Computer Science & Technology Series

XV Argentine Congress of Computer Science
Selected Papers

Computer Science & Technology Series

XV Argentine Congress of Computer Science
Selected Papers

Guillermo Simari | Patricia Pesado | José Paganini
(Eds)

Editorial de la Universidad de La Plata

De Giusti, Armando
Computer Science & Technology Series: XV Argentine Congress of Computer Science - Selected Papers.-
1a ed. - La Plata: Universidad Nacional de La Plata, 2010.
260 p.; 24x16 cm.

ISBN 978-950-34-0684-7

1. Informática. 2. Ciencias de la Computación. I. Título
CDD 005.3

Fecha de catalogación: 05/10/2010

Computer Science & Technology Series

XV Argentine Congress of Computer Science
Selected Papers

Diagramación: Andrea López Osornio

Editorial de la Universidad Nacional de La Plata
Calle 47 Nº 380 - La Plata (1900) - Buenos Aires - Argentina
Tel/Fax: 54-221-4273992
e-mail: editorial_unlp@yahoo.com.ar
www.unlp.edu.ar/editorial

La EDULP integra la Red de Editoriales Universitarias (REUN)

1º edición - 2010
ISBN Nº 978-950-34-0684-7
Queda hecho el depósito que marca la ley 11.723

Impreso en Argentina

Topics

X Intelligent Agents and Systems Workshop
Chairs Guillermo Simari (UNSur) Guillermo Leguizamón (UNSL) Laura Lanzarini (UNLP)

IX Distributed and Parallel Processing Workshop
Chairs Armando De Giusti (UNLP) Marcela Printista (UNSL) Jorge Ardenghi (UNSur)

VIII Information Technology Applied to Education Workshop
Chairs Cristina Madoz (UNLP) Miguel Fernández (UNER) María Eugenia Márquez (UNPA)

VII Graphic Computation, Imagery and Visualization Workshop
Chairs Silvia Castro (UNSur) Claudia Russo (UNLP-UNNOBA) Oscar Bría (INVAP)

VI Software Engineering Workshop
Chairs Patricia Pesado (UNLP) Elsa Estévez (UNSur) Alejandra Cechich (UNCOMA)

VI Database and Data Mining Workshop
Chairs Olinda Gagliardi (UNSL) Hugo Alfonso (UNLPam) Viviana Quincoces (UNJu)

IV Architecture, Nets and Operating Systems Workshop
Chairs Javier Diaz (UNLP) Antonio Castro Lechtaller (UTN) Hugo Padovani (UNMorón)

I Innovation in Software Systems Workshop
Chairs Pablo Fillottrani (UNSur) Nelson Acosta (UNCPBA) Uriel Cukierman (UTN)

Program Commitee

Abásolo María José (Spain)
Acero Alex (USA)
Acosta Nelson (Argentina)
Aguilar Castro José (Venezuela)
Alba Torres Enrique (Spain)
Alfonso Hugo (Argentina)
Ardenghi Jorge (Argentina)
Astudillo Hernán (Chile)
Baldassarri, Sandra (Spain)
Bertogna Leandro (Argentina)
Bertone Rodolfo (Argentina)
Bevilacqua Roberto (Argentina)
Bria Oscar (Argentina)
Brisaboa Nieves (Spain)
Buccella Agustina (Argentina)
Cabero Julio (Spain)
Cancela Héctor (Uruguay)
Castro Lechtaler Antonio Ricardo (Argentina)
Castro Silvia (Argentina)
Cechich Alejandra (Argentina)
Coello Coello Carlos A. (Mexico)
Collazos Ordóñez, César Alberto (Colombia)
Cukierman Uriel (Argentina)
De Alfonso Laguna Carlos (Spain)
De Giusti Armando (Argentina)
De Pablo Pons Juan (Spain)
De Petris Beatriz (Argentina)
Díaz Javier (Argentina)
Doallo Ramón (Spain)
Dosch Walter (Germany)
Dujmovic Jozo (USA)
Echaiz Javier (Argentina)
El Saddik Abed (Canada)
Errecalde Marcelo (Argentina)
Escarza Sebastián (Argentina)
Esquivel Susana (Argentina)
Estayno Marcelo (Argentina)
Estévez Elsa (Argentina)
Fabero Juan C. (Spain)
Fariña Antonio (Spain)
Feierherd Guillermo (Argentina)
Fernandez Miguel E. R. (Argentina)
Fillottrani Pablo (Argentina)
Finochietto Jorge (Italy)
Foti Antonio (Argentina)
Fusario Rubén Jorge (Argentina)
Gagliardi Olinda (Argentina)
García Garino Carlos (Argentina)
García Guibout Jorge (Argentina)
George Chris (United Nations)
Godo Luis (Spain)
González María Paula (Argentina)
Gorga Gladys (Argentina)
Gröller, Eduard (Austria)
Guerrero, Roberto (Argentina)
Gutierrez Claudio (Chile)
Hernández Peñalver Gregorio (Spain)
Iyad Rahwan (Dubay)
Janowski Tomasz (United Nations)
Jordan Mario (Argentina)
Juergen Dix (Germany)
Lanzarini Laura (Argentina)
Larrea Martín (Argentina)
Lecumberry Federico (Uruguay)
Leguizamon Guillermo (Argentina)
Luaces Miguel (Spain)
Lucero Margarita (Argentina)
Luque Emilio (Spain)
Luque Mónica (RITLA)
Madoz Cristina (Argentina)
Maguitman Ana (Argentina)
Malbrán María (Argentina)
Marcelo Carlos (Spain)
Margaleff Tomás (Spain)
Marín Mauricio (Chile)
Marquez María Eugenia (Argentina)
Marrone Luís (Argentina)
Martig Sergio (Argentina)

Program Commitee (cont.)

Martín María J. (Spain)
Mercado Gustavo (Argentina)
Merelo Juan Julián (Spain)
Motz Regina (Uruguay)
Naiouf Marcelo (Argentina)
Obac Roda Valentín (Brazil)
Otero Rita (Argentina)
Padovani Hugo (Argentina)
Paldao Carlos (RITLA)
Palomar Manuel (Spain)
Paolo Rosso (Spain)
Pardo Alvaro (Uruguay)
Pérez Carlos (Argentina)
Pesado Patricia (Argentina)
Pessaq Raúl (Argentina)
Piccoli María F. (Argentina)
Pina Alfredo (Spain)
Ponzoni Ignacio (Argentina)
Printista Marcela (Argentina)
Quincoces Viviana (Argentina)
Ramón Hugo (Argentina)
Rautek Peter (Austria)
Rexachs Dolores (Spain)
Riesco Daniel (Argentina)
Ripoll Ana (Spain)
Rodriguez de Souza Josemar (Brazil)
Rodriguez León Casiano (Spain)
Ronald Prescott Loui (USA)
Rossi Gustavo (Argentina)
Rosso Paolo (Spain)
Russo Claudia (Argentina)
Sanz Cecilia (Argentina)
Serón Arbeoloa Francisco (Spain)
Sierra Carles (Spain)
Simari Guillermo (Argentina)
Simari Patricio (Canada)
Sousa Pinto Jorge (Portugal)
Steinmetz Ralf (Germany)
Suppi Remo (Spain)
Tarouco Liane (Brazil)
Tartaglia Angelo (Italy)
Tinetti Fernando (Argentina)
Tirado Francisco (Spain)
Touriño Juan (Spain)
Vechietti Aldo (Argentina)
Vénere Marcelo (Argentina)
Verrastro Claudio (Argentina)
Vilanovai Bartroli, Ana (Holland)
Viola Ivan (Norway)
Vitturini Mercedes (Argentina)
Vizcaino Aurora (Spain)
Zamarro José Miguel (Spain)
Zbigniew Michalewicz (Australia)

Organizing Commitee

Universidad Nacional de Jujuy -
Facultad de Ingeniería
Argenitna

President
Paganini José Humberto

Coordinators
Laserre Cecilia Maria

Collaborators
Quincoces Viviana
Galvez Díaz Pilar
Ayusa Cristina
Aparicio Maria C.
Figueroa Sebastián Marcos
Méndez Sandra
Pérez Otero Nilda

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, Imagery and Visualization, Information Technology applied to Education, Databases and Data Mining, Innovation in Software Systems, Theory, Models and Optimization).

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 2009 in Jujuy

CACIC'09 was the fifteenth Congress in the CACIC series. It was organized by the School of Engineering of the National University of Jujuy.

The Congress included 9 Workshops with 130 accepted papers, 1 main Conference, 4 invited tutorials, different meetings related with Computer Science Education (Professors, PhD students, Curricula) and an International School with 5 courses. (http://www.cacic2009.fi.unju.edu.ar/cacic2009ing)

CACIC 2009 was organized following the traditional Congress format, with 9 Workshops covering a diversity of dimensions of Computer Science

Research. Each topic was supervised by a committee of three chairs of different Universities.

The call for papers attracted a total of 267 submissions. An average of 2.7 review reports were collected for each paper, for a grand total of 720 review reports that involved about 300 different reviewers.

A total of 130 full papers were accepted and 20 of them were selected for this book.

Acknowledgments

CACIC 2009 was made possible due to the support of many individuals and organizations. The School of Engineering of the National University of Jujuy, RedUNCI, the Secretary of University Policies, and the National Agency of Scientific and Technological Advancement 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
Dr. Guillermo Simari
RedUNCI

Table of Contents

X

Intelligent Agents and Systems Workshop

An argumentation framework with uncertainty management designed for dynamic environments

MARCELA CAPOBIANCO[1, 2], GUILLERMO R. SIMARI[1]

[1] Artificial Intelligence Research and Development Laboratory
Department of Computer Science and Engineering - Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina.
[2] Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina.
{mc,grs}@cs.uns.edu.ar

Abstract. *Nowadays, data intensive applications are in constant demand and there is need of computing environments with better intelligent capabilities than those present in today's Database Management Systems (DBMS). To build such systems we need formalisms that can perform complicate inferences, obtain the appropriate conclusions, and explain the results. Research in argumentation could provide results in this direction, providing means to build interactive systems able to reason with large databases and/or different data sources.*
In this paper we propose an argumentation system able to deal with explicit uncertainty, a vital capability in modern applications. We have also provided the system with the ability to seamlessly incorporate un-certain and/or contradictory information into its knowledge base, using a modular upgrading and revision procedure.

1. Introduction and motivations

Nowadays, data intensive applications are in constant demand and there is need of computing environments with better intelligent capabilities than those present in today's Database Management Systems (DBMS). Recently, there has been progress in developing efficient techniques to store and retrieve data, and many satisfactory solutions have been found for the associated problems. However, the problem of how to understand and interpret a large amount of information remains open, particularly when this information is uncertain, imprecise, and/or inconsistent. To do this we need formalisms that can perform complicate inferences, obtain the appropriate conclusions, and explain the results.
Research in argumentation could provide results in this direction, providing means to build interactive systems able to reason with large databases and/or different data sources, given that argumentation has been successfully used to develop tools for common sense reasoning [8, 4, 14].
Nevertheless, there exist important issues that need to be addressed to use argumentation in these kind of practical applications. A fundamental one concerns the quality of the information expected by argumentation systems: most of them are unable to deal with explicit uncertainty which is a vital

capability in modern applications. Here, we propose an argumentation-based system that addresses this problem, incorporating possibilistic uncertainty into the framework following the approach in [9]. We have also provided the system with the ability to seamlessly incorporate uncertain and/or contradictory information into its knowledge base, using a modular upgrading and revision procedure.
This paper is organized as follows. First, we present the formal definition of our argumentation framework showing its fundamental properties. Next, we propose an architectural software pattern useful for applications adopting our reasoning system. Finally, we state the conclusions of our work.

2. The OP-DeLP programming language: fundamentals

Possibilistic Defeasible Logic Programming (P-DeLP) [1, 2] is an important ex-tension of DeLP in which the elements of the language have the form (φ,α), where φ is a DeLP clause or fact. Below, we will introduce the elements of the language necessary in this presentation. Observation based P-DeLP (OP-DeLP) is an optimization of P-DeLP that allows the computation of warranted arguments in a more efficient way, by means of a pre-compiled knowledge component. It also permits a seamless incorporation of new perceived facts into the program codifying the knowledge base of the system. Therefore the resulting system can be used to implement practical applications with performance requirements. The idea of extending the applicability of DeLP in a dynamic setting, incorporating perception and pre-compiled knowledge, was originally conceived in [5]. Thus the OP-DeLP system incorporates elements from two different variants of the DeLP system, O-DeLP [5] and P-DeLP [9]. In what follows we present the formal definition of the resulting system.
The concepts of signature, functions and predicates are defined in the usual way. The alphabet of OP-DeLP programs generated from a given signature Σ is composed by the members of Σ, the symbol "~" denoting strong negation [11] and the symbols "(", ")", "." and ",". Terms, Atoms and Literals are defined in the usual way. A certainty weighted literal, or simply a weighted literal, is a pair (L,α) where L is a literal and $\alpha\in[0,1]$ expresses a lower bound for the certainty of φ in terms of a necessity measure.
OP-DeLP programs are composed by a set of observations and a set of de-feasible rules. Observations are weighted literals and thus have an associated certainty degree. In real world applications, observations model perceived facts. Defeasible rules provide a way of performing tentative reasoning as in other argumentation formalisms.

Definition 1. A defeasible rule has the form $(L_0 \prec L_1, L_2,.., L_k, \alpha)$ where L_0 is a literal, $L_1, L_2,.., L_k$ is a non-empty finite set of literals, $\alpha\in[0,1]$ expresses a lower bound for the certainty of the rule in terms of a necessity measure.

Intuitively a defeasible rule $L_0 \prec L_1, L_2,.., L_k$ can be read as $L_1, L_2,.., L_k$ provide tentative reasons to believe in L_0 [15]. In OP-DeLP these rules also have a certainty degree, that quantifies how strong the connection between the premises and the conclusion is. A defeasible rule with a certainty degree 1 models a strong rule.

