used as evidence that the process has been effectively implemented must be controlled as defined in *Record Control Procedure*. To meet documentation requirements, the ISO 90003 standard suggests using as records the documents related to compliance with requirements, proofs of operation, retention and disposition. The documents that are generated as established by COMPETISOFT will be valid as evidence.

As regards **Management Responsibilities** (section 5 of the Standard), the only reference that the ISO 90003 standard makes to software organizations

is in relation to **QMS Planning** (5.4.2), which mentions that planning must match the development life cycle and work documents must be defined. COMPETISOFT provides information for section 5.5.1 (**Responsibility and Authority**) with the description of role competencies, and for section 5.5.3 (**Internal Communication**) by establishing that a *Communications Plan* must be defined.

Under **Resource Management** (6), subsection 6.2 (**Human Resources**), COMPETISOFT offers a *Role Description* and a *Training Plan* as evidence. In section 6.3, **Infrastructure**, the ISO 90003 standard mentions the equipment, development tools, connection infrastructure (network, Internet, firewall, etc.), product licenses, etc. These items are also covered by the *Acquisitions Plan* in COMPETISOFT. The ISO 90003 standard does not provide any interpretation regarding section 6.4, **Work Environment**. It can be defined in the Quality Manual directly.

**Product Manufacture** (7), or service provision, is the most important aspect when comparing the ISO 9001 and COMPETISOFT, since this is where the planning, design, and development process is described. Table 1 details each item in section 7 of the Standard and their match in COMPETISOFT based on the interpretation of the Standard suggested by ISO 90003. Each item can be achieved with COMPETISOFT, in some cases through the documents established by COMPETISOFT, and in others by means of a combination of QMS documents and COMPETISOFT documents. On the other hand, some of the items correspond to QMS.

# ISO 9001		Origin	Document	P M	D E V
7.1	Planning product manufacture	QMS/COMPETISO FT	Project Description - Project Plan	2	
7.2	Customer-Related Processes	QMS/COMPETISO FT	Project Plan,		
7.2.1	Definition of product-related requirements	COMPETISOFT	Requirement Specification	1	3
7.2.2	Revision of product-related requirements	COMPETISOFT	Project Plan,	3	3
7.2.3	communication with the customer	COMPETISOFT	Communications Plan	2	1
7.3	Design and development	QMS			
7.3.1	Design and development planning	COMPETISOFT	Project Description - Project Plan	8	5
7.3.2	Input elements for design and development	COMPETISOFT	Description of the Project	1	
7.3.3	Design and	COMPETISOFT	Project Plan -	5	7

	development results		Requirement Specification		
7.3.4	Design and development review	COMPETISOFT	Review Report	3	1
7.3.5	Design and development verification	COMPETISOFT	Project Plan	2	2 1
7.3.6	Design and development validation	COMPETISOFT	Project Plan	2	7
7.3.7	Design and development changes control	COMPETISOFT	Change Request	1	
7.4	Purchases	QMS			
7.4.1	Purchase process	QMS/COMPETISOFT	Acquisitions Plan	1	
7.4.2	Purchases information	COMPETISOFT	Project Plan, Acquisitions Plan	3	
7.4.3	Verification of products purchased	COMPETISOFT	Project Plan, Acquisitions Plan	1	
7.5	Service production and provision	QMS			
7.5.1	Production control and service provision control	COMPETISOFT	Project Plan, Activity Reports, Acquisitions Plan	9	4 2
7.5.2	Validation of production processes and service provision processes	COMPETISOFT	Review Report	1	3
7.5.3	Identification and traceability	COMPETISOFT	Software configuration	1	
7.5.4	Customer property	QMS	Requirement Specification		2
7.5.5	Product preservation	COMPETISOFT	Project Plan, Repository	3	
7.6	Tracking and measurement equipment control	QMS			

**Table 1 Activities and documents used as evidence for ISO 9001**

The evidence of **Planning product manufacture (7.1)** is obtained from the document *Project Description - Project Plan* that is maintained by two activities of the Project Management (PM) process. For **Customer-Related Processes (7.2)**, it is obtained from the *Project Plan*, *Requirement Specification* and *Communications Plan* documents that are related to a total

of 6 activities of the PM process and 7 of the Software Development (DEV) process. **Design and Development** (7.3) is achieved through *Description of the Project, Project Plan, Requirement Specification, Review Report* and *Change Request* generated and maintained by 7 PM activities and 42 DEV activities. **Purchases** (7.4) is related to 5 PM activities and recorded in documents from *Project Plan and Acquisitions Plan*. **Service production and provision** (7.5) is documented better by COMPETISOFT through the *Project Plan, Activity Reports, Acquisitions Plan, Review Report, Software Configuration* and the *Repository*, managed by a total of 14 PM activities and 45 DEV activities.

**Service provision and measurement** (8) is mainly linked to ISO 9001, but COMPETISOFT provides a lot of information for the sections on **Tracking and measurement** (8.2) by means of documents *Tracking Report and Acceptance Documents*, and **Data Analysis** (8.4) by adding documents *Metrics and Improvement Suggestions Report and Lessons Learned*.

### Documented Procedures and Mandatory Records.

In section 4.2.1, mandatory documents are defined and documented procedures and mandatory records are mentioned. The Standard requires 6 documented procedures and 19 mandatory records. The documented procedures are more in relation to the QMS than software processes, since they detail and regulate their operation. The mandatory records correspond to sections 5.6.1, 6.2.2, 7.1, 7.2.2, 7.3.2, 7.3.4, 7.3.5, 7.3.6, 7.3.7, 7.4.1, 7.5.2, 7.5.3, 7.5.4, 7.6, 8.2.2, 8.2.4, 8.3, 8.5.2, 8.5.3. Most of the records corresponding to section 7 are covered by the documentation generated by COMPETISOFT, which would also provide information regarding sections 6.2.2 and 8.2.4. The remaining records are related to managing the QMS itself. Table 2 details the relationship between the mandatory records and the documents that provide the evidence.

#	Mandatory Records, ISO 9001		Origin	Documents
1	5.6.1	Review by Management	QMS	
2	6.2.2	Qualification, Training, Awareness	QMS/COMPETISOFT	Description of Roles - Training Plan
3	7.1	Planning product manufacture	COMPETISOFT	Project Description - Project Plan
4	7.2.2	Revision of product-related requirements	COMPETISOFT	Project Plan,
5	7.3.2	Input elements for design and development	COMPETISOFT	Description of the Project
6	7.3.4	Design and development review	COMPETISOFT	Review Report



7	7.3.5	Design and development verification	COMPETISOFT	Project Plan
8	7.3.6	Design and development validation	COMPETISOFT	Project Plan
9	7.3.7	Design and development changes control	COMPETISOFT	Change Request
10	7.4.1	Purchase process	COMPETISOFT	Acquisitions Plan
11	7.5.2	Validation of production processes and service provision processes	COMPETISOFT	Review Report
12	7.5.3	Identification and traceability	COMPETISOFT	Configuration Management
13	7.5.4	Customer property	QMS	
14	7.6	Tracking and measurement equipment control	QMS	
15	8.2.2	Internal Audit	QMS	
16	8.2.4	Product tracking and measurement	QMS/COMPETISOFT	Tracking Report
17	8.3	Non-compliant product control	QMS	
18	8.5.2	Corrective actions	QMS	
19	8.5.3	Preventive actions	QMS	

**Table 2.** Mandatory records in ISO 9001 and their relation to COMPETISOFT documents

#### 4. Conclusions and future work

A view of the ISO 9001 standard was presented with the guidelines for its application to the software development process as interpreted by ISO 90003. On the other hand, the document generated by the COMPETISOFT model and the activities that manage them were analyzed in detail.

From this analysis, it can be concluded that COMPETISOFT would generate all the necessary documents to comply with the requirements of section **Product Manufacture** (7) and that it would provide information to cover the requirements described in sections 5, 6 and 8. Therefore, COMPETISOFT would be a good tool to introduce software organizations to the improvement process needed to achieve an ISO 9001 certification.

We continue to analyze improvement models and helping organizations that want to optimize their processes. In particular, organizations that are using

the COMPETISOFT model to achieve this goal and are interested in obtaining a certification, such as Hospital Italiano de Buenos Aires. On the other hand, assistance is offered to software VSEs from the “La Plata Information Technology Development Area” that are interested in benefiting from the software promotion act.

## References

1. Piattini, Garcia, Caballero, “Calidad de Sistemas Informáticos”, Editorial Ra-Ma, Año 2007
2. Ley 25.922 - LEY DE PROMOCION DE LA INDUSTRIA DEL SOFTWARE  
[http://www.mincyt.gob.ar/multimedia/archivo/archivos/ley\\_25922\\_3.pdf](http://www.mincyt.gob.ar/multimedia/archivo/archivos/ley_25922_3.pdf)
3. IRAM-ISO. 9001:2008, “Sistema de gestión de la calidad. Requisitos” Segunda Edición 2008 IRAM
4. ISO/IEC 90003:2004, “Software engineering — Guidelines for the application of ISO 9001:2000 to computer” 2004, ISO
5. ISO/IEC 29110:2011, “Software engineering -- Lifecycle profiles for Very Small Entities (VSEs)” 2011, ISO
6. Oktaba Hanna, Alciquira Esquivel, Ramos. Modelo de Procesos para la Industria de Software versión 1.3, AÑO 2005
7. Piattini, Oktaba, Orozco, “COMPETISOFT. Mejora de procesos software para pequeñas y medianas empresas”, Editorial Ra-Ma, Año 2008
8. COMPETISOFT, Mejora de Procesos para Fomentar la Competitividad de la Pequeña y Mediana Industria del Software de Iberoamérica, Diciembre 2008

# Histogram-based intrinsic plagiarism detection

DARIO G. FUNEZ AND MARCELO L. ERRECALDE

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)  
Facultad de Cs. Físico, Matemáticas y Naturales, Universidad Nacional de San Luis  
e-mail: {dgfunez, merreca}@unsl.edu.ar

**Abstract.** *Intrinsic plagiarism detection (IPD) uses stylistic variations to detect sections of a text which are suspected of being written by a different author. In the present work, the feasibility of using local histograms to model the author's writing style and the detection of outliers is analyzed. In this context, detection of outliers is a key subtask of IPD which aims at detecting evident style changes in the histogram. The proposed approach, which was tested on the corpus of the Plagiarism Detection Competition PAN-PC-2011, shows an acceptable performance in comparison to other detectors participating in the competition.*

**Keywords:** *intrinsic plagiarism detection, histograms.*

## 1. Introduction

*Text plagiarism* refers to the use of another author's information, language, or writing, when done without proper acknowledgment of the original source [10].

For instance, a very frequent type of plagiarism is the exact copy of a text portion ("copy-paste" type) which is considered as the easiest to be detected. It is widely recognized the importance of providing tools that allow users *automatically* detect plagiarism [1]. The techniques needed in these cases are usually more elaborated than the required to detect the "copy-paste" type. This is evident if we consider that plagiarists also use more elaborated techniques to copy texts without being discovered such as changing the order of words, using synonyms, replace long sentences for shorter ones with the same meaning, etc.

Plagiarism detection is usually classified into two main types: *external* and *intrinsic* [10]. The former assumes that a reference collection of documents is given besides the document to be analyzed. The result in this case is the plagiarized sections of the document and the corresponding sections of the documents in the reference collection. The latter uses a completely different approach where no reference collection is given and plagiarized sections are identified by detecting style variations in the document. The output of the detector in this case corresponds to those sections which show an evident change in style with respect to the style of the remaining sections ("outlier detections").

Intrinsic plagiarism detection is an important and relatively new task that has gained increased interest in several research groups as it has been shown in PAN-PC-2011, a recent international competition on plagiarism detection. In this competition, the best performance was achieved by [9] with an approach that uses parameter's deviation of text segments with respect to the writing style of the whole document. The comparison among the segments and the whole document is implemented with vectors of word frequencies. A model (reference value) of the author's style is obtained on the whole document and those segments that significantly differ from this value are considered as plagiarized.

In this same competition, Kestemont [6] obtained the second best position with a stylographic model based on character tri-grams frequencies. His proposal involves using the more frequent tri-grams and a matrix of distances. Outlier detection is carried out by comparing the distance among each text window and the remaining windows.

On the other hand, in [4] very good results are obtained with histograms in authorship attribution tasks and the authors highlight the appeal of this technique to model writing style in plagiarism detection. Histograms are obtained with the *lowbow* framework [7], which has been used in different domains such as visualization of documents, segmentation of movies in chapters and scenes and discourse segmentation among others.

In this work, we focus on this research line and analyze the feasibility of using histogram-based representations in intrinsic plagiarism detection. To the best of our knowledge, this approach has not been previously used to model writing styles in intrinsic plagiarism detection tasks. The stylographic model in this case, significantly differs from other models previously used. It has available much more information about the whole graphic represented by the histogram, not only about isolated points as most popular outlier detection methods do.

Usually, an intrinsic detector needs to define the following three stages: text decomposition, style model construction and outlier identification [10]. In our work, the first two stages are carried out by the *lowbow* framework, which is provided with the number of samples determined by a text segmentation method; then the outlier identification selects the extreme points in the histogram graphic.

As we will see in Section 6, the obtained results with the PAN-PC-2011 corpus give firm evidence that using histograms for stylographic modeling is a feasible and interesting option.

The rest of this article is organized as follows. In Section 2 general aspects of intrinsic detection are introduced. Section 3 explains how the text segmentation algorithm used in our proposal works. The framework *lowbow* is described in Section 4. The implementation aspects of the proposed detector are presented in Section 5. The experimental work and the obtained results are described in Section 6. Finally, Section 7 draws some conclusions and future works.

## 2. Intrinsic Plagiarism Analysis

The intrinsic plagiarism detection problem may be posed as follows: given a text  $t$ , allegedly written by an author “A”, identify those sections written by an author “B”, “B”  $\neq$  “A” [10]. Obviously, it is assumed that the document  $t$  was written by only one author. Those cases where different authors collaboratively wrote the text would be very difficult to be detected.

Intrinsic plagiarism detection has gained increasing interest due to its few requirements to be implemented. It does not require any reference collection with potential sources for the plagiarized sections. This collection sometimes is impossible to be obtained because a lot of information is not available in the Web (or other digital media) so it is not possible to carry out the comparison against those files. Nevertheless, a drawback of this approach is the impossibility of clearly showing the plagiarism evidence because the source documents are not available.

Intrinsic plagiarism analysis is based on the idea that each author has its own writing style which is kept unchanged along the whole document [2]. This writing style of the author needs to be represented by a *model* that usually contains the *stylographic measures* that characterize the personal writing style of the author [10]. Intrinsic detection uses substantial variations from the model to obtain plagiarized sections. In this problem, the only class that is known is the one corresponding to the writing style of the author and, in consequence, it is considered as an *one class* classification problem [11] and the remaining writing styles are named *outliers*.

In [10], intrinsic detection task is considered a process composed by three different subtasks: a) *decomposition strategy*, b) *Style model construction* and c) *outlier identification* that are described below.

### 2.1 Decomposition strategy

The whole document needs to be split to obtain writing style information from different points in the document. This decomposition strategy must be carefully selected because the detector’s performance heavily depends on this stage.

Simple strategies, such as dividing the text in equal-length blocks, can be easily implemented and efficient in time execution. However, such simple strategies not always obtain good performances. Other strategies, like dividing the document in *structural* limits (for instance, *paragraphs* or *sentences*) may be a better option, because plagiarized sections could be complete paragraphs. An alternative that has achieved interesting results is to divide the text into cohesive segments using an algorithm of *text segmentation*, as we have previously proposed in [5].

## 2.2 Style model construction

To detect variations in the writing style, it is needed a model with stylographic information of the document's author. A writer, unconsciously maintains the same style throughout his writings. Therefore, a significant style variation from the model in a particular paragraph makes it candidate to be written by a different author. The most common stylographic models are built with a set of stylographic measures such as *readability indices*, frequencies of class of words like *adjectives* and *nouns*, the *richness of vocabulary*, etc. [10]. However, an interesting option, and the one used in our work, is to represent writing styles with local histograms of words. A similar approach, with character *n*-grams, has shown being effective in authorship attribution problems [4].

## 2.3 Outlier identification

With the information from the previous step, the writing style of the author is characterized in different parts of the text. In a text written by a single author, the writing style should remain the same with minor alterations throughout the entire text. His model is the only information that is available and thus it defines a *one class* classification problem [11], because it is not possible to characterize the writing style of all the remaining possible authors in the negative class. This type of problems are also called as *outlier detection* as they must choose those points in the text where the writing style is significantly different from the other samples taken from the text [10]. The outlier detection methods can be classified into three categories:

- *Density methods*, which approximate the probability density function of the target class. Outliers are considered uniformly distributed and the Bayes rule can be used to differentiate outliers from objects in the target class. These methods provide good results when the size of the sample is large.
- *Methods with a limit*, which try to define a region using distances between the elements in the target class. Outliers are those objects which are not included in this region.
- *Methods of reconstruction*, which need background knowledge about how the target objects are generated. Outliers are those objects that are hard to rebuild.

In the next subsection, some evaluation measures used to quantify the performance of a plagiarism detector are defined.

## 2.4 Evaluation measures

In order to evaluate the performance of a plagiarism detection algorithm, the *precision*, *recall* and *granularity* of the detections made by the algorithm need to be computed [8]. To define these measures we will use the following notation: 1)  $s$  represents a plagiarized section from the set  $S$  of all the plagiarized sections, 2)  $r$  denotes a detected section from the set  $R$  of all the detected sections, 3)  $S_R$  are the plagiarized sections that have been detected, 4)  $|s_i|$  and  $|r_i|$  denote the size (in number of characters) of the corresponding section and  $|S|$  and  $|R|$  denote the number of elements of these sets. Finally,  $\alpha(s_i)$  is the number of detected characters in  $s_i$ ,  $\beta(r_i)$  is the number of plagiarized characters in  $r_i$  and  $\gamma(s_i)$  is the number of plagiarized characters detected in  $s_i$ . These measures are used to define precision, recall, granularity, and global evaluation (*overall*),<sup>1</sup> as follows.

$$recall = 1/|S| \sum_{i=1}^{|S|} \alpha(s_i) / |s_i| \quad (1)$$

$$precision = 1/|R| \sum_{i=1}^{|R|} \beta(r_i) / |r_i| \quad (2)$$

$$granularity = 1/|S_R| \sum_{i=1}^{|S_R|} \gamma(s_i) \quad (3)$$

$$overall = F/(\log_2(1 + granularity)) \quad (4)$$

These measures can be interpreted as follows. Precision is the percentage of correct detections; recall is the percentage of plagiarism detected; a granularity close to 1 means that the algorithm will detect each plagiarism at most once. In all these measures, values close to 1 indicates a good behavior of the detection algorithm.

## 3. Text Segmentation

Text segmentation divides a text into units with the same topic. In [3], Freddy Choi implements this task in two phases on the full text. In the first stage, the *stops words* (articles, prepositions, connectors, etc.) are removed because they do not provide relevant information. Then, the stem of each word is

---

<sup>1</sup> In the global evaluation,  $F$  refers to the traditional  $F$ -measure, the harmonic mean of precision and recall:  $F = 2 * (precision * recall)/(precision + recall)$

obtained and its frequency in the text is stored in a vector. Thus, each sentence has associated a vector of frequencies and the frequency of the word  $j$  in the sentence  $i$  is denoted as  $f_{ij}$ .

The matrix resulting of applying the cosine similarity to each pair of vectors is named the *similarity matrix* [3]. In the graphics of this matrix, bright square regions along the diagonal represent cohesive text segments whose limits are hard to determine. To make this task easier, a *ranking* scheme is applied to the similarity matrix that consists in obtaining a rank matrix whose values are obtained by moving a *mask* (small square matrix) on the similarity matrix. To compute each entry in the rank matrix, all the values included in the mask are considered. The rank value ( $r$ ) for a position is obtained with the formula  $r = lvalue/N$ , where  $lvalue$  is the number of elements with low similarity and  $N$  is the total number of examined elements.

With this rank matrix, the last stage of the segmentation algorithm uses the Reynar's maximization algorithm to determine the boundaries of the segments. Before explaining this procedure, we will introduce the concept of *inside density* that will be defined using the following notation:

$s_{ij}$  : the sum of rank values in a segment between sentences  $i$  and  $j$ .

$a_{ij}$  : inside area of a segment, computed as  $a_{ij} = (j - i + 1)^2$ .

$B = b_1 \dots b_m$  is the list of  $m$  coherent segments.

$s_k$  y  $a_k$  are the sum of rank and area of segment  $k$  in  $B$ .

The *inside density* of  $B$  is now defined as:

$$D = \sum_{k=1}^m s_k / \sum_{k=1}^m a_k$$

The process first initializes  $B$  with a single segment that represents the whole document. Each step of the algorithm splits one of the segments in  $B$  selecting the split point that maximizes  $D$ . The number of segments  $m$  is automatically determined by using unusual variations in the gradient. The gradient is defined as follows:

$$\delta d^{(n)} = d^{(n)} - d^{(n-1)}$$

where  $d^{(n)}$  is the inside density of  $n$  segments. For a document with  $b$  potential boundaries, if  $u, v$  are the the mean and variance of  $\delta d(n)$ , with  $n \in 2, \dots, b + 1$ ,  $m$  is obtained by applying the threshold  $u + c \sqrt{v}$  to  $\delta d$ . A value  $c = 1.2$  is usually used in practice.

## 4. Lowbow

Lowbow is a framework implemented in Matlab that provides a sequential, continuous and differentiable representation of a document [7]. It has been used in authorship attribution problems with good results using local histograms of character



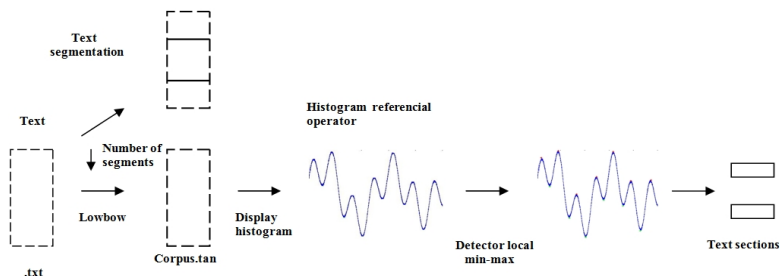
$n$ -grams [4]. Lowbow requires as input: a) the document to be analyzed, b) the number of parts or samples in which lowbow will split the document and c) a set of *kernels* whose values vary between 0 and 1.<sup>2</sup> Given an input text, this framework allows to obtain its histograms with different kernels. A kernel with values near to 0 preserves sequential information and this information is lost as these values increase. The vocabulary might also be provided to the framework, but this is an optional parameter that it is automatically obtained by the framework if it is not specified. Lowbow returns different files with distinct information about the histograms:

- Number of samples and kernels (*lowbow.info* file). It is used to display the histograms.
- Information about principal component analysis (*\*.proj* files).
- Information about speed, curvature, tangent vector, etc (*\*.tan* files).

Histograms-based representation gives sequential information, which is important to model the author's writing style that will be used in the detector explained in the next section.

## 5. Detector description

In Figure 1 all the steps involved in our proposal of intrinsic plagiarism detection are shown.



*Fig. 1. Detector's architecture.*

To use lowbow, the number of samples to be considered needs to be specified. This value is obtained by applying the text segmentation algorithm explained in Section 3 to the document to be analyzed, because it allows determining the number of segments in it. Different experiments were carried

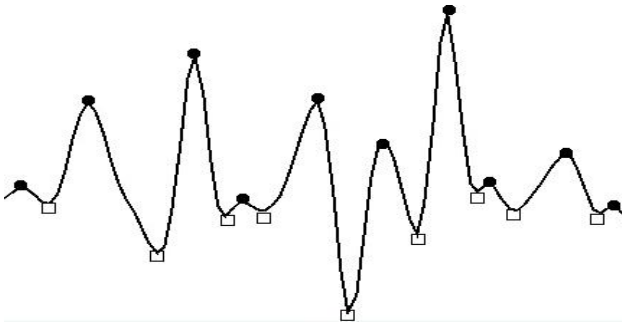
<sup>2</sup> Space constraints prevent us of giving a more detailed explanation of kernels, but the interested reader can find more details in [7].

out with distinct text decomposition techniques, such as splitting the text in uniform size segments and paragraphs but they obtained a lower performance. Therefore, considering the good results with text segmentation we had already obtained in [5], we used the Freddy-Choi's segmentation algorithm included in the Morphadorner library.<sup>3</sup>

Then, lowbow is executed with the following parameters: a) the number of samples which was determined by the previously explained procedure and b) the kernel that was experimentally determined with a 0.03 value. Standard lowbow produces, with those parameters, different output files corresponding to different histograms. From all the generated histograms, that one that represented the stylistic model the best was selected in a set of reference files. Figure 2 shows a histogram obtained by lowbow which, as it can be easily observed, corresponds to a continuous function.

Outlier detection identifies those text sections that produce significant variations in the histogram graphic. That was implemented with the *Peakdetect* script in Matlab, which detects peaks in a graphic, that is, local maximums and minimums, as it is shown in Figure 2. All segments in the texts that show those anomalous behaviors are suspected of plagiarism.

Once the sections provided by the outlier detector are available, it is necessary to verify if there are adjacent sections to join them into a single section, and get a better granularity. The detector output is a *.xml* file with information about the sections of the document that are suspected of plagiarism. Each section consists of information which is positioned in the text, that is, the offset from the beginning of the text in number of characters and its size.



*Fig. 2. Histogram of words.*

Below the differences between the proposed detector and the one presented in [5] are described:

---

<sup>3</sup> *Morphadorner* is an open-access Java library for PLN made available by the Northwestern University.

- Text segmentation is used to obtain the number of segments in the text required by lowbow, but not the sections provided by the algorithm.
- The stylistographic model of the whole text is represented by a single histogram. In [5] instead, a set of stylistographic measures are carefully selected for each segment in the document.
- Outlier detection is implemented with a new approach that detect outlier points when variations in the histogram graphic are observed. In the previous approach, outlier detection was implemented by using a Meda-based method.

## 6. Experiments

To evaluate the behavior of our proposed detector, the PAN-PC-2011 corpus provided for the competition was used [8]. Besides, the *perfmeasure.py* Python script used in this competition computed the precision, recall, granularity and *Plagdet score* or overall of the obtained results. This corpus is composed by 4.753 files that vary in length and the number of plagiarized sections. In the following table, the results obtained with the corpus of the competition are shown.

plag-det	Recall	Precision	Granularity
0.088486	0.167477	0.063944	1.064707

**Table 1.** Results with the PAN-PC-2011 corpus.

Below in Table 2, the results of the competitors in the intrinsic plagiarism competition of PAN2011 are shown.

Puesto	plag-det	Recall	Precisión	Granularidad
1	0.3254817	0.3397965	0.3123243	1.0000000
2	0.1679779	0.4279112	0.1075817	1.0329386
3	0.0841286	0.1277831	0.0664302	1.0549085
4	0.0693820	0.1080543	0.0783903	1.4787234

**Table 2.** Results of the participants in the PAN-PC-2011 competition.

As it can be observed, our detector would obtain the third position, with a better *recall* than the one obtained by the method in this position. However, one of the drawbacks of our proposal is that in many cases it returns plagiarized sections in documents without plagiarism (false positives). This problem decreases the global performance of the detector because the corpus has a high percentage of free-plagiarism files.

## 7. Conclusions and Future Work

From the experimental work, it can be concluded that there exists firm evidence of the feasibility of using histograms to model writing styles. While the results presented are preliminary and may be improved in the future, the proposed approach has shown being competitive with respect to other algorithms representative of the state of art in intrinsic plagiarism detection.

The proposed algorithm is fast because obtaining the model only requires recovering the histogram of the whole document. Outlier detection also is very simple and efficient and successfully contributes to a good global behavior of the detector.

As future work, an alternative would be using character *tri-grams* as vocabulary in lowbow instead of full words, an idea that has already produced good results in authorship attribution [4]. A second improvement is the modification of the lowbow code in order to use the information about the sections provided by the text segmentation algorithm. Finally, our immediate goal will be focused on improving the detector's performance with those documents without plagiarism, by using in this case a more accurate analysis of the variations in the peaks of the histograms.

## References

1. Enrique Vallés Balaguer. Empresa 2.0: Detección de plagio y análisis de opiniones. Master's thesis, Universidad Politécnica de Valencia, 2011.
2. Chien-Ying Chen, Jen-Yuan Yeh, and Hao-Ren Ke. Plagiarism detection using rouge and wordnet. *CoRR*, abs/1003.4065, 2010.
3. Freddy Y.Y. Choi. Advances in domain independent linear text segmentation. In *ANLP*, pages 26–33. The first conference on North American chapter of the Association for Computational Linguistics, Morgan Kaufmann, 2000.
4. Hugo Jair Escalante, Tamar Solorio, and Manuel Montes y Gómez. Local histograms of character n-grams for authorship attribution. In *ACL*, pages 288–298. The Association for Computer Linguistics, 2011.
5. Dario G. Funez and Marcelo L. Errecalde. Un ambiente de ejecución para la detección de plagio intrínseco usando la segmentación de texto. *Cacic 2011*, 2011.
6. Mike Kestemont, Kim Luyckx, and Walter Daelemans. Intrinsic plagiarism detection using character trigram distance scores. In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.
7. Guy Lebanon, Yi Mao, and Joshua Dillon. The locally weighted bag of words framework for document representation. *Journal of Machine Learning Research* 8(2007) 2405-2441, 2007.

8. Alberto Barrón-Cedeo Benno Stein Martin Potthast, Andreas Eiselt and Paolo Rosso. Overview of the 3rd international competition on plagiarism detection. *Notebook Papers of CLEF 2011 Labs and Workshops*, 2011.
9. Gabriel Oberreuter, Gaston LHuillier, Sebastián A. Ríos, and Juan D. Velásquez. Approaches for intrinsic and external plagiarism. In *CLEF (Notebook Papers/Labs/Workshop)*, 2011.
10. Benno Stein, Nedim Lipka, and Peter Prettenhofer. Intrinsic plagiarism analysis. *Language Resources and Evaluation*, 45(1):63–82, 2011.
11. David M. J. Tax. *One-class classification; Concept-learning in the absence of counter- examples*. PhD thesis, Delft University of Technology, 2001



# Automatic Text Classification through Word Similarity and Hidden Meanings

FABRICIO S. CATAE AND RICARDO L. A. ROCHA

Languages and Adaptive Techniques Laboratory  
Computer Engineering Department  
University of Sao Paulo (USP)  
{fcatae,rlarocha}@usp.br

***Abstract.** Latent Semantic Indexing (LSI) is a method to classify text using similarity metrics. The relationship among the documents is not restricted only by specific word co-occurrence, and LSI successfully identify the hidden similarities by meaning. In the beginning of LSI processing, there is a matrix dimensionality reduction to the commonly adopted value between 200 to 300 dimensions. In this paper, we explore the dimensionality choice and convert the LSI into a supervised learning algorithm in order to determine the ideal number of dimensions dynamically. The proposed work also move the actual coordinates into an isometric space in order to optimize the speed of the overall process.*

***Keywords:** clustering, information retrieval, latent semantic indexing.*

## 1. Introduction

Text classification organizes the available information into different categories to assist potential search capabilities. Besides that, document classification also assists in data mining, detection of adult content, spam detection, and other tasks. When a human is assigned as the subject matter expert, the decision making process depends on the knowledge and personal sense to define under which category the document belongs. The classification process may be streamlined as an automatic process in order to improve the precision and reduced the subjectivity associated with the manual process.

In the field of information retrieval, the content is usually represented as a non-ordered set of words, which is called bag of words (BOW). Salton [15] introduced the idea of representing documents as vectors, in which its components are calculated based on the weighted word count. These vectors are represented in a structure called Term-Document matrix. In this representation, we can find out the related documents by identifying specific words that are usually present in that same category. For example, since the terms “engine” and “pilot” frequently show up in topics related to “cars”, they have likely the same classification.

Naïve Bayes [13] is another way to classify documents by assigning probabilities to the word frequency, and then calculate the probability of a

certain document to be in a specific category. This model assumes the dimension independence, even though this is far from the truth. For example, when we refer to the term “car”, we are also referring to documents with the word “vehicle”. The words have inter-relationships that affect the counting, so the model has to consider that.

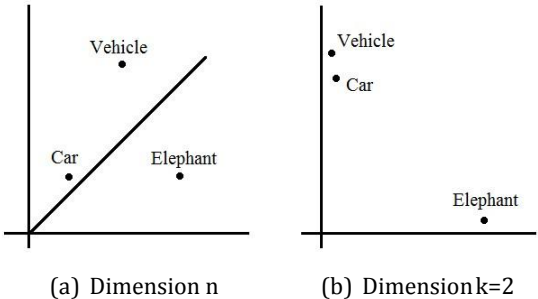
### 1.1 Latent Semantic Indexing

Landauer [10] searched for a relationship between the terms and the hidden meanings, which would be the “latent semantic” piece. That means the document is not limited only by the word presence, but the classification also considers the implicit meanings. The relationship dependency is modeled using a rotation of the space, and the actual vectors are transformed into new vectors. For example, consider the following words from a different perspective.

- (a) Car = 1.0 \* driver + 4.0 \* wheels + 2.0 \* doors + 1.0 \* engine
- (b) Vehicle = 1.0 \* driver + 4.0 \* wheels + 2.0 \* doors + 1.0 \* engine

In this case, even though the terms “car” and “vehicle” are written differently, their components make them equivalent. The latent semantic indexing (LSI) projects the vectors into a lower dimension space, also called as semantic space [6]. Because of it the word similarity are projected in the same space dimension.

For example, for k=2 dimensions, the matrix A is projected onto a plane. The points (“car”, “elephant” e “vehicle”) represent the documents, and they would be automatically grouped by similarity in this 2-dimension space (Figure 1).



**Fig. 1.** (a) Documents are randomly dispersed in the space; (b) From the projection perspective the documents are grouped.

Papadimitriou [14] provides a partial formal proof of this automatically grouping, and it could be extrapolated for the general case.