Ψ

```
(virus(b),0.7)
(local(b),1)
(local(d),1)
(~filters(b),0.9)
(~filters(c),0.9)
(~filters(d),0.9)
(black_list(c),0.75)
(black_list(d),0.75)
(contacts(d),1)
```

Δ

```
(move_inbox(X) -< ~filters(X),0.6)
(~move_inbox(X) -< move_junk(X),0.8)
(~move_inbox(X) -< filters(X),0.7)
(move_junk(X) -< spam(X),1)
(move_junk(X) -< virus(X),1)
(spam(X) -< black_list(X),0.7)
(~spam(X) -< contacts(X),0.6)
(~spam(X) -< local(X),0.7)
```

Figure 1: *an OP-DeLP program for email filtering*

A set of weighted literals Γ will be deemed as contradictory, denoted as $\Gamma \vdash\!\!\sim\!\bot$, iff $\Gamma \vdash\!\!\sim (l,\alpha)$ and $\Gamma \vdash\!\!\sim (\neg l,\beta)$ with α and $\beta > 0$. In a given OP-DeLP program we can distinguish certain from uncertain information. A clause (γ,α) will be deemed as *certain* if α=1, otherwise it will be *uncertain*.

Definition 2. [OP-DeLP Program] An OP-DeLP program $\mathcal{P}$ is a pair (Ψ,Δ), where Ψ is a non-contradictory finite set of observations and Δ is a finite set of defeasible rules.

Example 1. Fig.1 shows a program for basic email filtering. Observations describe different characteristics of email messages. Thus, virus(X) stands for "message X has a virus"; local(X) indicates that "message X is from the local host"; filters(X) specifies that "message X should be filtered" redirecting it to a particular folder; black_list(X) indicates that "message X is considered dangerous" because of the server it comes from; and contacts(X) indicates that "the sender of message X is in the contact list of the user".
The first rule expresses that if the email does not match with any user-defined filter then it usually should be moved to the inbox folder. The second rule indicates that unfiltered messages in the junk folder usually should not be moved to the inbox. According to the third rule, messages to be filtered should not be moved to the inbox. The following two rules establish that a message should be moved to the junk folder if it is marked as spam or it contains viruses. Finally there are three rules for spam classification: a message is usually labeled as spam if it comes from a server that is in the blacklist. Nevertheless, even if an email comes from a server in the blacklist it is not labeled as spam when the sender is in the contact list of the user. Besides, a message from the local host is usually not classified as spam.

In OP-DeLP the proof method, written $\vdash\!\!\sim$, is defined by derivation based on the following instance of the generalized modus ponens rule (GMP): $(L_0 \prec L_1 \wedge L_2 \wedge .. \wedge L_k, \gamma), (L_1,\beta_1),..,(L_k, \beta_k) \vdash (L_0, \min(\beta_1,.., \beta_k))$, which is a particular instance of the well-known possibilistic resolution rule. Literals in the set of observations

Ψ are the basis case of the derivation sequence, for every literal Q in Ψ with a certainty degree α it holds that (Q,α) can be derived from $\mathcal{P}=(\Psi,\Delta)$.
Given an OP-DeLP program $\mathcal{P}$, a query posed to $\mathcal{P}$ corresponds to a ground literal Q which must be supported by an *argument* [5,10].

Definition 3. [Argument – Subargument] Let $\mathcal{P}=(\Psi,\Delta)$ be a program $\mathcal{A} \subseteq \Delta$ is an argument for a goal Q with necessity degree $\alpha>0$, denoted as $\langle \mathcal{A}, Q, \alpha\rangle$, iff (1) $\Psi \cup \mathcal{A} \vdash (Q,\alpha)$; (2) $\Psi \cup \mathcal{A}$ is not contradictory; and (3) there is not $\mathcal{A}_1 \subseteq \Delta$ such that $\Psi \cup \mathcal{A}_1 \vdash (Q,\beta)$, $\beta >0$. An argument $\langle \mathcal{A}, Q, \alpha\rangle$ is a subargument of $\langle B, R, \beta\rangle$ iff $\mathcal{A} \subseteq B$.

As in most argumentation frameworks, arguments in O-DeLP can attack each other. An argument $\langle \mathcal{A}_1,Q_1,\alpha\rangle$ *counter-argues* an argument $\langle \mathcal{A}_2,Q_2,\beta\rangle$ at a literal Q if and only if there is a sub-argument $\langle \mathcal{A},Q,\gamma\rangle$ of $\langle \mathcal{A}_2,Q_2,\beta\rangle$, (called *disagreement subargument*), such that Q_1 and Q are complementary literals.

Defeat among arguments is defined combining the counterargument relation and a preference criterion "$\preceq$". This criterion is defined on the basis of the necessity measures associated with arguments.

Definition 4. [Preference criterion $\succeq$] [9] Let $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$, be a counterargument for $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$. We will say that $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ is preferred over $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ (denoted $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle \preceq \langle \mathcal{A}_1, Q_1, \alpha_1\rangle$. If it is the case that $\alpha_1 > \alpha_2$, then we will say that $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ is strictly preferred over $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle \prec \langle \mathcal{A}_1, Q_1, \alpha_1\rangle$. Otherwise, if $\alpha_1 = \alpha_2$ we will say that both arguments are equi-preferred, denoted $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle \approxeq \langle \mathcal{A}_1, Q_1, \alpha_1\rangle$.

Definition 5. [Defeat][9] Let $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ and $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ be two arguments built from a program $\mathcal{P}$. Then $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ defeats $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ iff (1) $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ counterargues $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ with disagreement subargument $\langle \mathcal{A}, Q, \alpha\rangle$ and (2) Either it is true that $\langle \mathcal{A}, Q, \alpha\rangle \prec \langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ in which case $\langle \mathcal{A}_1, Q_1, \alpha_1\rangle$ is a proper defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$ or $\langle \mathcal{A}, Q, \alpha\rangle \approxeq \langle \mathcal{A}_1, Q_1, \alpha_1\rangle$, in which case $\langle \mathcal{A}, Q_1, \alpha_1\rangle$ is a blocking defeater for $\langle \mathcal{A}_2, Q_2, \alpha_2\rangle$.

As in most argumentation systems [7], OP-DeLP relies on an exhaustive dialectical analysis which allows determining if a given argument is *ultimately* undefeated (or *warranted*) wrt a program $\mathcal{P}$. An *argumentation line* starting in an argument $\langle \mathcal{A}_0,Q_0,\alpha_0\rangle$ is a sequence $[\langle \mathcal{A}_0,Q_0,\alpha_0\rangle, \langle \mathcal{A}_1,Q_1,\alpha_1\rangle, \ldots, \langle \mathcal{A}_n,Q_n,\alpha_n\rangle, \ldots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). In order to avoid *fallacious* reasoning, argumentation theory imposes additional constraints on such an argument exchange to be considered rationally acceptable wrt an OP-DeLP program $\mathcal{P}$, namely:

1. **Non-contradiction:** given an argumentation line, the set of arguments of the proponent (resp. opponent) should be non-contradictory wrt $\mathcal{P}$.
2. **No circular argumentation:** no argument $\langle \mathcal{A}_j, Q_j, \alpha_j \rangle$ in the argumentation line is a sub-argument of an argument $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ such that $i < j$.
3. **Progressive argumentation:** every blocking defeater $\langle \mathcal{A}_i, Q_i, \alpha_i \rangle$ in the argumentation line is defeated by a proper defeater $\langle \mathcal{A}_{i+1}, Q_{i+1}, \alpha_{i+1} \rangle$ in this line.

An argumentation line satisfying the above restrictions is called *acceptable*, and can be proved to be finite. Given a program $\mathcal{P}$ and an argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, the set of all acceptable argumentation lines starting in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ (i.e. all possible dialogs rooted in $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$, formalized as a *dialectical tree*, denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$. Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ can be marked as *undefeated* and *defeated* nodes (U-nodes and D-nodes, resp.). A dialectical tree will be marked as an AND-OR tree: all leaves in $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ will be marked U-nodes (as they have no defeaters), and every inner node is to be marked as *D-node* iff it has at least one U-node as a child, and as *U-node* otherwise. An argument $\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle$ is ultimately accepted as valid (or *warranted*) iff the root of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0, \alpha_0 \rangle}$ is labeled as *U-node*.

Definition 6. [Warrant][9]Given a program $\mathcal{P}$, and a literal Q, Q is warranted wrt $\mathcal{P}$ iff there exists a warranted argument $\langle \mathcal{A}, Q, \alpha \rangle$ than can be built from $\mathcal{P}$.

To answer a query for a given literal we should see if there exists a warranted argument supporting this literal. Nevertheless, in OP-DeLP there may be different arguments with different certainty degrees supporting a given query. This fact was not considered in [9], but we are clearly interested in finding the warranted argument with the highest certainty degree.

Definition 7. [Strongest Warrant]Given a program $\mathcal{P}$, and a literal Q, we will say that is the strongest warrant degree of Q iff (1) there exists a warranted argument $\langle \mathcal{A}, Q, \alpha \rangle$ than can be built from $\mathcal{P}$, and (2) no warranted argument $\langle \mathcal{B}, Q, \beta \rangle$ such that $\beta > \alpha$ can built from $\mathcal{P}$.

Note that to find out the strongest warrant degree for a given literal Q we need to find the strongest warranted argument supporting it, that is, the warranted argument supporting Q with the higher certainty degree. Then, to find the strongest warrant degree for a literal Q we must first build the argument $\mathcal{A}$ that supports the query Q with the highest possible certainty degree and see if $\mathcal{A}$ is a warrant for Q. Otherwise we must find another argument $\mathcal{B}$ for Q with the highest certainty degree among the remaining

ones, see if it is a warrant for *Q*, and so on, until a warranted argument is found or there are no more arguments supporting *Q*.

Example 2. Consider the program shown in Example 1 and let *move_ inbox(d)* be a query wrt this program. The search for a warrant for *move_inbox(d)* will result in an argument $\langle \mathcal{A}$, *move_ inbox(d)*, 0.6$\rangle$, with $\mathcal{A}$ = {(*move_ inbox(d)* —< *~filters(d)*, 0.6)} allowing to conclude that message d should be moved to the folder Inbox, as it has no associated filter with a certainty degree of 0.6. However, there exists a defeater for $\langle \mathcal{A}$, *move_ inbox(d)*, 0.6$\rangle$, namely $\langle \mathcal{B}$, *~move_ inbox(d)*, 0.7$\rangle$, as there are reasons to believe that message *d* is spam:

$$\mathcal{B} = \{(\sim move\_\ inbox(d) \prec move\_\ junk(d), 0.8),$$
$$(move\_junk(d) \prec spam(d), 1), (spam(d) \prec black\ list(d), 0.7)\}$$

Using the preference criterion, $\mathcal{B}$ is a proper defeater for $\mathcal{A}$. However, two counterarguments can be found for $\mathcal{B}$ since message *d* comes from the local host, and the sender is in the user's contacts list:

– $\langle C$, *~spam(d)*, 0.6$\rangle$, where C = {(*~spam(d)* —< *contacts(d)*, 0.6)}.
– $\langle \mathcal{D}$, *~spam(d)*, 0.9$\rangle$, where $\mathcal{D}$ = {(*~spam(d)* —< *local(d)*, 0.9)}.

$\mathcal{B}$ defeats C but is defeated by $\mathcal{D}$. There are no more arguments to consider, and the resulting dialectical tree has only one argumentation line: $\mathcal{A}$ is defeated by $\mathcal{B}$ who is in turn defeated by $\mathcal{D}$. Hence, the marking procedure determines that the root node $\mathcal{A}$, is a U-node and the original query is warranted.

3. Dialectical graphs and pre-compiled knowledge

To obtain faster query processing in the OP-DeLP system we integrate pre-compiled knowledge to avoid the construction of arguments which were already computed before. The approach follows the proposal presented in [5] where the pre-compiled knowledge component is required to: (1) minimize the number of stored arguments in the pre-compiled base of arguments (for instance, using one structure to represent the set of arguments that use the same defeasible rules); and (2) maintain independence from the observations that may change with new perceptions, to avoid modifying also the pre-compiled knowledge when new observations are incorporated.

Considering these requirements, we define a database structure called dialectical graph, which will keep a record of all possible arguments in an OP-DeLP program P (by means of a special structure named potential argument) as well as the counterargument relation among them. Potential arguments, originally defined in [5] contain non-grounded defeasible rules, depending thus only on the set of rules in P and are independent from the set of observations.

Potential arguments have been devised to sum-up arguments that are obtained using different instances of the same defeasible rules. Recording every generated argument could result in storing many arguments which are structurally identical, only differing on the constants being used to build the corresponding derivations. Thus, a potential argument stands for several arguments which use the same defeasible rules. Attack relations among potential arguments can be also captured, and in some cases even defeat can be pre-compiled. In what follows we introduce the formal definitions, adapted from [5] to t the OP-DeLP system.

Definition 8. [Weighted Potential argument] Let Δ be a set of defeasible rules. A subset **A** of Δ is a potential argument for a literal Q with an upper bound γ for its certainty degree, noted as $\langle\langle A,Q,\gamma\rangle\rangle$ if there exists a non-contradictory set of literals ϕ and an instance $\mathcal{A}$ that is obtained finding an instance for every rule in **A**, such that $\langle A, Q, \alpha\rangle$ is an argument wrt (ϕ,Δ) $(\alpha \leq \gamma)$ and there is no instance $\langle \mathcal{B}, Q, \beta\rangle$ of **A** such that $\beta > \gamma$.

The nodes of the dialectical graph are the potential arguments. The arcs of our graph are obtained calculating the counterargument relation among the nodes previously obtained. To do this, we extend the concept of counterargument for potential arguments. A potential argument $\langle\langle A_1,Q_1,\alpha\rangle\rangle$ *counter-argues* $\langle\langle A_2,Q_2,\beta\rangle\rangle$ at a literal Q if and only if there is a non-empty potential sub-argument $\langle\langle A,Q,\gamma\rangle\rangle$ of $\langle\langle A_2,Q_2,\beta\rangle\rangle$ such that Q_1 and Q are contradictory literals.[1] Note that potential counter-arguments may or may not result in a real conflict between the instances (arguments) associated with the corresponding potential arguments. In some cases instances of these arguments cannot co-exist in any scenario (*e.g.*, consider two potential arguments based on contradictory observations). Now we can finally define the concept of dialectical graph:

Definition 9. [Dialectical Graph] Let $\mathcal{P}=(\Psi,\Delta)$ be an OP-DeLP program. The dialectical graph of Δ, denoted as G_Δ, is a pair (PotArg(Δ), C) such that: (1) PotArg(Δ) is the of all the potential arguments that can be built from Δ; (2) C is the counterargument relation over the elements of PotArg(Δ).