## 1.2 Singular Value Decomposition

LSI is based on linear algebra [4] and matrix decomposition. The idea is to obtain an approximation of the matrix  $A$  using the most significant singular values. Given the  $m \times n$  matrix  $A$ , there always is a matrix factorization of the form:

$$A = U\Sigma V^T$$

Where  $U$  and  $V^T$  are orthonormal matrixes ( $U^TU = V^TV = I_n$ )  $\Sigma$  is the diagonal matrix  $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ . The factorization is unique when the singular values  $\sigma_i$  are ordered from the highest to the lowest value.

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$$

The matrix  $A$  has a rank  $r$ , which it is defined as  $\text{rank}(A) = r$ .

$$\begin{cases} \sigma_k > 0 \text{ para } 1 \leq k \leq r \\ \sigma_{r+1} = \dots = \sigma_n = 0 \end{cases}$$

$A_0$  is a good approximation of  $A$  when the matrix  $\Sigma$  is left with the  $k$  highest singular values. Given  $U_0$ ,  $V_0$  and  $\Sigma_0$  as the truncated versions of  $U$ ,  $V$  and  $\Sigma$  matrices. Then,  $A_0$  is given by:

$$A_0 = U_0 \Sigma_0 V_0^T$$

For any given matrix  $M$  such that  $\text{rank}(M) = k$ , the error is calculated using the Frobenius form:

$$\|A - M\|^2 \geq \|A - A_0\|^2 = \sigma_{k+1}^2 + \dots + \sigma_n^2$$

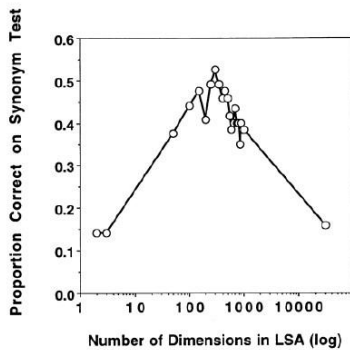
According to Eckart-Young Theorem [2], the matrix  $A_0$  is the best approximation of rank  $k$ .

## 1.3 k Parameter

Deerwester [4] conducted tests with MED dataset, which contains medical records. The test was to vary the number of dimension between 10 and 100 in order to find out the best precision. In the end, it was reported that the best result was achieved with 100 dimensions. According to the LSI reports in the Text Retrieval Conference (TREC), the number of dimension directly impacts the recall and precision of the documents [7][8][9]. In the first conference edition, it was used between 235 and 250 dimensions. In TREC-2, the best result was using between 199 and 204 dimensions. In the following edition, TREC-3, the experiment ran with different values for the  $k$  parameter: 199, 250, 300 and 346 dimensions.

As the number of dimension decreases (parameter  $k$ ), the document recall increases. On the other hand, spaces with excessive dimension reduction suffers from precision issues. Determining the correct number of dimension is a challenge. Dumais [5] suggests using a space 200 to 300 singular values, and after that there seems to be a cut point for the  $k$  parameter, in which the result is degraded as the dimensionality increases.

Landauer verified the impact of the number of dimension through an experiment with 80 questions from the ETS Test of English as a Foreign Language (TOEFL) [11]. The corpus was composed of available newsletter text from the Associated Press newswire, Grolier's Academic American Encyclopedia, and collection of children's reading. The  $k$  parameter was varied from 2 to 1032, achieving the best result at 300 and 325 dimensions.



*Fig. 2. The influence of  $k$  parameter in the synonym test using the TOEFL exam. It demonstrates the importance of the adequate choice of the number of dimensions.*

## 2. Objective

The goal of this research is to determine the ideal  $k$  parameter, which corresponds to the number of dimensions applied in LSI. The number of dimension directly affects the document precision and recall. Despite its importance, we have not found a reference on how to determine the ideal value for the dimension in a categorization task [3].

LSI is a non-supervised learning algorithm. During the training phase, both the training and the test documents are considered as a single set. The singular value decomposition is applied to this matrix to find a set of unitary and orthogonal vectors. The largest singular values will define the base vectors of the subspace representation. Until then the category has not being considered yet.

There LSI can be turned into a supervised learning algorithm. During the initial phase, we vary the  $k$  parameter and calculate the rate of successful

document classification. Then, we draw the function  $T(k)$  to represent the classifier precision against the number of dimensions. Assuming that the sets of training and tests are truly random, we can infer that the model will bring similar results for any set. The hypothesis is that  $T(k) = P(k)$ , where  $P(k)$  is the classifier precision using the test dataset. If the hypothesis is true, then we choose the  $k$  that maximizes the function  $T(k)$ . Therefore, the training dataset is used to calibrate the ideal  $k$  parameter, and it will also maximize the classifier with the test dataset.

### 3. Experiment

The extracted terms from the text collection are transformed into vectors with TF-IDF weighting. Despite the potential improvements, we have not considered any specific word treatment such as stop words, stemming, composite words, syntax validation and dictionary lookups. Terms with less than 3 characters or with less than 10 occurrences were dismissed.

The vectors created from the training and data sets are laid into a  $m \times n$  Term-Document matrix. During the experiments, we assumed the condition  $m > n$ . The singular value decomposition was done with SVDPACKC library [1] to factorize it into the matrices  $U$ ,  $S$  and  $V$ . The singular values were then truncated, and the space was reduced to only  $k$  dimensions. There we got to the matrices  $U_0$ ,  $S_0$ ,  $V_0$  and the matrix  $A_0$  of rank  $k$ :  $A_0 = U_0 S_0 V_0^T$

$A_0$  is an approximation for the Term-Document matrix and it will be used for prediction. Also in the training phase, the documents are classified using the nearest neighborhood method (kNN,  $k=1$ ). The similarity is given by the cosine of the angle formed between the corresponding vectors.

The process is repeated by varying the number of dimension between 2 and 1000 (we adopted this arbitrary value as the maximum). The result was a  $T(k)$  curve: precision x dimensions, based on the training dataset. We chose the  $k$  parameter that maximized the function  $T(k)$ , and the training phase is finished.

In order to validate the hypothesis, we calculate the precision  $P$  associated with the test dataset. We vary the number of  $k$  dimensions and apply the LSI method, then we trace the curve  $P(k)$ . This is expected to be equal to  $T(k)$ . The only difference is the data source:  $T$  is based on the training data set and  $P$  is based on the final data set. The ratio between  $P(k)$  and  $T(k)$  is expected to be a constant.

### 4. Data

We used the Reuters 21578 corpus [12] with original texts from Reuters 1987 manually classified by specialists in 135 categories. In “modApté” split, some documents are classified in zero or multiple subjects. In this paper, only the documents within a single category were considered.

If you have more than one surname, please make sure that the Volume Editor knows how you are to be listed in the author index.

## 5. Proposed Algorithm Optimization

The naive algorithm implementation is  $n$  times slower than the traditional LSI. During each iteration  $k = 2, 3, \dots, n$ , there is a SVD calculation and cosine similarity, which makes the process extremely slow. Thus, the repeated calculations can be converted into an incremental process.

### 5.1 Single SVD Calculation

The SVD factorization relies on the iterative Lanczos method implemented in SVDPACKC library. Since this is well optimized, we did not go through its implementation. At each iteration  $k = 2, 3, 4, \dots, n$ , there is a SVD factorization, repeating the matrix multiplication  $AA^T$  in order to find out the singular values. The process was changed to make a single call to SVDPACKC library. It gets all the singular values at once, and store the results in memory. For each iteration, the factorization is done by using the highest  $k$  singular values from the memory rather than calling SVDPACKC again. Even though storing the results in memory was a simple change, the time saving was significant.

### 5.2 Term-Document Matrix Rotation

The primary purpose of SVD is to find out the approximation  $A_0 = U_0 \Sigma_0 V_0^T$ . Then, we are able to calculate the document similarity using cosines:

$$\text{similar}(i, j) = \cos(d'_i, d'_j) \text{ , where } d'_i \text{ e } d'_j \text{ are the columns of } A_0$$

Let  $B = \Sigma_0 V_0^T$ . By definition, we have:

$$\begin{aligned} A_0 &= U_0 \Sigma_0 V_0^T \\ &= U_0 B \end{aligned}$$

Since  $U_0$  is orthonormal and  $B$  is a rotation of  $A_0$ , then the spaces are isometric. Therefore, the distance between points, scalar products and cosines are kept the same even after the transformation. Then, we have:

$$\text{similar}(i, j) = \cos(b_i, b_j) \text{ , where } b_i \text{ e } b_j \text{ are columns of } B$$

The optimization is to consider the  $B = \Sigma_0 V_0^T$  space instead of  $A_0 = U_0 \Sigma_0 V_0^T$ . Besides preventing a matrix multiplication, the  $\Sigma_0 V_0^T$  calculation is faster because  $\Sigma_0$  is a diagonal matrix.

### 5.3 Incremental Similarity Calculation

We adopted the vector space defined by  $B^k$  truncated at  $k$  dimensions. During the training phase, we vary the  $k$  value between 1 and  $n$  in order to trace the function  $T(k)$  and determine its peak value. For each iteration of  $k$ , we calculate the similarity of among all  $m$  documents.

$$\text{similar}(i, j, k) = \cos(b_i^k, b_j^k) = \frac{b_i^k \cdot b_j^k}{\|b_i^k\| \cdot \|b_j^k\|}$$

Considering the document representation  $b_i^k = (b_{i1}, b_{i2}, \dots, b_{ik})$ , we then define the product scalar and square module functions

$$\text{Prod}(i, j, k) = b_i^k \cdot b_j^k = \sum_t^k b_{it} b_{jt}$$

$$\text{Mods}(i, k) = \|b_i^k\|^2 = \sum_t^k (b_{it})^2$$

Then, we have:

$$\text{similar}(i, j, k) = \frac{\text{Prod}(i, j, k)}{\sqrt{\text{Mods}(i, k) \cdot \text{Mods}(j, k)}}$$

Each function *Prod* and *Mods* requires  $n$  operations (considering only multiplication, division and square root). The similarity function requires a total of  $3n + 3$  operations.

The idea is to calculate the similarity in an incremental fashion as the number of dimension increases. We change both the *Prod* e *Mods* to recursive functions, so that it is possible to get rid of the sum of products.

$$\text{Prod}(i, j, k) = \text{Prod}(i, j, k-1) + b_{ik} \cdot b_{jk}$$

$$\text{Mods}(i, k) = \text{Mods}(i, k-1) + b_{ik} \cdot b_{ik}$$

At each iteration, it takes only 5 operations (3 multiplications, 1 division and 1 square root). This is a significant reduction in the number of operations compared to the previous  $3n + 3$ .

## 6. Results

The experiment compares the impact of the proposed optimization and the naive algorithm implementation. Varying the number of dimension  $k$  from 10 to 400, we have the results seen in the Table 1.

**Table 1.** Time Comparison (measured in seconds)

Method	k=10	k=50	k=100	k=200	k=400
Naive	2919	14063	27071	54695	117169
Optimized	17	58	116	265	544

The result demonstrate the high amount of time spent in the naive method, which makes it impractical with no optimization.

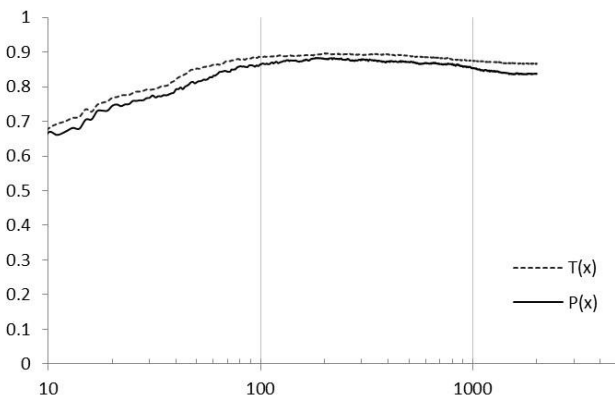
## 6.1 Precision

When we use the LSI method, the  $k$  parameter is usually adjusted to a value between 100 and 300. Based on the actual data, the results would lead to a precision of 0.87 to 0.88 (Table 2).

**Table 2.** LSI Precision (k dimensions)

Dimension	Precision
k=50	0.81
k=100	0.87
K=200	0.88
k=300	0.88
k=400	0.87
k=500	0.87
k=1000	0.85
k=2000	0.84

We used the training dataset to trace the curve  $T(k)$  by varying the number of dimensions  $k$ . As we see the peak of the curve (Fig. 3) takes place at the  $k = 201$ . We observe that the peak is between 200 and 400. Thus, the experiment correctly determines the  $k$  value within the expected range. We also observed that the ratio between  $P(k)$  and  $T(k)$  was lower than 5%, which makes the hypothesis of  $T(x) \approx P(x)$  true.

**Fig. 3.** Precision and the number of dimensions (logarithm scale).

## 7. Conclusion

In this paper, we presented the Latent Semantic Indexing (LSI) as an automatic text classification through word similarity and hidden meanings. Even when there is no term co-occurrence between documents, it is possible to classify them into the same category using the associated “latent semantic”.

The appropriate choice of the  $k$  parameter directly affects the classifier performance. Choosing values lower or higher than the ideal affects the precision. We propose to use the training dataset to support the decision of the ideal  $k$ , transforming the LSI into a supervised learning algorithm. In the experiment, we compare the curves  $T(k)$  and  $P(k)$  corresponding to the classifier precision for the training and test dataset respectively. We also observe that the difference between them was lower than 5%, so it is possible to determine the  $k$  parameter with only the training dataset. We also propose an algorithm optimization to allow the code to run within an acceptable time. This includes the idea of single SVD calculation, Term-Document matrix rotation, and incremental similarity calculation. As a future work, we propose the investigation of iterative Singular Value Decomposition calculation in line with the  $k$  identification. In this case, the partial SVD calculation would consume less time compared to the full SVD factorization.

## References

1. Berry, M., Do, T., Krishna, V., Varadhan, S.: Svdpackc (version 1.0) user's guide (1993)
2. Berry, M., Dumais, S.T.: Using linear algebra for intelligent information retrieval. *SIAM Review* 37, 573–595 (1995)
3. Bradford, R.B.: An empirical study of required dimensionality for large-scale latent semantic indexing applications. In: *International Conference on Information and Knowledge Management*. pp. 153–162 (2008)
4. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 391–407 (1990)
5. Dumais, S.T.: Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers* 23(2), 229–236 (1991)
6. Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S., Harshman, R.: Using latent semantic analysis to improve access to textual information. In: *Computer Human Interaction*. pp. 281–285 (1988)
7. Dumais, S.T.: Lsi meets trec: A status report. In: D. Harman (Ed.), *The First Text REtrieval Conference (TREC1)*, National Institute of Standards and Technology Special Publication. pp. 137–152 (1993)
8. Dumais, S.T.: Latent semantic indexing (lsi) and trec-2. In: *The Second Text REtrieval Conference (TREC-2)*. pp. 105–115 (1994)

9. Dumais, S.T.: Latent semantic indexing (lsi): Trec-3 report. In: Overview of the
10. Third Text REtrieval Conference. pp. 219–230 (1995)
11. Landauer, T.K., Dumais, S.T.: A Solution to Plato’s Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review* 104, 211–240 (1997)
12. Landauer, T.K., Foltz, P.W., Laham, D.: *Introduction to Latent Semantic Analysis* (1998)
13. Lewis, D.D.: Reuters-21578 text categorization test collection (1997), <http://www.daviddlewis.com/resources/testcollections/reuters21578>
14. Lewis, D.D.: Naive (bayes) at forty: The independence assumption in information retrieval. In: *Proceedings of ECML-98, 10th European Conference on Machine Learning*. pp. 4–15. Springer Verlag (1998)
15. Papadimitriou, C.H., Tamaki, H., Raghavan, P., Vempala, S.: Latent semantic indexing: a probabilistic analysis. In: *Symposium on Principles of Database Systems*. pp. 159–168 (1998)
16. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing.
17. *Commun. ACM* 18, 613–620 (November 1975), <http://doi.acm.org/10.1145/361219.361220>



# Design and Implementation of a Genetic Algorithm with Integer Number Coding for the Evolution of FPGAs in Space Applications

JUAN PABLO CAPOSSIO<sup>1</sup>, JUAN JORGE QUIROGA<sup>2</sup>, FRANCISCO PAZ<sup>3</sup>

<sup>1</sup> Departamento de Electrotecnia, Facultad de Ingeniería, Universidad Nacional del Comahue, Neuquén, Argentina  
juanpc\_23@hotmail.com

<sup>2</sup> Departamento de Electrotecnia, Facultad de Ingeniería, Universidad Nacional del Comahue, Neuquén, Argentina  
quirogajuanjorge@yahoo.com.ar

<sup>3</sup> Departamento de Electrotecnia, Facultad de Ingeniería, Universidad Nacional del Comahue, Neuquén, Argentina  
panchopazbustillo@gmail.com

***Abstract.** In [1] a form of representation of logic circuits by chains of integer numbers is presented. That type of representation is easy to simulate and to export to FPGA hardware in such a way that, by adding a genetic algorithm (GA), it can be used in an evolutionary process. Because regular GAs utilize binary number coding, one was designed, with all its operators and processes, that uses integer number coding. This evolvable hardware (EH) process was tested with more than 200 hours of runs to determine the effectiveness of the integer coded GA. Results show that, given the proper conditions, the GA is effective in finding solutions that fulfill the required needs of the target system and that this particular EH platform is suitable for applications where fault tolerance capability is required, such as space systems.*

***Keywords.** Genetic Algorithms, Evolvable Hardware, FPGA, Fault Tolerance.*

## 1. Introduction

Although real number representation, or, more precisely, integer or natural number representation, is often used in GAs because it is ideal to encode a wide spectrum of optimization problems, it has drawbacks when compared against binary number representation. According to Holland's theory of GAs, the main disadvantage is that it reduces the number of schemata which disfavors diversity and probability of forming good building blocks, which are the part of the chromosome that produces high aptitude. Lastly, integer number coding has a much shorter longitude than its binary counterpart.

## 2. Outline of the Algorithm

The basis of the GA design was taken from the chromosome model presented in [1]. Because, as mentioned before, regular GAs implement binary coding, the integer coded chromosome demanded a redesign of each GA operator and process to make them compatible with each other. In order to so, firstly, appropriate crossover and mutation operators were designed and, secondly, stages like evaluation, selection and replacement were implemented to work with vectors of integer numbers.

The GA was designed with the following parameters (which should be considered as standard):

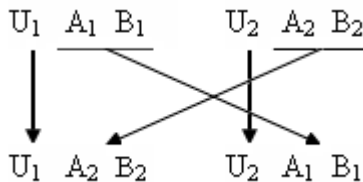
- One point crossover
- Roulette wheel selection
- 100% replacement
- Elite equal to one

The evaluation of each individual (i.e., possible solution to the optimization problem) is the degree of similarity, expressed as a percentage, between the represented circuit's output (obtained by simulation) and the target output signal. An aptitude of a 100% indicates the complete similarity between the simulated and the target output signals and an aptitude of 0% indicates the complete dissimilitude between the previously mentioned signals.

## 3. Genetic Operators

### 3.1 Crossover

As mentioned before, one point crossover was implemented. Figure 1 shows where the crossover point separates, on the one hand, matrices A and B (which encode routing) and, on the other, matrix U (which encodes the type of logic gate being used in the circuit). This way, the first descendant inherits the first progenitor's U matrix



*Fig. 1. One point crossover operator for integer number coding*

and matrices A and B from the second one; meanwhile the second descendant inherits the second progenitor's U matrix and matrices A and B from the first one.

### 3.2 Mutation

The mutation operator, as Figure 2 shows, works, firstly, choosing randomly a gene to mutate and, secondly, modifying it, also randomly, according to the possible values of the alphabet in use without the possibility of repeating the previous allele.

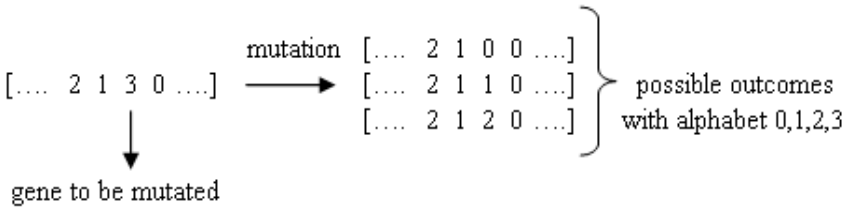


Fig. 2. Mutation operator for integer number coding

### 4. Search Space

The search space or problem domain is proportional to the size of the circuit that is to be evolved. For example, for a circuit that has three inputs ( $M=3$ ) and one layer of logic gates ( $N=1$ ), it results, for matrix U, in a search space of 729 ( $9^M$ ) and, for matrices A and B, a search space larger than  $262 \times 10^3$  ( $((M+1)^{M \cdot (2 \cdot N + 1)})$ ). This shows that, even though the amount of permutations of matrix U is modest, the number of different ways to interconnect those gates with the inputs and outputs is great.

### 5. Evolutionary Process

Once the GA was designed and implemented in a programming language, it was put to the test with different objectives of varied difficulty, thus developing an EH process. The first objective given to the GA was to implement the circuit expressed by Equation 1. As shown, it is a regular logic system with three functions: one AND, one

$$\begin{cases} Y_1 = X_1 \oplus X_3 \\ Y_2 = X_3 + X_2 \\ Y_3 = X_1 * X_3 \end{cases} \quad (1)$$

OR and one XOR. However, if U matrix's search space is analyzed, i.e., the search space of the matrix that encodes the intermediate layer of logic gates, it turns out that only a handful of gate combinations will yield the expected result. This is explained because there are no alternatives to implement the desired logic function. Later it will be clear that the AG's effectiveness to find a good result will be affected by this fact.

The second objective of the EH process was to implement the logic function given by Equation 2. It is a simpler logic function when compared to Equation 1 because it has two NOT gates and one AND gate. There are several ways a NOT logic function can be implemented with logic gates, for example with NAND and NOR gates (also known as universal logic gates). Due to this, there are also several alternatives to reach the desired goal. For this objective, 11% of the possible gate combinations could, if interconnected in a proper manner, provide with the right result.

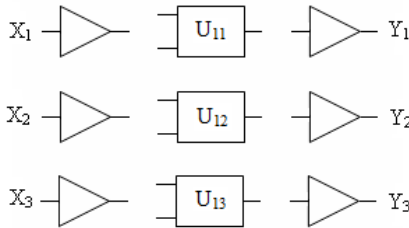
$$\begin{cases} Y_1 = \overline{X_2} \\ Y_2 = \overline{X_1} \\ Y_3 = X_1 * X_3 \end{cases} \quad (2)$$

The last objective, and the simplest of the three, consists only of two outputs instead of three -like the other two- and two logic functions: an OR and a NOT (see Equation 3). In this case it is clear that the number of gate combinations that can yield the correct result is the biggest of all, namely 44%.

$$\begin{cases} Y_1 = X_1 + X_3 \\ Y_2 = \overline{X_2} \end{cases} \quad (3)$$

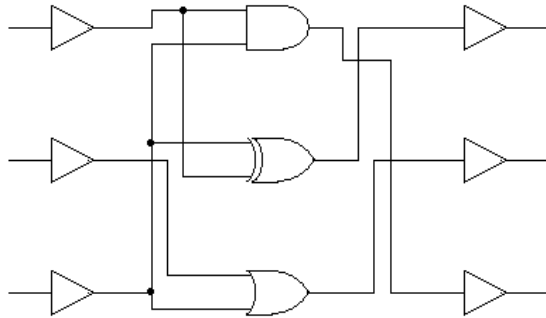
## 6. Analysis of the Successful Circuits

In Figure 3 the generic logic circuit to be evolved can be seen. It consists in three input and three output buffers (M=3), plus one intermediate layer (N=1) of three logic gates, which can be any of the nine available types.



*Fig. 3. Generic logic circuit to be evolved by the GA*

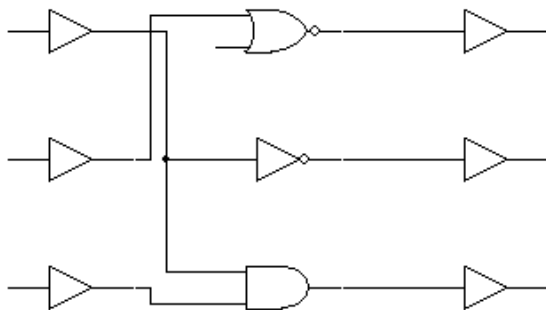
Figure 4 shows one of the circuits that meet the first objective and the chromosome that encodes it (U, A and B vectors one after the other). As mentioned above, there aren't many alternatives to implement the objective logic function, a fact that puts the GA in a disadvantageous position. Actually, only 6 out of 729 possible permutations of U matrix can, if interconnected properly, achieve objective 1. This could have been different if, for example, a second layer of intermediate logic gates would have been added to the circuit (N=2). Thus, the AND and OR gates could have been implemented with NAND and NOR gates. Also, if another layer of gates is added (N=3), the XOR gate could have been implemented with universal logic gates, but not without a high computational cost.



[ 3 7 4 1 2 3 3 2 1 1 2 1 2 1 2 3 0 1 0 3 0 1 0 1 0 ]

**Fig. 4.** Logic circuit that implements Equation 1's logic function

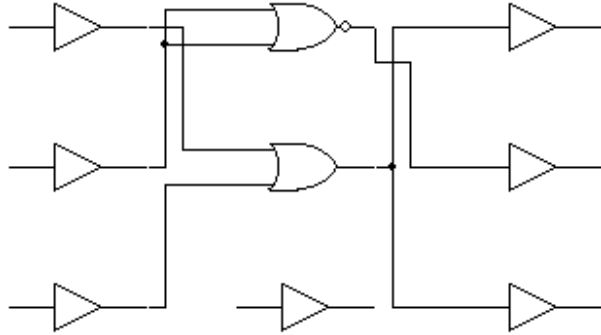
The second successful case we will analyze is portrayed in Figure 5, where the logic circuit and the chromosome that encodes it are shown. An interesting particularity of the circuit is that it uses an alternative implementation of the NOT gate (by using NOR gate) and, also, a NOT gate in itself.



[ 6 2 3 2 1 2 2 1 3 1 2 1 2 1 2 1 0 1 0 3 0 0 0 1 0 1 0 ]

**Fig. 5.** Logic circuit that implements Equation 2's logic function

Lastly, in Figure 6 one of the circuits that implements Equation 3's logic function is presented, along with the chromosome that encodes it. In the circuit, yet another different form to implement a NOT gate with a NOR gate can be seen.



[ 6 4 1 2 2 1 1 0 2 1 2 1 2 0 2 2 0 3 0 0 0 1 0 1 0 0 0 ]

*Fig. 6. Logic circuit that implements Equation 3's logic function*

## 7. Results, Effectiveness of the Algorithm

Table 1 presents the results of the different runs carried out for each one of the objectives previously discussed. GA's parameters were, for all runs and objectives, as follows:

- Population size 100
- Generations (iterations) 100
- Mutation 3%<sup>1</sup>

Parameter selection was made by performing initial test runs and by considering the size of the search space. Mutation rate is a controversial issue because some authors consider it should be low (less than 1%) and others consider a low mutation rate to be counterproductive. Thus, the chosen parameter value is a compromise selection.

As anticipated, the GA's effectiveness in finding the correct circuit for objective 1 is low. This is explained because the set of good solutions is very small when compared to the entire search space. Table 1 shows that in several runs with unsuccessful results, the average aptitude of the population is equal, or almost, to the best individual's aptitude. Thus, all individuals in the population are equal, a fact that results in no new genetic information generated by

<sup>1</sup> Additionally, 10 runs were carried out with a mutation rate of 5% only of objective 3

the crossover operator. In other words, the AG has been attracted to a local maximum and is trapped in it, leaving remote chances of it reaching a global maximum.

Results obtained for objective 2 are slightly better, mainly due to the fact that there are several ways to implement the NOT logic function. However, arguments similar to the first case can be stated to explain the second case's GA performance.

**Table 1.** Summary table of the GA's run results<sup>2</sup>

Objective	Run	Pop. Size	Generations	Mutation	Success?	Best Aptitude	Jumps	Avg. Aptitude
1	1	100	100	3	yes	99.2	4	99.06
	2				no	88.33	2	74.34
	3				no	74.83	2	74.58
	4				yes	99.2	4	96.34
	5				no	77.5	0	77.18
	6				no	91	3	85.25
	7				no	77.5	0	77.18
	8				no	91	3	85.25
	9				no	77.5	0	77.18
	10				no	77.5	0	77.18
					20.00%	85.356	1.8	82.354
2	1	100	100	3	yes	99.33	5	86.47
	2				no	93.9	5	93.2
	3				yes	99.33	4	93.37
	4				no	77.79	3	76.59
	5				no	88.5	3	87.84
	6				no	82.93	1	81.3
	7				no	82.93	2	74.23
	8				no	77.5	0	77.75
	9				no	88.5	2	88.17
	10				yes	99.33	6	98.16
					30.00%	89.004	3.1	85.708
3 (mutation 3%)	1	100	100	3	yes	99.46	2	97.61
	2				no	94.03	0	93.7
	3				no	94.03	1	94.03
	4				yes	99.46	3	98.47
	5				no	94.03	2	92.82
	6				no	94.03	0	93.39
	7				no	94.03	2	92.82
	8				yes	99.46	4	98.47
	9				yes	99.46	3	98.9
	10				yes	99.46	2	92.51
					50.00%	96.745	1.9	95.272

<sup>2</sup> In all successful runs the best individual's aptitude isn't 100% due to slight differences between the simulated output and the objective signal. Yet, all implement the right logic.

	1				no	94.03	1	91.32
	2				yes	99.46	1	99.46
	3				yes	99.46	3	97.4
	4				yes	99.46	1	92.18
3	5	100	100	5	no	94.03	2	92.11
(mutation	6				yes	99.46	3	92.34
5%)	7				yes	99.46	4	89.69
	8				no	94.03	2	88.25
	9				yes	99.46	2	91.98
	10				no	94.03	1	87.93
					60.00%	97.288	2	92.266

For objective 3 results are sensibly better. Success rates of 50 and 60% were obtained for mutation rates of 3 and 5% respectively (at this point it is important to remark that an increase in mutation rate didn't have any effect for the previous two objectives). Besides, an increase in the average aptitude compared to the other two objectives was obtained.

## 8. Efficiency of the Algorithm

A way to measure the efficiency of the GA is to compare two things: the size of the search space and the amount of circuits that are tested in each run. Taken the search space of matrices A and B ( $>262 \times 10^3$ ) and the amount of circuits tested in each run ( $100^2$ , although is clear that many are repeated) it turns out that, explicitly, only less than 4% of the search space needed to be explored in order to reach a solution that meets with the requirements of the proposed objective.

## 9. Problems and Perspectives

One of the most important problems to consider is the time it takes for the GA to complete an entire run, which is, in average, five hours in a home computer and bearing in mind the circuit to implement is rather small. One way around this problem is to reduce the amount of circuits being tested. It would be doable because the crossover operator not always produces new genetic material. Thus, a sort of marking has to be developed to indicate when an individual's aptitude is already known.

Although significant, the time problem is less important when the size of the hardware necessary to run the GA is taken into consideration, especially in space applications, where room is a great constraint for design. A solution would be to relocate the genetic processor outside the payload, thus controlling the reconfigurable hardware on board via telecommunications (if available), with either an intrinsic or extrinsic evolutionary process.

Once the GA is implemented, has its parameters adjusted and has a routine capable of transforming a vector of integer numbers coding a logic circuit



into VHDL digital circuit description language (i.e., VHDL export), the next step would be to implement an intrinsic evolutionary process, also named “*hardware in the loop*”. Furthermore, larger circuits can be tested; more complex logic functions implemented and fault tolerant capabilities can be tested.

## 10. Conclusions

Firstly, for the GA to perform with high levels of effectiveness, the FPGA circuit must have a certain amount of redundancies. If the relationship between the complexity of the logic function to implement and the amount of redundancies isn’t adequate, the effectiveness of the algorithm will be low (the more complex the function to implement, bigger the amount of redundancies that will be needed). On the other hand, if bigger circuits are used, the time needed for the GA to finish an entire run will be greater (scalability problem, see [8]). There has to be a proper compromise between these two requirements.

Secondly, due to the FPGA’s versatility and the GA’s effectiveness, this EH platform can be used as a multifunctional redundant system (see [1] and [2]) to improve reliability in systems where fault tolerance is essential to survival, like satellites.

Lastly, with a large number of runs performed, a good idea of the best algorithm parameters was obtained.

## References

1. Paz F., Quiroga J.J., Capossio J.P.: Diseño de Una Plataforma de Simulación para la Implementación de Algoritmos Genéticos en Módulos Redundantes Multifuncionales para Aplicaciones Espaciales. VI Congreso Argentino de Tecnología Espacial (2011).
2. Capossio J.P., Quiroga J.J.: Evolvable Hardware for Improving System Reliability in a Nanosatellite. 7<sup>th</sup> International Conference on Electric and Electronics Engineering Research, Mexico (2010).
3. Capossio J.P., Quiroga J.J., Paz F.: Análisis de Tolerancia a Fallos Mediante Hardware Evolucionable y Módulos Redundantes Multifuncionales para Aplicaciones Espaciales. VI Congreso Argentino de Tecnología Espacial (2011).
4. Coello Coello C.A., Christiansen A.D., Hernández Aguirre A.: Towards Automated Evolutionary Design of Combinational Circuits (2001).
5. Coello Coello C.A., Hernández Luna E., Hernández Aguirre A.: A Comparative Study of Encodings to Design Combinational Logic Circuits Using Particle Swarm Optimization (2004).

6. Coello Coello C.A., Hernández Aguirre A.: Design of combinationl logic circuits through an evolutionary multiobjective optimization aproach (2001).
7. Stomeo E., Kalganova T., Lambert C.: A Novel Genetic Algorithm for Evolvable Hardware. 2006 IEEE Congress on Evolutionary Computation.
8. Vassilev V. K., Miller J. F.: Scalability Problems of Digital Circuit Evolution - Evolvability and Efficient Designs. Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware (2000).