We have devised a set of algorithms to use the dialectical graph for improving the inference process. For space reasons these algorithms are not detailed in this work, the interested reader may consult [6] for a more detailed treatment of this subject. We have also compared the obtained algorithms theoretically with standard argument-based inference techniques (such as those used in P-DeLP).

At the inference process, we have found out that complexity is lowered from:

[1] Note that $P(X)$ and $P(X)$ are contradictory literals although they are non-grounded. The same idea is applied to identify contradiction in potential arguments.

$$O\left(2^{|\Delta'|^{3.(2^{|\Delta'|})/4}}\right) \text{ to } O(2^{|\Delta'|}.|\Delta'|).$$

4. A proposed architecture for OP-DeLP applications

Applications that use the OP-DeLP system will be be engineered for contexts where: (1) information is uncertain and heterogeneous, (2) handling of great volume of data ows is needed, and (3) data may be incomplete, vague or contradictory. In this section we present an architectural pattern that can be used in these applications.

Previous to proposing a pattern we started analyzing the characteristics of OP-DeLP applications. First, we found that data will generally be obtained from multiple sources. Nowadays the availability of information trough the Inter-net has shifted the issue of information from quantitative stakes to qualitative ones [3]. For this reason, new information systems also need to provide assistance for judging and examining the quality of the information they receive.

For our pattern, we have chosen to use a multi-source perspective into the characterization of data quality [3]. In this case the quality of data can be evaluated by comparison with the quality of other homologous data (i.e. data from different information sources which represent the same reality but may have contradictory values). The approaches usually adopted to reconcile heterogeneity between values of data are: (1) to prefer the values of the most reliable sources, (2) to mention the source ID for each value, or (3) to store quality meta-data with the data. We have chosen to use the second approach. In multi-source databases, each attribute of a multiple source element has multiple values with the ID of their source and their associated quality expertise. Quality expertise is represented as meta-data associated with each value. We have simplified this model for an easy and practical integration with the OP-DeLP system. In our case, data sources are assigned a unique certainty degree. For simplicity sake, we assume that different sources have different values. All data from a given source will have the same certainty degree. This degree may be obtained weighting the plausibility of the data value, its accuracy, the credibility of its source and the freshness of the data.

OP-DeLP programs basically have a set of observations and a set of rules. The set of rules is chosen by the knowledge engineer and remains fixed. The observation set may change according with new perceptions received from the multiple data sources. Nevertheless, inside the observation set we will distinguish a special kind of perceptions, those with certainty degree 1. Those perceptions are also codified by the knowledge engineer and cannot be modified in the future by the perception mechanism. To assure this, we assume that every data source has a certainty value such that $0 < \gamma < 1$.

Example 3. Consider the program in Example 2. In this case data establishing a given message is from the local host comes from the same data source and can be given a

certainty degree of 1. The same applies for *contacts(X)*. The algorithm that decides whether to filter a given message is another data source with a degree of 0.9, the filter that classifies a message as a virus is another data source with a degree of 0.7, and the algorithm that checks if the message came from some server in the blacklist is a different source that has a degree of 0.75. Note that we could have different virus filters with different associated certainty degrees if we wanted to build higher trust on this filter mechanism.

The scenario just described requires an updating criterion different to the one presented in [5], given that the situation regarding perceptions in OP-DeLP is much more complex. To solve this, we have devised Algorithm 1, that summarizes different situations in two conditions. The first one acts when the complement of the literal Q is already present in the set. Three different cases can be analyzed in this setting: (1) If both certainty degrees are equal it means that both *Q* and its complement proceed from the same data source. Then the only reason for the conflict is a change in the state of affairs, thus an update is needed and the new literal is added. (2) If $\alpha > \beta$ it means that the data sources are different, Thus we choose to add (Q, α) since it has the higher certainty degree. (3) If $\alpha < \beta$ we keep (comp(Q), β) (the complement of Q wrt strong negation). Note that (1) is an update operation [12] while (2) and (3) are revisions over. The difference between updating and revision is fundamental. Updating consists in bringing the knowledge base up to date when the world changes. Revision allows us to obtain new information about a static scenario [12].

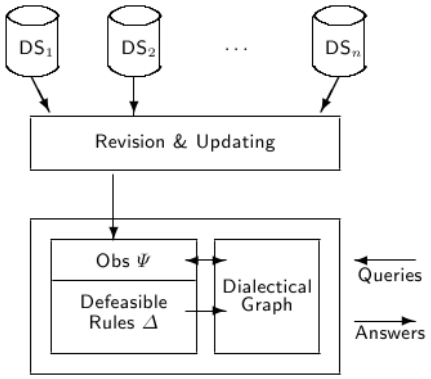


Figure 2: *Architecture for applications using OP-DeLP as underlying framework*

The second condition in Algorithm 1 considers the case when Q was in with a different certainty degree. Then it chooses the weighted literal with the highest degree possible. Note that the observations initially codified that have a certainty degree of 1 cannot be deleted or modified by algorithm 1.

Algorithm 1. UpdateObservationSet
Input: $\mathcal{P}$=(Ψ,Δ) , (Q, α)
Output: $\mathcal{P}$=(Ψ,Δ) (With Ψ updated)
If there exists a weighted literal (comp(Q), β) in Ψ such that $\beta \leq \alpha$ **Then**

delete((comp(Q), β))
add((Q, α))

If there exists a weighted literal (Q, β) in Ψ such that $\alpha \leq \beta$ **Then**
delete(Q, β))
add((Q, α))

Finally, fig. 2 summarizes the main elements of the O-DeLP-based architecture. Knowledge is represented by an OP-DeLP program $\mathcal{P}$. Perceptions from multiple sources may result in changes in the set of observations in $\mathcal{P}$, handled by the updating mechanism defined in algorithm 1. To solve queries the OP-DeLP inference engine is used. This engine is assisted by the dialectical graph (Def. 9) to speed-up the argumentation process. The final answer to a given query Q will be yes, with the particular certainty degree of the warranted argument supporting Q, or no if the system could not find a warrant for Q from $\mathcal{P}$.

5. Conclusions

In this work we have defined an argumentation-based formalism that integrates uncertainty management. This system was also provided with an optimization mechanism based on pre-compiled knowledge. Using this, the argumentation system can comply with real time requirements needed to administer data and model reasoning over this data in dynamic environments. Another contribution is the architectural model to integrate OP-DeLP in practical applications to administer and reason with data from multiple sources.

As future work, we are developing a prototype based on the proposed architecture to extend the theoretical complexity analysis with empirical results and to test the integration of the OP-DeLP reasoning system in real world applications. We are also working on the integration of OP-DeLP and database management systems by means of a strongly-coupled approach.

References

1. Alsinet, T., Chesñevar, C. I., Godo, L., Sandri, S. and Simari, G. R., Formalizing argumentative reasoning in a possibilistic logic programming setting with fuzzy uni cation, International Journal of Approximate Reasoning, 48 (3), 2008.
2. Alsinet, T., Chesñevar, C. I., Godo, L., Sandri, S. and Simari, G. R., A logic programming framework for possibilistic argumentation: Formalization and logical properties, Fuzzy Sets and Systems, 159 (10), 2008.
3. Berti, L., Quality and recommendation of multi-source data for assisting technological intelligence applications, in Proc. of 10th International Conference on Database and Expert Systems Applications, Italy, AAAI, 1999.

4. Bryant and Krause, An implementation of a lightweight argumentation engine for agent applications, Lecture Notes in Computer Science, 4160 (1), 2006.
5. Capobianco, M., Chesñevar, C. I. and Simari, G. R., Argumentation and the dynamics of warranted beliefs in changing environments, Journal of Autonomous Agents and Multiagent Systems, 11, 2005.
6. Capobianco, M. and Simari, G., A proposal for making argumentation computationally capable of handling large repositories of uncertain data, in Proceedings of the third international conference on scalable uncertainty management, 2009.
7. Chesñevar, C. I., Maguitman, A. G. and Loui, R. P., Logical Models of Argument, ACM Computing Surveys, 32 (4), 2000.
8. Chesñevar, C. I., Maguitman, A. G. and Loui, R. P., Argument-based critics and recommenders: A qualitative perspective on user support systems. Data & Knowledge Engineering, 59(2):293-319, 2006.
9. Chesñevar, C. I., Simari, G. R., Alsinet, T. and Godo, L., A logic programming frame-work for possibilistic argumentation with vague knowledge, in Proc. of Uncertainty in Artificial Intelligence Conference (UAI 2004), Ban, Canada, 2004.
10. García, A. and Simari, G., Defeasible Logic Programming: An Argumentative Approach, Theory and Practice of Logic Programming, 4 (1), 2004.
11. Gelfond, M. and Lifschitz, V., Classical negation in logic programs and disjunctive databases, New Generation Computing, 1991.
12. Katsuno, H. and Mendelzon, A., On the difference between updating a knowledge base and revising it, in P. Gardenfors (editor), Belief Revision, Cambridge University Press, 1992.
13. Prakken, H. and Vreeswijk, G., Logical systems for defeasible argumentation, in Handbook of Philosophical Logic, volume 4, 2002.
14. Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S. and Sonenberg, L., Argumentation-based negotiation, The Knowledge Engineering Review, 18 (4), 2003.
15. Simari, G. R. and Loui, R. P., A Mathematical Treatment of Defeasible Reasoning and its Implementation, Artifciial Intelligence, 53 (1-2), 1992.

An Immune Artificial Algorithm for Dynamic Optimization Problems: A Case of Study

VICTORIA S. ARAGÓN, SUSANA C. ESQUIVEL[1], CARLOS A. COELLO COELLO[2]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional*
Universidad Nacional de San Luis
Ejército de los Andes 950 (5700) San Luis, ARGENTINA
{vsaragon, esquivel}@unsl.edu.ar

[2] CINVESTAV-IPN (Evolutionary Computation Group)†
Departamento de Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco México D.F. 07300, MÉXICO
ccoello@cs.cinvestav.mx

Abstract. *In this paper, we present an algorithm inspired on the T-Cell model of the immune system, i.e., an artificial immune system (AIS). The proposed approach (called DTC) is intented to solve dynamic optimization problems. We realize a first study and we validate our algorithm using dynamic functions taken from the specialized literature. Results are promising when we compare them with respect to the results obtained by five AIS representative of the state of the art in this field.*

Keywords: *Artificial immune systems, dynamic optimization, heuristics optimization.*

1. Introduction

In general, the conditions of an optimization problem changes by one of the following reasons or a combination of both [1]: 1) The objective function changes itself, 2) The constraints change. In this paper the firts problematic is addressed.

In recent years, a bio-inspired metaheuristic known as the "artificial immune system" (AIS) has gained popularity in a wide variety of tasks [12]. The AIS is inspired on our natural immune system, which has a number of very interesting features, from a computational point of view, that make it a very good candidate to be modelled in a computer. For example, it is a distributed system, it is fault- tolerant, it has memory, it is able to distinguish between its own components and those which are foreign, and it learns by experience. AISs have been used for solving dynamic optimization problems (see for

* LIDIC is financed by Universidad Nacional de San Luis and ANPCyT (Agencia Nacional para promover la Ciencia y Tecnología).
† The third author acknowledges support from CONACyT project No. 45683-Y.

example [16, 7, 13]), but in most cases, it has been applied only to global optimization problems. In this paper, we precisely focus on solving dynamic optimization problems with an AIS that we have proposed, and which we believe that can be a viable alternative for solving these kind of problems.
The remainder of the paper is organized as follows. In Section 2, we describe the benchmark used to generate the dynamic functions. Section 3 describes the existing AIS for dynamic optimization. In Section 4, we describe our AIS, which is based on the T-Cell Model. In Section 5, we present our experimental setup, our results and we discuss them. Finally, in Section 6, we present our conclusions and some possible paths for continuing with this research line.

2. Moving Peaks Benchmark (MPB)

Moving Peaks Benchmark was proposed in [2]. MPB defines a dynamically changing fitness landscape $f : X \times T \longrightarrow \Re$, where T stands for the (discrete) time, and $X = x_1, \ldots, x_5$ is the set of admissible solutions. The landscape is build of a set of peaks (scenario 1) or cones (scenario 2). Every i^{th} peak or cone has its height h_i, width w_i, and the coordinates of its maximum $cmax_i$. All the parameters characterizing each peak are generated randomly from the corresponding interval. The fitness function for i^{th} peak is evaluated as follows:

$$f_i(x_1,...,x_5) = \frac{h_i}{1 + w_i \prod_{j=1}^{5} (x_j - c\max[j])^2}$$

while the ecuation for an i^{th} cone is:

$$f_i(x_1,...,x_5) = h_i - w_i \sqrt{\prod_{j=1}^{5} (x_j - c\max[j])^2}$$

Then the value of the overall fitness function $f(x_1, \ldots, x_5)$ is computed as: $f(x_1, \ldots, x_5) = \max_{i=1,\ldots,N} f_i(x_1, \ldots, x_5)$, where N is a number of peaks or cones defined in the scenario. For such a fitness landscape, all the three features of each peak or cone can be modified to perform landscape changes [16].

3. Previous Related Work

Gaspar et. al [7] proposed a Simple Artificial Immune System (Sais). Sais starts with an initial random population of B-Cells, each able to detect a given

antigen specified by a binary bits long string. Then, it applies at each generation three operators: Evaluation, Clonal Selection and Recruitment (elimination of undesirable B-cells). Sais was validated with pattern tracking problem.

Trojanowski K. in [14] analyze the efficiency of two mutation operators applied in a clonal selection based optimization algorithm (AIIA) for non-stationary tasks. Both operators use a α-stable random number generator. The author argues that appropriate tuning of the α parameter allows to outperform the results of algorithms with the traditional operators. The algorithms were tested with six environments generated with two test-benchmarks: a Test Case Generator and a Moving Peaks Benchmark.