# Simulation of Wireless Sensor Networks Using Cellular Automata

PABLO GODOY<sup>1,2</sup>, LUCAS IACONO<sup>1,2,3</sup>, RICARDO CAYSSIALS<sup>4</sup>, CARLOS GARCÍA GARINO<sup>1,2,3,5</sup>

<sup>1</sup>ITIC, Instituto universitario para las Tecnologías de la Información y las Comunicaciones, Universidad Nacional de Cuyo

<sup>2</sup>LAPIC, Laboratorio de Investigación y Desarrollo para la Producción Integrada por Computadora, Carrera de Redes y Telecomunicaciones, ITU

<sup>3</sup>Facultad de Ingeniería, Universidad de Mendoza

<sup>4</sup>Facultad de Ingeniería Eléctrica y de Computadoras, Universidad Nacional del Sur

<sup>5</sup>Facultad de ingeniería, Universidad Nacional de Cuyo

**Abstract.** *Wireless Sensor Networks (WSN) are a useful tool for remote sensing of agricultural and environmental variables. WSNs are composed by nodes that communicate among them through radio frequency signals.*

*The cellular automata based models are suitable for modeling WSNs, due to every node acts as an autonomous entity, has a attribute set, and interacts with other nodes according to rules established in the communication protocol.*

*In this paper we propose a model for WSN based on cellular automata. This model proposes to model nodes, events and data messages through objects interacting among them, using the cellular automata theory. As an example, we perform an implementation of the model, using the modeling and simulation framework Repast Symphony.*

## 1. Introduction

### 1.1 Wireless Sensor Networks

A WSN is a system composed of devices called nodes. These nodes are generally composed by four main components:

- A microcontroller, which includes a microprocessor and memory modules.
- A wireless communication module or radio.
- Sensors to measure different variables.
- Power supply, usually composed by commercial AA batteries.

WSNs have a large number of applications such as environmental monitoring, measurement of agricultural variables, industrial process automation, building automation, military applications, etc. It is expected that the number of applications and the number of WSNs grow significantly in the next few years [1][2][3][4].

A WSN may be composed by different types of nodes, every type of node with different functions. A WSN may have nodes with special functions depending on the application, but in general, all WSNs have the following types of nodes:

- Base station: Its main functions are to control and manage the WSN, collect information about the nodes (nodes in operation, faulty nodes, etc.), and initialize the WSN.
- Sink node: It is responsible for collecting data generated by other nodes. The base station generally performs this function, although it may be a different node.
- Router nodes or simply nodes: Their functions are to collect data and transmit them to the sink node. Furthermore, they have the function of forwarding data generated by other nodes, so that these data can reach their final destination. In general, all nodes of most WSNs fulfill this function (except the base station).

The WSN have very strict design requirements. These requirements include:

- Large number of nodes to achieve adequate coverage. This requirement imposes the need of low-cost nodes. Therefore, nodes have limited processing power and memory.
- Adequate useful life, according to the application, with no or minimal maintenance.
- Suitable robustness to withstand environmental conditions in which nodes must operate.
- Low energy consumption, due to nodes are powered by batteries.
- High reliability for preserving the data integrity under different operation environmental conditions.
- Suitable autonomy according to the application. A WSN must be able to adapt itself to changes in topology due to relocation of nodes, nodes that break down, etc. Additionally, the WSN must be able to resolve these faults.

The design requirements listed above and the growing number of applications have led to researchers and companies to develop hardware and software platforms specific for WSN.

## 1.2 Communication Protocols

Nodes that compose a WSN communicate via radio frequency signals. Communication is the function which usually produces the highest energy consumption. As an example, for the Tmote Sky nodes [5] the consumption is:

- 2 mA when the radio is off
- 23 mA when the radio is on and on data reception mode

- 21 mA when the radio is on and on data transmission mode

In the above example can be observed that the radio consumes more than 10 times the energy consumed by the other components of the node. Since the energy consumption depends on the transmission distance, the transmission range of each node is limited to 100 m. Only nodes close to the base station can communicate directly with the base station, other nodes send their messages to others, and these nodes forward these messages to other nodes, until the messages reach the base station.

Both nodes and the base station can generate data messages. These messages may contain information about measured variables, detected events, or commands sent by the base station to nodes. Messages may be generated at time points predefined by the application (eg temperature measurements every 10 minutes), or when a event occurs in the area of interest (eg, fire, natural phenomena, presence of animals, etc.). The goal of communication protocols is that the messages generated by nodes or base station reach their destination.

### 1.3 Cellular Automata

A cellular automaton is a system that evolves in discrete time steps. They are formed by a grid of objects or cells that interact and evolve according to a rule set. For a given object, these rules determine the object state in the next time period, according to the state of adjacent objects and its own current state. Cellular automata are used to model the behavior of systems consisting of a large number of components which follow a rule set. They are used to model various types of physical systems, social systems, etc.

There are several software tools for modeling and simulating agent-based systems and cellular automata. Among these tools, two very popular are NetLogo [**¡Error! No se encuentra el origen de la referencia.**] and Repast [**¡Error! No se encuentra el origen de la referencia.**].

Repast is a software platform based on Java. Most of commercial nodes for WSNs use C language. But the transcription from Repast code to C language seems to be simple. Thinking on future works, which may include software tools to perform this transcript, we have selected Repast to implement the simulation described in section 3.

### 1.4 Paper Structure

The rest of the paper is organized as follows. Section 2 describes in detail the proposed model for modeling the behavior of WSNs through cellular automata. Section 3 describes, as an example, an implementation of the proposed model through simulation. Finally, Section 4 presents possible improvements and concludes this work.

## 2. WSNs Modeling through Cellular Automata

This paper proposes a methodology for modeling WSNs using cellular automata. Other papers that propose models based on cellular automata for WSNs are [8][9][10].

The cited papers present models for modeling only the nodes of a WSN using cellular automata. We propose to model the nodes composing a WSN, messages that are transmitted through the WSN, and events that may occur in the area that is being monitored. These elements are represented as different types of objects that are part of a system that behaves like a cellular automaton. The proposed models for every elements are explained below.

### 2.1 Nodes

Every node is modeled as an object or element of a static cellular automaton arranged in a grid shape (future works could use mobile nodes). Every node has properties and rules that the node must comply. Due to there are different types of nodes, we propose different types of objects for modeling each type of node.

In general, each node can be in one of several states:

- **Radio on:** A node whose radio is on can communicate with its neighboring nodes, be aware of the status of them, and receive, send or forward data messages. Its next state depends on its current state, the state of the nodes in its neighborhood, and whether it has messages to send or not. Additionally, the model may consider the case of nodes that modify the transmission power level, in order to save energy and be able to deal with changes on the network operating conditions.
- **Radio off:** A node whose radio is off can not communicate with its neighbors, so it can not know the status of them, nor receive or forward messages. As a result, the next state of a node whose radio is off can not depend on the state of its neighboring nodes, since it can not know the status of them, can only depend on its own current state. The microcontroller can be on or off, this determines the ability of the node to detect events or not.
- **Low-power modes:** a node may be in different states of low power consumption, in which its functions are restricted, or the operating frequency of its processor is reduced, in order to reduce energy consumption.

The nodes have two fundamental properties, energy level and a timer. Due to the measurements are result of analog-to-digital conversions, the measured power level is an integer number between 0 and a maximum. The model proposed in this paper decreases the value of the energy level  $N$  units for every  $M$  time units, where  $M$  and  $N$  are function of the node state. If the energy level of a node reaches zero, the node is considered out of service.

When the node radio is off, the node can not know the state of its neighbors, so the only way to “wake it up” is through an internal timer, which switches on the radio after a certain amount of time units, or an event that occurs in the node itself.

As a result, the transition from radio off towards radio on depends exclusively on the node itself, regardless the state of neighboring nodes. Whereas, the transition from radio on to radio off depends on the node itself and the state of neighboring nodes, according to the rules implemented by the communication protocol.

Sink nodes follow similar rules, except that they are always on. Energy of sink nodes never runs out, due to they are usually powered by a not finite power source.

## **2.2 Messages**

Data messages are also modeled using objects. The model assumes that every node covers a certain area of the field that is being monitored. When an event occurs at a certain area, the node that monitors this area will act according to the rules established by the communication protocol. This node may generate a message indicating that an event has occurred. Messages can move passing from one node to another, as long as both nodes, ie the sender and receiver of the message, are on.

There may be different types of messages, for example:

- Data messages: contain data of measurements or information about detected events. If a data message arrives at a node that is not the sink node, this node forwards the message to another node, until the message reaches the sink node.
- State information messages: These messages are generated by the nodes that are on, to notify this condition to their neighbors. These messages are not retransmitted by nodes that receive them.
- Commands generated by the base station: These messages are generated by the base station and may be destined to any node on the network, a group of nodes or all nodes.

It can be seen that every message type has its own characteristics and behavior, so that each one must be modeled using different types of objects.

## **2.3 Events**

Events may be synchronous, for example, when a node takes measurements at regular time intervals, or asynchronous, for example, when nodes detect sporadic events such as fire, presence of animals, etc. These events can be generated at any point in the network, and may motivate to other network elements to perform certain actions, for example, generate messages. Therefore, the events are also modeled through objects, which are generated in specific places and time points, or with a certain occurrence probability.

These events interact with the other elements of the system. This interaction leads to the other elements of the system to perform the actions specified in the communication protocol.

### 3. Implementation of the Model Using the Repast Simulator

#### 3.1 Implementation of Model Components

We implemented a network formed by 122 nodes, however, the number of nodes can easily be varied, from a few to hundreds of thousands. The data sink node was located in the center of the network. The nodes are arranged in a grid. We assume that each simulation step lasts a second. Each node has an initial energy value of 1000. For each time step the energy value is decremented one unit if the node radio is on. This is equivalent to saying that a node permanently turned on would have a lifetime of 1000 seconds.

#### 3.2 Nodes

We define a class to implement routers nodes and the sink node. We create objects from this class to simulate the nodes. This class has the following properties:

- State: May be “on” when the radio is on, “off” when the radio is off, or “out of service” when the node is without power.
- Energy: is decremented one unit for every time step when the node is “on”. It is left unchanged when the node is “off”.
- Time: Variable that indicates the time remaining before a node that is “off” changes to the “on” state. It is increased one unit, from zero, every time step of the simulation when the node is “off”. When this variable reaches a certain value, the node passes to “on” state. For the simulations implemented in this work, this variable has a value of 120 (2 minutes), but this value can be modified according to the needs of each experiment.

The position of all nodes is static. This position is fixed at the beginning of the simulation.

The protocol that controls the nodes behavior establishes the following rules:

- For every node it is defined a neighborhood, formed by those nodes which are at a distance of one unit, excluding the node itself.
- If a node is in “on” state, will continue in “on” state if none or only one of its neighbors are “on”. Otherwise the node will switch “off”.
- If the node has a message to be transmitted, then it can not be switched “off”, regardless the state of its neighbors, until the message has been transmitted to another node.
- If a node is “off”, it will be switched “on” only when its timer reaches a predefined value.



A node knows the number of neighbors in “on” state thanks to state information messages that receives.

### 3.3 Messages

Messages are generated periodically at random points of the grid. Messages move seeking to reach the sink node. The simulation assumes two types of messages: data messages and state information messages.

The protocol requires at every message to try two position changes for every time step. The first position change attempt is in horizontal direction toward the sink node (trying to come near to the sink node). The second position change attempt is in vertical direction, also towards the sink node. Both change position attempts are independent, may take place only one, both or neither. Every movement only takes place if both the source node and the destination node are in “on” state.

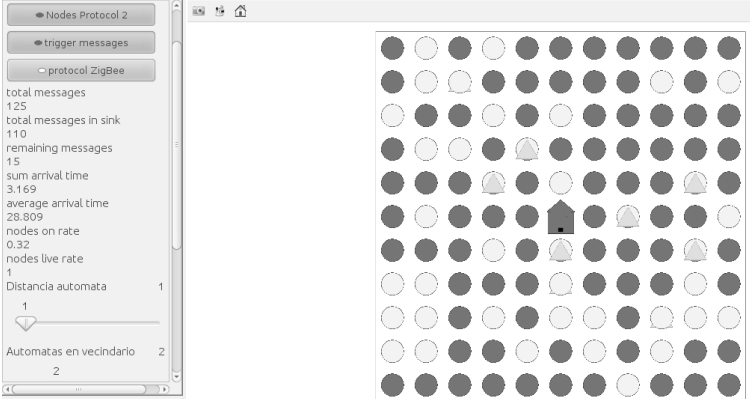
### 3.4 Experiments

**Comparison Metrics.** The experiments are designed to evaluate the performance of the proposed communication protocol. We first define two comparison metrics:

- Relationship between the WSN lifetime using the protocol under test, compared to the WSN lifetime when all nodes are permanently “on”. This metric allows to compare easily different communication protocols in terms of their effectiveness to maximize the WSN lifetime.
- Average latency: average time that messages need to reach the destination node.

The first metric is of great importance because the useful life is a key parameter in WSN. The idea is that this metric allows to compare results obtained by different users. It requires the definition of the network lifetime. One drawback is that in the literature there are different definitions for lifetime. As a result, to compare communication protocols through this metric requires that experiments use the same lifetime definition.

**Results.** Figure 1 is a graphical representation of the simulated WSN generated by Repast. The nodes are represented by circles, whose color indicates their status: white when the node is “on”, black if the node is “off”, and gray if the node is “out of service”. Messages are represented by triangles.

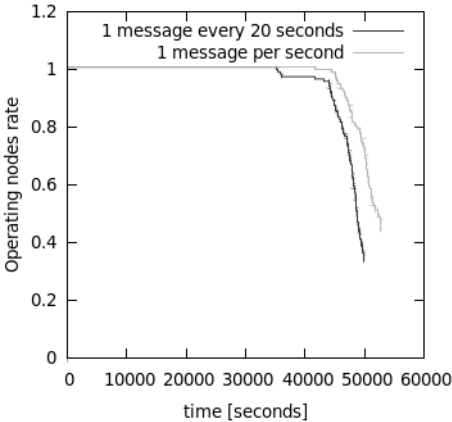


**Fig. 1.** Graphical representation of the simulated system

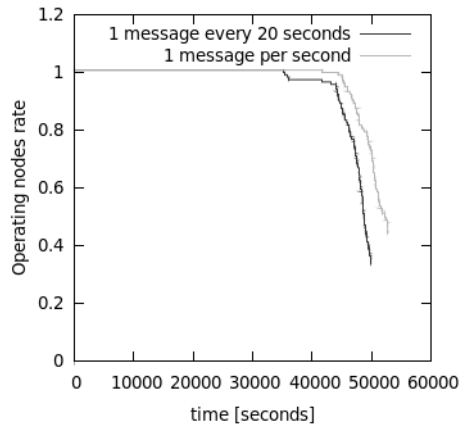
Figures 2 and 3 represent the ratio of operating nodes (nodes with energy) to total nodes (operating nodes rate) and the average arrival time to the sink node of the messages, for two different message generation rates, a message every 20 seconds, and a message per second.

Figure 2 shows that the average lifetime is about 50,000 seconds, ie 50 times higher than the lifetime of a network whose nodes are permanently “on” (see section 3.1).

Figure 3 shows that the average arrival time for each message is 100 seconds.



**Fig. 2.** Operating nodes rate



*Fig. 3. Average arrival time*

Finally, from both figures we can conclude that the change on the message generation rate has no significant influence neither on the average arrival time nor on the network lifetime for the WSN simulated.

#### 4. Conclusions and Future Work

This paper presents a model of WSN implemented by cellular automata. In this model, the components of the WSN (nodes, messages and events) are modeled using different types of objects. An implementation of the model was performed with the Repast software tool to show the feasibility of the proposal.

In addition to the proposed model, an additional contribution of this paper is the metric proposed to compare communication protocols for WSN, with respect to their efficiency in terms of maximizing the nodes lifetime.

The modeling of WSN using cellular automata and agents, and the corresponding simulations, can be useful to design software and communication protocols for WSN.

As future work we propose the implementation of these models on real nodes, and to compare the proposed communication protocol against current communication protocols for WSN. There are several WSN testbeds that could be used to test the model [11].

**Acknowledgements.** The authors acknowledge the financial support received from the Instituto de Desarrollo Industrial Tecnológico y de Servicios (IDITS) and the Agencia Nacional de Promoción Científica y Tecnológica (ANPCyT) through the project PID-PAE 146 Grid Embebida y de Sensores. Pablo Godoy and Lucas Iacono thank doctoral fellowships awarded by the

ANPCyT through the program PRH funded by the ANPCyT and UNCuyo, and project PID PAE-146 respectively.

## References

1. J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.
2. Th. Arampatzis, J. Lygeros, and S. Manesis. A survey of applications of wireless sensors and wireless sensor networks. In *Intelligent Control, 2005. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation*, pages 719 –724, June 2005.
3. Q. Ho and T. Le-Ngoc. A wireless sensor network testbed. In *Communication Networks and Services Research Conference (CNSR), 2010 Eighth Annual*, pages 304 –309, May 2010.
4. H. Hellbruck, M. Pagel, A. Kroller, D. Bimschas, D. Pfisterer, and S. Fischer. Using and operating wireless sensor network testbeds with wisebed. In *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, pages 171 –178, June 2011.
5. Moteiv Corporation. *Tmote Sky, Ultra low power IEEE 802.15.4 compliant wireless sensor module*, 2006.
6. Center for Connected Learning (CCL) and Computer-Based Modeling. Netlogo website, <http://ccl.northwestern.edu/netlogo/>.
7. Argonne National Laboratory. Repast simphony website, <http://repast.sourceforge.net/>.
8. I. Maity, G. Bhattacharya, S. Das, and BK Sikdar. A cellular automata based scheme for diagnosis of faulty nodes in wsn. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 1212–1217. IEEE, 2011.
9. S. Misra, V. Tiwari, and M. Obaidat. Lacas: learning automata-based congestion avoidance scheme for healthcare wireless sensor networks. *Selected Areas in Communications, IEEE Journal on*, 27(4):466–479, 2009.
10. X. Xu, X. Zhang, and L. Wang. Simulating energy efficient wireless sensor networks using cellular automata. In *Simulation Conference (WSC), Proceedings of the 2011 Winter*, pages 3202–3211. IEEE, 2011.
11. L.P. Steyn and G.P. Hancke. A survey of wireless sensor network testbeds. In *AFRICON, 2011*, pages 1 –6, Sept. 2011.

# Assessment of Writing Text in Mobile Devices

RODRIGO CAPA-ARNAO<sup>1</sup>, CRISTINA MANRESA-YEE<sup>1</sup>,  
RAMON MAS-SANSÓ<sup>1</sup>, MARTIN LARREA<sup>2</sup>

<sup>1</sup> Math and Computer Science Department.  
Universitat de les Illes Balears. SPAIN

<sup>2</sup> Laboratorio de Investigación y Desarrollo en Visualización y Computación  
Gráfica (VyGLab), Departamento de Ciencias e Ingeniería de la Computación,  
Universidad Nacional del Sur, ARGENTINA  
correo@derodrigo.com, {cristina.manresa, ramon.mas}@uib.es,  
mll@cs.uns.edu.ar

***Abstract.** The input of text on mobile devices has evolved from physical keyboards to virtually displayed touch keyboards. The aim of the paper is to assess and compare two mobile phone text input methods (mini-QWERTY and on-screen keyboards) to analyze which one is more efficient and effective. We review previous works related to text entry metrics and studies on mini-QWERTY physical and virtual keyboards.*

***Keywords:** HCI, usability, text input, mobile phones, mini-QWERTY*

## 1. Introduction

Mobile phones have evolved to become multiple-featured communicating devices. The increase of the processing power and storage capacity, the introduction of new and more complex operating systems and the use of larger screen sizes allow current mobile devices to perform tasks traditionally devoted to computers.

One of the main restrictions of mobile devices is the input of text, a task that is very easily and comfortably done with personal computers using traditional keyboards. Mobile devices manufacturers have addressed this problem from two different perspectives: to use a mini-QWERTY physical keyboard (i.e. BlackBerry and some high-end Symbian based Nokia phones) or to use a tactile keyboard on the touch screen (i.e. IOS and Android devices).

At first, and if we not take into account the 12-key keypad, mobile devices used either a virtual keyboard or character recognition systems. Virtual keyboards were operated using a stylus because of the resistive nature of the touch screens. Character recognition was offered as an alternative input text method, but high error rates combined with a long learning-training process highly reduced its usability and performance [1, 2, 3, 4].

Current devices introduce new and more sophisticated text input techniques combined with text prediction and correction. For example, Paek et al. [5] and Riadi [6] compared input techniques considering advances in text prediction and on-screen keyboards of several formats.

Although mini-QWERTY physical keyboards are proven to be faster and less prone to errors they are not always the users' choice [4]. In this paper we review the text-input metrics and we compare the use of touch screen to physical mini-QWERTY keyboards in mobile phones.

## 2. Previous Work

### 2.1 Performance measure

The Words per minutes (WPM) value is a measure of writing speed. As a rule-of-thumb a word is considered to be five characters long (in average) [5, 7]. It is important to highlight that WPM does not count keystrokes but only output characters. This means that if the user writes "Alamo" or "alamo" both count as one word, even if the first character in the first word requires two keystrokes (SHIFT+'a') hence requiring more time to write than just a "a". This difference becomes relevant when writing in a language that requires special characters, like Spanish with characters such as 'ñ' or 'á'.

The rate of keystrokes over characters (KSPC) [8] is a measure of the user's errors when writing. If the user does not insert any errors when writing, then the KSPC is equal to one. This means that there was one keystroke for each character. If the text entry has a mistake then there was more than one keystroke involved, then increasing the KSPC. When using traditional writing methods, the KSPC can never be lower than one. But, with predicted writing systems, a word can be introduced with less keystrokes than characters in it. This results in a KSPC lower than one.

The WPM metric does not reflect the user's errors and therefore it cannot be used as a real measure of performance. A way of improving this metric is to weight in the number of errors. For example, by subtracting the number of errors from the WPM.

Another method used to measure performance is the minimum string distance (MSD) [9]. This metric computes the minimum distance between two strings defined in terms of editing primitives. The primitives are *insertion*, *deletion*, and *substitution*. Given two character strings, the idea is to find the smallest set of primitives that applied to one string produces the other. The number of primitives applied is the MSD. This measurement can be used as a performance metric by using the MSD as an indicator of the number of corrections required to fix a misspelled word. Then, the user's error rate, based on the MSD, can be calculated as the rate between the MSD and the length of the sentence (Equation 1).

$$error\ rate = \frac{MSD(A,B)}{\max(|A|,|B|)} \quad Eq. 1$$

For example, considering the following sentences:

*This phrase is correct*  
*This phrase nis correct*

In order to transform the second sentence into the first one, we need to eliminate one character, the 'n'. The MSD for this example is 1. If we rate the MSD with the length of the second sentences, 23, the result is the error rate. In this case is 0.043. This metric is interesting because it considers both the omission as well as the inclusion of characters.

## **2.2 Performance analysis based on a theoretical model**

The analysis of input methods on mobile devices has been done from different point of views. One of them is Fitts' law (1954), which involves the time of movement, the distances travelled and the size of the object to select. There are many studies on writing performance based on Fitts' law but none of them cover all the available devices [10]. In order to include some of these results, we will describe only three of them.

The first one is from Silfverberg et. al. [11], in this work the authors compared the performance of multitap input (one character is obtained after several keystrokes) versus a predicted T9 dictionary on a Nokia 5110 mobile device. They concluded that a user using the T9 dictionary and with an optimal training increase his writing speed between 52% and 109% over the multitap method.

Another study [12] compared user's performance on a QWERTY keyboard versus an OPTI one, both projected onto a touch screen and operated using a stylus. They concluded that the maximum writing speed on the QWERTY keyboard was 43 WPM, and 58 on the OPTI. In both cases, the results are theoretical. The authors then tested several real users using these keyboards and got 40 WPM for the QWERTY and 45 WPM for the OPTI.

Finally, in 2002 MacKenzie and Soukoreff [13] studied a mini-QWERTY keyboard and concluded that the average input speed was between 50 and 60 WPM.

## **3. Methodology**

This section aims at evaluating the current text entry mechanisms in mobile phones. Two systems will be compared by using the Curran et al. [4] experiment in laboratory conditions.

### **3.1 Apparatus**

Two families of mobile devices which count with a tactile screen are globally present: iPhone and Android. iPhones do not count with devices with mini-QWERTY, but there are Android devices which count with mini-QWERTY

keyboards plus on-screen tactile keyboards. Therefore, an HTC Desire was used to test both text entry strategies using the same device (see Fig. 1).

The application MyTextSpeed was used to carry out the tasks. In the superior half screen, the phrase to copy is presented and in the low half screen, there is space to write the sentence. The writing duration is automatically registered, starting when the user introduces the first character and finishing when the user introduces the Enter key. In Curran's experiment this last character is not introduced, but for the system to register automatically the duration is the simplest mode. This software does not allow computing the error rate using MSD, so after finishing each subtask, a photograph of the screen was taken to control errors.



*Fig. 1. Mobile phone with mini-QWERTY*

### 3.2 Procedure

This experiment involved a within-group study, that is, all users had to try both input strategies (mini-QWERTY and tactile). At the beginning of the session, the experiment conductors described and explained the evaluation objectives to the users and described the two input strategies.

There were two tasks with four subtasks each: users had to write four phrases with both input strategies. The phrases were in English, and in order for it not to have an influence on the users, this criterion was taken in to account to select the users. Before writing the phrases, users could read them as many times as desired.

The phrases were:

1. I have never sent a text message before
2. Your flying lesson's cancelled today. Call Andrew from 7:00pm onwards to arrange another lesson.
3. Plane gets in at 10:00pm to Gate 11. Aerlingus flight no. EI 987. Can you meet me? My e-mail address is biggles@hotmail.com!



#### 4. let me no where u r and il pic u up 18r

As Curran et al [4] state, the first phrase is simple and containing no punctuation characters. The second phrase is more complex because it includes punctuation, numbers and lower and upper case letters. Then, the third phrase is more complex and uses more letters, numbers and punctuation characters. Finally the fourth phrase had abbreviations for words and phrases. The learning bias that could influence the repetition of the task with the second input strategy was reduced by selecting as the first input device the one that the user used usually, in this way, the level of complexity increased gradually. Users before starting the tasks could try the input strategy as long as desired and they usually trained between three and five minutes.

Fatigue or boredom did not appear because the test was of short duration (less than five minutes each task).

At the beginning of the test, users answered a questionnaire compiling demographics information (age, gender), their experience level using Android devices and using both input strategies, kind of tasks frequently carried out and style of writing with mobile phones (use of predictive text, correction of errors, number of fingers and fingers used when writing)

### 3.3 Participants

Ten users were recruited from the general campus population to test the game. Participant demographics included ages ranging from 18 to 40, and 4 men and 6 women. The criteria followed to select the users were:

- Users had to have an average level of English
- Users had to be Android users
- Users had to know how to write with the QWERTY layout.

## 4. Results

In Table 1 the results of the speed and error rate are shown. It can be observed that the mini-QWERTY text entry strategy is slightly faster than the on-screen keyboard and with less errors.

Observing the results regarding the users' experience is interesting to highlight those users with experience with mini-QWERTY as they obtained faster results with this system, but the error rate does not vary. Regarding those users without experience, their performance in speed is similar for both systems but inserting more errors when working with the on-screen keyboard. We could observe that users who relied usually on the predicting mechanism for on-screen did not notice errors on words, therefore increasing the error rate (see Table 2 and 3).

**Table 1.** Average and standard deviations of error rate and WPM.

Input strategy	WPM(mean)	MSD (mean)	WPM (SD)	MSD (SD)
Mini-QWERTY	14,22	1,07	10,67	1,42
On-screen keyboard	13,32	1,48	7,8	2,03

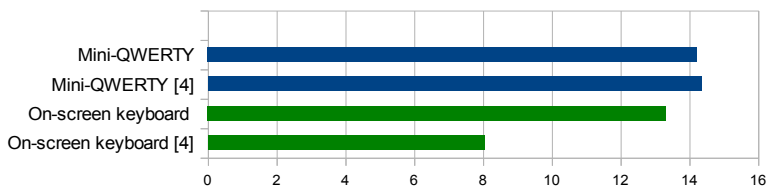
**Table 2.** Average and standard deviations of WPM regarding miniQWERTY experience.

Experience with mini-QWERTY	WPM mini-QWERTY (m)	WPM on-screen keyboard (m)	WPM mini-QWERTY(SD)	WPM on-screen keyboard (SD)
Yes	16.05	14.32	3,55	2,67
No	11.48	11.85	9,41	14,08

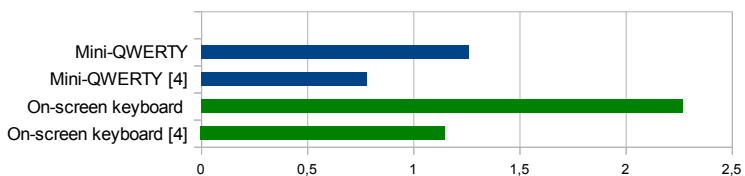
**Table 3.** Average and standard deviations of error rate regarding miniQWERTY experience.

Experience with mini-QWERTY	MSD mini-QWERTY (m)	MSD on-screen Keyboard (m)	MSD mini-QWERTY(SD)	MSD on-screen Keyboard (SD)
Yes	0,95	0,95	1,95	1,05
No	1,26	2,27	0,93	2,96

If we compare the results of our experiment with the ones obtained by Curran et al [4], the results for the mini-QWERTY strategy are very similar regarding the speed but results for WPM on-screen keyboard differ being much faster in our experiment than in Curran's (see Fig. 2). This can be due to the difference in the device used, as they used a Motorola PDA. On the other hand, error rate is higher in our experiment for both systems; being greater the error rate for on-screen keyboards (see Fig. 3).



**Fig. 2.** WPM comparison



*Fig. 3. Error rate comparison*

## 5. Discussion and conclusions

The most efficient input device for text-entry is the mini-QWERTY, as users have achieved better writing speed and fewer errors. However, modern devices tend to include only on-screen keyboards due to aesthetics, weight and size reduction and building costs.

Moreover, as users acquire experience with the on-screen keyboard their typing speed gets closer to the mini-QWERTY, nevertheless inserting more mistakes particularly due to the predictive text.

Nowadays new text input strategies appear to work faster with on-screen keyboards such as sliding your finger through the keyboard instead than pressing each key. Following the evolution, it seems that in a short period the on-screen keyboard with new text input strategies may allow trained users to write faster than with mini-QWERTY, and hopefully with less errors.

**Acknowledgments.** This work is partially supported by Spanish MAEC-AECID A1/037910/11.

## References

1. Pila Suomalainen, L. K., Pääkkönen, R. 2010. A comparison of the usability of a laptop, communicator, and handheld computer. *Journal of Usability Studies* 5, 111–123.
2. Yatani, K., Truong, K. N. 2007. An evaluation of stylus-based text entry methods on handheld devices in stationary and mobile settings. In *Proceedings of the 9th international conference on human computer interaction with mobile devices and services. MobileHCI '07*. ACM, New York, NY, USA, 487–494
3. Commarford, P. M. 2004. An investigation of text throughput speeds associated with pocket pc input method editors. *Int. J. Hum. Comput. Interaction* 17, 3, 293–308.

4. Curran, K., Woods, D., Riordan, B. O. 2006. Investigating text input methods for mobile phones. *Telematics and Informatics* 23, 1, 1–21.
5. Paek, T., Chang, K., Almog, I., Badger, E., Sengupta, T. 2010. A practical examination of multimodal feedback and guidance signals for mobile touchscreen keyboards. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. MobileHCI '10. ACM, New York, NY, USA, 365–368.
6. Riadi, D. 2005. An overview and usability measurement of virtual keyboard.
7. Arif, A.S. and Stuerzlinger, W. 2011. Analysis of Text Entry Performance Metrics. In Proc. IEEE TIC-STH 2009. IEEE New York (2009), 100–105
8. MacKenzie, I. S., Soukoreff, R. W. 2002b. Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction* 17, 2-3, 147–198.
9. Soukoreff, R. W., MacKenzie, I. S. 2001. Measuring errors in text entry tasks: an application of the levenshtein string distance statistic. In CHI '01 extended abstracts on Human factors in computing systems. CHI EA '01. ACM, New York, NY, USA, 319–320.
10. Soukoreff, R. W., MacKenzie, I. S. 2004. Towards a standard for pointing device evaluation, perspectives on 27 years of fitts' law research in hci. *International Journal of Human-Computer Studies* 61, 6, 751–789. Fitts' law 50 years later: applications and contributions from human computer interaction.
11. Silfverberg, M., MacKenzie, I. S., Korhonen, P. 2000. Predicting text entry speed on mobile phones. In Proceedings of the SIGCHI conference on Human factors in computing systems. CHI '00. ACM, New York, NY, USA, 9–16.
12. MacKenzie, I. S. Zhang, S. X. 1999. The design and evaluation of a high-performance soft keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems: the CHI is the limit*. CHI '99. ACM, New York, NY, USA, 25–31.
13. MacKenzie, I. S., Soukoreff, R. W. 2002a. A Model of Two-Thumb Text Entry. In Proc. Graphics Interface. 117–124.

# Expressibility of the Logic $SO^F$ on Classes of Structures of Bounded FO Types

Alejandro L. Grosso<sup>1</sup> and José M. Turull Torres<sup>2</sup>

<sup>1</sup> Dpto. de Informática, Universidad Nacional de San Luis, Argentina,  
agrosso@unsl.edu.ar

<sup>2</sup> ICTIC, Universidad de la Cuenca del Plata, Corrientes, Argentina  
and

Dpto. de Informática, Universidad Nacional de San Luis, Argentina  
J.M.Turull@massey.ac.nz

**Abstract.** We introduce a new property for classes of structures (or relational database instances), that we call bounded FO types. Then we prove that on such classes the expressive power of  $SO^F$  collapses to first order logic FO. As a consequence of this we prove that  $SO^F$  is strictly included in SO.

**Keywords:** Finite Model Theory, Descriptive Complexity, Relational Machines

## 1. Introduction

Significant research has been done in the last few decades in regard to the relationship between finite model theory and computational complexity theory. There is a close relationship between computational complexity, the amount of resources we need to solve a problem over some Turing machine, and descriptive complexity, the logic we need to describe the positive instances of the problem. The most important result about this relationship was the result of Fagin [3]. This result establishes that the properties of finite relational structures (or relational database instances) which are defined by existential second order sentences coincide with the properties that belong to the complexity class NP. This result was extended by Stockmeyer [10] establishing a close relationship between second order logic and the polynomial hierarchy.

In [7] we introduced the logic  $SO^F$  and proved that the logic  $SO^\omega$  defined by A. Dawar [2] and further studied by F. Ferrarotti and the second author [6], which is a fragment of the infinitary logic  $\mathcal{L}_{\infty,\omega}^\omega$ , is strictly included in  $SO^F$ . In  $SO^\omega$  the second order quantifiers range over  $k'$ -ary relations closed by the equivalence relation  $\equiv^{k'}$  for  $k' \leq k$ , whereas in  $SO^F$  the second order quantifiers range over  $k$ -ary relations closed by the equivalence relation  $\equiv^{FO}$  (see section 2 for definitions of  $\equiv^k$  and  $\equiv^{FO}$ ). We showed that in  $SO^F$  the rigidity query can be expressed. This property means that the structures only has one automorphism which is the identity function. Rigidity belongs to co-NP. In [7] we also

characterize the existential fragment  $\Sigma_1^{1,F}$  of  $SO^F$  with a modified version of the relational machine defined in [1].

In [8] we added oracles to our version of relational machine, and we introduced the polynomial time hierarchy PHF. We then showed the correspondence between the fragments  $\Sigma_i^{1,F}$  of  $SO^F$  and the levels of the polynomial time hierarchy PHF. That result is analogous to the L. Stockmeyer's characterization of polynomial time hierarchy on Turing machine [10] and also to the F. Ferrarotti and the second author's characterization of the relational polynomial time hierarchy on relational machines [6].

Using the semantics of  $SO^F$  and fixing a  $\sigma$ -structure  $\mathcal{A}$  it is possible to assign an FO formula equivalent to each  $SO^F$  formula. In this reduction lies the idea to define, for a fixed  $\sigma$ -structure  $\mathcal{A}$ , each possible  $k$ -ary relation closed under FO types for  $k$ -tuples using an FO formula. This match between sentences of  $SO^F$  and sentences of FO can be established for classes of structures which have a finite number of isolating formulae for FO types. This property holds for the class of structures with unary vocabulary.

As a consequence, the expressive power of  $SO^F$  collapses to FO over classes with a finite number of isolating formulae including the case of structures with unary vocabulary. It is well known that parity is expressible in SO but is not expressible in FO, then parity over sets is not expressible in  $SO^F$ . Therefore,  $SO^F$  is strictly included in SO.

## 2. Preliminaries

We only consider finite relational structure.

A vocabulary  $\sigma$  is a set of relational symbols  $\{P_1, \dots, P_s\}$  with associated arities,  $r_1, \dots, r_s \geq 1$ . A  $\sigma$ -structure (also called model or relational database instance)  $\mathcal{A} = \langle A, P_1^{\mathcal{A}}, \dots, P_s^{\mathcal{A}} \rangle$  consists of a non empty set  $A$  called domain of  $\mathcal{A}$  and a relation  $P_i^{\mathcal{A}} \subseteq A^{r_i}$  for each relation symbol  $P_i$  in  $\sigma$  for  $1 \leq i \leq s$ . The domain of  $\mathcal{A}$  is denoted with  $A$  or  $dom(\mathcal{A})$ .

An  $m$ -ary query  $q$ , for  $m \geq 1$  is a function which maps structures of a fixed vocabulary  $\sigma$  to  $m$ -ary relations on the domain of the structures, and which preserves isomorphisms, i. e., when  $f$  is an isomorphism from  $\mathcal{A}$  to  $\mathcal{B}$  then  $\bar{t} \in q(\mathcal{A})$  iff  $f(\bar{t}) \in q(\mathcal{B})$ . A 0-ary query, also called *Boolean query*, is a function from a class of  $\sigma$ -structures to  $\{0,1\}$  and can be identified with a class of  $\sigma$ -structures. By a class of structures in  $B_\sigma$  we means a class closed under isomorphism.

For the definitions of syntax and semantics of FO see [9] among others.

The truth value of a formula  $\varphi$ ,  $\mathcal{A} \models \varphi(x_1, \dots, x_n)[v]$ , with free variables  $x_1, \dots, x_n$ , only depends on the values assigned by the valuation  $v$  to the free variables. Therefore, we denote with  $\mathcal{A} \models \varphi[a_1, \dots, a_n]$  the truth value of the  $\varphi$  in the structure  $\mathcal{A}$  for a valuation that assigns to the free variable  $x_i$  the value  $a_i$  for  $1 \leq i \leq n$ . Then a formula  $\varphi(x_1, \dots, x_n)$  with  $n$  free variables defines an  $n$ -ary relation on  $\mathcal{A}$ ,  $\varphi^{\mathcal{A}} = \{(a_1, \dots, a_n) \in A^n \mid \mathcal{A} \models \varphi(x_1, \dots, x_n)[a_1, \dots, a_n]\}$ .

$FO^k$  is the fragment of FO where we use up to  $k$  different variables.

In second order logic we add a set of second order variables which range over relations instead of elements of the structure domain.

Another way to extend the logic FO is by allowing conjunction and disjunction over sets of formulae with arbitrary cardinality, then we have the infinitary logic  $\mathcal{L}_{\infty,\omega}$ . The logic  $\mathcal{L}_{\infty,\omega}^k$  is the fragment of  $\mathcal{L}_{\infty,\omega}$  where we use up to  $k$  different variables. The logic  $\mathcal{L}_{\infty,\omega}^\omega$  is the union of the fragments  $\mathcal{L}_{\infty,\omega}^k$  for  $k \geq 1$ .

### 2.1. Element Types

Let  $\mathcal{A}$  be a structure and  $\bar{a}$  be an  $l$ -tuple of elements of  $A$  for  $l \geq 1$ , we define the FO type of  $\bar{a}$  in  $\mathcal{A}$ , denoted by  $type_{\mathcal{A}}^{FO}(\bar{a})$ , as the set of FO formulae,  $\varphi$ , with free variables among  $x_1, \dots, x_l$  such that  $\mathcal{A} \models \varphi[a_1, \dots, a_l]$ . A set  $\tau$  of FO formulae is an FO type iff  $\tau$  is the FO type for some tuple in some structure. If  $\tau$  is an FO type, we say that the tuple  $\bar{a}$  realize  $\tau$  in  $\mathcal{A}$  iff  $\tau = type_{\mathcal{A}}^{FO}(\bar{a})$ .

Let  $\mathcal{A}$  and  $\mathcal{B}$  be  $\sigma$ -structures and  $\bar{a}, \bar{b}$  be two tuples of the same length in the structures  $\mathcal{A}$  and  $\mathcal{B}$  respectively.  $(\mathcal{A}, \bar{a}) \equiv^{FO} (\mathcal{B}, \bar{b})$  iff  $type_{\mathcal{A}}^{FO}(\bar{a}) = type_{\mathcal{B}}^{FO}(\bar{b})$ . That is, two tuples of possibly different structures have the same FO type when they satisfy the same (maximally consistent) set of FO formulae. Let  $k \geq 1$ , and  $\tau$  be an FO type, a formula  $\varphi(\bar{x}) \in \tau$  is called an *isolating formula* of FO type  $\tau$  for  $k$ -tuples when for all pairs of structures  $\mathcal{A}, \mathcal{B}$  in  $B_\sigma$  and tuples  $\bar{a} \in A^k, \bar{b} \in B^k$  it holds that  $\mathcal{A} \models \varphi(\bar{a})$  and  $\mathcal{B} \models \varphi(\bar{b})$  iff  $\tau = type_{\mathcal{A}}^{FO}(\bar{a}) = type_{\mathcal{B}}^{FO}(\bar{b})$ .

In a similar way we can define  $type_{\mathcal{A}}^k(\bar{a})$  and  $\equiv^k$  for the logic  $FO^k$ .

**Fact 1** Let  $l \geq 1$ ,  $\mathcal{A}$  a finite  $\sigma$ -structure and  $\bar{a}$  and  $\bar{b}$  be two  $l$ -tuples on  $\mathcal{A}$ .  $(\mathcal{A}, \bar{a}) \equiv^{FO} (\mathcal{A}, \bar{b})$ , if and only if, there is an automorphism  $f$  such that  $f(a_i) = b_i$  for  $1 \leq i \leq l$ .

## 3. Semantic Restrictions of SO

A. Dawar in [2] introduced a restriction on second order logic ( $SO^\omega$ ) by restricting the class of relations that the quantified second order variables can be assigned to by valuations. In  $SO^\omega$ , the second order variables can only contain relations which are closed under the equivalence relation  $\equiv^k$  for some  $k \geq 1$ . This implies that we cannot assign arbitrary relations to the variables. The relations have to be unions of  $FO^k$  types, i.e., they have to be unions of equivalence classes of  $\equiv^k$ .

In [7] we introduced the logic  $SO^F$  as a restriction of the second order logic where the second order quantifiers range over relations closed under the equivalence relation  $\equiv^{FO}$ , i.e., the quantifiers range over relations which are unions of FO types. These relations are redundant in the sense of [5].

Let  $k \geq 1$ . For a  $k$ -ary relation variable  $R$ , we define the second order quantifier  $\exists^F R$  with the following semantics:  $\mathcal{A} \models \exists^F R \varphi[v]$  if and only if there exists a relation  $S \subseteq A^k$  such that  $S$  is closed under the equivalence relation  $\equiv^{FO}$  in  $\mathcal{A}$  for  $k$ -tuples, and  $\mathcal{A} \models \varphi[v \frac{S}{R}]$ . As usual  $\forall^F R \varphi$  is an abbreviation of  $\neg \exists^F R \neg \varphi$ .

We add the following formation rules to the FO formation rules to obtain the formulae of  $SO^F$ : 1) If  $R$  is a  $k$ -ary second order variable, for  $k \geq 1$ , and

$x_1, \dots, x_k$  are first order variables, then  $R(x_1, \dots, x_k)$  is a formula (atomic) of  $SO^F$ . 2) If  $\varphi$  is an  $SO^F$  formula, and  $R$  is a  $k$ -ary second order variable, then  $\exists^F R\varphi$  and  $\forall^F R\varphi$  are formulae of  $SO^F$ . The fragment  $\Sigma_i^{1,F}$  of  $SO^F$  consists of the formulae of  $SO^F$  which have a prefix of  $i$  alternated blocks of second order quantifiers followed by an FO formula. The prefix must begin with an existential quantifier block. Then, we can define:  $SO^F = \bigcup_i \Sigma_i^{1,F}$ .

#### 4. Collapse of $SO^F$ to FO

Before we show the reduction to FO we will see that the FO types for  $r$ -tuples realized in a particular structure can be expressed by means of FO formulae with  $r$  free variables.

When we use second order quantification we extend the structure with relations. In  $SO^F$  we extend the structure with redundant relations in the sense that has been studied in the work of F. Ferrarotti, A. Paoletti and the second author in [5]. In the extended structure  $\langle \mathcal{A}, R \rangle$  where  $R$  is an  $r$ -ary relation closed under FO types for  $r$ -tuples for  $r \geq 1$ , the equivalence relation  $\equiv^{FO}$  for  $r$ -tuples is the same for the original structure and the extended structure. That is, for all  $\bar{a}, \bar{b} \in A^r$ ,  $(\mathcal{A}, \bar{a}) \equiv^{FO} (\mathcal{A}, \bar{b})$  iff  $(\langle \mathcal{A}, R \rangle, \bar{a}) \equiv^{FO} (\langle \mathcal{A}, R \rangle, \bar{b})$ . This is not true in SO where the quantified relations can break the FO types. The following lemma is from [5].

**Lemma 1.** *Let  $\mathcal{A}$  be a  $\sigma$ -structure. Let  $R$  be a  $r$ -ary relation closed by FO types for  $r$ -tuples in  $\mathcal{A}$ . Let  $\bar{a} \in R$  and  $\bar{b} \in A^r$ . There is formula  $\varphi_{\bar{a}}(x_1, \dots, x_r)$  of  $FO(\sigma)$  such that  $\mathcal{A} \models \varphi_{\bar{a}}(x_1, \dots, x_r)[\bar{b}]$  iff  $tp_{\mathcal{A}}^{FO}(\bar{a}) = tp_{\mathcal{A}}^{FO}(\bar{b})$ .*

The proof uses the diagram,  $\Delta_{\mathcal{A}}$ , of the structure  $\mathcal{A}$ . Let  $|dom(\mathcal{A})| = n$  and  $v: \{x_1, \dots, x_n\} \rightarrow dom(\mathcal{A})$  be an injective valuation such that  $v(x_{i_1}) = a_1, \dots, v(x_{i_r}) = a_r$  for  $1 \leq i_1, \dots, i_r \leq n$ , then  $\varphi_{\bar{a}}(y_1, \dots, y_r) \equiv \exists x_1 \dots \exists x_n (\delta_{\mathcal{A}} \wedge (\bigwedge_{1 \leq i < j \leq n} x_i \neq x_j) \wedge \forall x_{n+1} (\bigvee_{1 \leq i \leq n} x_{n+1} = x_i) \wedge (x_{i_1} = y_1 \wedge \dots \wedge x_{i_r} = y_r))$  where

$$\delta_{\mathcal{A}} = \bigwedge_{R \in \sigma} \{R(x_{i_1}, \dots, x_{i_r}) : \mathcal{A} \models R(x_{i_1}, \dots, x_{i_r}) y_1 \leq i_1, \dots, i_r \leq n\} \wedge \bigwedge_{R \in \sigma} \{\neg R(x_{i_1}, \dots, x_{i_r}) : \mathcal{A} \not\models R(x_{i_1}, \dots, x_{i_r}) y_1 \leq i_1, \dots, i_r \leq n\}$$

that is,  $\delta_{\mathcal{A}}$  is the conjunction of the atomic formulae and negated atomic formulae that hold in  $\mathcal{A}$ .

The formula above, without the conjunction  $(x_{i_1} = y_1 \wedge \dots \wedge x_{i_r} = y_r)$ , is the diagram of  $\mathcal{A}$ . It is known that for all structure  $\mathcal{B}$ ,  $\mathcal{B} \models \Delta_{\mathcal{A}}$  iff  $\mathcal{B}$  is isomorphic to  $\mathcal{A}$ .

The Lemma below is well known and holds for many logic used in Finite Model Theory. We include a proof for  $\Sigma_k^{1,F}$  because we will make use of the construction later.



**Lemma 2.** Let  $\mathcal{A}$  be  $\sigma$ -structure and  $\psi$  be a sentence in  $\Sigma_k^{1,F}$  for  $k \geq 1$ . There exists a sentence  $\hat{\psi}_{\mathcal{A}}$  in  $\text{FO}(\sigma)$  such that  $\mathcal{A} \models \psi$  iff  $\mathcal{A} \models \hat{\psi}_{\mathcal{A}}$ .

*Proof.* Let  $\psi \equiv \exists^F X_{11} \dots \exists^F X_{1s_1} \forall^F X_{21} \dots \forall^F X_{2s_2} \dots Q X_{k1} \dots Q X_{ks_k} \varphi(X_{11}, \dots, X_{1s_1}, X_{21}, \dots, X_{2s_2}, \dots, X_{k1}, \dots, X_{ks_k})$  where  $Q = \exists^F$  when  $k$  is odd or  $Q = \forall^F$  when  $k$  is even.

Let  $R_1^{\mathcal{A}}, \dots, R_{n_r}^{\mathcal{A}}$  be all the  $r$ -ary relations closed by FO types for  $r$ -tuples on  $\mathcal{A}$ , then we can define each relation with an FO formula using the formulae defined in the Lemma 1.

$(\mathcal{A}, R_j^{\mathcal{A}}) \models R_j(y_1, \dots, y_r)[b_1, \dots, b_r]$  iff  $\mathcal{A} \models \bigvee_{\bar{a} \in R_j^{\mathcal{A}}} \varphi_{\bar{a}}(y_1, \dots, y_r)[b_1, \dots, b_r]$ , for  $1 \leq j \leq n_r$ . We define  $\hat{\phi}_j \equiv \bigvee_{\bar{a} \in R_j^{\mathcal{A}}} \varphi_{\bar{a}}(y_1, \dots, y_r)$ .

Then  $\mathcal{A} \models \exists^F X_{11} \dots \exists^F X_{1s_1} \forall^F X_{21} \dots \forall^F X_{2s_2} \dots Q X_{k1} \dots Q X_{ks_k} \varphi(X_{11}, \dots, X_{1s_1}, X_{21}, \dots, X_{2s_2}, \dots, X_{k1}, \dots, X_{ks_k})$  iff  $\mathcal{A} \models \hat{\psi}_{\mathcal{A}}$ , with

$$\begin{aligned} \hat{\psi}_{\mathcal{A}} \equiv & \bigvee_{1 \leq j_{11} \leq n_{r_{11}}} \dots \bigvee_{1 \leq j_{1s_1} \leq n_{r_{1s_1}}} \bigwedge_{1 \leq j_{21} \leq n_{r_{21}}} \dots \bigwedge_{1 \leq j_{2s_2} \leq n_{r_{2s_2}}} \dots \left( \bigvee / \bigwedge \right)_{1 \leq j_{k1} \leq n_{r_{k1}}} \\ & \dots \left( \bigvee / \bigwedge \right)_{1 \leq j_{ks_k} \leq n_{r_{ks_k}}} \varphi(\hat{\phi}_{j_{11}}/X_{11}, \dots, \hat{\phi}_{j_{1s_1}}/X_{1s_1}, \dots, \hat{\phi}_{j_{21}}/X_{21}, \dots, \\ & \hat{\phi}_{j_{2s_2}}/X_{2s_2}, \dots, \hat{\phi}_{j_{k1}}/X_{k1}, \dots, \hat{\phi}_{j_{ks_k}}/X_{ks_k}), \end{aligned}$$

where  $n_{r_{tu}}$ , for  $1 \leq t \leq k$  and  $1 \leq u \leq s_t$ , is the number of different  $r_{tu}$ -ary relations closed by FO types for  $r_{tu}$ -tuples.

In this way the second order existential quantifiers are replaced by disjunctions over all  $r_{tu}$ -arity relations  $R_{j_{tu}}^{\mathcal{A}}$  closed by FO types for  $r_{tu}$ -tuples. These relations are definible by the formulae  $\hat{\phi}_{j_{tu}}$  belonging to  $\text{FO}(\sigma)$ . Similarly, the second order universal quantifiers are replaced by conjunctions.  $\square$

Looking at the proof of Lemma 2, note that the formula  $\hat{\psi}_{\mathcal{A}}$  depends on the  $\sigma$ -structure  $\mathcal{A}$ . That is, given the set of formulae  $\hat{\Psi} = \{\hat{\psi}_{\mathcal{A}} \in \text{FO}(\sigma) \mid \mathcal{A} \in B_{\sigma}\}$ , if we fix the  $\sigma$ -structure  $\mathcal{A}$ , then the formula  $\psi \in \text{SO}^F(\sigma)$  is equivalent to the formula  $\hat{\psi}_{\mathcal{A}} \in \hat{\Psi}$ .

We will see now a property that we can define over a class  $\mathcal{C}$  in order to reduce a formula  $\psi \in \text{SO}^F(\sigma)$  to an equivalent formula  $\hat{\psi}_{\mathcal{C}} \in \text{FO}(\sigma)$  that holds for every structure in  $\mathcal{C}$ , not only for a single structure  $\mathcal{A} \in \mathcal{C}$  (module isomorphism).

**Definition 1.** Let  $\mathcal{C} \subseteq B_{\sigma}$  be a class of structures and  $k \geq 1$ . A finite set of formulae  $\Phi^k$  is a set of intra-isolating formula for FO types for  $k$ -tuples over  $\mathcal{C}$  when:

- i) For every structure  $\mathcal{A} \in \mathcal{C}$ , every  $\bar{a}, \bar{b} \in A^k$  and  $\varphi^k \in \Phi^k$  it holds: if  $\mathcal{A} \models \varphi^k(x_1, \dots, x_k)[\bar{a}]$  and  $\mathcal{A} \models \varphi^k(x_1, \dots, x_k)[\bar{b}]$  then  $\text{type}_{\mathcal{A}}^{FO}(\bar{a}) = \text{type}_{\mathcal{A}}^{FO}(\bar{b})$
- ii) For every structure  $\mathcal{A} \in \mathcal{C}$ , and every  $\bar{a}, \bar{b} \in \text{dom}(\mathcal{A})^k$ , if  $\text{type}_{\mathcal{A}}^{FO}(\bar{a}) = \text{type}_{\mathcal{A}}^{FO}(\bar{b})$  then there exists  $\varphi_i \in \Phi^k$  such that  $\mathcal{A} \models \varphi_i(x_1, \dots, x_k)[\bar{a}]$  and  $\mathcal{A} \models \varphi_i(x_1, \dots, x_k)[\bar{b}]$ , and for all  $\varphi_j \in \Phi^k$  with  $j \neq i$  it holds  $\mathcal{A} \not\models \varphi_j(x_1, \dots, x_k)[\bar{a}]$  and  $\mathcal{A} \not\models \varphi_j(x_1, \dots, x_k)[\bar{b}]$ .

iii) For every structure  $\mathcal{A} \in \mathcal{C}$ , and every  $\bar{a} \in \text{dom}(\mathcal{A})^k$  there exists  $\varphi^k \in \Phi^k$  such that  $\mathcal{A} \models \varphi^k(x_1, \dots, x_k)[\bar{a}]$ .

A formula  $\varphi^k \in \Phi^k$  can express that two  $k$ -tuples over a structure in  $\mathcal{C}$  have the same FO type even if  $\varphi^k$  is not an isolating formula for FO type for  $k$ -tuples. Note that Def. 1 is based on tuples of the same structure and does not consider the case when the tuples belong to two different structures. Then, it may happen that the same intra-isolating formula is satisfied by two tuples from two distinct structures with different FO types. For example over the class of  $r$ -ary full trees with depth  $h$  (see example 1), for  $r \geq 1$ , the elements in the same level have the same FO type. One intra-isolating formula  $\varphi^1(x_1)$  can just express that the element  $x_1$  has depth  $d$  with  $0 \leq d \leq h$ . Given elements  $a$  and  $b$  with the same depth from full trees  $\mathcal{T}_r$  ( $r$ -ary) and  $\mathcal{T}_{r+1}$  ( $(r+1)$ -ary) respectively, then  $a$  and  $b$  satisfy the intra-isolating formula  $\varphi^1(x_1)$ , but they don't have the same FO type, i.e.,  $\text{type}_{\mathcal{T}_r}^{FO}(a) \neq \text{type}_{\mathcal{T}_{r+1}}^{FO}(b)$ . They have different FO type because, for example, they have a different number of siblings.

Then, if a class  $\mathcal{C}$  has a set  $\Phi^k$ , we can define each  $k$ -ary relation closed by FO types for  $k$ -tuples over all structures in  $\mathcal{C}$ . We recall that a relation closed by FO types over  $\mathcal{A}$  is the union of FO types realized in  $\mathcal{A}$ . And by making the disjunction of different intra-isolating formulae of  $\Phi^k$  we have unions of FO types.

**Lemma 3.** *Let  $\mathcal{C}$  be a class of structures,  $k \geq 1$  and  $\Phi^k$  be a set of intra-isolating formulae for FO types for  $k$ -tuples over  $\mathcal{C}$ . Then every  $k$ -ary relation closed by FO types for  $k$ -tuples  $R^{\mathcal{A}} \subseteq \text{dom}(\mathcal{A})^k$  with  $\mathcal{A} \in \mathcal{C}$  is definable from  $\Phi^k$ .*

*Proof.* By definition of  $\Phi^k$ , for each  $\mathcal{A}$ , and each  $\bar{a} \in \text{dom}(\mathcal{A})^k$  there exist a formula  $\varphi_{\bar{a}}^k \in \Phi^k$  such that  $\mathcal{A} \models \varphi_{\bar{a}}^k(x_1, \dots, x_k)[\bar{a}]$ . By Def. 1,  $\varphi_{\bar{a}}^k(x_1, \dots, x_k)$  is satisfied for  $k$ -tuples on  $\mathcal{A}$  that have the same FO type as  $\bar{a}$ . Then every relation closed by FO types  $R^{\mathcal{A}} \subseteq \text{dom}(\mathcal{A})^k$  is definable with the following formula:  $\varphi(x_1, \dots, x_k) \equiv \bigvee_{\bar{a} \in R^{\mathcal{A}}} \varphi_{\bar{a}}^k(x_1, \dots, x_k)$ .  $\square$

**Lemma 4.** *Let  $\mathcal{C}$  be a class of structures,  $k \geq 1$  and  $\Phi^k = \{\varphi_1, \varphi_2, \dots, \varphi_{n_k}\}$ , with  $n_k \geq 1$ , be a set of intra-isolating formulae for FO types for  $k$ -tuples over  $\mathcal{C}$ . Then, the set  $\{R^{\mathcal{A}} \mid \mathcal{A} \in \mathcal{C}, R^{\mathcal{A}} \subseteq \text{dom}(\mathcal{A})^k \text{ and } R^{\mathcal{A}} \text{ is closed by FO types}\}$  is equal to the set  $\{\varphi^{\mathcal{A}} \mid \varphi(x_1, \dots, x_k) \equiv \bigvee_{i \in D} \varphi_i(x_1, \dots, x_k), D \neq \emptyset, D \subseteq \{1, 2, \dots, n_k\} \text{ and } \mathcal{A} \in \mathcal{C}\} \cup \{\emptyset\}$ .*

*Proof.* ( $\Rightarrow$ ) Let  $R^{\mathcal{A}}$  be a non empty  $k$ -ary relation closed by FO types for some  $\mathcal{A} \in \mathcal{C}$ . Then, by the Lemma 3,  $R^{\mathcal{A}} = \varphi^{\mathcal{A}}$  such that  $\varphi(x_1, \dots, x_k) \equiv \bigvee_{\bar{a} \in R^{\mathcal{A}}} \varphi_{\bar{a}}(x_1, \dots, x_k)$  and  $\varphi_{\bar{a}}(x_1, \dots, x_k) \equiv \varphi_i(x_1, \dots, x_k)$  for some  $i \in \{1, \dots, n_k\}$ .

( $\Leftarrow$ ) Every formula  $\varphi(x_1, \dots, x_k) \equiv \bigvee_{i \in D} \varphi_i(x_1, \dots, x_k)$ , for some  $D \subseteq \{1, 2, \dots, n_k\}$  and  $D \neq \emptyset$ , define a relation closed by FO types since the disjunction implies the union of the FO types which are isolated by the formulae  $\varphi_i(x_1, \dots, x_k)$ . If  $R^{\mathcal{A}} = \emptyset$ , then  $\varphi(x_1, \dots, x_k) \equiv x_1 \neq x_1 \wedge \dots \wedge x_k \neq x_k$ .  $\square$

**Definition 2.** Let  $k \geq 1$ , the class of structures  $\mathcal{C}$  is a class with bounded FO types for  $k$ -tuples if  $\mathcal{C}$  has a set  $\Phi^k$  of intra-isolating formulae for  $k$ -tuples.  $\mathcal{C}$  is a class with bounded FO types when for each  $k \geq 1$  it holds that  $\mathcal{C}$  is a class with bounded FO types for  $k$ -tuples.

**Theorem 1.** Let  $\mathcal{C}$  be a class with bounded FO types, then  $SO^F$  is equivalent to FO on  $\mathcal{C}$ .

*Proof.* Let  $\mathcal{A}$  be  $\sigma$ -structure in  $\mathcal{C}$ , and  $\psi \equiv \exists^F X_{11} \dots \exists^F X_{1s_1} \forall^F X_{21} \dots \forall^F X_{2s_2} \dots Q X_{k1} \dots Q X_{ks_k} \varphi(X_{11}, \dots, X_{1s_1}, X_{21}, \dots, X_{2s_2}, \dots, X_{k1}, \dots, X_{ks_k})$  where  $Q = \exists^F$  when  $k$  is odd and  $Q = \forall^F$  when  $k$  is even. Let  $\Phi^{r_{tu}} = \{\varphi_1, \dots, \varphi_{n_{r_{tu}}}\}$  be a set of intra-isolating formulae for FO types for  $r_{tu}$ -tuples for  $\mathcal{C}$  with  $n_{r_{tu}}$  the cardinality of such set.

We define the set  $\hat{I}_{r_{tu}} = \{\hat{\gamma}_{r_{ij}} | \hat{\gamma}_{r_{tu}} = \bigvee_{s \in D} \varphi_s(x_1, \dots, x_{r_{tu}}) \text{ for } D \subseteq \{1, \dots, n_{r_{tu}}\}, D \neq \emptyset \text{ and } \varphi_s \in \Phi^{r_{tu}}\} \cup \{x_1 \neq x_1 \wedge \dots \wedge x_{r_{tu}} \neq x_{r_{tu}}\}$ . Note that  $\hat{I}_{r_{tu}}$  is the set of FO( $\sigma$ ) formulae that define every possible  $r_{tu}$ -ary relation closed by FO types for  $r_{tu}$ -tuples on  $\mathcal{C}$ .

Then

$$\mathcal{A} \models \exists^F X_{11} \dots \exists^F X_{1s_1} \forall^F X_{21} \dots \forall^F X_{2s_2} \dots Q X_{k1} \dots Q X_{ks_k} \varphi(X_{11}, \dots, X_{1s_1}, X_{21}, \dots, X_{2s_2}, \dots, X_{k1}, \dots, X_{ks_k}) \text{ iff } \mathcal{A} \models \hat{\psi}_{\mathcal{C}}.$$

where

$$\hat{\psi}_{\mathcal{C}} \equiv \bigvee_{\hat{\gamma}_{r_{11}} \in \hat{I}_{r_{11}}} \dots \bigvee_{\hat{\gamma}_{r_{1s_1}} \in \hat{I}_{r_{1s_1}}} \bigwedge_{\hat{\gamma}_{r_{21}} \in \hat{I}_{r_{21}}} \dots \bigwedge_{\hat{\gamma}_{r_{2s_2}} \in \hat{I}_{r_{2s_2}}} \dots \left( \bigvee / \bigwedge \right)_{\hat{\gamma}_{rk1} \in \hat{I}_{rk1}} \dots \left( \bigvee / \bigwedge \right)_{\hat{\gamma}_{rk s_k} \in \hat{I}_{rk s_k}} \varphi(\hat{\gamma}_{r_{11}}/X_{11}, \dots, \hat{\gamma}_{r_{1s_1}}/X_{1s_1}, \dots, \hat{\gamma}_{r_{21}}/X_{21}, \dots, \hat{\gamma}_{r_{2s_2}}/X_{2s_2}, \hat{\gamma}_{rk1}/X_{k1}, \dots, \hat{\gamma}_{rk s_k}/X_{ks_k}).$$

We note that all the sets  $\hat{I}_{r_{tu}}$ , for  $1 \leq t \leq k$  and  $1 \leq u \leq s_t$ , are finite by Lemma 4. Then, since all disjunctions and conjunctions are finite,  $\hat{\psi}_{\mathcal{C}} \in \text{FO}$ .  $\square$

*Example 1.* Fixing  $h \geq 1$ , the class of full trees with arbitrary arity  $r$  and depth  $h$  is an example of a class with bounded FO types. This class has an infinite number of FO types but has a finite set of intra-isolating formulae. Given a tuple  $\bar{a}$  in an  $r$ -ary full tree  $\mathcal{T}_r$  and a tuple  $\bar{b}$  in an  $(r+1)$ -ary full tree  $\mathcal{T}_{r+1}$ , the FO type of  $\bar{a}$  is different to the FO type of  $\bar{b}$  but they can satisfy the same intra-isolating formula. Nevertheless, for a particular full tree  $\mathcal{T}_r$  if two tuples on  $\mathcal{T}_r$  satisfy the same intra-isolating formula then they have the same FO type. In every full tree there are  $h+1$  different FO types for elements. Since the elements at the same level have the same FO type. As an example, let  $h = 2$  and  $r = 3$ , then we have the following FO types for pairs of elements:

We have an equivalence class  $C_1$  for the pairs of elements which are siblings at level 2. This class is symmetrical. The equivalence class  $C_2$  consists of pairs

which are “cousins” at level 2. This class is symmetrical. The equivalence class  $C_3$  consists of pairs of elements which are siblings at level 1.  $C_3$  is symmetrical. The equivalence class  $C_4$  consists of pairs where the first component is an element in level 1 and the second component is a child in level two.  $C_4$  has an associated class  $C_5$  with the symmetrical pairs. The class  $C_6$  consists of pairs where the first component is the root node and the second component is an element of level 2.  $C_6$  has an associated class  $C_7$  with the symmetrical pairs. The class  $C_8$  consists of the pairs where the first component is an element of level 1 and the second component is an element of level two which is not a child of the element of level 1.  $C_8$  has an associated class  $C_9$  with the symmetrical pairs. The class  $C_{10}$  consists of pairs where the first component is the root node and the second component is a child of root node.  $C_{10}$  has an associated class  $C_{11}$  with the symmetrical pairs. The class  $C_{12}$  consists of a reflexive pair with the root node. The class  $C_{13}$  consists of reflexive pairs with elements of level 1. The class  $C_{14}$  consists of reflexive pairs with elements of level 2. All of these classes of pairs can be expressed with FO formulae with two free variables and all of them satisfy an unique intra-isolating formula.

When we consider 3-tuples instead of 2-tuples the combinatorics grow but they remain finite independently of the arity of the tree. Then all queries in  $SO^F$  on the class of full tree with fixed depth can be translated to equivalent queries in FO.

## 5. $SO^F$ over Sets

A special case of structures with bounded FO types is any recursive class of structures with a unary vocabulary. Note that in these structures the relations are sets. We can build formulae which isolate the FO types for elements in a given structure considering how each element participates in the different sets. Using this formulae we can build formulae which isolate the FO type for  $k$ -tuples in a given structure. It is interesting to see that, fixing a unary vocabulary, there is a finite set of intra-isolating formulae for FO types for  $k$ -tuples.