Nanas et. al [10] compared Evolutionary Algorithms and Artificial Immune Systems under Multimodal Dynamic Optimization. They review the basic evolutionary and immune-inspired approaches to multimodal dynamic optimization and they identify correspondences and differences and point out essential computational elements.

Trojanowski K. in [13] analyze the efficiency of the B-Cell algorithm applied to Moving Peaks Benchmark. The algorithm starts with a population of solutions randomly generated and performs the process of iterated improvement of the solutions by the repetition of: 1) affinity evaluation and 2) clonal selection and expansion.

Trojanowski K. et. al compared in [16] five instances of AISs: 1) Artificial Im- mune Iterated Algorithm (AIIA) [15], 2) B-Cell Algorithm (BCA) [8], 3) Clonal Selection Algorithm (CLONALG) [4], 4) opt-Ainet algorithm [11], and a Simple Artificial Immune System (Sais) [7]. All of them implement non-deterministic iterated process of search and all of them work with a population of antibodies or B-cells. These represent candidate solutions to the problem. The coordinates of these points are represented by real numbers or can be coded as bit strings. Each algorithm starts with a population of randomly generated tentative solutions which are iteratively improvement. The authors tested those approaches with seven types of mutations. M_1 to M_7 (see [16] for details). The algorithms were tested with six environments generated with two test-benchmarks: a Simple Test Case Generator and a Moving Peaks Benchmark.

4. T Cell Theory

In this paper, we present what we believe to be a new adaptive immune system model based upon the immune responses mediated by the T-cells. Our model is called TCELL, and it considers many of the processes that the T cells suffer from their origin in the hematopoietic stem cells in the bone marrow until they become memory cells.

T cells belong to a group of white blood cells known as lymphocytes. They play a central role in cell-mediated immunity. They present a special receptor on their cell surface called T cell receptors (TCR[3]).
T cells can be classified in different populations according to the antigen receptor they express, TCR-1 or TCR-2. Additionally, TCR-2 cells express CD4 or CD8[4].
Also, T cells can be divided into three groups according to their maturation or development level (phylogenies of the T cells [5]). Virgin cells are those which had never been activated (i.e., they did not suffer proliferation or differentiation). At the beginning, these cells do not express CD4 nor CD8. However, later on, they develop and express both marks, CD4 and CD8. Finally, virgin cells mature and express only one mark, either CD4 or CD8. Before these cells release the thymus, they are subject to both positive selection [6] and negative selection [6]. Positive selection guarantees that the only survivors are the cells with TCRs that present a moderate affinity with respect to the self MHC. Negative selection eliminates the cells with TCRs that recognize self components unrelated to the MHC.
Effector cells are a type of cells that express only one mark, CD4 or CD8. They can be activated by co-stimulating signals plus their ability to recognize an antigen [3, 9]. The immune cells interact through the secretion of cytokines[5].
Cytokines allow cellular communication. Thus, an immune cell c_i influences the activities (proliferation and differentiation) of another cell c_i by the secretion of cytokines, modulating the production and secretion of cytokines by c_i [5]. In order to activate an effector cell, a co-stimulated signal is necessary. Such signal corresponds to the cytokines secreted from another effector cell. The activation of an effector cell implies that it will be proliferated and differentiated. The proliferation process has as its goal to replicate the cells and the differentiation process changes the clones in order to they acquire specialized functional properties.
Finally, the memory cells are cells that persist into the host even when the infection or danger have been overtaken, so that in the future, they are able to get stimulated by the same or by a similar antigen. It is worth noting that, although the effector and memory cells are proliferated, they are not subject to somatic hypermutation. For the effector cells, the differentiation process is subject to the cytokines released by another effector cell. In our model, the differentiation process of the memory cells relies on their own cytokines.

[3] TCRs are responsible for recognizing antigens bound to major histocompatibility complex (MHC) molecules.
[4] Lymphocytes express a large number of surface molecules that can be used to mark different cellular populations. CD means Cluster Denomination and indicates the group to which lymphocytes belong.
[5] Proteins act as signal transmitters between cells, and also induce growth, differentiation, activation, etc.

4.1 Our Proposed Algorithm Based on TCELL

DTC (Dynamic T-Cell) is an algorithm inspired on our TCELL model, which we propose to solve dynamic optimization problems. DTC operates on four populations, corresponding to the groups in which the T-cells are divided: (1) Virgin Cells (VC), (2) Effector Cells with cluster denomination CD4 (CD4), (3) Effector Cells with cluster denomination CD8 (CD8) and (4) Memory Cells (MC). Each population is composed by a set of T-cells whose characteristics are subject to the population to which they belong.

Virgin Cells (VC) do not suffer the activation process. They have to provide diversity.Virgin cells are represented by 1) a TCR (TCR b): represented by a bitstring using Gray coding and 2) a TCR (TCR r): represented by a vector of real numbers.

Effector Cells are composed by 1) a TCR b or TCR r, if belongs to CD4 or CD8, respectively, 2) a proliferation level and 3) a differentiation level. The goal of this type of cell is to explore in a global way the search space.

The goal of memory cells is to explore the neighborhood of the best found solutions. These cells are represented by the same components that CD8.

For our propose the TCR identifies the decision variables of the problem, independently of the TCR representation. The proliferation level indicates the number of clones that will be assigned to a cell and the differentiation level indicates the number of bits or decision variables (according with the TCR representation) that will be changed, when the differentiation process is applied.

The activation of an effector cell, ce_i, implies the random selection of a set of potential activator (or stimulating) cells. The closer one to ce_i, using Hamming or Euclidean distance according to the TCR, in the set is chosen to become the stimulating cell, say ce_i. Then, ce_i proliferates and differentiates.

At the beginning, the proliferation level of each stimulated cell, ce_i, is given by a random value between [1, 5], but then it is determined taking into account the proliferation level of its stimulating cell (ce_i). If the ce_i is better than ce_i, then ce_i keeps its own proliferation level; otherwise, ce_i receive a level which is 10% lower than level of ce_i.

Memory cells proliferate and differentiate according to their proliferation level (random between 1 and the size of MC) and differentiation level (random between 1 and the 90% of the number of decision variables), respectively. Both levels are independent from the others memory cells.

Each type of cell has its own differentiation process, which is blind to their representation and population.

Differentiation for CD4: the differentiation level is determined by the Hamming distance between the stimulated (ce_i) and stimulating (ce_i) cells. Each decision variable and the bit to be changed are chosen in a random way. The bit changes according to a mutation (or reaction) probability $prob_{mut-C\,D4}$.

Differentiation for CD8: the differentiation level from each cell is related to its stimulating cell (ce_i). If ce_i is better than the stimulated cell ce_i, then the level (for ce_i) is a random between [1, |dv|]; otherwise is a random between [1, |dv|/2], where |dv| is the number of decision variables of the problem.

Each variable to be change is chosen in a random way and it is modified according to the following expression: $x' = x \pm U(0, lu - ll)^{U(0,1)}$, where x and x' are the original and the mutated decision variables, respectively. lu and ll are the upper and lower bounds of x, respectively. At the moment of the differentiation of a cell (ce_i), the value of the objective function of its stimulating cell (ce_i) is taken into account. In order to determinate if $r = U(0, lu - ll)^{U(0,1)}$, will be added or subtracted to x, the following criteria are considered: 1) if ce_i is better than ce_i and the decision variable value of ce_i is less than the value of ce_i, or if ce_i is better than ce_i and the decision variable value of ce_i is less than the value of ce_i, then r is subtracted to x; otherwise r is added to x. Both criteria are motivated by our aim to guide the search towards the best solutions found so far.

Differentiation for MC: Each variable to be change is chosen in a random way and it is modified according to the following expression: $x' = x \pm (U(0, lu-ll)/100iter)^{U(0,1)}$, where x and x' are the original and the mutated decision variables, respectively. lu and ll are the upper and lower bounds of x. iter indicates the number of iterations until reach the amount maximum of evaluations for a change. In a random way, we decide if $r = (U(0, lu-ll)/100iter)^{U(0,1)}$ will be added or subtracted to x.

In both differentiation processes U(0,w) refers to a random number with a uniform distribution in the range (0,w).

The general structure of our proposed algorithm for dynamic optimization problems is given in Algorithm 1.

Algorithm 1 DTC Algorithm

```
1: Initialize Function();
2: Initialize-Evaluate VC();
3: Assign Proliferation();
4: Divide CDs();
5: Positive Selection CD4 CD8(); //eliminate the cells in CD4 and CD8 with
   worst objective function value
6: Negative Selection CD4 CD8(); //eliminate the most similar cells in CD4
and CD8
7: while Repeat a predetermined number of change do
8:  while Repeat a predetermined number of evaluations do
9:        Active CD4();
10:       Sort CD4();
11:       Active CD8();
12:       Sort CD8();
13:       Insert CDs en MC();
14:       while Repeat a predetermined number of times (repMC) do
15:           Active MC();
```

```
16:        end while
17:        Sort CM();
18:   end while
19:     Statistics();
20:     Change Function();
21:     Re-evaluate Populations();
22: end while
```

The algorithm works in the following way. At the beginning TCR b and TCR r from virgin cells are initialized in a random way. Then, they are evaluated. The negative and positive selections are applied, the first eliminates the similar cells (keeping the best cells) and second eliminate a 10% of the worst cells. Next, the best virgins cells are divided in order to compose the populations CD4 (TCR b) and CD8 (TCR r).

Once the CD4 and CD8 populations have been activated (proliferation and differentiation) the best solutions from these populations are inserted or replace the worst solutions in MC (if MC is empty or not, respectively). When a cell from CD4 has to be inserted into MC, the cell first has to be converted in its real-value vector through the application of the following equation: $dv_j = l_{lj} + (\sum_{i=0}^{Lj} 2^{Lj-i} dv'_{ij}(l_{uj} - l_{lj}))/(2_j^L - 1)$ where dv_j is the j^{th} decision variable with j = 1, . . . , number of decision variables of the problem, Lj is the number of bits for the j^{th} decision variable, l_{uj} and l_{lj} are the upper and lower limits for the decision variable dv_j and dv'_{ij} is the i^{th} -bit of the binary string that represents dv_j.

Next, the cells from MC are activated a predefined number of times (rep MC).

The algorithm finishes when a predefined number of evaluations for each change is reached. We assume that the algorithm knows when the environment has changed, in order to re-evaluate the populations.

5. Numerical Experiments

In order to evaluate the performance of the algorithm we use the measure [16] Offline error (oe), it represents the average deviation of the best individual evaluated since the last change from the optimum. It is defined by:

$$oe = \frac{1}{\mathrm{Nc}} \sum_{j=1}^{Nc} \left(\frac{1}{Ne(j)} \sum_{i=1}^{Ne(j)} \left(f_i^* - f_{ji}^* \right) \right)$$

where N_c is the number of fitness landscape changes within a single experiment, $N_e(j)$ is the number of solution evaluations performed for the j^{th} state of the landscape, f^*_j is the value of the optimal solution for the j^{th} landscape and f^*_{ji} is the current best fitness value found for the j^{th} landscape [16].

To validate DTC, we use two fitness landscapes created with MPB. Each landscape consist of a number of peaks, changing in a random way their height, width and location. The two environments were used according to standard settings given in the web page[6]: scenarios 1 (5 dimensions - 5 peaks) and 2 (5 dimensions - 50 peaks, see the web page for details). For each scenario the function changes every 5000 evaluations of the objective function.
The required parameters for DTC are: size of VC, CD4, CD8 and MC; number of repetitions MC (rep MC); percentage of replacement for MC; Probability of mutation $prob_{mutC\ D4}$. For the experiments, we adopted the following parameters, which were empirically derived alter numerous experiments, we used a population size, for VC, of 100 cells. For CD4 and CD8 we adopted a population size of 50 and 70 cells, for scenario 1 and 2 respectively. For MC 5 cells. The mutation probability $prob_{mut}$ was 0.07. rep MC was set to 10 and 100, for scenario 1and 2 respectively. Finally, 50% replacement was adopted for replacing from CDs to MC. 50 independents runs and 110 change of the objective function were performed for each scenario. For both scenarios we use a binary Gray code (for VC and CD4) with 40 bits for each decision variable.
Our results were compared respect to the results obtained for the best combination AIS-mutation operator presented in [16] for each scenario, they are: AIIA-M6-α=2.0, CLONALG-M2, Sais-M3 and opt-Ainet -M2 for scenario 1 and AIIA-M5 - α = 1.5, CLONALG-M3, Sais-M2 and opt-Ainet - M1 for scenario 2. BCA- M5- α = 1.75, for scenario 1 and - α = 2.0 for scenario 2.

5.1 Analysis of Results

Comparing our proposed DTC with respect to the AIS presented in [16] (see Table 1), our approach obtained better results in scenario 1and competitive results for scenario 2.
Fig. 1a shows the average of the best found solutions, of each population (CD4, CD8 and MC) and the optimum before the objective function change, for scenario 1. Here, we can see that the solutions found by CD4 are better than the solutions found by CD8. MC performs a good local search for the solutions provide by CD4. The performance of DTC does not deteriorate in the presence of a change. While for scenario 2 (see Fig. 1b), we can observe that both populations, CD4 and CD8, provide good solutions in order to MC find solutions near the optimum and its performance is affected. This fact makes us think that scenario 2 requires more diversity due to the number of peaks, fifty peaks while scenario 1 has only five peaks.
Fig. 1c shows the average of the offline error, over the 50 runs, for both scenarios. The presence of a change affects the performance of DTC worst in scenario 2 than scenario 1.

[6] http://www.aifb.uni-karlsruhe.de/ jbr/MovPeaks/movpeaks/.