**Definition 3.** Let  $\sigma = \{S_1, \dots, S_t\}$  be a unary vocabulary. A binary conjunction is a formula  $\varphi_{i_1 \dots i_t}(x) = l_{i_1} \wedge \dots \wedge l_{i_t}$  with  $l_{i_j} = S_j(x)$  when  $i_j = 1$ , and  $l_{i_j} = \neg S_j(x)$  when  $i_j = 0$  for  $1 \leq j \leq t$  and  $i_j \in \{0, 1\}$ .

For example, for  $\sigma = \{S_1, S_2, S_3\}$  we have the following binary conjunctions:

$$\begin{aligned} \varphi_{000}(x) &= \neg S_1(x) \wedge \neg S_2(x) \wedge \neg S_3(x) \\ \varphi_{001}(x) &= \neg S_1(x) \wedge \neg S_2(x) \wedge S_3(x) \\ &\vdots \\ \varphi_{111}(x) &= S_1(x) \wedge S_2(x) \wedge S_3(x) \end{aligned}$$

**Lemma 5.** Let  $\sigma = \{S_1, \dots, S_t\}$  be a unary vocabulary with  $t \geq 1$ . Let  $\mathcal{C} \subseteq B_\sigma$  and  $\Phi^1 = \{\varphi_i(x) | i \in \{0, 1\}^t \text{ and } \varphi_i \text{ is a binary conjunction}\}$ . Then,  $\Phi^1$  is a set of intra-isolating formulae for FO types for elements on  $\mathcal{C}$ .

*Proof.* We must prove that  $\Phi^1$  satisfies Def. 1.

Let  $\mathcal{A} \in \mathcal{C}$ . First we will see that condition *i*) holds. Let  $a_1$  and  $a_2$  be elements in  $\text{dom}(\mathcal{A})$  such that  $\mathcal{A} \models \varphi_i(x)[a_1]$  and  $\mathcal{A} \models \varphi_i(x)[a_2]$  for  $i \in \{0, 1\}^t$ , then the bijective function  $f$  that exchanges  $a_1$  with  $a_2$  and fixes the other elements in  $\text{dom}(\mathcal{A})$  is an automorphism on  $\mathcal{A}$ . For  $S_j \in \sigma$ , with  $1 \leq j \leq t$  it holds that  $\mathcal{A} \models S_j(x)[a_1] \Leftrightarrow \mathcal{A} \models S_j(x)[a_2]$  since  $S_j$  is in  $\varphi_i(x)$  either positively or negatively. All elements  $a_l \notin \{a_1, a_2\}$  trivially satisfy  $\mathcal{A} \models S_j(x)[a_l] \Leftrightarrow \mathcal{A} \models S_j(x)[a_l]$ . Then for all  $a_l \in \text{dom}(\mathcal{A})$  and for all  $S_j \in \sigma$  it holds  $\mathcal{A} \models S_j(x)[a_l] \Leftrightarrow \mathcal{A} \models S_j(x)[f(a_l)]$ . Therefore  $\text{type}_{\mathcal{A}}^{FO}(a_1) = \text{type}_{\mathcal{A}}^{FO}(a_2)$ .

For condition *ii*) Let  $a$  and  $b$  satisfy  $\text{type}_{\mathcal{A}}^{FO}(a) = \text{type}_{\mathcal{A}}^{FO}(b)$ , let  $i = i_1 \dots i_t$  with  $i \in \{0, 1\}^t$  and  $i_j = 0$  when  $\mathcal{A} \not\models S_j(x)[a]$  and  $i_j = 1$  when  $\mathcal{A} \models S_j(x)[a]$  for  $1 \leq j \leq t$ , then  $\mathcal{A} \models \varphi_{i_1 \dots i_t}(x)[a]$  and  $\mathcal{A} \models \varphi_{i_1 \dots i_t}(x)[b]$  but  $\mathcal{A} \not\models \varphi_r(x)[a]$  and  $\mathcal{A} \not\models \varphi_r(x)[b]$  for  $r \in \{0, 1\}^t$  and  $r \neq i$ .

We prove condition *iii*). For  $i$  and  $j$  as above, let  $a$  be an element in  $\text{dom}(\mathcal{A})$ . Let  $i_j = 0$  when  $\mathcal{A} \models \neg S_j(x)[a]$  and  $i_j = 1$  when  $\mathcal{A} \models S_j(x)[a]$ . Then  $\mathcal{A} \models \varphi_{i_1 \dots i_t}(x)[a]$  and  $\varphi_{i_1 \dots i_t}(x) \in \Phi^1$ .  $\square$

**Lemma 6.** Let  $k \geq 1$ ,  $t \geq 1$ ,  $\sigma = \{S_1, \dots, S_t\}$  a unary vocabulary and  $\mathcal{C} \subseteq B_\sigma$ . Let  $\Phi^k = \{\varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k) \mid \varphi_{u_l}$  is a binary conjunction,  $u_l \in \{0, 1\}^t$  and  $1 \leq l \leq k\}$ .  $\Phi^k$  is a set of intra-isolating formulae for FO types for  $k$ -tuples on  $\mathcal{C}$ .

*Proof.* Let  $\mathcal{A} \in \mathcal{C}$ . We prove the condition *i*) of Def. 1. Let  $\bar{a}, \bar{b} \in \text{dom}(\mathcal{A})^k$  and  $\varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k) \in \Phi^k$  such that  $\mathcal{A} \models \varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k)[\bar{a}]$  and  $\mathcal{A} \models \varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k)[\bar{b}]$ . Let  $f_l$  be the bijective function that exchange  $a_l$  with  $b_l$  for  $1 \leq l \leq k$  and fixes the other elements in  $\text{dom}(\mathcal{A})$ ,  $f_l$  is an automorphism as we show in Lemma 5. Then the composition  $f_1 \circ \dots \circ f_k$  is an automorphism that exchanges the  $k$ -tuple  $\bar{a}$  with the  $k$ -tuple  $\bar{b}$ , and fixes the other elements in  $\text{dom}(\mathcal{A})$ . Therefore  $\text{type}_{\mathcal{A}}^{FO}(\bar{a}) = \text{type}_{\mathcal{A}}^{FO}(\bar{b})$ .

For condition *ii*) we suppose that for  $k$ -tuples  $\bar{a}$  and  $\bar{b}$  it holds that  $\text{type}_{\mathcal{A}}^{FO}(\bar{a}) = \text{type}_{\mathcal{A}}^{FO}(\bar{b})$ . Let  $\varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k)$  with  $u_l \in \{0, 1\}^t$  and  $u_l = i_1 \dots i_t$  such that  $\varphi_{i_1 \dots i_t}(x)$  is as the proof of Lemma 5 and  $1 \leq l \leq k$ . Then  $\mathcal{A} \models \varphi_{u_l}(x_l)[a_l]$  and  $\mathcal{A} \models \varphi_{u_l}(x_l)[b_l]$  and for all  $r \in \{0, 1\}^t$  such that  $u_l \neq r$ ,  $\mathcal{A} \not\models \varphi_r(x_l)[a_l]$  and  $\mathcal{A} \not\models \varphi_r(x_l)[b_l]$  for  $1 \leq l \leq k$ . Therefore  $\mathcal{A} \models \varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k)[\bar{a}]$  and  $\mathcal{A} \models \varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k)[\bar{b}]$ , with  $\varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k) \in \Phi^k$  and for all  $\varphi_{v_1}(x_1) \wedge \dots \wedge \varphi_{v_k}(x_k) \in \Phi^k$  with  $v_1 \dots v_k \in \{0, 1\}^t$  and  $u_1 \dots u_k \neq v_1 \dots v_k$  it holds that  $\mathcal{A} \not\models \varphi_{v_1}(x_1) \wedge \dots \wedge \varphi_{v_k}(x_k)[\bar{a}]$  and  $\mathcal{A} \not\models \varphi_{v_1}(x_1) \wedge \dots \wedge \varphi_{v_k}(x_k)[\bar{b}]$ .

We prove condition *iii*). Let  $\bar{a} = (a_1, \dots, a_k) \in \text{dom}(\mathcal{A})^k$ . Let  $u_{l_j} = 0$  when  $\mathcal{A} \models \neg S_j(x_l)[a_l]$  and  $u_{l_j} = 1$  when  $\mathcal{A} \models S_j(x_l)[a_l]$  for  $1 \leq l \leq k$  and  $1 \leq j \leq t$ , then  $\mathcal{A} \models \varphi_{u_l}(x_l)[a_l]$  for  $u_l = u_{l_1} \dots u_{l_t}$ . Therefore  $\mathcal{A} \models \varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k)[\bar{a}]$  and  $\varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k) \in \Phi^k$ .  $\square$

**Theorem 2.** Let  $\mathcal{C}$  be a class of structures with unary vocabulary, then  $SO^F$  is equivalent to FO on  $\mathcal{C}$ .

*Proof.* Let  $t \geq 1$ , and  $\sigma = \{S_1, \dots, S_t\}$ , and  $\mathcal{C} \in B_\sigma$  then, for all  $k \geq 1$ , by Lemma 6,  $\Phi^k = \{\varphi_{u_1}(x_1) \wedge \dots \wedge \varphi_{u_k}(x_k) \mid \varphi_{u_l}$  is a binary conjunction,  $u_l \in$

$\{0, 1\}^t$  and  $1 \leq l \leq k$ , is a set of intra-isolating formulae for FO types for  $k$ -tuples over  $\mathcal{C}$ . Therefore, by Theorem 1,  $SO^F$  is equivalent to FO on  $\mathcal{C}$ .  $\square$

Among other queries, the parity query<sup>3</sup> over sets is not expressible in  $SO^F$  since as it is well known parity is not expressible in FO. However this query is expressible in SO for structures of arbitrary vocabulary, in particular for unary vocabulary. Then, the following Corollary is immediate.

**Corollary 1.**  *$SO^F$  is strictly included in SO.*

## 6. Conclusion

In the existential fragment  $\Sigma_1^{1,F}$  of  $SO^F$  we can express co-NP problems . In [4] we proved that there are NP complete problems that can be expressed in  $\Sigma_1^{1,F}$ . However we cannot express in full  $SO^F$  the parity query which is in P. Then we can conclude that different logics allow orthogonal classifications of the problems with respect to the classic classification of computational complexity. These orthogonal classifications can be used to refine the classic computational complexity classes, providing us more information about certain problems.

## References

1. Abiteboul, S., Vianu, V.: Datalog extensions for database queries and updates. J. Comput. System Sci. 43, 62-124 (1991)
2. Dawar. A.: A Restricted Second Order Logic for Finite Structures. Information and Computation 143, 154-174 (1998)
3. Fagin, R.: Generalized first-order spectra and polynomial-time recognizable sets. Complexity of Computation (R. M. Karp Ed.), SIAM-AMS Proceedings. 7, 43-73 (1974)
4. Ferrarotti F. A., Grosso A. L., Turull Torres J. M.: Semantic Restrictions over Second-Order Logic. To be presented 6th International Workshop on Semantics in Data and Knowledge Bases (2012)
5. Ferrarotti F. A., Paoletti A. L., Turull Torres J. M.: Redundant Relations in Relational Databases: A Model Theoretic Perspective. in the Journal of Universal Computer Science Vol. 16, 20, 2934-2955 (2010)
6. Ferrarotti, F. A., Turull Torres J. M.: The Relational Polynomial-Time Hierarchy and Second-Order Logic, invited for "Semantics in Databases", edited by K-D. Schewe and B. Thalheim, Springer, LNCS 4925, 48-76 (2008)
7. Grosso, A. L., Turull Torres J. M.: A Second-Order Logic in which Variables Range over Relations with Complete First-Order Types. 2010 XXIX International Conference of the Chilean Computer Science Society (SCCC). IEEE 270 - 279 (2010)
8. Grosso, A. L., Turull Torres J. M.:  $SO^F$  : A Semantic Restriction over Second-Order Logic and Its Polynomial-Time Hierarchy . Conceptual Modelling and Its Theoretical Foundations (LNCS). Springer. ISBN 978-3-642-28278-2 (2011)
9. Libkin L.: Elements of Finite Model Theory. Springer (2004). ISBN 3-5402-1202-7.
10. Stockmeyer, L.: The polynomial-time hierarchy. Theoret. Comput. Sci 3, 1-22 (1976)

---

<sup>3</sup> class of  $\sigma$ -structures with an even domain

# Real time images processing using embedded technology. Application in visual rehabilitation.

MARCELO RAPONI<sup>1,2</sup>, EZEQUIEL PAWELKO<sup>3</sup>, LEANDRO ARCUSIN<sup>4</sup>,  
CLAUDIO DELRIEUX<sup>5</sup>

<sup>1</sup> Escuela de Ciencia y Tecnología, ECyT-UNSAM, Buenos Aires, Argentina

<sup>2</sup> Asociación Argentina de Baja Visión, Mar del Plata, Buenos Aires, Argentina

<sup>3</sup> Instituto Universitario de la Policía Federal Argentina, IUPFA, CABA, Argentina

<sup>4</sup> Facultad Regional Buenos Aires, UTN, CABA, Argentina

<sup>5</sup> Departamento de Ingeniería Eléctrica y Computadoras, UNS, IIIE (CONICET), Bahía Blanca, Buenos Aires, Argentina  
raponimarcelo@gmail.com

***Abstract.** A visual rehabilitation system able to improve the life quality of patients with severe visual disorders (low vision) is presented. The device acquires and processes images in real time, does a selective enhancement of visual information and translates that information to an appropriate stimulation pattern for each patient. The software was entirely developed using open-source libraries (OpenCV & Qt) and is modularized enough to allow a fast adaptation to new embedded devices and an effective redesign adjusting the platform to the patients's pathology evolution. The system was designed to run on embedded system such as the BeagleBoard development board, which is easily configurable, portable and inexpensive. The tools used to implement different processing, are described, and the results obtained by applying them on reals signals, are presented.*

***Keywords:** Low vision, visual rehabilitation, image processing, embedded systems, OpenCV.*

## 1. Introduction

The sense of vision is essential for carrying out the different activities that humans perform daily. Tasks as common as crossing a street, reading a book, watching TV, among many others, are severely affected by eye diseases that cause partial or complete blindness in millions of human beings. Worldwide, a lot of people have severe visual dysfunctions without becoming completely blind (condition known as low vision). WHO (World Health Organization) defines low vision as "loss of visual acuity (VA) and/or visual field (VF), which is incapable of performing daily life's tasks, even after treatment and/or standard refractive correction." Patients with severe reduction of visual field have a decreased mobility (they are unable to avoid obstacles or identify terrain's defects effectively). If the peripheral vision is intact but the visual acuity is affected, activities that require high visual acuity (such as reading, writing, sewing, etc.) are seriously affected. Through

the use of optical and electronic systems (video magnifiers, CCTV, etc.) is possible to achieve some degree of visual rehabilitation on these patients. Access to these devices is usually restricted due to their high cost, and not always results in a substantial benefit to the patient. In general, they are narrowly focused on a specific kind of deficiency, and therefore do not get adapted to the patient's evolution.

In this paper, the development of a vision rehabilitation device designed to improve the life quality of people with severe visual dysfunctions is presented. The system is characterized as being reconfigurable, portable and inexpensive. It is capable of acquiring and processing images in real time and making selective enhancement of visual information, mapping this information into a stimulation pattern suitable for each patient. The software is based on free platforms (open source) and is modularized enough to allow a fast adaptation to new embedded devices or libraries, and an effective redesign to adjust it to patient's pathology evolution.

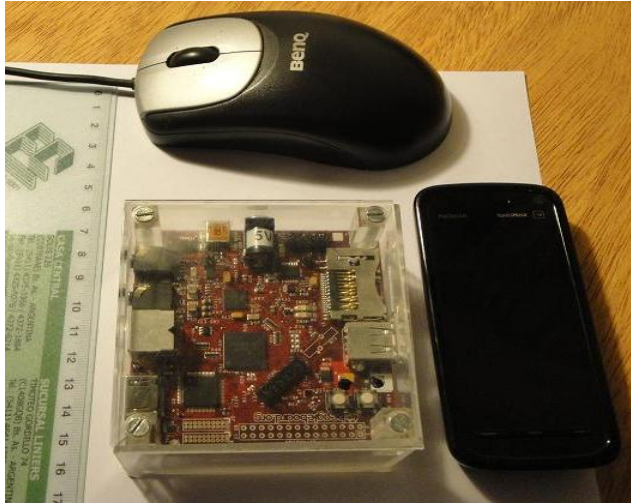
## 2. Materials and Methods

The developed device to assist visually to people with visual impaired, consist of a reconfigurable system based on embedded technology and digital image processing algorithms. The framework has three components: a video signal acquisition module, a processing module whose core is an ARM (Advanced RISC Machines) [1] and a display module. Acquisition is performed by a mini-camera (Logitech Pro-9000, 1600x1200 pixels of spatial resolution and adjustable focus) which is connected to the control unit via a USB 2.0 port. The output device is a video glasses (video eyewear EVG920V) which has two graphics TFT LCD display (640x480 pixels, VGA resolution), that generate virtual images of 80" approximately at 2 meters away. It accepts composite video signal (NTSC /PAL /SECAM), consume less than 1.1 W and its battery lasts around 4 hours.

The core of the system is a BeagleBoard development board [2] with an ARM ® Cortex™ A8 working at 600 MHz, 256 Mb of RAM, a C64x + DSP and a graphics accelerator. The board has connectivity with several peripherals: keyboard, USB mouse, webcam, LCD/LED display, TV, SD memory, etc. Preliminary tests show that is possible to acquire, process and display in real time digital images with very satisfactory results [3]. For the design of the rehabilitation device we took into account a number of requirements, including:

*Size:* it was necessary to minimize the device's size to improve its portability and adapt it to a wider range of everyday situations. To do this we chose a system on chip (SoC) that brings multiple functions in a single platform. Fig 1 shows the control unit dimensions (located inside a specially designed acrylic box) relative to the size of a mouse and a phone.





*Fig. 1. Control and processing unit size. The Beagleboard development board is inside a specially designed acrylic box.*

*Energy consumption:* the device's dimensions also affect the battery's size and therefore the usage time without recharging. ARM type processors are a good choice due to their high computational power and low energy consumption.

*Input image quality:* standard webcams have low resolution and high blurring (between video frames) sensors. To achieve autonomous operation, an auto-focus mechanism and at least 640x480 spatial resolution are required.

*Processing capabilities:* graphics pipeline operations - specially dewarping (to remove spherical distortion) and optical character recognition (OCR) - require large computational power. This is provided by the DSP (Digital Signal Processor) and the Beagleboard's graphics accelerator.

*Graphic interface latency and usability:* to work in real time demands a reduced latency, so, the design of blocking reduction mechanisms regarding the user interface, implementing threads for concurrent operations is required.

The main features of the software include: reconfigurability (it fits to the patient and the stage of their disease), simplicity, versatility (multi-function), wide connectivity (to video glasses, PC monitor, LCD/LED TV, etc.), adjustable digital zoom and autofocus, enhanced vision, OCR + TTS (character recognition and text-to-speech, etc.) Acquisition, processing and visualization of images involve an operating system, kernel drivers, low-level graphics libraries, graphics toolkits and user space tools. The embedded operating system Angstrom v2.6.32 [4] was selected and images acquisition was performed using V4L2 drivers (called from the highgui OpenCV library). OpenCV library [5, 6] provides the data structures and algorithmic

elements both initial and advanced. In Fig. 2 a software solution scheme is shown.

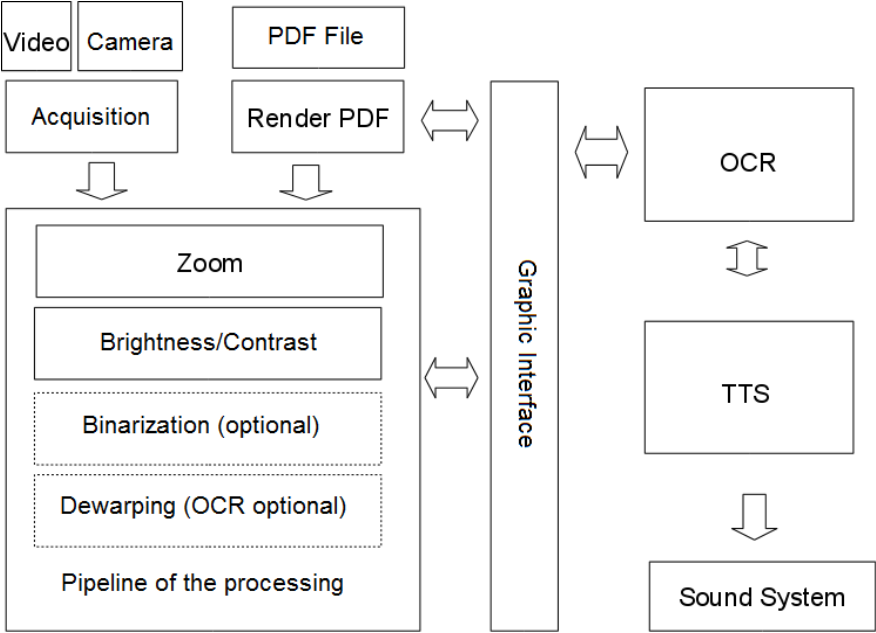


Fig. 2. Software architecture.

### 3. Results

Several algorithms were implanted. Some of one will be described below. Since the Linux driver does not provide an autofocus algorithm it was developed for the application. It consists on calculating the F operator which gives an idea of the current focus of the camera. Equation 1 is used to calculate F:

$$F = \left( \frac{\sum |S(I)|}{n} \right)$$

(1)

where S is a 3x3 Sobel filter applied on each frame of the video signal (matrix I). The mean value of the filtered image is calculated, being n the number of pixels. The F value is related to the presence of high spatial frequencies, the more focused the filtered image is, the greater the F value. The selected webcam enables users to manage its focus by sending numbers between 0 and 255. In order to generate an autofocus, an algorithm with the a

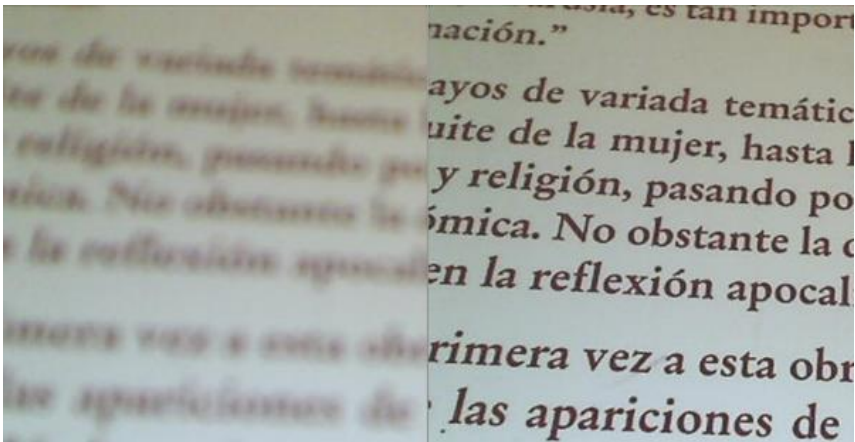
few iterations was designed. We describe the algorithm for calculating the optimal focus sequentially:

```
Set variable max=0.0
Set maxFoco=0
for (i=MinFocusValue;i<= MaxFocusValue;i=i+Step)
{
  SetFocus(i)
  GetImage
  derivative= Sobel (imagen)
  results= mean ( abs ( derivative) )

  if (results > max)
  {
    max=current;
    maxFocus=i ;
  };
};
```

*Pseudo code to determine the best approach.*

An improvement to this autofocus algorithm would be to use a zigzag searching technique in order to avoid return to the starting point in each iteration. In Fig. 3 the results of the autofocus algorithm application are shown.



**Fig. 3.** *Effect of applying autofocus algorithm on text sharpness.*

In order to improve the features distinction of the captured images, a combination of algorithms that allow to highlight the edges of the object present in the visual scene was used. The processing is done in three steps: the first one consist of blurring the image to reduce jpg compression effects which generates certain artifacts (mobile blocks). After this an adaptive

thresholding and a morphological operator (erosion) are applied in order to enhance divisions between the features highlighted (Fig. 4). As extra operation, the extreme luminance values (above the threshold) are equalized to an intermediate value to preserve the specific color information, avoiding the rings generation in the image which could be expected for areas with excessive difference of luminance.



(a)



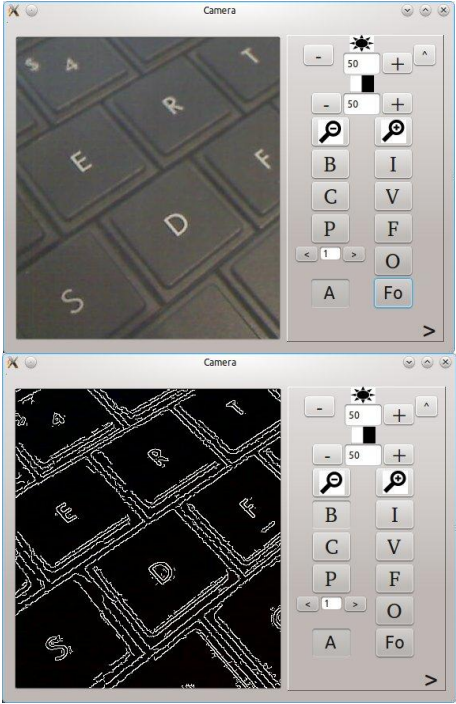
(b)



(c)

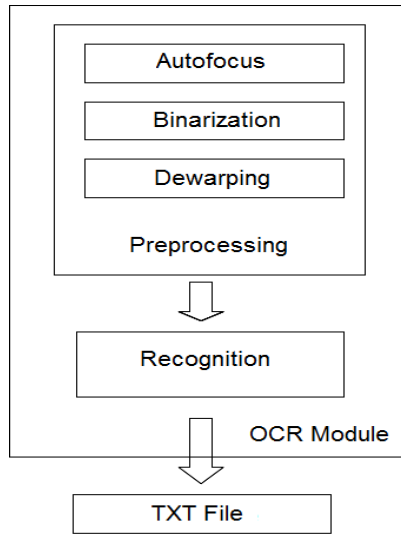
**Fig. 4.** a) Original image, b) Image with maximum luminance and application of morphological operator (erosion 5x5), c) Image with average luminance and erosion 3x3.

To highlight the edges of objects in visual scenes, a Canny filter was implemented. Fig 5 shows the prototype interface with webcam acquisition mode and the result of applying a Canny filter to the video signal in real time, to detect edges (other filters that can be applied are the Laplacian filter the Sobel filter).



**Fig. 5.** Edge detection applied to the video signal captured by the webcam: (left) Original image and (right) filtered image.

Another function developed is the text recognition (OCR, Optic Character Recognition) and its translation into a TXT file that would be read by a TTS engine (Text to Speech) (Fig. 6). In order that the operator OCR works properly, image features should be separated from its background and the image should have a minimum of transformations. Among pre-processing algorithms for OCR, the binarized operation plays a key role because it is responsible of separating the background of the characters.

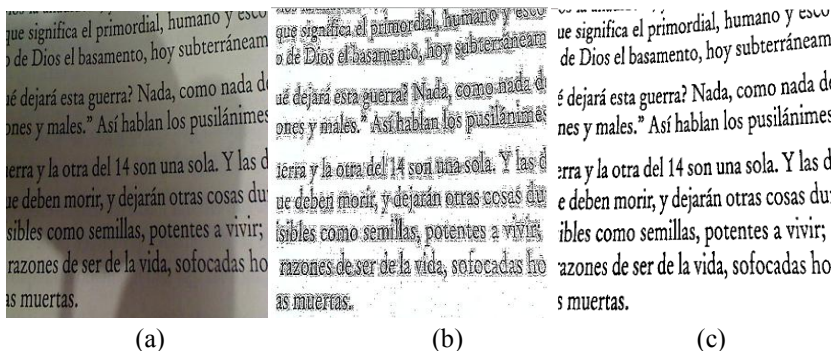


*Fig. 6. Image pre-processing algorithms for OCR.*

Sauvola algorithm [7] has shown a great performance in documentary images binarization due to its effectiveness and simplicity. In this technique, the binarization threshold  $t(x,y)$  is calculated using the mean  $\mu(x,y)$  and the standard deviation  $\sigma(x,y)$  of the intensities of the pixels in a window  $W_{xy}$  centered around the pixel with coordinates  $x,y$  (equation 2):

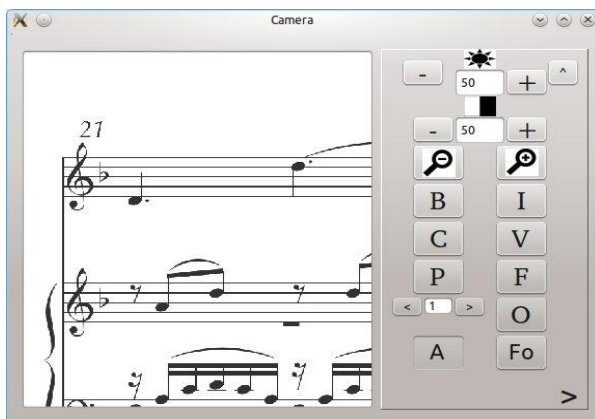
$$t(x,y) = \mu(x,y) \left[ 1 + k \left( \frac{\sigma(x,y)}{R} \right) - 1 \right] \quad (2)$$

where  $R$  is the maximum value of the standard deviation ( $R = 128$  for a gray scale document) and  $k$  is a parameter that takes positive values. Current computational implementation of this binarization is an efficient variation on the one implemented in the OCRopus package (<http://code.google.com/p/ocropus/>). Fig. 7 shows a comparison between the adaptive threshold algorithm – used in the second step of the traditional binarization – and the Sauvola algorithm applied to a JPG compressed image (which is common in webcam images). As can be seen, Sauvola algorithm is very resilient to local perturbations such as ringing and blocking effects own jpg files.



**Fig. 7.** Comparison between binarization algorithms: (a) Original image, (b) adaptive threshold, (c) Sauvola algorithm.

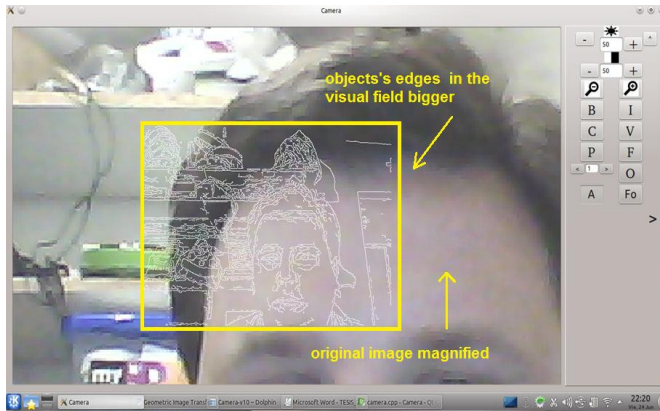
In addition to the previous mentioned algorithms, the interface developed allows manipulating PDF (portable document format) files. Fig. 8 shows a music score in PDF format which is processed in different ways: color inversion, edge detection, etc. Note that the user can scroll through the different pages of the document using a simple control located in a sidebar.



**Fig. 8.** Viewing a PDF file (musical score)

An important algorithm developed is the augmented vision, which provides patients with tunnel vision, the necessary information about objects located in its periphery (outside the patient's visual field, which varies according to the disease stage), improving their mobility without compromising residual central vision. To compose the images shown on the portable display, video frames are processed using edge detection filtering. In a same frame the original zoomed image and the edges of objects (in white or other user selected color) are presented (Fig. 9). The purpose of this computational tool is to provide the patient with information about the objects in a wide field of view (with less resolution or detail, but good enough for moving), and at the same time letting enjoy high resolution residual central vision.





*Fig. 9. Augmented vision mode*

Images acquired by a mobile camera have commonly deformations due to multiple sources: perspective from which you take, lens curvature (webcams are designed to take faces from a distance larger than 30-40 cm) and deformation effects due to binding. In order to solve such disturbances, two algorithms are being developed: an aspect ratio correction by means of the estimate of a trapezoidal container and its four associated control points, and curls correction (based on the Bukhari's coupled snakelets [8]).

## Conclusions

The development of a visual rehabilitation device to improve the life quality of patients with severe visual disorders was presented. The device was designed to acquire and process images in real time, carry out a selective enhancement of the visual information and translate that information to an appropriate stimulation pattern for each patient. Details on hardware and software implementation were shown. The developed system uses only open source libraries and it was designed to allow fast adaptation to new embedded devices and different pathologies. Preliminary results were obtained in real situations, which shown that development is really applicable as visual rehabilitation device, and it is currently in testing stage with volunteer patients. A future work that is being considered is the use of implants stimulators to use this device in cases of complete blindness.