Table 1: *Offline Error obtained for each Approach.*

Approach	Scenario 1	Scenario 2
AIIA	0.71	3.44
CLONALG	11.71	10.53
Sais	12.40	11.57
BCA	0.39	2.69
opt-Ainet	2.39	4.76
DTC	0.37	3.02

6. Conclusions and Future Work

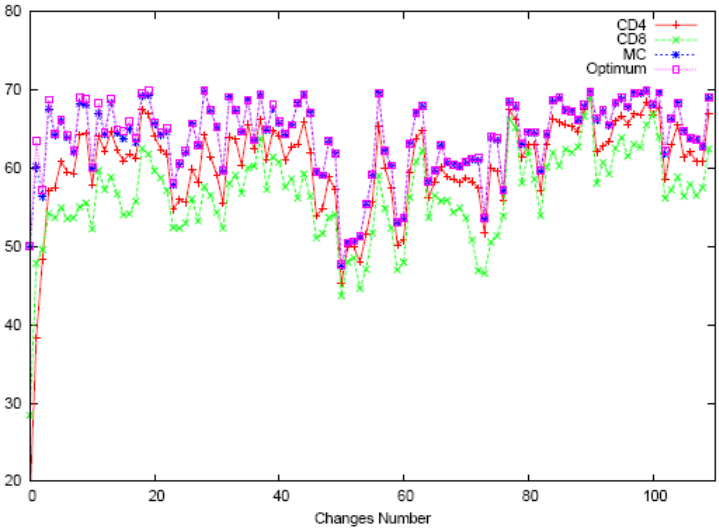
We have presented an artificial immune system based on the T-Cell model, which has been proposed to solve dynamic optimization problems. The proposed approach is inspired on the processes suffered by the T-Cells within our immune system. In this first study, the proposed approach has been tested with two scenarios generated by a generator of dynamic functions (MPB). The results obtained by our proposed approach are promising, resulting better in one of the cases to those generated by the other algorithms with respect to which it was compared. This fact encourages us, as future work, to deep this study over different kind of dynamic environments.

References

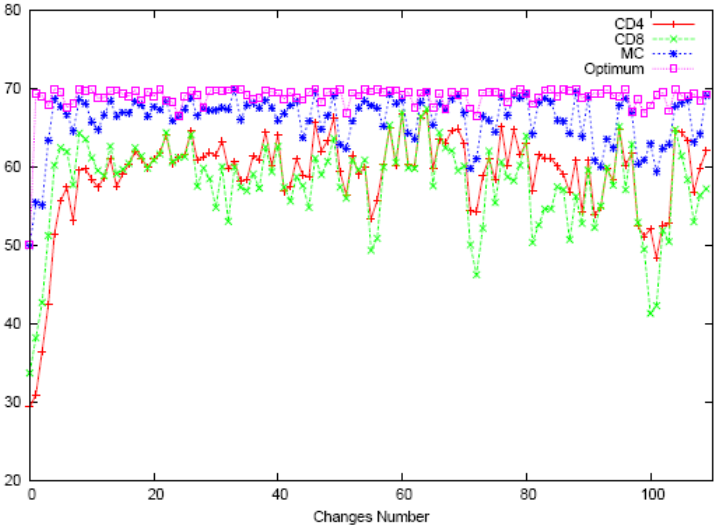
1. How to Solve It: Modern Heuristics, Springer, December, 2004.
2. Branke, J., "Memory enhanced evolutionary algorithms for changing optimization problems", in Congress on Evolutionary Computation CEC99, IEEE, 1999.
3. Bretscher, P. and M. Cohn, A theory of self-nonself discrimination, Science, 169, 1970.
4. Nunes De Castro, L. and Von Zuben, F. J., The clonal selection algorithm with engineering applications. In GECCO 2002 - Workshop Proceedings, Morgan Kaufmann.
5. Dasgupta, D. and Nino, F., Immunological Computation: Theory and Applications, Auerbach Publications, Boston, MA, USA, 2008.
6. David, B. and Roth, D. M., Brostoff, J. and Roitt, I., Inmunologia, Else-vier Science Health Science div, Madrid, 7a edition, Harcourt Brace, 1997.
7. Alessio, G. and Collard, P., From gas to artificial immune systems: Improving adaptation in time dependent optimization. In Proceedings of the Congress on Evolutionary Computation, IEEE Press, 1999.
8. Kelsey, J. and Timmis, J., Immune inspired somatic contiguous hypermutation for function optimisation, In E. Cantu-Paz, J. A. Foster, K. Deb, L. Davis, R. Roy, U. O'Reilly, H. Beyer, R. K. Standish, G. Kendall,

S. W. Wilson, M. Har- man, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. F. Miller, editors, Genetic and Evolutionary Computation- GECCO 2003, Genetic and Evolutionary Computation Conference, Chicago, Illinois, USA, July 2003. Springer. Lecture Notes in Computer Science Vol. 2723.
9. Matzinger, P., Tolerance, danger and the extend family. Annual Review of Immunology, 12, April 1994.
10. Nanas, N. and De Roeck, A. N., Multimodal dynamic optimization: From evolutionary algorithms to artificial immune systems, in ICARIS, 2007.
11. Nunes de Castro, L. and Timmis, J., An artificial immune network for multimodal function optimization, in Proccedings of the 2002 Congress on Evolutionary Computation (CEC'2002), volume 1, Honolulu, Hawaii, May 2002.
12. Nunes de Castro, L. and Timmis, J., Artificial Immune Systems: A New Computational Intelligence Approach, Springer-Verlag, New York, 2002.
13. Trojanowski, K., B-cell algorithm as a parallel approach to optimization of moving peaks benchmark tasks. Computer Information Systems and Industrial Management Applications, International Conference on, 0, 2007.
14. Trojanowski, K., Clonal selection approach with mutations based on symmetric alpha stable distributions for non-stationary optimization tasks, in ICAN-NGA '07: Proceedings of the 8th international conference on Adaptive and Natural Computing Algorithms, Part I, Berlin, Heidelberg, Springer-Verlag, 2007.
15. Trojanowski, T. and Wierzchon, S. T., Studying properties of multipopulation heuristic approach to non-stationary optimisation tasks, in IIS, 2003.
16. Trojanowski, K. and Wierzchon, S. T., Immune-based algorithms for dynamic optimization, Inf. Sci., 179(10), 2009.

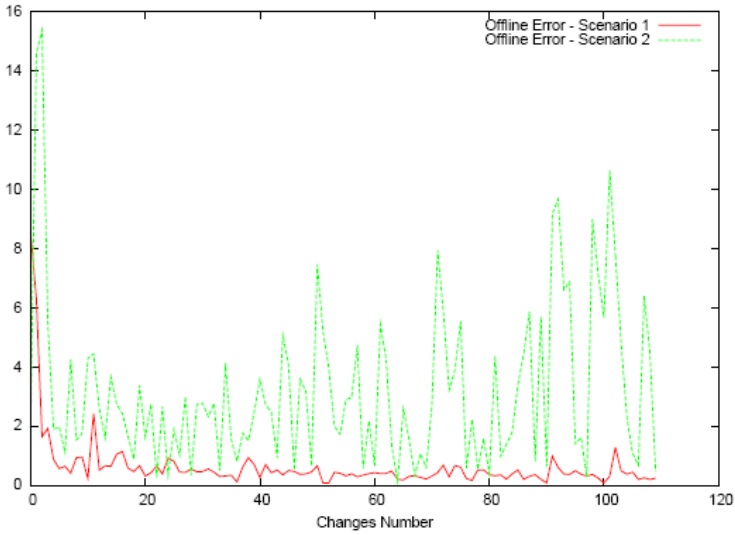
Figure: *Optimum, Best Solutions and Offline Errors for DTC.*



a. Optimum and Best Solutions Before a Change Found by CD4, CD8 and MC for Scenario 1



b. Optimum and Best Solutions Before a Change Found by CD4, CD8 and MC for Scenario 2



c. Offline Error Obtained by DTC

GA and PSO Applied to Wind Energy Optimization

MARTIN BILBAO[1], ENRIQUE ALBA[2]

[1] Universidad Nacional de la Patagonia Austral, Caleta Olivia, Argentina
mbilbao@uaco.unpa.edu.ar
[2] Universidad de Málaga, España
eat@lcc.uma.es

Abstract. *In this article we analyze two kinds of metaheuristic algorithms applied to wind farm optimization. The basic idea is to utilize CHC (a sort of GA) and GPSO (a sort of PSO) algorithms to obtain an acceptable configuration of wind turbines in the wind farm that maximizes the total output energy and minimize the number of wind turbines used. The energy produced depends of the farm geometry, wind conditions and the terrain where it is settled. In this work we will analyze three study farm scenarios with different wind speeds and we will apply both algorithms to analyze the performance of the algorithms and the behavior of the computed wind farm designs.*

Keywords: *CHC, Geometric Particle Swarm Optimization, Optimization, Wind Energy, Metaheuristics.*

1. Introduction

Nowadays, using renewable energies is an increasing area of research and development in all the world, because they are important alternatives to generate free and clean power. The raise of this energy is clear in Europe and America, being a strategic part of development for many countries like Argentina and Spain. A capital interest resides in combining a maximum of energy generation at the same time as reducing the total cost of the wind farm. A farm is a set of wind turbines, every one being costly, whose position is a strategic decision in orden to maximize the produced energy. One of the most important aspects of wind farm design is to obtain an optimal location of the wind turbines, because they recieve lower wind speed and less energy captures if e.g. they are located behind each other. This effect is called *the wake effect* [1]. The wake effect can be reduced by optimizing the geometry of the wind farm. Then, obtaining a maximum annual profit means taking into account the number of wind turbines and their proper positioning simultaneously. Therefore, an effective algorithm is necessary to get an optimal solution by using a mathematical model of the wind farm as close as possible to a real world complex problem.
Simulated Annealing and Distributed Genetic Algorithms have been used in the past to solved this kind of problem [2][3]. In this work we use other techniques that have provided in the past good solutions in problems like

RND (Radio Network Design) that share some points in common to our work [4] and Geometric Particle Swarm Optimization (GPSO) [5] will be applied and analyzed here, showing also that they can provide new state of the art solutions to optimal wind farm design applications.

The rest of the article is structured as follows: Section 2 we will explain the wake model, the power model and the cost model used. Section 3 will detail CHC and GPSO the proposed algorithms. In section 4 we will show the experimental studies and discuss on the results obtained and in Section 5 the conclusions and future work.

2. Wind Farm Modelling

In this section we describe the mentioned inter-turbine wake effect model, the power model, and the cost model for our further mathematical manipulations. This are the basic components to deal with a realistic farm design, and they are combine together for the needed guidance offered to the design algorithms in their quest for an optimal farm configuration.

2.1 Wake Effect Model

The used model in this work is similar to the wake decay model developed by Katic [6]. Depending of the farm geometry, the wind turbine that is upwind of other wind turbine results in lower wind speeds than the one downwind, as shown in Fig. 1. The *velocity deficit* measures this effect [6]:

$$dV = U_0 - U_t = U_0 \frac{1-\sqrt{1-C_t}}{(\frac{1+2kX}{D})^2}, \quad (1)$$

where U_0 is the initial free stream velocity, U_t is the velocity in the wake at a distance *X* downstream of the upwind turbine, C_t is the thrust coefficient of the turbine, *D* is the diameter of the upwind turbine, and *k* is the wake decay constant. This model assumes that the kinetic energy deficit of interacting wakes is equal to the sum of the energy deficits of the individual wakes. Thus, the velocity deficit at the intersection of several wakes is:

$$U_t = U_0 \left[1 - \sqrt{\sum_{i=1}^{N} (1 - \frac{U_i}{U_0})^2} \right], \quad (2)$$

where U_i is the free strem velocity of the individual wake, and N is the number of wind turbines in the wind farm.

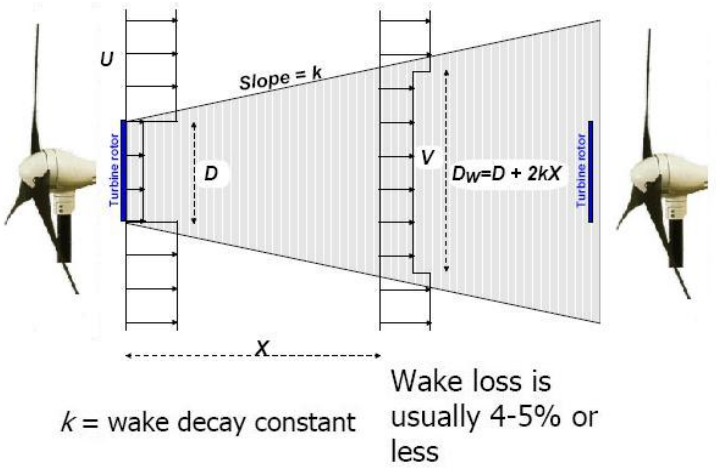


Figure 1: *Wake model for interaction between two wind turbines*

2.2 Power Model

The wake model directly defines the power model, that is to be maximized. The power curve for the wind turbine under consideration in our work follows here:

$$P_i(kW) = \begin{cases} 0 & \text{for} U_x < 3m/s, \\ 0.3U_x^3 & \text{for} 3m/s \le U_x < 13m/s, \\ 750 & \text{for} 13m/s \le U_x \le 25m/s, \\ 0 & \text{for} 25m/s < U_x \end{cases} \quad (3)$$

where Ux is the wind speed on the wind turbine.

the total power generation for all the wind turbines in the wind farm is:

$$P_{tot} = \sum_{i=1}^{N} P_i, \quad (4)$$

where N is the total number of wind turbines.

2.3 Cost Model

In our case, only the number of wind turbines influences the total cost to be minimized. The total cost per year for the entire wind farm, assuming a predefined and constant number of wind turbines, can be expressed as follows:

$$cost_{tot} = cost_{gy} N(2/3 + 1/3e^{-0.00174N^2}), \quad (5)$$

where $cost_{gy}$ represents the cost per wind turbine per year, and its value in this work is € $400{,}000$.

3. CHC and GPSO Algorithms

In this section we will explain the algorithms that we will use to solve the optimization problem of optimally design a wind farm. We have selected two well-known algorithms, a good feature found in a previous work [2][3].

3.1 CHC

The CHC algorithm was designed to work with populations coded as binary strings. CHC is a type of genetic algorithm that does not use mutation to produce new solutions; insteads it uses a mechanism called *HUX* crossover. The selection of individuals to complete the next generation is under only an elitist approach between parents and children.The *R* best solutions are retained and will be present in the next generation. When stagnation in the population is detected, a cataclysmic method of restart is used. The population tends to be homogeneous due to the absence of mutation and the elitist approach because there is no diversity; in order to solve this problem CHC implements a mechanism called *incest prevention*. The parents are selected randomly, but crossover takes place only if the individuals are not too close between them (Hamming distance) exceeds a certain threshold called *the threshold of incest*. As the population evolves, fewer individuals have the condition of not incest; in this case it is necessary to reduce the threshold. Every time that no change appears in the population (after one iteration) the threshold reduces in one unit.
The mechanism of crossover HUX also preserves diversity. This crossover copies in the two offspring all bits matched in both parents, and then copies half bits different in each offspring, such the Hamming distance between children and between children and parents is high. Once that the threshold of incest is 0, if *q* iterations pass without any new solution has entered the population, it means that the population has converged and the algorithm has stagnated, thus requiring a restart. All individuals except the best are modified by

a mutation by bit inversion with very high probability (in our case is 50%). Fig.2 shows an example of crossover HUX. It generates a mask with the common bits from the parents and non-common bits are assigned randomly to each child taking into account that each one must take half of the bits not common.

The pseudocode of the CHC algorithm is shown in Algorithm 1.

Algorithm 1 CHC

$t \leftarrow 0$; /* evaluation */
$initialize(Pa, Distance)$ /*Initialize the population and the distances */
while $not\ stop\ criterion(t, Pa)$ **do**
$Parents \leftarrow selected(Pa)$; /* Selected parent */
$Offspring \leftarrow HUX(Parents)$ /* Crossover HUX */
$evaluate(Pa, Offspring)$ /*evaluate Offspring*/
$Pa \leftarrow elitism(Offspring, Pa)$
if $Pa\ no\ change$ **then**
$distance \leftarrow distance - 1$;
if $distance == 0$ **then**
$reset(Pa)$
$initialize(distance)$
endif
endif
$t \leftarrow t + 1$ /* One more generation */
Return: best solution found.

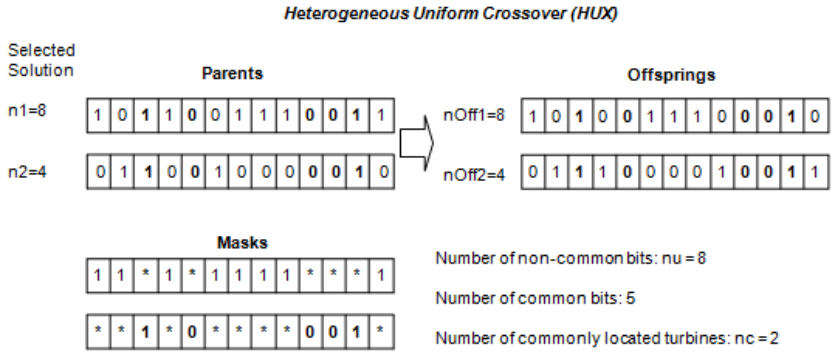


Figure 2: *Crossover HUX for CHC algorithm*

3.2 Geometric Particle Swarm Optimization

The Geometric Particle Swarm Optimization (GPSO) enables us to generalize PSO to virtually any solution representation in a natural and straight-forward way, extending the search to other spaces, such a combinational ones. This property was demostrated for the cases of Euclidean, Manhattan and Hamming landscapes [7].

The key issue in this approach consists in using a multi-parental recombination of particles which leads to the generalization of a *mask-based crossover operation*, proving that it respects four requirements for being a *convex combination* in a certain space. This way, the mask-based crossover operation substitutes the classical movement in PSO, based on the *velocity* and *position update* operations, only suited for continuous spaces.

The pseudocode of the GPSO algorithm for Hamming spaces is shown in Algorithm 2. For a given particle *i*, three parents take part in the 3PMBCX operator (line 13). The current position x_i, the social best position g_i and the historical best position found h_i (of this particle). The weight values w_a, w_b and w_c indicate for each element in the crossover mask the probability of having values from the parents x_i, g_i or h_i respectively. A constriction of the geometric crossover forces w_a, w_b and w_c to be non-negative and add up to one.

Algorithm 2 GPSO

$S \leftarrow InitializeSwarm()$; /* Initialize Swarm */
While *not stop criteria* **do**
 for *each particle* x_i *of* S **do**
  $evaluate(x_i)$
  if $fitness(x_i) >= fitness(h_i)$ **then**
   $h_i \leftarrow x_i$;
  end if
  if $fitness(h_i) >= fitness(g_i)$ **then**
   $g_i \leftarrow h_i$;
  end if
end for
for *each particle* x_i *of* S **do**
  $x_i \leftarrow 3PMBCX((x_i, w_a), (g_i, w_b), (h_i, wc))$
  $mutation(x_i)$
end for
end while
Return: best solution found.

For Hamming spaces, which is the focus of this work, a *three-parent mask-based crossover* (3PMBCX) was defined as follows: given three parents *a*, *b* and *c* in $\{0,1\}^n$, generate randomly a crossover mask of length *n* with symbols from the alphabet $\{a,b,c\}$. Build the offspring o filling each position with the bit from the parent appearing in the crossover mask at the considered position.

In a convex combination, the weights w_a, w_b and w_c indicate for each position in the crossover mask the probability of having the symbols *a*, *b* o *c*. Fig. 3 shows an example of this kind of crossover.

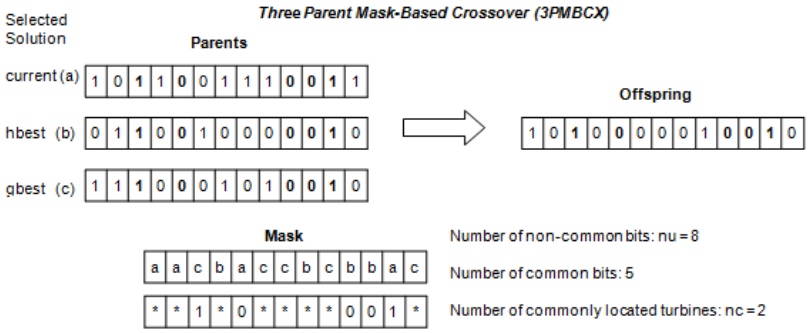


Figure 3: *Crossover 3PMBCX for the algorithm GPSO*

4. Instantiating the algorithms for the Problem

In this section, we will explain how our approach works: we will introduce the fitness function, the representation used, and the customizing of CHC and GPSO for the problem.

4.1 Objective Function

The objective function that we are maximizing is the annual profit got from the wind farm, defined as follows [8]:

$$profit = \left[st - \left(\frac{cost_{tot}}{P_{tot}}\right) \right] P_{tot} \tag{6}$$

where st represents the estimated selling price for a KWh of electrical energy on the market in (in this work it value is 0.1 €/KWh), P_{tot} represents the total expected energy output (kWh) of the wind farm per year, and $cost_{tot}$ is given by equation 5. The number of wind turbines is unknown and here also to be found by the used optimization algorithms.

4.2 Representations of Wind Turbine Locations

As other existing approches for the problem of Wind Energy Optimization we discretize the terrain in a matrix. A wind farm is logically divided into many small square like cells. Each cell in the wind farm grid can have two possible states: it contains a turbine (represented by 1) or it does not contain a turbine (represented by 0). A 10×10 grid is used here as the ground platform to place the wind turbines, and shown in Fig. 4. A binary string with 100 bits represents the location of the wind turbines in the wind farm. There are 2^{100} candidate solutions. The width at each cell, in the center of which a turbine would be placed, is equal to five times rotor diameter, $5D$ (or 220 m). Thus, the resulting dimension is $50D \times 50D$. The $5D$ square grid size also satisfies the rule of thumb of spacing requirements in the vertical and horizontal directions.

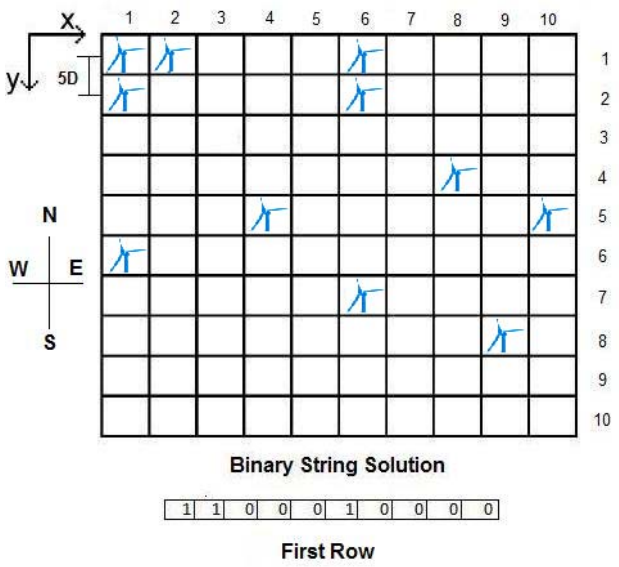


Figure 4: *Example of wind farm layout and the binary string representation*

4.3 Customizing CHC and GPSO for the Problem

In this problem, GPSO was developed as follows: each particle *i* of the swarm consists of a binary vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ representing the terrain (10×10) where the wind farm will be installed; each element x_{ij} can have a wind turbine (represented by 1) or be empty (represented by 0). In this particular case (10×10) each partíele has a length (*n*) of 100 elements.
CHC was developed as follows: each individuals consists of a binary vector $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ in the same representation than GPSO, and the same criteria for the positioning of the wind turbines.

5. Experimental Study

In this work we investigate three farm scenarios, in all of them we consider the case of uniform wind coming from the North, with different speeds for each case. Our aim is to analyse different wind farms and try to generalize our conclusions to guide designer in similar configurations. The first case we assume a wind speed of 12 m/s, in the second case a wind speed of 18 m/s, and the third case a wind speed of 25 m/s. We have selected these three scenarios based on the properties of wind profit of the mathematical model.
We show the different configurations for each case with the average fitness values, standard deviation of the fitness, total annual power output, average power output, number of wind turbines, average efficiency of the park, average execution time of each algorithm and the number of evaluation needs to find the better solution. We have also computed a statistical study comparing the average fitness values, and execution time of of each algorithm and we calcule the *p-value* with the *Kruskal-Wallis* test to conclude if it exists statistical significance between average fitness values and between average execution times. Each algorithm was executed 30 independent times with a stop criteria of $5{,}000{,}000$ evaluations. All the algorithms are executed in a MultiCore $2\times$ QuadCore 2 GHz and for the implementation of the algorithms we have used the library of optimization MALLBA [9].
For each scenario we used the properties of wind turbines and the parameters of the each algorithm shown in Table 1.

[Wind Turbine Property]

Description	Parameter	Value
Nominal Power	P	750 *KWh*
Rotor Diameter	D	44 *m*
Trust Coefficient	Ct	0.88
Wake Decay Constant	k	0.11
Cut-in Velocity	V_i	13 *km/h*
Cut-Out Velocity	V_p	90 *km/h*

[Parameters of GPSO]

Description	Value
Population Size	128
Size of the Swarm	100
Crossover	3PMBCX
Probability of Mutation	0.1%
Frecuency of Mutation	Bit flip 0.2%
Selection of Parent	x_i , g_i y h_i
Selection of New Generation	Elitist
Weight values w_a , w_b y w_c	0.2+0.1+0.7

[Parameters of CHC]

Description	Value
Population Size	128
Crossover	HUX
Cataclismic Mutation	Bit Flip 50%
Preserved Population	5%
Initial Threshold	25% of instance size
Convergence Value *Q*	1
Selection of Parents	Randomly
Selection of New Generation	Elitist

Table 1: *Properties of wind turbines and parameters used in CHC and GPSO*

5.1 Scenario *(a)*: Wind Speed of 12 m/s

For this scenario we have executed both algorithms (CHC and GPSO) with the parameters shown in Table 1tab:chc and 1tab:pso respectively, and we obtained the best configuration of the farm ilustrated in the Fig. 6 and the numerical values shown in Table 2.

Table 2: *Results of scenario* (a)

Description	CHC	GPSO
Average Fitness Values (€)	**3,608,160** (± 10,985.8)	3,544,900 (± 39,926.4)
Average Power Output (KWH)	**14,205.13**	14,132.89
Annual Power Output (MW)	124,471.36	124,471.36
Average Efficiency (%)	**91.28**	90.86
Number of Wind Turbines (N)	30	30
Average Execution Time (s)	1.54	**1.15**
Average Evaluation of Best Solution Found	259,735	**107,725**

In this scenario CHC obtained better average fitness value, better power output and better efficiency.CHC needs more execution time and more evaluations to find the best solution than GPSO. GPSO obtained smaller values but with less execution time and evaluations. We calcule the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it value is $2.28e^{-08}$. This value is smaller than $0{,}05$, so we conclude that it exists statistical significance between average fitneses and that CHC is more accurate and sligthly slower than GPSO. The *p-value* for the average execution time is 0.17, it is higher than 0.05, so we conclude that it does not exists statistical significance between average execution times.

Fig. 5a shows the evolution of the fitness fig:caso1fitness and the power output obtained fig 5b. The configuration of the farm found for each algorithms is ilustrated in Fig. 6. We can see that the solution uses 30 wind turbines and they are aligned in rows keeping a constants distance between them, and in an orthogonal position with respect to the wind direction.