## References

1. ARM, descripción del procesador Cortex A8. (www.arm.com/products/processors/cortex-a/cortex-a8.php)



2. Coley, G.: BeagleBoard System Reference Manual Rev C4 (2009). ([www.beagleboard.org/static/BBSRM\\_latest.pdf](http://www.beagleboard.org/static/BBSRM_latest.pdf))
3. Raponi, M.M., Bonnin, R.O.: Dispositivo de rehabilitación visual basado en sistemas embebidos del tipo ARM. Congreso Argentino de Sistemas Embebidos, CASE. UTN-FRBA, Argentina. ISBN 978-987-9374-69-6, pp. 85--88 (2011)
4. The Ångström Distribution Embedded power ([www.angstrom-distribution.org](http://www.angstrom-distribution.org))
5. Bradski, G., Kaebler, A.: Learning OpenCV. Ed. O'Reilly Media Inc., ISBN 978-0-596-51613-0 (2008)
6. Laganière, R.: OpenCV 2 Computer Vision Application Programming Cookbook. Ed. Packt Publishing. ISBN 978-1-849513-24-1 (2011)
7. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recognition, 33 (2), pp. 225--236 (2000)
8. Bukhari, S.S., Shafait, F., Breuel, T.M.: Coupled Snakelet Model for Curled Textline Segmentation of Camera-Captured Document Images. Proc. 10th Int. Conf. on Document Analysis and Recognition, ICDAR'09, Barcelona, Spain (2009)

# Preliminary Work to Get Synchronized Times in Multicore Node Clusters

FERNANDO ROMERO, HORACIO VILLAGARCÍA<sup>1</sup>, FERNANDO G. TINETTI<sup>2</sup>

Instituto de Investigación en Informática LIDI (III-LIDI)  
School of Computer Science, UNLP  
{fromero, hvw, fernando}@lidi.info.unlp.edu.ar

***Abstract.** This paper presents an expansion of the methodology and tool for instrumenting parallel programs in distributed computer platforms that has been described in previous works. Specifically, this expansion considers multicore hardware that is currently used in the individual nodes of clusters. In the context of instrumentation in distributed environments, it is essential that all involved clocks are synchronized. The basic synchronization algorithm remains almost unchanged, using the classic strategies that are commonly used in distributed environments for cluster configurations, with an interconnection network with exclusive (or controlled) access to all communications among the synchronized computers. This environment is specifically that of parallel computing environments in clusters.*

***Key words:** Parallel Performance Instrumentation, Process Synchronization, Distributed Clocks, Parallel and Distributed Systems, Intracluster and Intercluster Parallelism, Internal and External Synchronization.*

## 1. Introduction

This paper is a continuation of the research line on computer clock synchronization aimed at carrying out program instrumentation [5] [6] [7] [8] [9] [10] [11] [12]. The previously presented tool is expanded to include multicore systems. The strategies used in monoprocessor systems are not suitable for these multicore systems. The time reference used, originally given by the hardware device “TimeStamp Counter” (TSC), has two limitations:

- 1) Each core has its own separate TSC, which is not synchronized with the others.
- 2) It is possible that energy saving systems alter the reference update frequency in TSC. This means that no assumptions can be made regarding the constant update frequency of TSC (beyond the physical features of the TSC itself and the environment).

---

<sup>1</sup> Profesional Principal CICPBA

<sup>2</sup> Investigador CICPBA

These limitations have led to the search for an alternative time reference that, while meeting all requirements upon which the design of the instrumentation tool was based [8], would be free of these issues. The original requirements are as follows:

- A tool that can be initially used in a cluster of PCs and allows being extended to clusters in general and then to even more general distributed platforms.
- It has a high resolution, that is, it can be used to measure short times, of the order of the microseconds.
- It does not alter the operation of the application being tested, or the alteration is minimal and known so that all necessary performance tests can be done.
- It uses the interconnecting network in a predictable way. More specifically, the time intervals during which the network will be used can be determined from the application. Thus, the use of the interconnecting network can be decoupled, since there will be time intervals used for synchronization and time intervals used for the execution of parallel programs.

By analyzing the last changes in peripheral devices, the use of “*High Precision Event Timers*” (HPET) registers [4] as time reference is considered. All requirements mentioned above are met, and a unique reference is provided for all cores within a node in each cluster. Evidently, separate nodes are independent from each other and have to be synchronized; to do so, the standard methodology/algorithm will be used.

## 2. Hardware and HPET

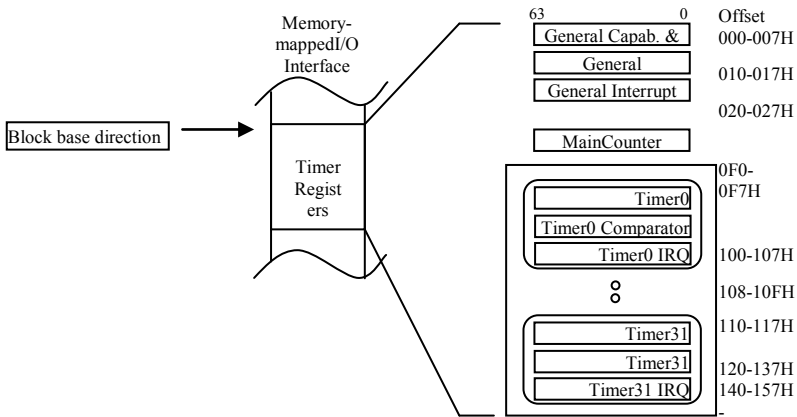
Industry specifications in manuals for OEM users and BIOS providers that use series of Intel chipsets as the basis to their products [3] describe the “Platform Controller Hub” (PCH) as a generic term encompassing I/O support features and capabilities. These manuals describe in detail the features for the signals (time and electric), packages, memory mapping and registers, as well as the various interfaces and their configuration registers. Among the features and capabilities that are traditionally used, there are specifications for PCI, USB, SATA, PCI Express, and LAN. New capabilities are also added, including additional timer hardware to complement and eventually replace the features of time interval generation and periodic interrupts provided by 8254 Programmable Interval Timer and Real Time Clock in personal computers based on Intel architectures (IA-PC). High Performance Event Timers (HPET) are defined [4] as a set of timer records divided in blocks that the operating system can use, and in the future it will even be able to assign them to be used specifically by an application. Their use is proposed for audio/video synchronization; task, thread, or process scheduling by generating interrupts; and time marking operations in multiprocessor platforms.

## 2.1 Timer Registers

Each timer can be configured to generate a separate interrupt. The architecture allows for 32-timers blocks and supports up to 8 blocks, i.e. up to 256 timers. The minimum support is set at 3 timers. Timers are implemented with a single ascending counter (*main counter*) and a set of comparators and match registers. The ascending counter increases its count monotonically, that is, when two consecutive counter readings are made, the value of the second reading is always greater than that of the first reading, except in roll over situations. Each timer includes a comparator and a match register. Each timer can generate an interrupt when its match register value is equal to the value of the ascending counter. The interrupt generated by each timer will have well-defined configuration and routing characteristics in a related register that can, for instance, enable the periodic generation of that interrupt. The registers that are associated to the timers are assigned to the memory space in a similar way to that used for the advanced input/output interrupt controller (I/O APIC, *Advanced Programmable Interrupt Controller*). However, they are not implemented as a standard PCI feature. The BIOS reports to the operating system the memory location assigned to the timer block. The operating system is not expected to change timer location once it is established by the BIOS.

## 2.2 Memory Mapping

The CPU can directly access each timer register because these are mapped to the memory. Timer register space is 1024 bytes. They are aligned in 64-bit bounds to simplify their implementation with 64-bit processors. The register model allows each timer block to have up to 32 timers, where each timer consists of a comparator plus a match register. Figure 1 shows a general view of timer registers and their mapping to memory. As it can be observed, HPET block operation is defined in three general registers (capabilities and identification, configuration, and interrupt generation) and a single ascending counter. These are located from the base location of the block, on shifts 0H, 10H, 20H, and F0H, respectively. The first of these registers (general capabilities and identification register) includes definitions for the period used by the ascending counter and the number of timers implemented in the hardware.

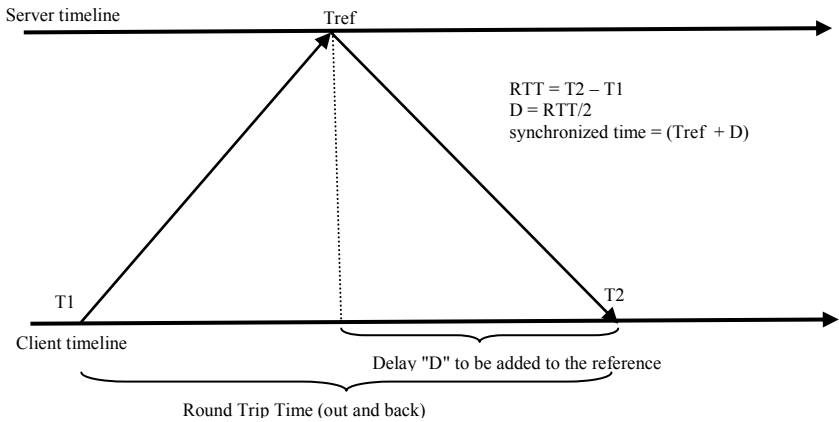


**Figure 1.** Register Model and Their Relative Memory Location

The memory positions that follow the previous ones hold the registers related to each timer, in successive order, starting with timer 0. For every timer, there are 3 64-bit registers (configuration, matching and interrupt routing) that are placed every 32 bytes. The relative location of the registers corresponding to each timer can be determined using the formula  $(20 * n + 100H)$ , where  $n$  is the timer number.

### 3. The Problem of Clock Synchronization

The same as in most distributed systems, the nodes in a cluster cannot share one single clock. For this reason, the possibility of accessing each node's local clock has to be implemented, and then keep all local clocks synchronized with a server clock (internal synchronization) [2]. Computer clock synchronization in distributed environments is based on limiting time differences at any given point in time. To do so, an initial reference must be communicated to all machines. From that reference, the drift of each individual clock has to be calculated and then corrected. This in turn requires an exchange of more time references between server and client. The communication of these references through the underlying communications network faces the issue of variation in communication times. This is limited by means of an algorithm based on communication time statistical methods [1] [2]. This algorithm records statistics for message communication times between the various nodes. After establishing the most frequent time (mode time), only those messages that arrive within that time are considered to be valid. Reference time validation is carried out at the node that sends the message when it receives the corresponding acknowledgment by calculating the RTT (*Round Trip Time*). Figure 2 shows that the reference will arrive with a delay  $D = RTT/2$ , if out and back times are considered to be equal (or symmetric), which is highly likely in cluster environments and local networks. Communication time symmetry variation is part of the synchronization errors that will be effectively present in the final implementation.



**Figure 2:** Reference communication times between nodes

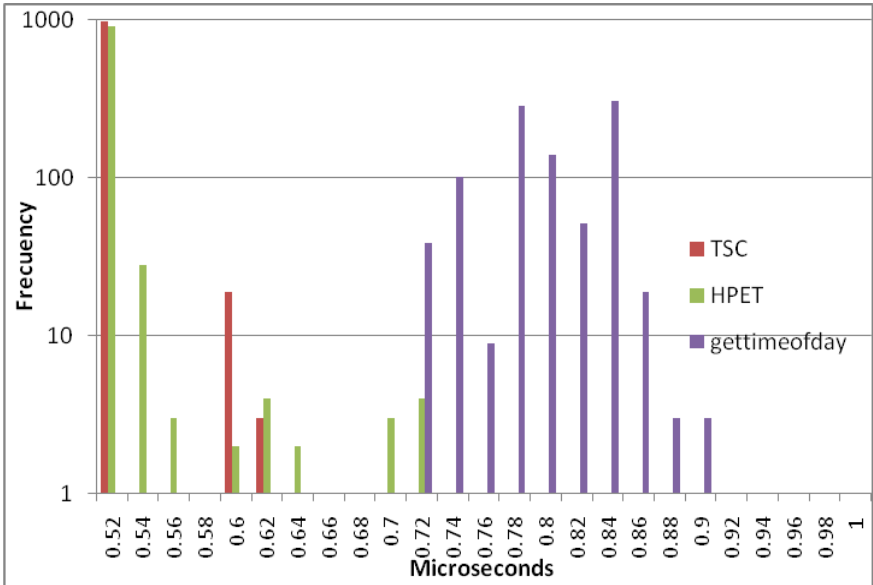
#### 4. Experiments to Determine Overhead and Resolution

When instrumenting distributed applications, the idea is to improve their performance by analyzing the sequence of events at runtime. To correctly analyze the sequence of events of a parallel application, a unique time reference, or at least limited time differences, is required for all nodes in the cluster. The use of such reference should not introduce any significant performance (or operation, in general) differences with respect to the performance or operation of the application alone (intrusion). Experiments were carried out to determine the overhead and resolution of the various time sources, and check that these requirements are met. All experiments were carried out in a cluster of homogeneous nodes formed by Intel XEON E5405 2GHz processors, 2GB RAM, and Linux 2.5.32-5 as operating system. Table 1 shows the statistical results for overhead times of various local time sources, in microseconds.

**Table 1.**Overhead for Various Time Sources.

Source	Max.	Min.	Average
gettimeofday	0.980918	0.674944	0.79087915
HPET	0.989917	0.506958	0.5185626
TSC	0.614949	0.059995	0.07464648

Figure 3 shows overhead frequency histograms for each of the methods that can provide local time references. As it can be seen, the reading time values for HPET and TSC are mostly near the minimum, and those corresponding to gettimeofday are more widespread.



*Figure 3. Overhead Time Distribution for gettimeofday, HPET and TSC.*

This is probably because gettimeofday implies a system call, and therefore, there are several factors that determine its runtime. Even though the lowest overhead time is obtained with TSC, HPET is used to have one single reference for all cores in all processors of each node.

As regards resolution, in most systems, the HPET counter is updated at a frequency of the order of 14MHz, but if successive measurements are made at maximum speed, a time between measurements of 0.55 microseconds is obtained, which could limit resolution to this value. Even if this were the case, the resolution is still within the requirement of being less than or equal to 1 microsecond.

## 5. Conclusions and Future Work

Multiprocessor systems are still evolving, which means that research must continue to find new ways of providing this type of machines time references that meet the requirements mentioned above. In this article, the possibility of using HPET has been shown and, based on the results obtained, the corresponding overhead can be considered to be within acceptable limits. It would appear that having a single time reference for all cores in all possible multiple processors in a node would allow synchronizing all nodes in a cluster. As usual, the immediate work to be done as continuation of this line of research will be the implementation in an entire cluster. This implementation will have to take into account synchronization limits (errors).

Once the synchronization is implemented, it will be used to assess the performance of parallel applications in clusters. In all cases, the interconnection network, which will be used both for synchronization and by the parallel application itself, will have to be taken into account. As in previous works, the use of the interconnection network will be known and controlled before carrying out any experiments. If network performance improves, it could also be possible to reduce synchronization error limits.

## Bibliography

1. Christian F. "Probabilistic Clock Synchronization", *Distributed Computing*, 3: 146-158, 1989.
2. Fetzer C., Christian F., "Integrating External and Internal Clock Synchronization", *Journal of Real-Time Systems*, Vol. 12, Issue 2 pp. 123-171 (1997).
3. Intel Corporation, Intel® C600 Series Chipset Data Sheet, Document Number: 326514-001, March 2012.
4. Intel Corporation, IA-PC HPET Specification, Revision 1.0a (2004).
5. F. L. Romero, A. E. De Giusti, F. G. Tinetti. "Sincronización de Relojes para Evaluación de Rendimiento: Experiencias en un Cluster Utilizado para Cómputo Numérico". XIV Congreso Argentino de Ciencias de la Computación, Univ. Nac. de Chilecito, Chilecito, Argentina, October 2008. ISBN 978-987-24611-0-2.
6. F. G. Tinetti, F. L. Romero, A. E. De Giusti "Clock Synchronization in Clusters for Performance Evaluation: Numeric/Scientific Computing". Proc. 2009 World Congress on Computer Science and Information Engineering, IEEE Computer Society, March 2009, Los Angeles, USA, ISBN 978-0-7695-3507-4/08.
7. F. Romero, "Sincronización de Relojes en Ambientes Distribuidos", Master Thesis in Data Networks, School of Computer Science, UNLP, May 2009.
8. F. G. Tinetti, F. L. Romero, A. E. De Giusti "Evaluación de la Sincronización Periódica de Relojes de Computadoras para Instrumentación". XV Congreso Argentino de Ciencias de la Computación (XV CACIC) Univ. Nac. de Jujuy, San Salvador de Jujuy, Argentina, October 2009. ISBN 978-897-24068-4-1, pp.301-310 (2009).
9. F. G. Tinetti, F. L. Romero, "Herramientas para Instrumentación de Programas Paralelos en Ambientes Distribuidos", XII Congreso Argentino de Ciencias de la Computación, Univ. Nac. de San Luis, Potrero de los Funes, San Luis, Argentina, October 2006. ISBN 950-609-050-5. pp. 1414-1423.
10. Fernando L. Romero, Fernando G. Tinetti. "Sincronización de Relojes en Ambientes Distribuidos" IX Workshop de Investigadores en Ciencias de la Computación. WICC 2007. Facultad de Ingeniería - UNPSJB - Trelew - Chubut. May 3 & 4, 2007. ISBN 978-950-763-075-0. pp. 638-642. National Congress.



11. Work P., Nguyen K., “Measure Code Sections Using The Enhanced Timer”, <http://www.intel.com>, October 2005.
12. Zhao Y., Zhou W., Huang J, Yu S., “Self Adaptive Clock Synchronization for Computational Grid”, Journal of Computer Science and Technology, 2003 Volume: 18 Issue: 4 pp. 434-441.

# An analysis of network traffic characteristics for Botnet detection

MARIA JOSE ERQUIAGA<sup>1</sup>, CARLOS CATANIA<sup>1</sup>  
AND CARLOS GARCIA GARINO<sup>1,2</sup>

<sup>1</sup> Instituto para las Tecnologías de la Información y las Comunicaciones (ITIC)

<sup>2</sup> Facultad de Ingeniería Universidad Nacional de Cuyo, Mendoza, Argentina  
mariajoseerquiaga@gmail.com  
{ccatania,cgarcia}@itu.uncu.edu.ar

***Abstract.** The fast evolution of Botnet malware made it extremely difficult to detect. Despite it can be just considered as a tool, nowadays it has become one of the most dangerous threats for system administrators. Botnets are used as the starting point for different kind of attacks, such as SPAM, Denegation of Service, key logging and traffic sniffing, among others. In this paper we analyze some of the most relevant network traffic characteristics used for Botnet recognition. We have reviewed the most important works in the field of Botnet detection and have carried out an analysis in order to establish which are more appropriate to describe the Botnet behavior. Our final goal is to provide to network administrators the bases for building tools that can help them in their daily fight against this security threat.*

## 1. Introduction

Malware is a kind of software that damage computers and compromise the information inside them. The problem increases due to the accelerate development in the informatics field. With the important growth of malware along with the progress made in Artificial intelligence (AI) have resulted in the arising of Botnets.

A Botnet can be defined as net of bots, that is malware software installed on compromised computer with the ability of automatically propagating itself to new computers hosts. All the infected computers conform a Net of Bots with synchronized behavior. Botnet operation can be remotely controlled by a Botmaster to perform different malicious activities. Since a Botnet can be view as nothing more than a tool, there are several potential criminal uses for them.

Botnets have the capacity of continuously changing their behavior. This evolving capacity made them more difficult to detect. However, it is possible to recognize their presence in the network. For detecting Botnet activity on local networks it is important to know how Botnets work and which are the

common characteristics of these nets. By studying Botnet behavior, we might be able to detect and mitigate their effects.

In this work we intend to analyze different network traffic features capable of recognizing the presence of Botnets. We analyze the most relevant Botnet detection methods [4,6,5,11,1] in order to find out the relation between the network traffic features they use and their relation with Botnets behavior. The goal is to establish which are the attributes with the higher discriminate power. Network administrators may follow the behavior of features in the network, by using tools such as MRTG [8] or RRDTOOL [10]. Eventually, they can find some anomalies in the traffic, and it might indicate that there is an attack, or a bot in the local network. The contribution of our work is to analyze certain attributes, in order to describe the relation between them and Botnets behavior. In that sense, it will be helpful to identify anomalies in traffic that leads us to detect Botnets and possible network attacks.

This work is organized as follows: in section 2, we present a background, it defines Botnets and gives a short explanation of how do they work. Then, we introduce the Intrusion Detection Systems (IDS), follow by the concept of network traffic feature and their classification. In section 4, we present the Botnets detection features, according to the given classification. Finally, in section 5, we present a conclusion.

## 2. Background

### 2.1 Botnets

As the word suggests, Botnet, is a net composed of bots. The word bot comes from RoBot, bots are « smart » programs that can be automatically executed and can perform different actions according to the orders given by someone remote controlling them [3]. No human intervention is required by the bots to perform their tasks. These nets composed of bots, have the purpose of attacking and taking new hosts in order to conform the botnet. Making this process automatically, the botnet grows. Thus, a botnet, also known as zombie army, is a net of bots.

A Botnetmaster is the person who control Botnets trough remote control. They use their zombie army with different purposes, for example Distributed Denegation of Service (DDoS) attacks or spamming among others.

Furthermore, botnetmasters manage their Botnets using different control mechanisms. Those mechanisms include the protocols and the commands used by them to control his Botnet. There are two kinds of network architectures: centralized and decentralized ones. As shown in Fig. 1 (a), centralized nets usually have a central server where the bots are connected to. Decentralize Botnets have a server as well, but they differ in that they use peer-to-peer communication (P2P), this is shown in Fig. 1(b).

Moreover, there are different kinds of servers, the ones who use the IRC protocol, Fig. 2(a) and the others: web servers, that use the HTTP protocol Fig. 2(b).

## 2.2 Intrusion detection

An NIDS (Network Intrusion Detection System) is a software that detects unauthorized access to a network by sniffers and analyzing the network traffic. The goal of this kind of systems is to detect intrusions in a network. Hence, a NIDS analyses the activity of a certain network, looking for possible intrusions and threats.

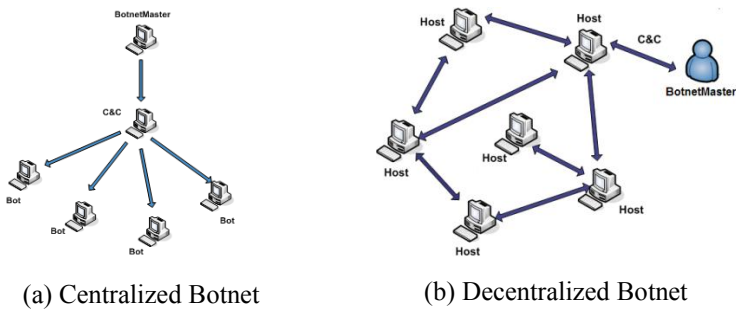


Fig. 1: Botnet Topologies

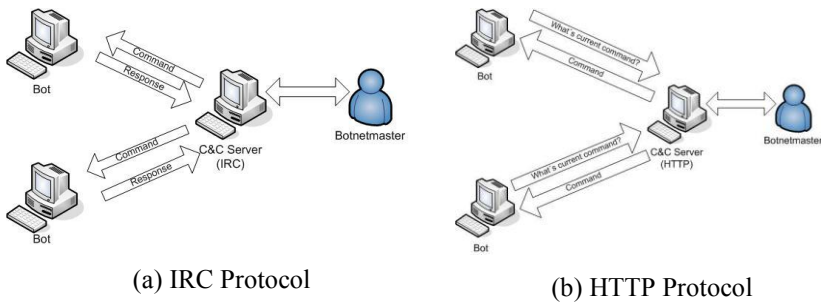


Fig. 2: Botnet underlying protocols

There are two main intrusion detection methods: anomaly-based and misuse based [2]. The first method defines a model of normal traffic and compares it with the network traffic to be analyzed. If there are any differences with the normal traffic, it is considered an anomaly and it might be a threat or a possible attack. The normal traffic model is built by following certain

parameters, for example, the number of connections, packet distribution according to protocol, etc. In opposition, the second method defines a model of intrusions, and waits for them to occur. These methods use different techniques. For example, anomaly based may use statistics, machine learning or data mining. On the other hand, misuse-based methods usually use pattern recognition, implication rules or data mining [2].

### 2.3 Features

The term attributes or features is usually related to data mining or machine learning process [12]. However, for the purpose of our study, features are defined as certain characteristics of a set of data that can be obtained from network traffic captures. The analysis of network traffic features are the bases for network-based IDS.

There are two attributes classifications, one related to the computational resources needed to be obtained and the other one to the network traffic source [2]. In the first classification, there are two cases: *low-level* features that can be acquired from raw traffic captures (as IP headers, or protocol) and *high-level* features that are the results of the traffic capture processing. For instance: Bytes per packet, packets per second, etc. These features might be obtained from low-level attributes.

The second classification mentioned, has three possibilities: packet, flow and payload features. These are obtained from packet headers, the information of network connections and packet payload (application layer) respectively. To clarify this classification, we present a diagram in Fig. 3.

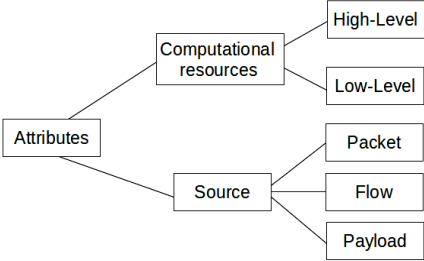


Fig. 3: Attributes classification

### 3. Network traffic features used in Botnet detection

In this section we present the features for Botnet detection, as suggested by the most relevant authors in the Botnet detection field [4,5,6,11,1] All these authors also implement different algorithms in order to detect Botnets.

However, our analysis is only focused on Botnets features. We will describe those attributes following the classification related to the network traffic source (packet, flow, payload). It should be noted that some features are included in more than one category, for instance an attribute might be packet and flow based.

### 3.4 Packet based features

Some of the features for Botnet detection are included in this category. In particular, a number of authors such as Gu [4], Karasadiris [5], Strayer [11] and Livadas [6] have worked with the Bytes-per-packet (Bpp) feature, and Strayer [11] mentioned that *Bpp* is the attribute that present the most discriminatory power. Moreover, Gu [4] utilized the features: average number of bytes per packets (Bpp) and number of packets-per\_flow (ppf). This last one, can be considered a flow feature as well, because it takes into account the information of the flow.

Binkley [1] proposed the metric TCP work weight, which is obtained by using the following formula:  $w = (S_s + F_s + R_s) / T_{sr}$ , where  $S_s$  are the SYN's plus SYNACKs sent,  $F_s$  the FIN's sent,  $R_s$  the RESETS and  $T_{sr}$  are the total number of TCP packets. This information is obtained from the TCP packets. The value of this metrics is expressed as a percentage. If this value is closer to 100% (percent), there are more possibilities for some kind of anomaly to occur. Another packet feature considered by [1] is IP source. This information is obtained from the packet header. Thus, it is consider a low-level attribute.

Furthermore, Karasadiris [5] considers the feature packets-per-flow (ppf), which is also a flow feature, and bytes-per-packet (Bpp). Likewise, this last feature is also suggested in [11].

In addition, Livadas informs that the features with most discriminatory power are: the percentage of packets that are pushed (PctPktsPushed), and the variance in the Bytes-per-packet (varBpp), besides Bpp.

### 3.5 Flow based features

The flow features suggested by Gu [4] are: the number of packets per flow (ppf), the number of flows per hour (fph) and the average number of bytes per second (bps). Karasadiris [5] besides using packets-per-flow (ppf) also uses flows-per-address (fpa). This feature is obtained making an association between a local IP address and their local port with several remote IP addresses and remote ports. Then, [11] expresses that the features with high discriminatory values are: duration, role (it means, who started the flow: the client or the server), average bits per second (bps), and average packets per second (pps).

### 3.6 Payload based features

Binkley [1], selected several payload features using two lists: Channel Name (is the name of the IRC channel), Joins (to the channel), Private Messages, Hits (the joins plus the private messages), number of IPs in the channel and the list of the IP numbers. All those features are related to a channel, that is why they are called *Channel List*. The other list is the *Node List* related to the metrics obtained from any IP address in any IRC channel. The flow features in this list are: Total Messages, Joins, Pings, Pongs, Private Messages, Channels and Server Hits (the number of messages sent to/from a host).

### 3.7 Analysis

In Table 1, we summarize the information discussed in previous sections. The table shows information regarding the authors, the features utilized by them, the kind of attributes, and the detection method selected by each.

**Table 1.** Botnet Detection Feature analysis

Author	Features	Type of feature	Detection technique
Guofrei Gu [4]	number of flows per hour (fph)	high level, flow	Anomaly-based and misuse-based
	number of packets per flow (ppf)	high level, flow, packet	
	average number of bytes per packets (bpp)	high level, packet	
	average number of bytes per second (bps)	high level, flow	
Karasaridis [5]	flows-per-address (fpa)	high level, flow	Anomaly-based
	packets-per-flow (ppf)	high level, flow, packet	
	bytes-per-packet (bpp)	high level, packet	
Strayer [11], Livadas[6]	duration	high level, flow	Misused-based
	role	high level, flow	
	average bytes per packet (Bpp)	high level, packet	
	average bits per second (bps)	high level, flow	
	average packets per second (pps)	high level, packet	
	% of packets that are pushed (PctPktsPushed)	high level, packet	
	variance in the Bytes-per-packet (varBpp)	high level, packet	
Binkley[1]	TCP work weight	high level, packet	
	Chanel Name (CHANNAME)	high level, payload	
	joins	high level, payload	
	hits	high level, payload	

	private messages (PRIVMSG)	high level, payload	Anomaly-based
	number of IPs in the channel (NOIPS)	high level, payload	
	list of the IPs (IP_LIST)	high level, payload	
	IP source (IPSCR)	high level, payload	
	total messages (TOTALMSG)	high level, payload	
	Pings	high level, payload	
	pongs	high level, payload	
	private messages (PRIVMSG)	high level, payload	
	channels	high level, payload	
	server hits	high level, payload	

In order to understand the features presented by each author, it is important to know how do Botnets works. For example, we can say that Botnet behavior is synchronized [4,5]. In that sense, we present the features and analyze their relation with Botnet operation.

The feature Bpp, is selected for several authors for Botnet detection [4,6,5,11]. In the case of Botnets, the value of Bpp is low because the messages between the bots and the Botnetmaster are short commands. So, the size of the packets are approximately 1KB [7]. Furthermore, the duration of this communication is short, so it is another feature to be considered.

Moreover, the number of flows per hour (fph) suggested by Gu et al. [4] has a relation with Botnet behavior too. When the Botnetmaster send commands to their bots, he do it at the same time, so suddenly a lot of flows may appear. And the number of flows-per-address (fpa) will be high as well.

In the case of Botnets, the attribute packets-per-flow (ppf), does not reach a high value because there are a few packets that the botmaster sends to the bots with instructions. Bytes-per-second reaches a low number too as the messages sent are short. Regarding the role feature, considered by Strayer [11], the importance is on who initiates the communication, which is usually the server (in this case, the Botnetmaster).

Blinkley's [1] analysis focuses on \_nding IRC-based Botnets, by using two list: the channel list and the node list. The \_rst one, is composed by the following features:  $\{CHANNNAME, HITS, JOINS, PRIVMSGS, NOIPS, IP\_LIST\}$ , which have already been described. The node list, mentioned in subsection 3.3, contains the features:  $\{IPSCR, TOTALMSG, JOINS, PINGS, PONGS, PRIVMSGS, SERVERHITS\}$ . To identify Botnets, the author proposed to find certain values or characteristics in these features. One of them, is to identify the evil channels that must be the ones with a lot of hosts with a high value of TCP work weight. This attribute, TCP Work Weight, is an indicator of a possible scanning or the presence of a worm.

As we have seen, the features selected have a relation with Botnet behavior. However, our goal is to find the most useful features for Botnet detection. These ones are not related to a particular kind of Botnet. For instance: IRC based, HTTP-based, centralized, etc. In that sense, the features selected by



[1] would not be the best ones, because it intent to detect only IRC-based Botnets. Therefore, an important feature to consider is Bpp, because the communication between the bots and the Botnet master always include short messages, while data transfer packets in normal traffic are longer. Another attribute to highlight is the number of flows per hour (fph), as we have indicate, this feature have also a relation with Botnets operation. In general the messages are sent to all the bots simultaneously. Since Botnets are larger, commands are given to the entire net, and not to individuals bots [11]. Since communication between the Botnetmaster and their bots have a small interval of time, we consider that duration is another helpful attribute to detect Botnets as well.

#### **4. Concluding remarks and future work**

Today, Botnets are one of the most important threats in network security. They are originated by the combination of malware and IA. Since they are constantly evolving, it is really difficult to detect and mitigate them. In that sense, it is crucial to understand the behavior of Botnets, in order to find the network traffic features that can potentially help system administrators to detect them.

Due to Botnets behavior changes, it is difficult to detect them, however in the past years, several complex Botnet detection methods have been proposed. These methods were based on machine learning, pattern recognition and clustering among others. Still, we believe a much simpler approach is possible. Given some traffic network features with high discrimination power, it is possible to utilize certain network analyzer tools as MRTG[8] or RRDTTools [10] in order to help system administrator in the Botnet detection process. Moreover, it is also possible to use such features for writing a proper signature rules for using with a NIDS such as Snort [9]. Notice that we are aware this approach will be not enough for fully Botnets detection, but we think it can potentially provide good enough information for helping system administrators in their daily fight against intruders.

Therefore, regardless of the detection method utilized by the different authors, our goal with this work was to understand how the attributes are related with Botnet behavior. We analyzed network traffic attributes under two major aspects. First, from the point of view of computational resources requirements. Second, how the attributes are related with Botnet behavior

From our analysis came out that, since the most interesting attributes are the ones related to Botnets behavior. On the other hand, attributes such as the protocol used, or the connection ports are useless. Moreover, we consider that, the attributes with most discriminatory power can be, in the first place Bpp, because the kind of messages send by the bots and the Botnetmaster are particularly short. In second place we consider flows per hour (fph) as another network traffic feature with high potential for Botnet detection.

Finally, connection duration, is also an important network traffic characteristic for recognizing Botnet behavior. These network traffic features provide a good trade off between their discriminative power and the computational resources they need. Therefore, we believe system administrators can exploit the benefits of these network traffic features in his daily battle against Botnet propagation.

In a future work, we intent to evaluate the performance of the features selected in this survey in order to find the optimal performance in Botnet detection.

## Acknowledgments

The authors would like to thank the financial support received by the projects PAE-PICT 2312 granted by ANPCyT and 06/M036 granted by SECTYP, UNCuyo.