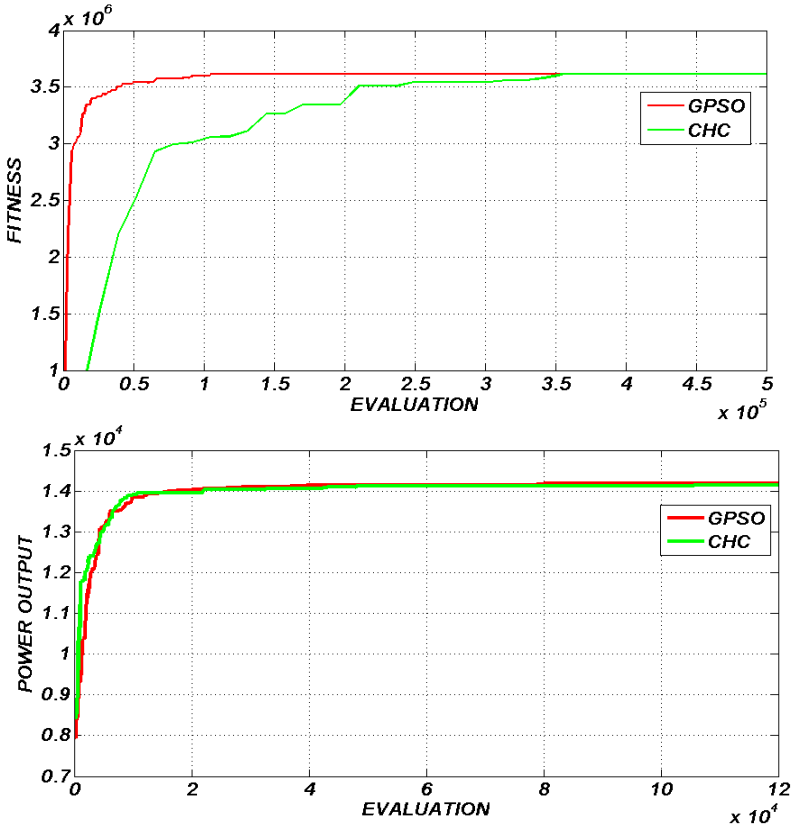


Figure 5: *Evolution of fitness values and power output for scenario* a

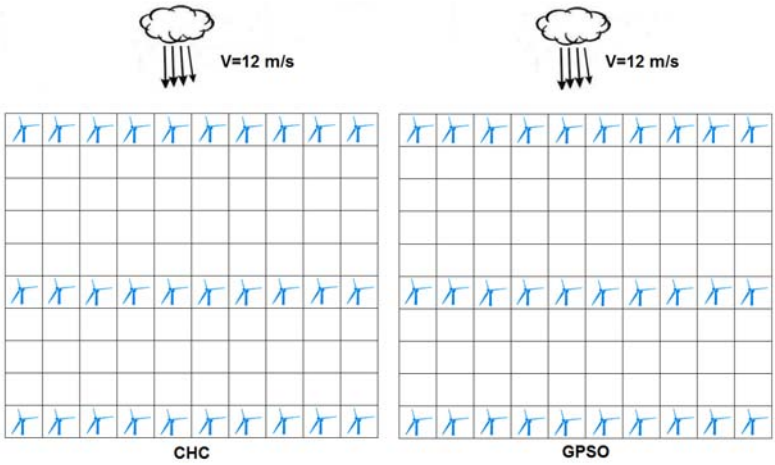


Figure 6: *Best configuration of the wind farm for both algorithms in scenario* a

5.2 Scenario *(b)*: Wind Speed of 18 m/s

For this scenario we have executed both algorithms CHC and GPSO with the parameters shown in Table 1tab:chc and 1tab:pso respectively, and we obtained the best configuration of the wind farm ilustrated in the Fig. 8, with the numerical values shown in Table 3

Table 3: *Results of scenario* (b)

Description	CHC	GPSO
Average Fitness Values (€)	15,283,300(± 0)	15,283,300 (± 0)
Average Power Output (KWH)	27,532.9	27,532.9
Annual Power Output (MW)	241,188.2	241,188.2
Average Efficiency (%)	91.77	91.77
Number of Wind Turbines (N)	40	40
Average Execution Time (s)	**0.12**	0.25
Average Evaluation of Best Solution Found	**18,890**	23,706

In this scenario CHC and GPSO obtained the same average fitness value, better power output and better efficiency. However CHC needed less execution time as it needed less evaluations than GPSO. We calculed the *p-value* with the *Kruskal-Wallis* test for the average fitness values and it results higher than 0.05, so we conclude that it does not exist stadistical significance between average fitnees values. The *p-value* for the average execution time is 0.002, it is smaller than 0.05, so we conclude that it exists statistical significance between average execution times.

Fig. 7 shows the evolution of the fitness fig 7a and the power output obtained fig 7b. The best configuration of the wind farm found for each algorithms is ilustrated in Fig. 8, where we can see that the number of wind turbines are 40, they forming two rows in the center and in the opposite way with the wind sense.

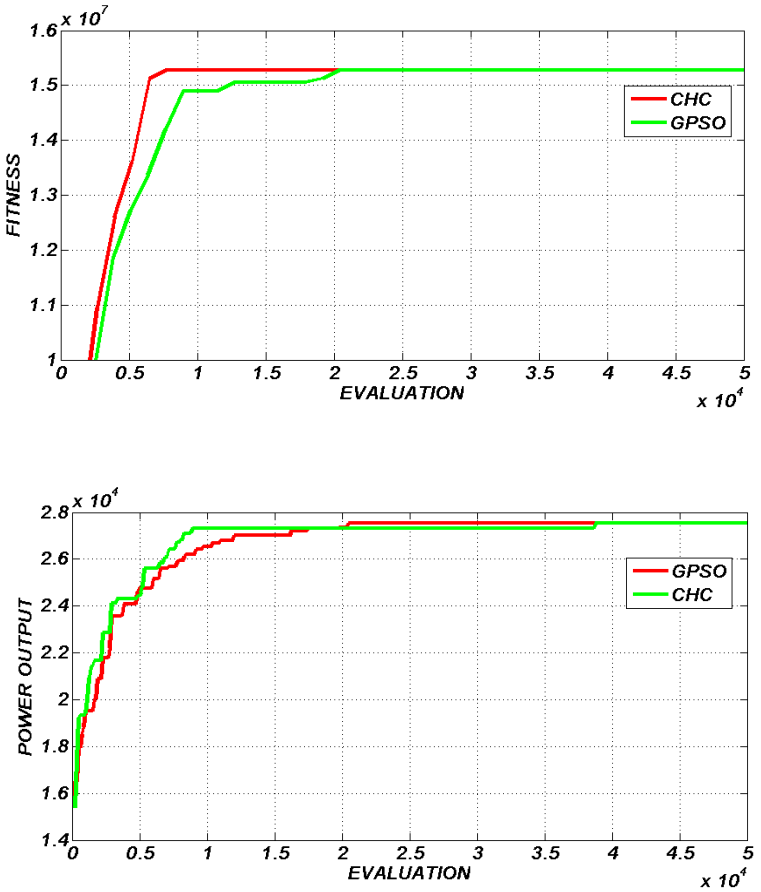


Figure 7: *Evolution of fitness values and power output for scenario* b

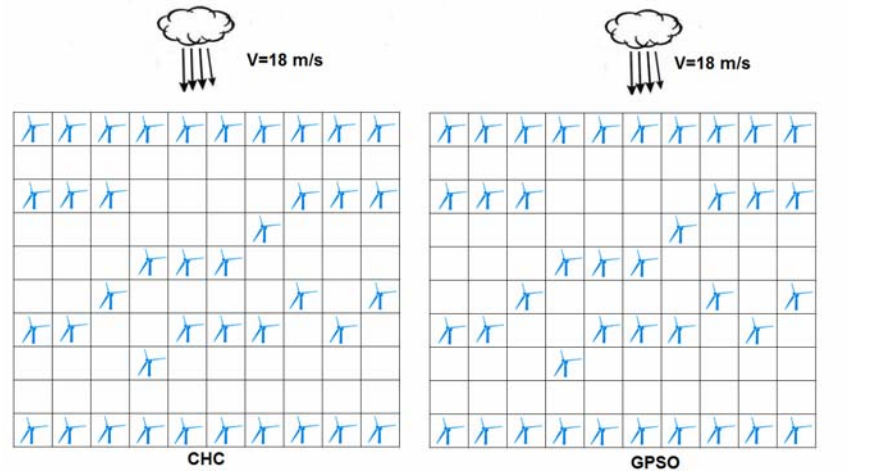


Figure 8: *Best configuration of the park for both algorithms in scenario* b

5.3 Scenario *(c)*: Wind Speed of 25 m/s

For this scenario we have executed both algorithms, CHC and GPSO, with the parameters shown in Table 1tab:chc and 1tab:pso respectively, and we obtained the best configuration of the wind farm ilustrated in Fig. 10 and the numerical values shown in Table 4.

Table 4: *Results of scenario* (c)

Description	CHC	GPSO
Average Fitness Values (€)	**19,345,000** (± 132,283)	19,094,000 (± 295,600)
Average Power Output (KWH)	**32,169.51**	31,882.98
Annual Power Output (MW)	282,654.54	282,654.54
Average Efficiency (%)	**85.76**	85.01
Number of Wind Turbines (N)	50	50
Average Execution Time (s)	**0.55**	1.54
Average Evaluation of Best Solution Found	**96,061**	201,024

In this scenario CHC obtained again better values in most of metrics than GPSO, although the final configuration for the wind farm is the same for both algorithms. We have calculed the *p-value* with the *Kruskal-Wallis* test for the

average fitness values and it results is $7.526e^{-05}$. This value is smaller than 0.05, so we conclude that it exists statistical significance between average fitnees values, then CHC is better than GPSO. The *p-value* for the average execution time is 0.023, it is smaller than 0.05, so we conclude that it exists statistical significance between average execution times.

Fig. 9 shows the evolution of the fitness fig 9a and the power output obtained fig 9b. The best configuration of the wind farm found for each algorithms is ilustrated in Fig. 10, where we can see that the number of wind turbines are 50 and they all form the expected three rows in the center, in the opposite way than the wind direction.

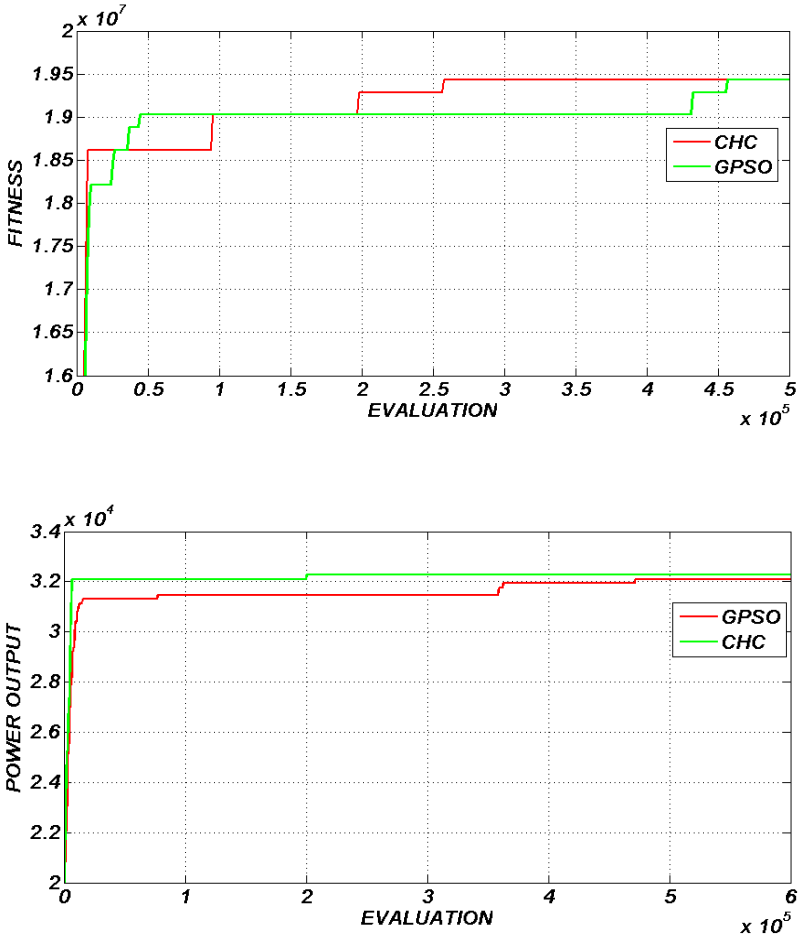


Figure 9: *Evolution of fitness values and power output for scenario* c

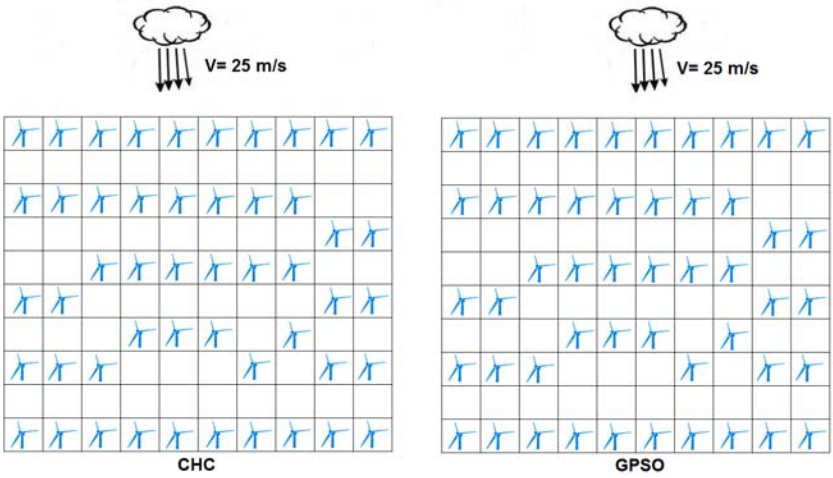


Figure 10: *Best configuration of the wind farm for both algorithms in scenario* c

6. Conclusions and Future Work

We have here solved the problem of optimal placement of wind turbines in a wind farm with the objective to maximize the power energy produced with the less number of wind turbines to reduce the overall cost. Both algorithms are very competitive. In the first scenario CHC obtained better values in average fitness values, average efficiency and averare power output than GPSO. Both obtained the same final configuration of the wind farm but GPSO did it in less execution time and less number of evaluations. In the second scenario CHC and GPSO obtained the same preformance in the majority of metrics except in execution time and number of evaluation where CHC had better performance. In the third scenario CHC had a better performance than GPSO in all metrics, in this case both algorithms obtained the same configuration of the wind farm. We obtained the same configuration compared with previous work for the scenario *a*. Aparently cost function allow to find different solution and power function keeps similar evolution in both algorithms. In second and third scenario may need more time to find the optimal solution. As a future work we will consider additional farm models, including more real world factors, such as terrain effect and the esthetic impact. Also, we intend to study the scalability of this problem with bigger instances of the wind farm and new parameters of the wind turbines. Finally we plan to solve this problem as multiobjective consider two contrast function, the cost of design the wind farm and the produced energy.

7. Acknowledgements

We here acknowledge partial funding from Project UNPA-29/B105, University of Patagonia Austral Argentina, DIRICOM Project Nª P07-TIC-03044 and M* Project Nª TIN2008-06491-C04-01, University of Málaga Spain.
M. Bilbao acknowledge the co-operation of the University of Málaga for providing new ideas and constructive criticisms. Also to the University of Patagonia Austral, and the ANPCYT (National Agency to Promote Science and Technology) from which we receive continuous support.