## References

1. J. R. Binkley and S. Singh. An algorithm for anomaly-based botnet detection. In *Proceedings of the 2nd conference on Steps to Reducing Unwanted Traffic on the Internet - Volume 2*, SRUTI'06, pages 7-7, Berkeley, CA, USA, 2006. USENIX Association.
2. C. A. Catania and C. García Garino. Automatic network intrusion detection: Current techniques and open issues. *Computers and Electrical Engineering*, 2012. Accepted. In Press. DOI:10.1016/j.compeleceng.2012.05.013.
3. M. J. Erquiaga. Botnets: Mecanismos de control y de propagacion. In *CACIC 2011. XVII Congreso argentino de ciencias de la computacion*, pages 1076-1085, 2011.
4. G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer: clustering analysis of network traffic for protocol - and structure-independent botnet detection. In *Proceedings of the 17th conference on Security symposium*, SS'08, pages 139-154, Berkeley, CA, USA, 2008. USENIX Association.
5. A. Karasaridis, B. Rexroad, and B Hoeflin. Wide-scale botnet detection and characterization. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, HotBots'07, pages 7\_7, Berkeley, CA, USA, 2007. USENIX Association.
6. C. Livadas, R. Walsh, D. Lapsley, and W.T. Strayer. Usilng machinelearning techniques to identify botnet tra\_c. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 967-974, nov. 2006.
7. M.M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K.W. Hamlen. Flowbased identifification of botnet traffic by mining multiple

- log files. In *Distributed Framework and Applications, 2008. DFMA 2008. First International Conference on*, pages 200-206, oct. 2008.
8. T. Oetiker. MRTG - the multi router tra\_c grapher. In *Proceedings of the 12th USENIX conference on System administration, LISA '98*, pages 141-148, Berkeley, CA, USA, 1998. USENIX Association.
  9. M. Roesch. SNORT-lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX conference on System administration, LISA '99*, pages 229-238, Berkeley, CA, USA, 1999. USENIX Association.
  10. J. Sellens. RRDTool: Logging and graphing. In *USENIX Annual Technical Conference, General Track*. USENIX, 2006.
  11. W.T. Strayer, R. Walsh, C. Livadas, and D. Lapsley. Detecting botnetswith tight command and control. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 195-202, nov. 2006.
  12. I. Witten and E. Frank. *Data Mining, Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2nd edition, 2005.

# Using Facebook as a Support Tool in an Undergraduate Computer Science Course

VICTORIA MARTÍNEZ DE LA CRUZ AND CARLOS IVÁN CHESÑEVAR

Department of Computer Sciences and Engineering  
Universidad Nacional del Sur  
Avenida Alem 1253, Bahía Blanca, Buenos Aires, Argentina. CP 8000

victoria@vmartinezdelacruz.com  
cic@cs.uns.edu.ar

***Abstract.** During the last years social networks have become a cultural phenomenon bringing to a new dimension the potential provided by ICTs. The massive usage of this networks as an interaction tool between peers, particularly by the age group in which are located the current college students, brings a new and powerful tool for the college professor. This article addresses the usage of a social network, specifically Facebook, in the context of classroom education in a college course. The hypothesis underlying our research is based on the idea that social networks enhance and empowers various aspects in education, such as motivational, recreational and educational. This belief was confirmed by an online survey carried out with students along two different years.*

***Keywords:** ICTs, Social Networks, College teaching.*

## 1. Introduction and motivations

During the last years social networks have become a cultural phenomenon bringing to a new dimension the potential provided by ICTs. The massive usage of this networks as an interaction tool between peers, particularly by the age group in which are located the current college students, brings a new and powerful tool for the college professor. This vision of social interactions has created a new paradigm for information dissemination [1] as well as novel types of communication between people, companies, institutions, organizations and interest groups. Most of the digital press, to mention one example, has incorporated readers as part of the process of opinions formation, providing different interaction mechanisms through networks like Twitter and Facebook.

In spite of this remarkable increase of impact in business and everyday life, social networks remain relatively far from institutional use in the process of classroom education at the college [7]. Even though social networks are commonly used as one-way communication mechanism for universities to broadcast news, announcements and institutional and academic information,

its use as a pedagogical tool within the classroom is atypical, using instead other technologies such as static web pages. This is of particular interest to Argentina since it is estimated that it will be the third country in the world with the highest penetration of Facebook in 2014, staying ahead of the U.S. and Canada.<sup>1</sup>

This article presents an analysis of the use of Facebook as technological support in the context of classroom education in an undergraduate college course. Our hypothesis is based in the fact that social networks enhance and improve motivational, recreational, didactic and relationship aspects in the classroom space. We assessed this hypothesis on the basis of a survey carried out among different students in two courses.

The remainder of the paper is organized as follows. First, in Section 2 we will discuss the features of the proposed use of Facebook, stressing some of its advantages. Then, in Section 3, we will present a survey conducted on undergraduate students of the *Licenciatura en Ciencias de la Computación* (Licentiate in Computer Sciences) and *Ingeniería en Computación* (Computer Systems Engineering) degrees at the Universidad Nacional del Sur. We will discuss the obtained results and their relationship with the hypothesis previously mentioned. In Section 4 we will explore different papers related to our research. Finally, in Section 5, we will present the main conclusions obtained as well as some future research lines.

## 2. Use of Facebook in an undergraduate course: Approach

The creation of a Facebook group for a course is an extremely simple process and allows a quick integration on an interactive virtual platform of students and professors that share the classroom space during the course. The professor, teaching assistant or another person designated by the course staff, simply has to create a new group linked with the course defining it as a “closed group”. The latter is a security measure requiring that the participation is constrained to those members who are accepted by the coordinator. The “lifetime” of the Facebook group can exceed the length of the course and remain available for subsequent use of the subscribed members.

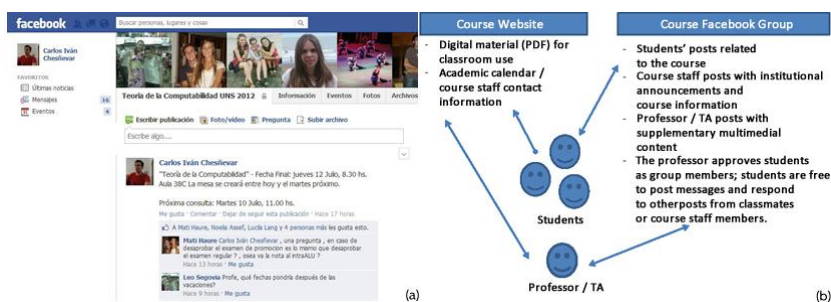
The functionality of that Facebook group is to complement in a dynamic way the information provided by the course website<sup>2</sup>, which has a read-only access for the students, that is, it's only possible to check news and download the material used in classes. In contrast, the Facebook group aims to provide

---

<sup>1</sup> Emarketer.com (2012) *Argentina Ranks Third Worldwide in Facebook Penetration*. [online] Available at <http://www.emarketer.com/Article.aspx?R=1008972> [Accessed July 9, 2012]

<sup>2</sup> Teoría de la Computabilidad (2012) Departamento de Ciencias e Ingeniería de la Computación. [online] Available at: <http://cs.uns.edu.ar/~drg/tc/> [Accessed July 10, 2012]

a simple interactive environment in which both the course staff members and the students can ask questions, establish conversations and publish multimedia content. To contextualize this we attach as an example screenshots of the Facebook group of Teoría de la Computabilidad, year 2012. Figure 1(a) shows the general appearance of the Facebook page; Figure 1(b) summarizes in a diagram the various facilities provided by the both channels in relation to the interaction in the classroom. The bidirectional arrows indicate that the student can enter and access to the content; the unidirectional arrows denote that the information is read only.



**Fig. 1.** (a) Appearance of the Facebook page for the course “Teoría de la Computabilidad” (Departamento de Cs. e Ing. de la Computación, Universidad Nacional del Sur, 2012); (b) Outline of the interaction facilities provided to the coordinator and the students through the website and the Facebook page of the course.

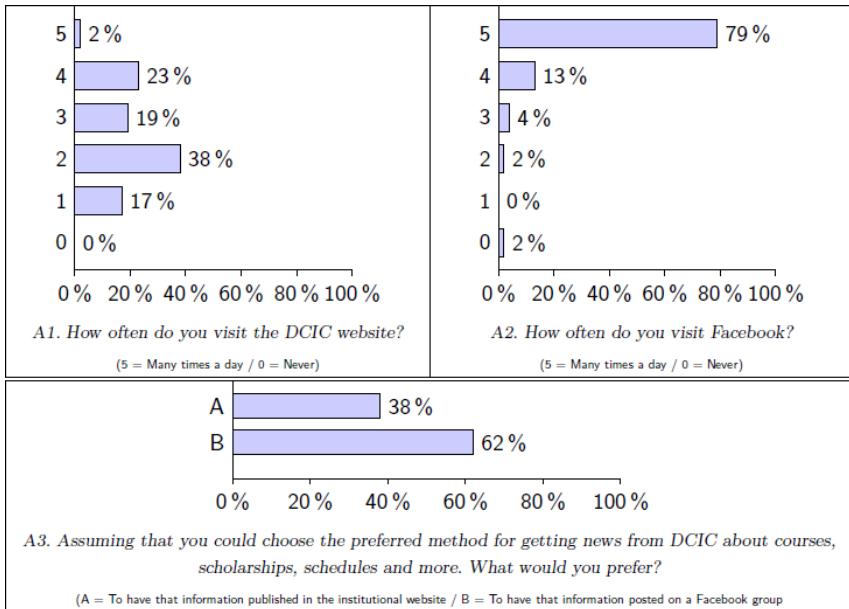
On the one hand, students not only used the group to read the news or comments issued by the course staff, but they turned it into a space with their own identity in which they generated discussions, personal thoughts and disseminated multimedia content, thus establishing a communicational bond that transcended the purely academic one. This conclusion was reinforced by perceiving that in moments before midterms several students externalized and shared their moods, feelings, etc. In this regard, it should be noted that in the group there was no preestablished regulation from the course staff, but there was an implicit balance of respect and virtual coexistence similar to the one in the regular classroom. From the course staff’s point of view it was possible to appreciate that using a Facebook group turned out to be a positive experience, a conclusion also drawn from the comments provided by the students. This motivated the interest of assessing in a more formal way the impact of the use of Facebook in the course, performing a survey discussed in the next section.

### 3. Study case: Survey and results

To conduct a survey of the impact of Facebook in relation to the course “*Teoría de la Computabilidad*” of the Departamento de Ciencias e Ingeniería de la Computación (DCIC) of the Universidad Nacional del Sur (UNS), a poll among students who took that course in the year 2012 was carried out. The course of the mentioned year was composed of 105 students, of which 87 participated in the Facebook group. At the end of the course 63 students passed the course, 15 did not pass the course and 26 were absent. All the students who were part of the Facebook group regularly attended the on-site classroom lectures.

The surveys were divided according their scope, more specifically on the use of Facebook in relation to institutional information (A1-A3) and the use of Facebook in relation to the course (B1-B7). Both surveys had a similar impact: for the survey A received 50 responses and the survey B received 49 responses, a 57% and a 56% of the students with Facebook respectively. The obtained results are shown in Figures 2, 3 and 4. In several questions it was possible to establish a degree of satisfaction from 0 to 4, with 0 being the denial or disagreement and 4 the acceptance or agreement. Values in the range of 3 to 4 are considered to be positive results and the ones from 0 to 2 are taken as negative results. The survey B also allowed respondents to make additional comments related to their experience of using Facebook in the course staff.

It can be noted at first glance that results were encouraging since in most questions students expressed unanimously the interest and the satisfaction of involving Facebook in their academic lives. The first survey (figure 2) analyzed the students look on institutional communication from the college. Questions A1 and A2 measure the frequency with which students visit the official website of DCIC in contrast with the frequency of visits to Facebook. A1 shows that only the 25% of the students visited the website of the department regularly, while in A2 we see that 92% confirms using Facebook more intensely. To complement A1 and A2 the question A3 was formulated, in which it was established a preference of students for the usage of Facebook, with a 62% of positive feedback. These values allowed us to confirm a strong interest in the use of Facebook as a channel for the institutional information, over static websites traditionally used as an information portal.

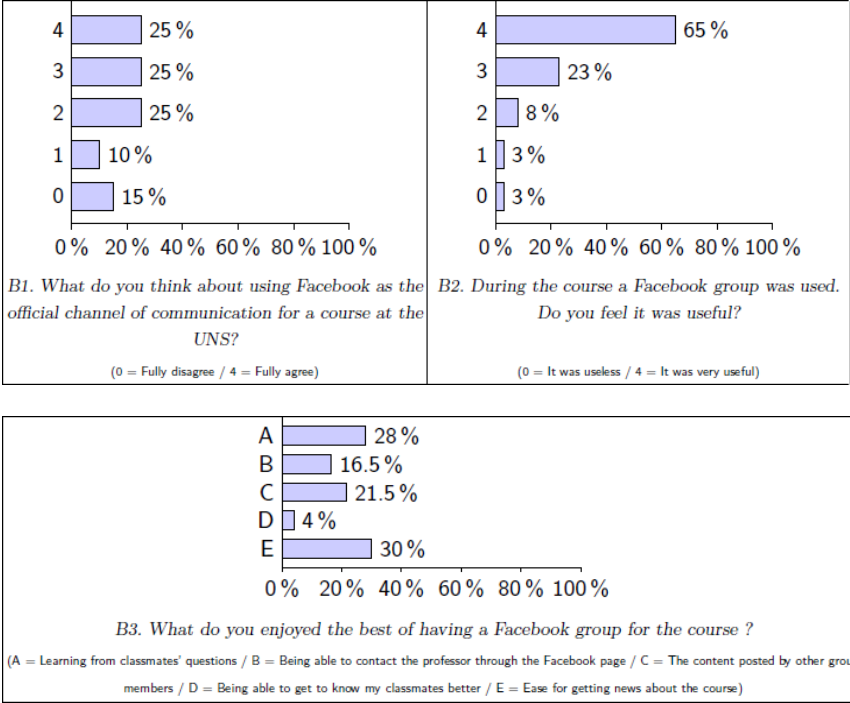


**Fig. 2.** Facebook in relation to the institution / Survey to students of “Teoría de la Computabilidad”, year 2012 (Universe of respondents: 50 students; 57% of students that attended the course and were members of the course Facebook group.)

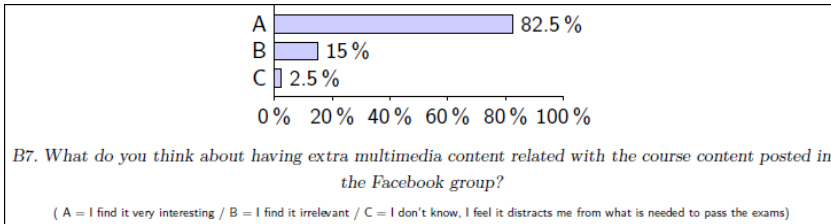
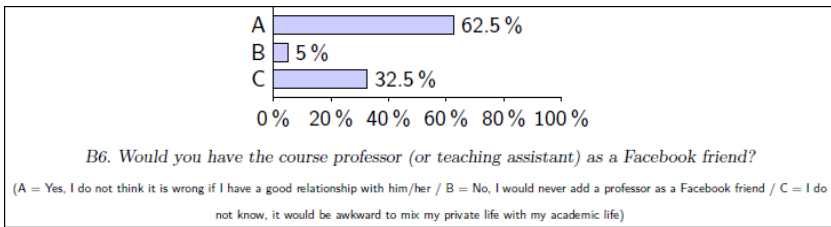
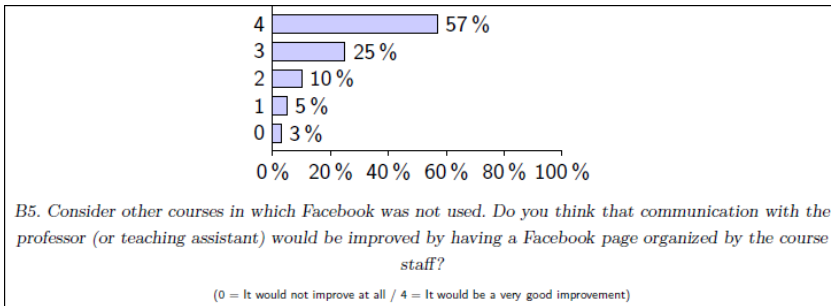
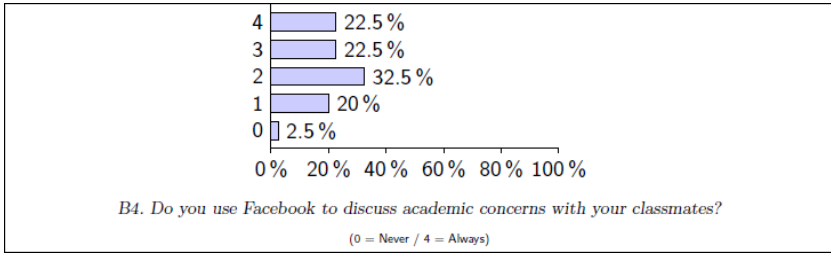
Subsequently we analyzed the use of Facebook in relation to the course. In this sense, there were mixed views since 50% of students agreed with establishing the use of this social network as a way of communication from the course staff (Fig. 3, question B1), but the other 50% expressed not to be convinced with the idea. This position is supported by a fact suggested by some students in the comments section (the required usage of a commercial platform such as Facebook as a access channel to the information). Beyond this, 88% of the respondents rated as positive or very positive the usefulness of the Facebook group (Fig. 3, questions B2 and B3), providing greater value to two aspects: the ease of getting news and comments from the course staff (30%) and the apprenticeship from fellow students questions (29%). Some of the students surveyed mentioned that the latter was reinforced by the access to Facebook through smartphones, whose usage is getting increasingly more widespread. To help to complete the vision of integration of a Facebook group to the courses, students were also asked about other courses the same students took in which there was no Facebook space (Fig. 4, question B5). 82% of the students surveyed stated that the communication with the course staff of those courses would be better if there were a dedicated Facebook group. Furthermore, we also determined that 45% of the students use Facebook to discuss academic matters (Fig. 4, question B4). Having a group linked to the course staff enable students to use that space to share their concerns and ideas with the benefit of having the course staff and their



colleagues to make a contribution. In addition, 82.5% was interested with the possibility of having additional multimedia content related with the subject (Fig. 4, question B7), so that the universe of shared information in the same space is not limited to what is known by the course members nor the contents taught in the on-site class. In the particular case of “*Teoría de la Computabilidad*”, additional material was included, such as pictures, videos, links to academic and career information as well as useful web applications. A noticeable result of the survey was that 62.5% of the students reported agreement on including their professors as “friends” on Facebook. This aspect can be controversial and it was not encouraged from the course staff (the membership to a group does not determine the students being “friends” with their professors), but highlights how students perceive their personal space in relation to the institution, and to what extent they are willing to share their private life with people occupying a different hierarchical role. Clearly this subject deserves a more detailed analysis that is beyond the scope of this paper.



**Fig. 3.** Facebook in relation to the course (part 1 of 2) / Survey to students of “*Teoría de la Computabilidad*”, year 2012 (Universe of respondents: 49 students; 56% of the students who attended the course and were member of the course Facebook group).



**Fig. 4.** Facebook in relation to the course (part 2 of 2) / Survey to students of “Teoría de la Computabilidad”, year 2012 (Universe of respondents: 49 students; 56% of the students who attended the course and were member of the course Facebook group).

#### 4. Discussion and related work

There are several researches related to Facebook and the academia, mostly recently published. At the beginning the interest was focused in analyzing if it was possible that the work in education could be enhanced through the use of Facebook. In [11] significant results were determined that supported this

idea and led to its application. Based on this, we analyzed as well different case studies and approaches, more precisely [2] and [6], in conjunction with the approach discussed in this paper. In [4] the authors analyzed and contrasted degree students who used Facebook with others who did not, finding that in many circumstances the academic performance of the former was lower than the latter. Similar conclusions are also discussed in [3]. In contrast, in a recent doctoral research [10] five complementary and independent studies were conducted to measure the impact of Facebook in degree students of different cultures. The acquired results showed that the presence and cognitive absorption of Facebook does not threaten directly the students' academic performance. In [9] an interesting analysis was performed about the underlying motivations for using Facebook. Coincidentally with the obtained results of our surveys, the communication motivation was a critical element for the choice of this social network, rather than factors related to the users' personality. As mentioned before, the survey enabled the students to add their personal comments about the experience. In this space many respondents noted the importance of the compatibility of Facebook with mobile interfaces. Being able to access Facebook through their smartphones implies that communications can be done instantly and permanently between professors and students, the ideal setting for reporting situations or last minute changes, as the situation in which a lecture is cancelled or exam schedule changes. They also pointed out that they consider an undesirable aspect the usage of a commercial platform, in this case Facebook, which somehow compels students to be part of it in order to communicate with the course staffs.

## **5. Conclusions and future work**

In this work we examined and discussed the implications of the use of a Facebook group for a second year degree course, in the context of careers related to Computer Sciences. It has been performed a survey to evaluate various relevant elements, finding many interesting conclusions that reflect the positive aspects of having a Facebook group associated with the course staff. In our analysis we can find the following significant aspects:

1. The daily use of Facebook among students as a tool to access information, perceiving the mentioned social network as very useful in relation to a course (questions A2 and B2);
2. The emphasis placed on Facebook in the dynamics of learning and the appreciation given to the complementary multimedia information (questions B3, B4 and B7);

3. The important assessment of the professor-student bond through a social network, with a unique additional component which results in the student's disposal to "be friends" with the professor through Facebook (questions B5 and B6).

This last aspect is definitely debatable, but it highlights the trend imposed by social networks in the students' thinking, blurring the hierarchies and strengthening horizontal bonds<sup>3</sup>.

It should also be mentioned that for the professor responsible for the course staff, the Facebook group provided a powerful tool to assess different communicational aspects of the course students, e.g. questions formulation, solution seeking to problems in practical works and participation through media material contributions. In this regard the Facebook page complemented the professor's knowledge over the students, helping him to determine different aspects of their personalities and behavior, such as their degree of extroversion and their clarity to formulate ideas, and their relationship with the teaching techniques used to address various topics in class. It is also important to mention that the virtual participation of the students was much more frequent and with greater interactivity than the classroom participation. As a final conclusion it can be stated that Facebook provides an important communication platform with a high penetration into society and particularly prevalent in the age group for college students. The experience of its use as a complementary tool has been highly successful, as corroborated from the survey and the collected data. As a future work, we expect to make an analysis of new variables in the evaluation of the Facebook application, e.g. correlation between the success rate in the courses and the frequency of access to Facebook, similar that the one conducted in [4]. Another dimension to explore is the inference of social ties between students, i.e. who are those who post messages and who are the ones who respond to others' messages. These latter aspects have been investigated recently in [8], and would allow us to identify sociograms associated with group discussions. Research in this direction has already been done with online communities, e.g. in [12].

---

<sup>3</sup> The "friends" relationship in Facebook between teachers and students can be controversial and some reports have recently emerged on this direction (e.g. Wakefield, K. (2012) *Should Professors Use Facebook to Communicate with Students?* - Faculty Focus. [online] Available at: <http://www.facultyfocus.com/articles/trends-in-higher-education/should-professors-use-facebook-to-communicate-with-students/> [Accessed July 9, 2012]). Some additional considerations about the institutional role in this context may be further explored in specific Facebook sites for educators (e.g. Facebookfor educators.org (2011) *Facebook for educators*. [online] Available at: <http://facebookforeducators.org/> [Accessed July 9, 2012]).

**Acknowledgments** This paper would not have been possible without the collaboration of the students of Universidad Nacional del Sur who attended the course “Teoría de la Computabilidad” in 2012 and provided the data associated to the survey discussed in this paper.

## References

1. Bakshy, E., Rosenn, I., Marlow, C., Adamic, L.A.: The role of social networks in information diffusion. In: Mille, A., Gandon, F.L., Misselis, J., Rabinovich, M., Staab, S. (eds.) WWW. pp. 519-528. ACM (2012).
2. Deed, C., Edwards, A.: Using social networks in learning and teaching in higher education: An Australian case study. *IJKSR* 1(2), 1-12 (2010).
3. Junco, R.: Too much face and not enough books: The relationship between multiple indices of facebook use and academic performance. *Computers in Human Behavior* 28(1), 187-198 (2012).
4. Kirschner, P.A., Karpinski, A.C.: Facebook and academic performance. *Computers in Human Behavior* 26(6), 1237-1245 (2010), <http://www.sciencedirect.com/science/article/pii/S0747563210000646>
5. Moore, K., McElroy, J.C.: The influence of personality on facebook usage, wall postings, and regret. *Computers in Human Behavior* 28(1), 267- 274 (2012), <http://www.sciencedirect.com/science/article/pii/S0747563211002020>
6. Ractham, P., Firpo, D.: Using social networking technology to enhance learning in higher education: A case study using facebook. In: HICSS. pp. 1-10. IEEE Computer Society (2011).
7. Roderá, A.: Profesores 2.0 en la Universidad del Siglo XXI: Criterios para la Integración Educativa de la Web Social en la Universidad. Ph.D. thesis, Universitat Oberta de Catalunya, Barcelona, España (2012).
8. Rodríguez, S.S., Redondo, R.P.D., Vilas, A.F., Arias, J.J.P.: Using facebook activity to infer social ties. In: Leymann, F., Ivanov, I., van Sinderen, M., Shan, T. (eds.) CLOSER. pp. 325-333. SciTePress (2012).
9. Ross, C., Orr, E.S., Sisic, M., Arseneault, J.M., Simmering, M.G., Orr, R.R.: Personality and motivations associated with facebook use. *Computers in Human Behavior* 25(2), 578-586 (2009), <http://www.sciencedirect.com/science/article/pii/S0747563208002355>
10. Rouis, S.: Impacts of Facebook hedonic usage on students’ academic achievement: the role of culture and personality characteristics. Ph.D. thesis, Luleå University of Technology, Sweden, [http://pure.ltu.se/portal/files/36613764/Thesis\\_Sana\\_Rouis\\_April\\_27\\_2012.pdf](http://pure.ltu.se/portal/files/36613764/Thesis_Sana_Rouis_April_27_2012.pdf) (2012).
11. Yu, A.Y., Tian, S.W., Vogel, D.R., Kwok, R.C.W.: Can learning be virtually boosted? an investigation of online social networking impacts. *Computers & Education* 55(4), 1494-1503 (2010).
12. Zhao, J.: A sociogram analysis on group interaction in an online discussion forum. In: Li, F.W.B., Zhao, J., Shih, T.K., Lau, R.W.H., Li, Q., McLeod, D. (eds.) ICWL. Lecture Notes in Computer Science, vol. 5145, pp. 377-389. Springer (2008).

# A multiplatform interpreter to introduce structured and concurrent programming

BEATRIZ DEPETRIS<sup>1</sup>, GUILLERMO FEIERHERD<sup>1</sup>,  
DANIEL AGUIL MALLEA<sup>1</sup>, GERMÁN TEJERO<sup>1</sup>

<sup>1</sup> Universidad Nacional de Tierra del Fuego, Instituto de Desarrollo Económico e Innovación,  
Hipólito Irigoyen 880, 9410 Ushuaia, Tierra del Fuego, Argentina

{bdepetris, gfeierherd, daguilmallea, gtejero}@untdf.edu.ar

***Abstract.** The process of teaching and learning computer programming has always been a challenge for students and teachers. Throughout time, the challenge has become tougher because now the concepts related to concurrent programming must be added to the traditional concepts of programming. The tools that show the performance of algorithms have been of great help, although they must be used carefully. This article describes the development and use of an update of the Visual Da Vinci (an environment used in different institutions to introduce computer programming) and discusses the reasons why programming (and software design in general) usually arises issues. It also justifies the current importance of concurrent programming and the difficulties it adds to traditional programming. Further on in the text, the Concurrent Da Vinci is analysed and commented on, showing an example of how it is used to solve a classical problem of concurrency.*

***Keywords:** teaching computer programming, teaching concurrent programming, algorithm visualization.*

## 1. Introduction

Having a flair for managing abstract ideas is a distinctive characteristic of good computing professionals. It is an ability made up of two complementary elements: the ability to simplify, removing unnecessary details, and the ability to derive generalizations that highlight the common and essential aspects of a group of specific cases.

This ability is necessary because software is essentially abstract, therefore its design and development have to do mainly with abstractions. As Devlin states, “Once you realize that computing is all about constructing, manipulating, and reasoning about abstractions, it becomes clear that an

*important prerequisite for writing (good) computer programs is the ability to handle abstractions in a precise manner.” [1]*

Along the same line, when Kramer wonders “*why is it that some software engineers and computer scientists are able to produce clear, elegant designs and programs, while others cannot?*”, he reaches the conclusion that “*the key lies in abstraction: The ability to perform abstract thinking and to exhibit abstraction skills.*” [2]

According to Piaget, humans develop the abilities related to abstraction at the fourth (and last) stage of cognitive development: the formal operational stage, which starts approximately at 12 years old. Nevertheless, reality shows that many university students (at least among the ones that chose courses related to informatics) have a low development of these abilities.

Piaget’s theory backs up the assertion of Rutherford and Ahlgren, quoted by Dann and Cooper in their article about Alice: “*students’ learning progression is usually from the concrete to the abstract. Young people can learn most readily about things that are tangible and directly accessible to their senses—visual, auditory, tactile, and kinesthetic. With experience, they grow in their ability to understand abstract concepts, manipulate symbols, reason logically, and generalize. These skills develop slowly, however, and the dependence of most people on concrete examples of new ideas persists throughout life. Concrete experiences are most effective in learning when they occur in the context of some relevant conceptual structure.*” [3]

## **1.1 Teaching Computer Programming**

Given that humans understand abstract ideas basing on concrete ideas, tools aimed for the introduction of young people to programming have been created and used for some time. The common characteristic of these tools is the visualization of the algorithms’ execution. These tools are usually used in introductory courses to computer programming of the university courses that deal with informatics (CS1 and CS2). Nevertheless, because of the relevance computing thought has gained lately, in some cases the tools are used with under-age students, or even with students that are not planning to venture into informatic courses. While some of these tools have specific purposes (for example, visualizing the action on trees), others, even if they can be of use for people with different ages and interests, are more general (like Alice, Greenfoot and Scratch). [4]

At the ex Ushuaia headquarters of the Universidad Nacional de la Patagonia San Juan Bosco (UNPSJB), currently part of the newly created Universidad Nacional de Tierra del Fuego, we have been using Visual DaVinci (VDV) for several years. The environment VDV is a tool designed and developed in the Instituto de Investigación en Informática LIDI of the Universidad Nacional de La Plata. This institution started to use it, and it was later implemented in other National Universities. [5]

One of the main characteristics of the Visual DaVinci language is that it can receive instructions in Spanish. In order to be able to visualize the execution, it uses a city made compose of streets (horizontal) and avenues (vertical), a robot (that can move through the city) and objects (flowers, papers and obstacles), located in the intersection of streets and avenues, with which the robot can interact.

Before executing a program, the location of flowers, papers and obstacles can be determined. During the execution, the robot (Lubo-I), which has a bag to take flowers and papers, moves along the city obeying a group of primitives (*mover*, *derecha*, *tomarFlor*, *depositarPapel*, etc.). Besides, there is access to certain variables of the system (*HayFlorEnLaEsquina*, *HayPapelEnLaBolsa*, etc.). [6]

Compared to any other visual programming language available in the market, Visual DaVinci is a limited tool, but it is precisely this characteristic which helps avoiding the distraction that a great number of possibilities causes, and diminishing the time it takes to learn how to use the environment. Thanks to this students can focus on learning to design algorithms and express them in a short period of time. Obviously, these limitations restrict both the possible data and the algorithms that can be developed and visualized.

Finally it should be noted that Visual DaVinci uses a syntax similar to Pascal, which facilitates the task in that courses of programming and algorithmic that use it as a language.

## **1.2 Previous Experiences at the Ushuaia Headquarters**

VDV has been used at the Ushuaia Headquarters since 1999. At first, it was used in an optional course of Expression of Problems and Algorithms that used to take place simultaneously with the Maths Leveling Course, which is required to start any course at the Faculty of Engineering of the UNPSJB.

The optional course had two basic aims. First, to work as a vocational guidance for the applicants to the course. Second, to start working with some contents of Algorithms and Programming (subjects of the second semester of the first year), allowing the concepts to sink in during the first semester.

The results of the experiences were most satisfactory: the failure level in Algorithms and Programming dropped among the students that had taken the optional course [7]. As a consequence, in the new syllabus implemented since 2010, the subject Expression of Problems and Algorithms was added to the first semester of the first year. At first, the subject works with VDV, and then starts with Pascal.

## **1.3 Concurrent Programming**

Concurrency is a characteristic of some problems which allows them to be transformed into less serious sub-problems that, to a certain extent, can be



solved simultaneously. However, given that these solutions are partial collaborations to the general solution, they are not completely independent. Depending on the problem, they have to *share resources* and *synchronize*. When this technique is applied to problems that must be solved by computers, it is called concurrent programming.

It is important to point out that concurrent programming does not mean *real simultaneity*. In fact, if there is one processor only, the simultaneous execution is only apparent. Besides, when the execution of the programs that solve partial solutions can take place really simultaneously, using a group of machines (distributed system), a machine with many processors (multicore), or some combination of both options, we prefer to call it parallel programming.

The construction of operating systems was a field where concurrency techniques sparked great interest and experienced significant improvements. As a consequence, in many universities the introduction to concurrent programming is usually a part of the subject that deals with the different aspects of Operating Systems. Afterwards, in many cases, both concurrent and parallel programming, receive little further treatment.

However, in recent times, hardware developments that took the ability of multiprocessing to cheaper devices generated an added interest in this way of solving problems. In the case of our syllabus, this has been reflected in some changes made to the syllabus, implemented since 2010. Some of the changes are the transformation of the concurrency topics of the subject Operating Systems into a new subject (Introduction to Concurrency) and the creation of curricular spaces specifically devoted to the issue (Parallel Systems-S).

To the already mentioned issues of traditional programming, concurrent programming adds new ones. Some of them are the need to establish a communication between the processes that contribute to the solution, the control of the access to shared resources and synchronization. However, we think that the greatest difficulty comes up when checking the processes' correctness, as a consequence of *nondeterminism*. As it is generally known, this characteristic causes that, starting from the same initial conditions (input data) the consecutive executions of the processes vary. This brings about difficulties to reproduce errors and detect them.

#### **1.4 A single environment**

Bearing in mind the success of the VDV as a method of introducing students to programming, the idea to use it to introduce concurrent programming came natural. The advantages are obvious: the students already know the environment and have experience with it, which allows them to focus on the specific matters of the topic.

At first, the idea to modify VDV and add facilities that would allow its use to solve concurrent programming issues was considered. The result of this

analysis was that it should be fully reprogrammed. Some of the reasons behind this decision were:

- VDV is programmed in Delphi and only works with Windows. To produce a multiplatform version was a priority due to the fact that many students use free software and would not be able to install it.
- To create a full version would allow the gaining of experience and the improvement of some details (better messages in case of error, more coherent language, replace the enunciation of some primitives that can become ambiguous in other contexts, etc.).
- Finally, to structure VDV from its roots so that it can accept the extensions that allows its use to introduce students to concurrent programming.

### 1.5 Other aspects to consider

Even if experience shows that the tools which allow the visualization of the algorithms' execution are of great help when starting programming, it is important that the teacher pays special attention to the design of the activities that will be presented to the students.

The advantages of this tool stem from the fact that it makes code debugging easier. When the errors are visible, the students become less dependent on their teacher. Nevertheless, this can lead to the problems being solved by trial and error. Without taking anything away from this method, truth is that students need to develop other strategies to solve problems.

## 2. Characteristics of the Concurrent DV

En primer lugar cabe señalar que la versión de DaVinci First of all, it is important to highlight that the Concurrent DaVinci version, in addition to all the elements it has to make the teaching of concurrent programming easier and that will be briefly described below, adds a group of extensions to the Visual DaVinci, the most important being:

- Multiplatform implementation.
- Improvement of the error messages when compiling and executing.
- Incorporation of the string data type.
- Reading variables when the execution is taking place.
- Incorporation of a group of primitives (random numbers, convert text to numbers and numbers to text, etc.).
- Replace the indentation for the keywords *comenzar* and *fin* to limit code blocks.
- Admission of a distribution of flowers, papers and obstacles established by the student, and the possibility to preserving it for future executions.

The main extensions for concurrency are:

- Incorporation of the abstract data type *semaphor*, both general and binary. *Semaphores* solve the typical issues of concurrency. This data type only works with the primitives *iniciarSemaforo*, *esperar* and *avisar*.
- Incorporation of the most common short-term schedulers (FIFO, Round-robin, random). Although a concurrent program should work properly regardless the type of scheduler the operating system uses, the possibility to choose among different schedulers allows a practical observation of how they influence the execution.
- Incorporation of the possibility to manipulate the logical execution sequence. Concurrency introduces the problem of non-determinism. That is why it is essential to have a mechanism that allows both the exact reproduction of a concurrent execution that is finished (correctly or incorrectly) as the ability to execute traces established by the user (“forced traces”) that lead to error situations and that, regarding non determinism, can fail to happen even when a high number of program executions is made.

### 3. Example

Concurrent programming is usually taught through a group of traditional examples, each of which work as a metaphor for real situations that come up in the operating systems environment (original cradle of concurrent programming), and that then can extend to other domains.

These examples help to set out the typical issues that come up when accessing to shared resources and when synchronizing the different processes that help to the solution of the problem is needed. One of these examples is the Bounded Buffer Problem.

#### 3.1 The Bounded Buffer Problem

This problem is first considered taking into account a couple of processes that share a repository of predefined size. They are usually identified as *producer* (the process that adds elements to the repository) and *consumer* (the process that takes elements from the repository). Later, the problem can be generalized into *n producers* and *m consumers*.

The problem presents requirements from the point of view of accessing to the use of the shared resource (the repository), and also regarding the synchronization between processes: a producer cannot add elements to a full repository and a consumer cannot take elements from an empty repository.

In DVC, the simplified problem has been represented using two robots: one is the producer and the other one, the consumer. The repository is an area of the city (Slots for Consumer in Fig. 1). The elements of the repository are

flowers, which the producer picks up from other area of the city (Producer Resources Quadrant in Fig. 1), where the flowers have been placed before the execution. For the process to be executed indefinitely, once the consumer takes an element of the repository, he must place it back in the quadrant from which the producer takes it. The code of the example is the following:

```

programa ProductorConsumidor

variables //globales
  lleno:semaforoGeneral
  vacio:semaforoGeneral
  turno:semaforoBinario
  turnoCuadrante:semaforoBinario
  avenidaProd:numero
  avenidaCons:numero

  //Constantes
  AVFINAL : numero
      CALLECONS : numero
  CALLEPROD : numero

subprogramas

  procedimiento tomarFlorDeCuadrante(en
calle:numero; en avFin:numero)
  comenzar
    Pos(1,calle)
    mientras ! HayFlorEnLaEsquina
      Pos((posAv % avFin)+1,posCa)
    tomarFlor
  fin

hilos
  hilo productor (en caPro:numero;en
caCon:numero;en avFin:numero)
  comenzar
    iniciar
    mientras v
      comenzar
        esperar(turnoCuadrante)
        tomarFlorDeCuadrante(caPro, avFin)
        avisar(turnoCuadrante)
        esperar(vacio)
        esperar(turno)

```

```

        Pos (avenidaProd,caCon)
        depositarFlor
        avenidaProd := (avenidaProd % avFin)+ 1
        avisar (turno)
        avisar (lleno)
        fin
    fin

hilo consumidor (en caPro:numero;en
caCon:numero;en avFin:numero)
comenzar
    iniciar
    mientras v
        comenzar
        esperar (lleno)
        esperar (turno)
        Pos (avenidaCons,caCon)
        tomarFlor
        avenidaCons := (avenidaCons % avFin)+ 1
        avisar (turno)
        avisar (vacio)
        Pos (aleatorio (avFin)+1,caPro)
        depositarFlor
        fin
    fin

comenzar
    //constantes
    AVFINAL := 10
    CALLECONS := 2
    CALLEPROD := 8

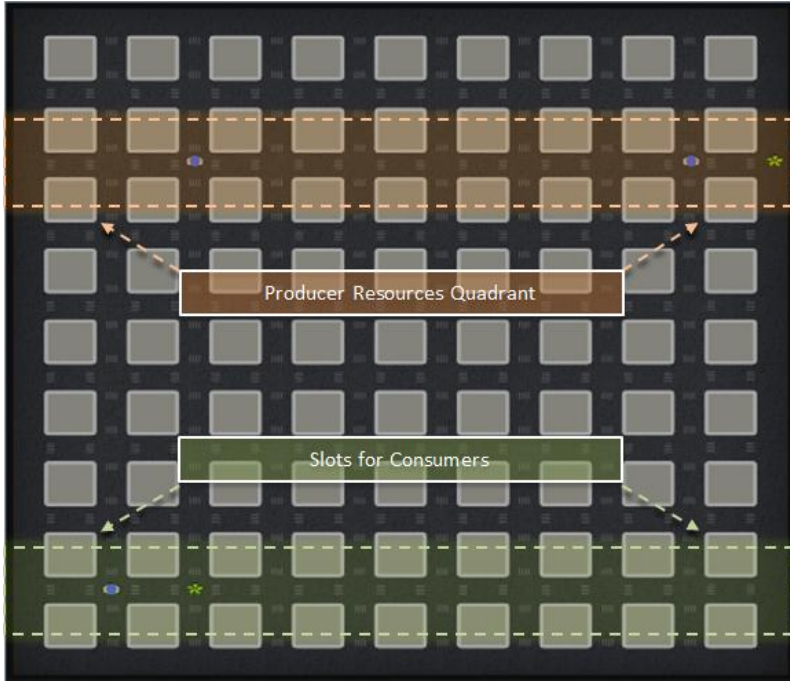
    //recursos compartidos
    avenidaprod := 1
    avenidacons := 1

    //inicilizacion semaforos
    iniciarSemaforo (lleno,0)
    iniciarSemaforo (vacio,AVFINAL)
    iniciarSemaforo (turno,1)
    iniciarSemaforo (turnoCuadrante,1)

    //arrancamos los hilos
    arrancar productor (CALLEPROD, CALLECONS,
AVFINAL)

```

```
arrancar consumidor(CALLEPROD, CALLECONS,  
AVFINAL)  
fin
```



*Fig. 1.*

#### 4. Future projects

The interpreter has been developed as part of a thesis. In order to show its functionality, an elementary development environment has also been created. In the future, we are planning to work on the following aspects, among others:

- Improvement of the development environment: the plan is to change the elementary environment the current version has. The new environment will have characteristics similar to the modern environments, like NetBeans and Eclipse, and will provide useful tools for configuration, code editing and debugging and it will also improve the visualization of the city and its robots. The fact that the new environment will be similar to the currently used will be an extra advantage.

- Language extension: the extension of the specification is planned in order to study in depth different concepts related to computer programming in general. Some of these extensions are: the possibility to declare functions, the incorporation of array data type, the improvement of the robots' characteristics (making it possible to uncoupling them from the threads of execution).
- Incorporation of other primitives and mechanisms for concurrency. It is contemplated incorporate at least messages and monitors.
- Verification of the "forced traces": currently, it is the users' responsibility to make sure that a "forced trace" is coherent with the code. In the future, the interpreter will have the ability to perform this task.

## References

1. Devlin, K.: Why universities require computer science students to take math, *Communications of ACM* 46, 9, 37-39 (september, 2003).
2. Kramer, J.: Is abstraction the key to computing?, *Communications of ACM* 50, 4, 36-42 (April 2007).
3. Dann, W., Cooper, S.: Education: Alice 3: concrete to abstract. *Communications of ACM* 52, 8, 27-29 (August 2009).
4. Utting, I., Cooper, S., Kölling, M., Maloney, J., Resnick, M.: Alice, Greenfoot, and Scratch-A Discussion, *Transactions on Computer Education* 10, 4, Article 17, 11 pages (November 2010).
5. Champredonde, R., De Giusti, A.: Design and Implementation of Visual Da Vinci. In: *III Congreso Argentino en Ciencias de la Computación (CACiC 1997)*, La Plata (1997).
6. De Giusti, A. y otros. *Algoritmos, Datos y Programas*. Pearson Education, Buenos Aires (2001).
7. Feierherd, G., Depetris, B., Jerez, M.: Una evaluación sobre la incorporación temprana de algorítmica y programación en el ingreso a informática. In: *VII Congreso Argentino de Ciencias de la Computación (CACiC 2001)*, El Calafate (2001).

# Open Access to Scientific Literature and Research Data: A Window of Opportunity for Latin America

MARÍA DEL ROSARIO MOLFINO<sup>1,2</sup>, CLAUDIA M. GONZÁLEZ<sup>1,3</sup>

<sup>1</sup>Universidad Nacional de La Plata, School of Humanities and Education Sciences.  
Library Science Department

<sup>2</sup>Universidad Nacional de La Plata. School of Information Technology Library

<sup>3</sup>Institute of Research in Humanities and Social Sciences (IdIHCS - UNLP/CONICET La Plata)  
48 e/6 y 7, La Plata, Argentina

{rmolfino, cgonzalez}@fahce.unlp.edu.ar

***Abstract:** The advance that the international open access movement has had in the last decade may seem to suggest that we are witnessing an important change in the model of scientific communication. This paper introduces the fundamental concepts of this movement, and in turn tries to measure the impact it has had in Latin America based on the development of different strategies.*

***Keywords:** Open Access, Open Access Movement, Academic and Scientific Production, Primary Data, Repositories.*

## 1. Introduction

The issue of *Open Access* to scientific information (OA) becomes strong in the 90s, although it can be traced back to 1966 [1]. It emerges in a scenario marked by the economic difficulty of the state research system to maintain its information services based on the subscription to paid resources. The movement takes shape when institutions understand that they are trapped in a system where, in addition to subsidizing the research processes, they have to pay for access to sources of information held by publishing houses that, paradoxically, become holders of the intellectual property rights of a large part of the results of their activity. Consequently, the researchers start to suffer the non-access to what is produced by their own –local or disciplinary– community.

The solution seems to be on a new model that, based on the technology of Internet, calls on the scientific and academic community to consent the free distribution of their intellectual property. Most searches are for authors of non-published works –as the case of most post-graduate theses– and authors of journal articles who usually receive no profit in exchange. This situation clearly leaves them out of the debate on open access to books, music and



movies. The most important argument is full compatibility with the traditional publishing system with respect to peer review. The model proposes that institutions invest in structuring, on the one hand, low-cost digital publishing systems to produce free access journals with referees who donate their work (which is indeed the usual fashion), and on the other hand, it proposes the development of digital repositories to store their production, i.e. the creation of platforms that facilitate their authors the self-archiving of materials once they have been through the –traditional or not– publishing circuit, guaranteeing some kind of peer review [2].

This initiative is an integral part of a group of movements which may be gathered under the large umbrella of open access: *open access to knowledge* [3], *open access to information* [4], *open access to scientific data* [5], *free software* [6], *open source* [7], *open-source hardware* [8], locally known as *common property* [9]. Even though the process we are analyzing here may be classified under this “free culture”, it is worth mentioning that, at present, it has already involved the political decision of several governments, the commitment of resources of several scientific organizations and a major pragmatic effort of the world leading publishing houses in order to advance to new global management and business models as survival strategy.

This paper is intended to introduce the main aspects of the *OA* movement, starting with the first conceptualization which was formally presented in the Budapest Initiative in 2001 and which was later consolidated in the Bethesda Statement and the Berlin Declaration in 2003. We also intend to specify the media proposed to achieve these goals and present current data which will show the global situation as well as the movement's particular situation in Latin America. A special interest is taken in the most recent derivation of the movement addressed to *open science data*, and we conclude by drafting of a window of opportunity for our environment.

## 2. Main Concepts

### 2.1 Cost Free

From the financial point of view, the cost-free availability of scientific content in Internet means a radical change in the scientific communication model. Since 1665 –the year when the first scientific journal was established–, this kind of publication gained a position at the core of the research system by becoming the quintessential means for knowledge sharing. The possibility of examining in academic articles the results obtained by others has allowed, for the past 350 years, a greater flow of ideas, information and data, which may be subjected to observation and improvement. According to Ulrich's Periodicals Directory –a global directory with information about journals– there are currently over 93,000 active titles in the world [10], producing about 1,7 million academic articles [11].

However, and notwithstanding the value that the scientific journals represent for the development of science, it is a well-known fact that for the past 20 years, the costs established by the publishing market have undermined the sustainability of this model. The Harvard University Library—one of the best financed libraries in the world— issued a memorandum in April 2012 stating its concern regarding the cost faced for the subscription to services from the main journal publishers. With \$3.75 million dollars being only a 10% of what the library acquires on a yearly-basis, they denounce a 145% increase in the past six years imposed by the main publishing groups (Elsevier, Springer y Willey) [12]. In Argentina, the Science and Technology Electronic Library (Biblioteca Electrónica de Ciencia y Tecnología), financed by the nation, reports a 58% increase in its 2010/2011 budget keeping a consistent subscription package, comprising barely fewer than 11,000 titles [13]. Considering that knowledge is a type of possession that may be shared without exhaustion and that the ubiquity offered by Internet is a fact for scientists around the world, the opportunity to boost the scientific communication model to its highest expression may be found within reach. If the cost of generating new knowledge and preparing a scientific article is fully afforded by the authors and their institutions, the cost of *peer review*—the other fundamental element of the model— and on-line publishing, seems to fail to justify the price imposed by publishers. For promotion agencies, that invest in generating science and then pay to have access to their results, adhering to and promoting an economic change in the model is a way of demanding a fairer balance between their input and output. For libraries, promoting free access is a way of breaking down barriers in order to guarantee the access to information as a universal right. Finally, for authors, this implies the opportunity to read and be read without restrictions. Although it is worth mentioning that any kind of article openly published on the Internet cannot be read without a considerable investment in hardware, software and networks—an infrastructure that is directly afforded by users themselves or indirectly by the taxes they pay to sustain the public system—, the underlying intention is to break down the barrier of payment for content, so that more people can gain greater access to research literature.

## **2.2 Freedom of Use and Re-Use**

Another basic element of OA, in addition to the cost free guarantee, is the abolition of certain restrictions in the rights of exploitation of works. As it is well-known, copyrights imply moral rights and exploitation rights. Moral rights are basically two-fold: acknowledgement of authorship and respect for the integrity of the work, and they do not expire. However, exploitation rights do expire. For a period of about 70 years after the death of the author, depending on the legislation of the country, the successors in title preserve their reproduction, distribution, public communication and work transformation rights. Based on the OA movement, the way in which this

legislation has been transmitted to society has been one of the factors why authors have been hostages to publisher's commercial interests for a long time.

Scholars are not interested in selling, their effort is rather put on gaining acceptance of their ideas and progress by the community. For most scientific authors, a personal financial retribution as consideration for their contribution to knowledge is not as important as a compensation through peer acknowledgement and prestige. It may seem that while exploitation rights may lack importance, the other component of copyrights is vital to obtain the acknowledgement of "Who's who" in science. By recognizing this duality, OA initiatives have worked on creating standardizing instruments that allow a real change in the way knowledge is spread, and they call on authors to exercise their right of freedom to choose what they want to do or not with their works.

The main contribution is provided by open licenses, including Copyleft in the field of Information technology and Creative Commons in the field of publications –which are the most popular ones. Both instruments generally seek to extend their application based on regulations themselves, i.e. imposing or suggesting for derived works the application of the same type of license. This would allegedly entail a cascade propagation effect that would end up changing the information ecosystem around us. The Creative Commons license –which is much less rigid than Copyleft– recognizes authorship, but it is more flexible because it offers alternatives regarding the creation of derived works.

As mentioned in the previous section, regarding scientific progress and its close relationship with the possibility of improving the ideas and results obtained by others, offering immediate (i.e. cost-free) access and admitting the re-use of information are two vital aspects. Being free not only to read what has been produced, but also to manipulate such content in order to produce new enriched works, is one of the substantial changes of the new model.

### **2.3 Gold Road and Green Road**

Considering that OA to scientific literature entails that users may read, download, copy, distribute, print, search and/or link the complete text of scientific articles and use them for any other legitimate purpose, without any financial, legal, or technical limitations other than gaining access to Internet itself, there are basically two roads to reach open access: the authors publish their works in open access journals (gold road), and/or the authors self-archive their works in their personal websites or in an open access institutional or thematic repository (green road).

The first open access scientific journals appear in the early 1990, and according to Ulrich's Periodicals Directory, they currently comprise over 11,000 titles; 6,798 of which have external review. Whether considering the

total number of scientific journals or only the journals with peer review system, the percentage of OA titles is between the 12-13% of the total number of titles in the world [10]. The Directory of Open Access Journal (DOAJ), a multidisciplinary directory of open access journals supported by Lund University, currently includes over 7800 OA journals. This category includes: open and cost free for both readers and authors journals, and those which are paid only by authors, also journals that combine the traditional subscription model permitting authors to choose if they want to pay for open access publishing, and finally, journals that offer contents for free after an embargo period [14]. All of them pursue the same quality standards as the traditional subscription model journals, and are therefore starting to be indexed by the main specialized directories.

Institutional repositories, in turn, are collections of diverse academic materials digital objects, grouped following an institutional production or similar subject criteria, and they always follow a well-defined policy. Some of them, among other features, admit self-archiving by the authors; guarantee interoperability and adjust to the Open Archives Initiative (OAI) protocol that enables their resources to be accessed by other systems; and are committed to long-term preservation. Their purpose is larger than the one pursued by gold road journals as they seek, in addition to spreading knowledge, to provide visibility to their institutional production, organize it and preserve it. Their origin dates back to 1991, when the thematic repository arXiv is founded, created by physicians to openly share their pre-prints. At present, according to the *OpenDOAR* directory of OA repositories, there are over 2100; distributed as follows: 47% in Europe, 21% in USA, 18% in Asia, 7% in South America and 7% in the rest of the world.

The fashion in which these two roads have evolved is subject to permanent study, mainly because leading science managers feel the need to define stimulus and funding policies in order to involve their institutions in OA projects. Recent examples are the report of the Danish Agency for Science [15] and the report prepared for the UK Ministry of Science, known as Finch Report [16].

## 2.4 Research Data

There is a branch of the *OA* movement that stands out mainly because of the advantage it presents related to the feature of re-use of information. This is a trend that works to make the data generated during research processes openly available. This is an major proposal because, in addition to generally maintaining OA elements, it includes a change in the scientific communication system by suggesting another form of distribution of new knowledge different from the hegemonic form that has followed the scientific journal article in the past three centuries.

Having access to data and models that scientists gather and create during their research involves depositing said data in an *ad hoc* repository. These data may have an associated journal article or not, since one of the aspects

proposed by the model is to reduce the time span between the production of information and the moment when it reaches the community of interest to be reviewed. It is well-known that in many disciplines, such as medicine, the time required to complete the circuit until data reaches publication may be detrimental for the scientific community to solve an urgent problem. However, this possibility of obtaining faster feedback from colleagues, in a more collaborative proposal, involves developing effective communication mechanisms that turn data intelligible for others. Although all repositories require digital objects to be stored together with their corresponding meta-data records, in primary data repositories, this kind of record becomes paramount as it will provide information regarding where the data come from, how the data were acquired and treated, on which date, how the data were analyzed for a scientific purpose and how the data must be used. This matter becomes a real challenge when the purpose is also to reach multidisciplinary audiences. On the other hand, this same OA trend proposes that data provided in scientific journal works should be accessible to be re-used and should be presented in a way that allows for a direct correspondence with the parts of the article in question. Both the repository record and the journal article should instruct under which conditions data can be used.

### **3. Actors in Latin America**

In Latin America we are witnessing not only the growth in the number of open access repositories and journals, but also the development of other related expressions and initiatives that have emerged in the domain of science and technology institutions. In order to have a more complete view of our regional reality, some statistical data extracted from reference websites will be shown and the main progress achieved will be described.

#### **3.1 Growth Aspects**

With regard to the green road, i.e. the development of institutional or thematic repositories, such as the one shown in Table 1, Latin America (excluding Mexico), is a bit behind as compared to other continents and North America. If we take into account the 20 repositories kept by Mexico (which in the Table are included in North America), Latin America holds 203 repositories, a 9.4% of the total number of repositories registered in *OpenDOAR* [17]. At the top is Brazil with 62 (30.5%), followed by Colombia, Mexico and Argentina with 20 (9.9%) each, then there are Ecuador with 17 (8.4%), Peru and Venezuela with 13 (6.4%) and Chile with 10 (4.9%). Institutional repositories clearly prevail over disciplinary ones. Considering only the aforementioned countries, out of 175 repositories, 142 (81.1%) are institutional, 16 (9.1%) are disciplinary and the remaining 17 (10%) are aggregating and governmental repositories.

Europe	1018	47.1%
North America	463	21.4%
Asia	382	17.7%
South America	159	<b>7.4%</b>
Australasia	60	2.8%
Africa	54	2.5%
Caribbean	13	<b>0.6%</b>
Central America	11	<b>0.5%</b>
Other	2	0.1%
TOTAL	2162	100.0%

Table 1. Number of Repositories per Continent. Source OpenDOAR (28 June, 2012)

In terms of the gold road, we can observe that all regions are showing growth. Although Europe stands out, we can also note that Latin America has kept a growth rate similar to that of North America.

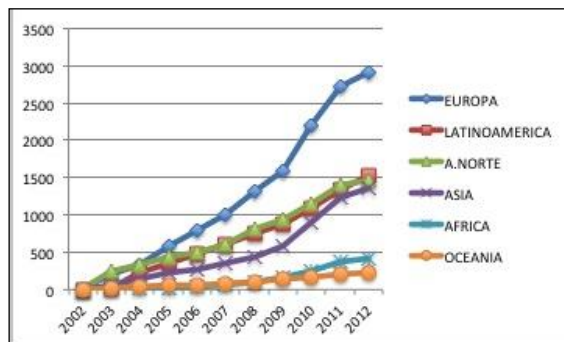


Fig. 1- Number of OA Journals per Continent. Source DOAJ (28 June, 2012)

If we estimate the annual growth rate between 2003 and 2011, Latin America is the region that accounts for the most significant growth (0.73), followed by Africa (0.62), Asia (0.46), Europe (0.38), Oceania (0.36) and North America (0.25). The high rate shown by Latin America is explained by the growth that exceeded a 1200% between 2003 y 2004.

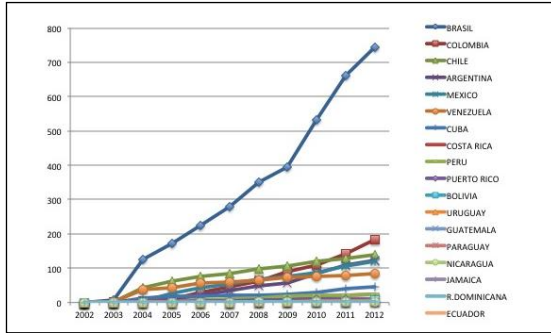


Fig. 2 - Number of OA Journals per Latin American Countries. Source DOAJ (28 June, 2012)

When analyzing the increase per countries of *OA* journals in Latin America, the sharp difference shown by Brazil as compared to the rest is easily noticed. However, the annual growth rate between 2004 and 2011, exhibits better figures for Argentina (0.59), followed by Colombia (0.49) and Mexico (0.23).

### 3.2 Regulatory Aspects

The Berlin Declaration of 2003 is a milestone that initiates the political/governmental recognition at world level of the *OA principles*. The major German scientific organizations committed thereby to call on their scientists to publish in compliance with these principles, i.e. to follow any (or both) road(s). In turn, said institutions would guarantee publications of this kind to be recognized during the scientific career evaluation processes and they took a commitment to manage and maintain digital repositories with long term preservation of said materials.

This major agreement later drove other scientific and technical organizations from several countries to adopt measures that generally range from signing adherence, preparing statements, establishing mandates, creating lines of credit for *OA* projects, to enacting laws. Similarly, in Latin America, Brazil and Argentina have bills of law which are being discussed at the Congress. Among the essential points included in said bills of law is the obligation to deposit works in open repositories for research financed with public funds. Peru has also started its way down this road with the Bill of law No. 01188 of May, 2012.

The Argentine bill of law defines as “*scientific-technical production the set of documents resulting from the scientific-technical activities that go through a quality review process, whether they are published or not. Primary research data may include, but are not limited to, text, numbers, equations, algorithms, images, audio and video, animations and software tools*”, Brazil's bill of law fails to consider primary data. Likewise, in Argentina, the

Ministry of Science, Technology and Productive Innovation (Ministerio de Ciencia, Tecnología e Innovación Productiva, MINCyT) recently concluded the formalization and creation of the National Science and Technology Digital Repositories System (Res.469/2011).

On the other hand, academic institutions that keep institutional repositories have found in the concept of *mandated* deposit in their own repositories a way of putting into practice *OA* statements. And they do this not only because it provides greater accessibility to information, but also because the results of the research they produce and/or fund may achieve greater visibility and impact. ROARMAP (*Registry of Open Access Repository Material Archiving Policies*), a registry of *OA* repositories archiving policies at international level that provides systematic information regarding their most salient aspects, in June 2012, contained 419 registered mandates, 16 out of which belonged to Latin America. In Argentina, for example, two university mandates of a different nature are the UNLP mandated theses deposit (resolution 78/2011) and the obligation to use *Creative Commons* licenses for UNC academic publications (resolution 116/2010) –the existence of which is known to us, despite the nonresistance of their registration in ROARMAP.

The progress of *OA* to scientific information would receive major support and boost were it established by national legislation.

Both Brazil and Argentina have bills of law which are under the process of discussion at the Congress, “*Dispõe sobre o processo de disseminação da produção técnico-científica pelas instituições de ensino superior no Brasil e dá outras providências*” and “*Creación de repositorios digitales abiertos de ciencia y tecnología. Ciencia Abierta Argentina 2010*”, respectively.

In Argentina, beginning in 2009 policies of *OA* to scientific knowledge start to spread and became established in the academic field, helped by the boost provided by the Ministry of Science, Technology and Productive Innovation (MINCyT), which recently concluded with the formalization and creation of the National Science and Technology Digital Repositories System (Res.469/2011), and the drafting of the bill of law that has recently been passed by the House of Representatives.

### 3.3 Funding Aspects

The region of Latin America has already some important projects related to *OA* which involve significant funding. The Network of Scientific Journals from Latin America and the Caribbean, Spain and Portugal (Red de Revistas Científicas de América Latina y el Caribe, España y Portugal) [18], promoted by Universidad Autónoma del Estado de México, and the SciELO (Scientific Electronic Library Online) [19] platform, developed by FAPESP-BIREME, which has been implemented by eight Latin American countries, gather their most relevant national production.

The Latin American Cooperation of Advanced Networks (Cooperación Latinoamericana de Redes Avanzadas, RedCLARA), with the support of the



Inter-American Development Bank, is promoting a project to create the Latin American Federated Network of Institutional Repositories (Red Federada Latinoamericana de Repositorios Institucionales), which intends to share and provide visibility to scientific production from higher education and scientific research institutions of Latin America. The initiative has been supported by eight countries of the region: Colombia, Argentina, Brazil, Chile, Ecuador, Mexico, Peru and Venezuela.

## 4. Conclusion

Having an environment rich in Open Access intellectual products at Latin American level allows us to think of a scenario of collective knowledge development with greater capacity to set out the strategic problems of the region. The fact that researchers have the chance to access not only articles, but also the data sets on which research is based is a key point to increase scientific capacity. This involves taking care of the systematic organization of the information produced in the region, assuming a commitment regarding long term preservation and developing information retrieval systems capable of showing said information in the context where it was produced providing a real opportunity for re-use. Similarly, these results will not only be beneficial for the society, but in the medium term, caring for the technological and organizational infrastructure requirements that this kind of activities involve will also turn to be beneficial; and this also tend to dissolve the barriers between more and less developed countries.

## References

1. Open Access Directory (OAD): Timeline of Open Access. (2011). <http://oad.simmons.edu/oadwiki/Timeline> [Accessed on: 25 March, 2011]
2. Suber, P.: Open access overview. (2010). <http://www.earlham.edu/~peters/fos/overview.htm> [Accessed on: 25 March, 2011].
3. Open Knowledge Foundation (OKF): Open knowledge definition (2011). <http://www.opendefinition.org/> [Accessed on: 25 March, 2011].
4. Assange, J.: What is WikiLeaks? (2011) <http://213.251.145.96/About.html> [Accessed on: 25 March, 2011].
5. Murray-Rust, P.: Open data in science. (2008) <http://precedings.nature.com/documents/1526/version/1/files/npre20081526-1.pdf> [Accessed on: 25 March, 2011].
6. Stallman, R.: The GNU manifesto. (1985). <http://www.gnu.org/gnu/manifesto.html> [Accessed on: 25 March, 2011].
7. Open Source Initiative (OSI): The Open Source definition. (2011). <http://www.opensource.org> [Accessed on: 25 March, 2011].

8. Bewens, M. P2P Foundation (2011) <http://blog.p2pfoundation.net> [Accessed on: 25 March, 2011].
9. Verselli, A., Thomas, H.: Repensando los bienes comunes: análisis socio-técnico sobre la construcción y regulación de los bienes comunes (2008). <http://www.bienescomunes.org/lectura/> [Accessed on: 25 March, 2011].
10. Abadal, E.: Retos de las revistas en acceso abierto: cantidad, calidad y sostenibilidad económica. Hipertext.net, 10 (2012). <http://www.upf.edu/hipertextnet/numero-10/retos-revistas-en-acceso-abierto.html> [Accessed on: 25 March, 2011]
11. Willetts, D.: Public access to publicly-funded research. Publishers Association annual general meeting, London (2012). <http://www.bis.gov.uk/news/speeches/david-willetts-public-access-to-research> [Accessed on: 23 June 2012].
12. Harvard Faculty Advisory Council. Faculty Advisory Council memorandum on journal pricing, <http://isites.harvard.edu/icb/icb.do?keyword=k77982&tabgroupid=icb.tabgroup143448> [Accessed on: 23 June 2012].
13. BECyT (Biblioteca Electrónica de Ciencia y Tecnología. MINCyT. Argentina), <http://www.biblioteca.mincyt.gov.ar> [Accessed on: 23 June 2012].
14. Abadal, E., Melero, R.: Las universidades y el apoyo institucional al Open Access. ThinkEPI, (2008) <http://www.thinkepi.net/las-universidades-y-el-apoyo-institucional-al-open-access>. [Accessed on: 28 June 2012]
15. FI & DEFF: Access to Research and Technical Information in Denmark <http://www.fi.dk/publikationer/2011/adgang-til-forskningsresultater-og-teknisk-information-i-danmark> (2011) [Accessed on: 26 June 2012]
16. Finch, J.: Accessibility, sustainability, excellence: how to expand access to research publications. Report of the Working on Expanding Access to Research Findings. <http://www.researchinfonet.org/publish/finch/> (2012) [Accessed on: 25 June 2012].
17. OpenDOAR - The Directory of Open Access Repositories, <http://www.opendoar.org/> [Accessed on: 25 June 2012].
18. Red de Revistas Científicas de América Latina y el Caribe, España y Portugal, <http://redalyc.uaemex.mx/>[Accessed on: 23 June 2012].
19. SciELO, Scientific Electronic Library Online, <http://www.scielo.org/>[Accessed on: 23 June 2012].



Esta edición de 150 ejemplares  
se terminó de imprimir en Estudiocentro,  
Bolívar, Buenos Aires, Argentina,  
en el mes de octubre de 2013.





Its objectives are:

Coordinate academic activities related to the improvement of the teachers' training as well as the curricular update and the use of shared resources to assist the development of both the Computer Sciences careers and the Technology careers in Argentina" and "To establish a cooperative framework for the development of Postgraduate activities in Computer Sciences and Technology, in order to optimize the assignation and use of the resources.

## **RedUNCI:**

This Network was formally created through an Agreement signed in November 1996 by five National Universities (UNSL, UBA, UNLP, UNS y UNCPBA), during the second edition of CACIC.

Actually 51 argentine Universities are active members of this network.

## **Regular Activities of the RedUNCI**

- Arrangement of an Annual Congress on Computer Science (CACIC) since 1995.
- Arrangement of an Annual Workshop for Researchers on Computer Science (WICC) since 1999.
- Meetings for university professors of Computer Science, for Postgraduate Dissertators and for specialists in certain areas, to promote the debate of common interest topics.
- Publication of *the Journal on Computer Science & Technology* by agreement with ISTECS (Iberoamerican Science and Technology Education Consortium).
- Annual Congress on Technology in Education and Education in Technologies (TE&ET) since 2006.
- Publication of the *Iberoamerican Journal of Technology in Education and Education in Technology*, since 2007.



ISBN 978-987-1985-20-3



9 789871 985203