References

1. Manwell, J. F., McGowan, J. G., and Rogers, A. L., Wind Energy Explained-Theory, Design and Application, 1st ed., Reprint with correction, Jhon Wiley & Sons Ltd., 2003.
2. Bilbao, M., Alba, E., "Simulated Annealing for Optimization of Wind Farm Annual Profits", 2nd International Symposium on Logistics and Industrial Informatics, Linz, Austria. 2009.
3. Huang, H. S., "Distributed Genetic Algorithm for Optimization of Wind Farm Annual Profits", Intelligent Systems Applications to Power Systems, ISAP, International Conference on Volume, Issue, 5-8 Nov. 2007.
4. Alba, E., Molina, G., Chicano, F., "Optimal Placement of Antennae using Metaheuristics", Numerical Methods and Applications (NM&A-2006), Borovents, Bulgaria, 2006.
5. Alba, E., García-Nieto, J., Taheri, J., Zomaya, A., "New Research in Nature Inspired Algorithms for Mobility Management in GSM Networks", Fifth European Workshop on the Application of Nature-inspired Techniques to Telecommunication Networks and other Connected Systems, EvoWorkshops 08, Springer-Verlag, Napoli Italy, 2008.
6. Katic, I., Hojstrup, J. and Jensen, N. O., "A Simple Model for Cluster Efficiency", European Wind Energy Association Conference and Exhibition, Rome-Italy, 7-9 October, 1986.
7. Moraglio, A., Di Chio, C., Poli, R., "Geometric Particle Swarm Optimization", In Ebner, M., O'Neill, M., Ek, A., Vanneschi, L., Esparcia-Alcázar, A.I. (eds.), EuroGP 2007, LNCS, vol. 4445, Springer, Heidelberg, 2007.
8. Ozturk, U. A. and Norman, B. A., "Heuristic methods for wind energy conversion system positioning", Electric Power Systems Research, vol.70, 2004.
9. Alba, E., Almeida, F., Blesa, M., Cotta, C., Díaz, M., Dorta, I., Gabarró, J., León, C., Luque, G., Petit, J., Rodríguez, C., Rojas, A., Xhafa, F., "Efficient Parallel LAN/WAN Algorithms for Optimization. The MALLBA Project", Parallel Computing 32, 2006.

Approximations on Minimum Weight Pseudo-Triangulations using Ant Colony Optimization Metaheuristic

EDILMA OLINDA GAGLIARDI, MARIA GISELA DORZÁN, MARIO GUILLERMO LEGUIZAMÓN[1], GREGORIO HERNÁNDEZ PEÑALVER[2]

[1] Facultad de Ciencias Físico Matemáticas y Naturales,
Universidad Nacional de San Luis, Argentina
{oli,mgdorzan,legui}@unsl.edu.ar
[2] Facultad de Informatica, Universidad Politécnica de Madrid, España
gregorio@fi.upm.es

Abstract. *Globally optimal pseudo-triangulations are difficult to be found by deterministic methods as, for most type of criteria, no polynomial algorithm is known. In this work, we consider the Minimum Weight Pseudo-Triangulation (MWPT) problem of a given set of n points in the plane. This paper shows how the Ant Colony Optimization (ACO) metaheuristic can be used to find optimal pseudo-triangulations of minimum weight. For the experimental study presented here we have created a set of instances for MWPT since no reference to benchmarks for these problems was found in the literature. We assess through the experimental evaluation the applicability of the ACO metaheuristic for MWPT.*

Key words: *Pseudo-Triangulation, Minimum Weight, Computational Geometry, ACO Metaheuristic.*

1. Introduction

In Computational Geometry there are many problems that either are NP-hard or no polynomial algorithms are known. Therefore, it is interesting to find approximate solutions using metaheuristics. The optimization problems related to special geometric configurations, such as triangulations and pseudo-triangulations, are interesting to research due to their use in many fields of application, e.g., visibility, ray-shooting, kinetic collision detection, rigidity, guarding, etc. The pseudo-triangulations, like triangulations, are planar partitions. Minimizing the total length has been one of the main optimality criteria. The related problems are the Minimum Weight Triangulation (MWT) and the Minimum Weight Pseudo-Triangulation (MWPT) that minimize the sum of the edge lengths, providing a quality measure for determining how good a structure is. The complexity of computing a minimum weight triangulation has been one of the most long-standing open problems in Computational Geometry, introduced by Garey and Johnson [5] in their open problems list, and various approximation

algorithms were proposed over time. Mulzer and Rote [7] recently showed that MWT problem is NP-hard. The complexity of MWPT problem is unknown, but Levcopoulos and Gudmundsson [6] show that a 12-approximation of an MWPT can be computed in $O(n^3)$ time. They give an $O(log\ n.\ w(MST))$ approximation of an MWPT, in $O(n\ log\ n)$ time, where $w(MST)$ is the weight of the minimum Euclidean spanning tree, which is a subset of the obtained structure.

Given the inherent difficulty of these problems, the approximate algorithms arise as alternative candidates. These algorithms can obtain approximate solutions to the optimal solutions, and they can be specific for a particular problem or they can be part of a general applicable strategy in the resolution of different problems. The metaheuristic methods satisfy these properties.

In this work, we consider the MWPT problem of a given set of n points in the plane. This paper shows how the Ant Colony Optimization (ACO) metaheuristic can be used to find optimal pseudo-triangulations of minimum weight. For the experimental study presented in this work we use the Ant Colony Optimization (ACO) metaheuristic.

This paper is organized as follows. In the next section we present the theoretical aspects of pseudo-triangulations. In Section 3, we present the general overview of the ACO metaheuristic. Section 4 presents the proposed ACO algorithms for the MWPT problem. Finally, we describe the MWPT instances used and the details and results of the experimental study in which we analyze the sensitivity of some important parameters on the performance of the proposed ACO algorithm.

2. Minimum Weight Pseudo-Triangulation

Let S be a set of points in the plane. A pseudo-triangulation PT of S is a partition of the convex hull of S into pseudo-triangles whose vertex set is exactly S. A pseudo-triangle is a planar polygon that has exactly three convex vertices, called corners. The weight of a pseudo-triangulation PT is the sum of the Euclidean lengths of all the edges of PT. The pseudo-triangulation that minimizes this sum is named a *Minimum Weight Pseudo-Triangulation* of S and it is denoted by $MWPT(S)$.

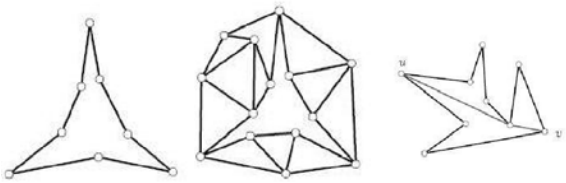


Figure 1: *(a) A pseudo-triangle, (b) a pseudo-triangulation of a point set and (c) a pseudo-triangulation of a simple polygon, including a geodesic path from u to v [9].*

The concept of pseudo-triangulation was introduced by Pocchiola and Vegter in [8] on the analogy of the arrangements of pseudo-lines; see [9] for a survey with many results of pseudo-triangulations. As we mentioned in Section 1, there exists a set of points for which any triangulation will have weight $O(nwt(M(S)))$. A natural question is whether there exist a similar worst-case bounds for pseudo-triangulations. Rote et al.,[10] were those who asked if the MWPT is a NP-hard problem, stimulating the search of exact or approximate algorithms. Gudmundsson and Levcopoulos [6] considered the problem of computing a minimum weight pseudo-triangulation of a set S of n points in the plane, presenting an $O(n \log n)$-time algorithm that produces a pseudo-triangulation of weight $O(\log n\ wt(M(S)))$ which is shown to be asymptotically worst-case optimal. That is, there exists a point set S for which every pseudo-triangulation has weight $\Omega(\log n\ wt(M(S)))$, where $wt(M(S))$ is the weight of a minimum spanning tree of S. Also, they presented a constant factor approximation algorithm running in cubic time, and they gave an algorithm that produces a minimum weight pseudo-triangulation of a simple polygon. Previous works about approximations on mentioned problems using metaheuristic, were presented in [2] and [3], where we described the design of the ACO algorithms and gave the first steps in this research.

3. Ant Colony Optimization Metaheuristic

The ACO metaheuristic involves a family of algorithms in which a colony of artificial ants cooperate in finding good solutions to difficult discrete optimization problems. Cooperation is a key design component of ACO algorithms. The choice is to allocate the computational resources to a set of relatively simple agents (artificial ants) that communicate indirectly by stigmergy. Thus, good quality solutions are an emergent property of the agents cooperative interaction.

An artificial ant in an ACO algorithm is a stochastic constructive procedure that incrementally builds a solution by adding opportunely defined solution components to a partial solution under construction. Therefore, the ACO metaheuristic can be applied to any combinatorial optimization problem for which a constructive graph can be defined. Each edge (i, j) in the graph represents a possible path and it has associated two information sources that guide the ant moves: pheromone trails and heuristic information. The pheromone trail, denoted by τ_{ij}, encodes a long-term memory about the entire ant search process, and is updated by the ants themselves. The heuristic information, denoted by η_{ij}, represents a priori information about the problem instance or run-time information provided by a source different from the ants. In many cases is the cost, or an estimate of the cost, of adding the component or connection to the solution under construction. These values are used by the ants to make probabilistic decisions on how to move on the graph. The ants act concurrently and independently and although each ant is complex enough to find a solution to the problem, which is probably poor, good-quality solutions can only emerge as the result of the collective interaction among the

ants. This is obtained via indirect communication mediated by the information ants read or write in the variables storing pheromone trail values. It is a distributed learning process in which the single agents, the ants, are not adaptive themselves but, on the contrary, adaptively modify the way the problem is represented and perceived by other ants [4].
There are two additional process for updating pheromone and the daemon actions. The pheromone updating is the process by which the pheromone trails are modified. The trail values can either increase, as ants deposit pheromone on the components or connections they use, or decrease, due to pheromone evaporation. The daemon procedure is used to implement centralized actions which cannot be performed by single ants. Examples of daemon actions are the activation of a local optimization procedure, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a nonlocal perspective. The daemon can observe the path found by each ant in the colony and select one or a few ants, like those that built the best solutions in the algorithm iteration that allowed to deposit additional pheromone on the connections they used.

3.1 The general ACO algorithm

In this section we present a general ACO algorithm (Algorithm 1) and a description of the main components. After that, the next section describes in detail the specific component of the general ACO algorithm (function BuildSolutionk) that has to be adapted for MWPT. Main components of Algorithm 1:

Algorithm 1 General-ACO

Initialize
for $c \in \{1, \ldots, C\}$ **do**
    for $k \in \{1, \ldots, K\}$ **do**
        BuildSolutionk
        EvaluateSolution
    end for
    SaveBestSolutionSoFar
    UpdateTrails
end for
ReturnBestSolution

Initialize: this process initializes the parameters considered for the algorithm. The initial trail of pheromone is associated to each edge, τ_0; it is an small positive value, in general, the same for all edges. The quantity of ants of the colony, K. The weights that de ne the proportion in which they will affect the

heuristic information and pheromone trails in the probabilistic transition rule, named respectively β and α. C is the maximum number of cycles.
BuildSolutionk: this process begins with a partial empty solution which is extended at each step by adding a feasible solution component chosen from the current solution neighbors; i.e., to find a route on the construction graph guided by the mechanism that defines the set of feasible neighbors with regard to the partial solution. The choice of a feasible neighbor is done in a probabilistic way in every step of the construction, depending on the used ACO variant. In this work, the selection rule for the solutions construction is based on the following probabilistic model:

$$P_{ij} = \begin{cases} \dfrac{\tau_{ij}^{\alpha} . \eta_{ij}^{\beta}}{\sum_{h \in F(i)} \tau_{ih}^{\alpha} . \eta_{ih}^{\beta}}, \text{ j} \in \text{F(i)} \\ 0, \quad \text{otherwise} \end{cases} \qquad (1)$$

F(i) is the set of feasible points for point *i*. τ_{ij} is the pheromone value associated to edge (*i, j*). η_{ij} is the heuristic value associated to edge (*i, j*). α and β are positives parameters for determining the relative importance of the pheromone with respect to the heuristic information.
EvaluateSolution: evaluates and saves the best solution found by ant *k* in the current cycle.
SaveBestSolutionSoFar: saves the best solution found for all cycles so far.
UpdateTrails: increases the pheromone level in the promising paths, and is decreased in other case. First, all the pheromone values are decreased by means of the process of evaporation. Then, the pheromone level is increased when good solutions appear. The following equation is used:

$$\tau_{ij} = (1 - p)\tau_{ij} + \Delta\tau_{ij} \qquad (2)$$

- $\rho \in (0, 1]$ is the factor of persistence of the trail.
- $\Delta\tau_{ij} = \sum_{k=1}^{K} \Delta^k \tau_{ij}$ is the accumulation of trail, proportional to the quality of the solutions.
- $\Delta^k \tau_{ij} = \begin{cases} Q / L_k & \text{when ant k used edge (i, j)} \\ 0 & \text{in other case} \end{cases}$
- Q is a constant depending of the problem; usually is set to 1.
- L_k is the objective value of the solution *k*.

Pheromone evaporation avoids a fast convergence of the algorithm. In addition, this way of forgetting allows the exploration of new areas of the search space. The update of the pheromone trail can be done according to one of the following criteria: elitist and non-elitist. In the elitist case, the best found solution is used to give an additional reinforcement to the levels of pheromone. The non-elitist one uses the solutions found by all the ants to give an additional reinforcement to the levels of pheromone.

3.2 The proposed ACO algorithm for MWPT (ACO-MWPT)

For ACO-MWPT, each ant in *BuildSolutionk* function builds a pseudo-triangulation, starting with one face. This face has the edges obtained by the convex hull of the points set *S*, i.e., *CH*(*S*). For the solution construction, each ant performs a process of partitioning *CH*(*S*) in faces. This process finishes when all faces are pseudo-triangles without interior points. A face is divided into two faces when it has interior points or is not a pseudo-triangle.
Thus, the partition can be done if *i*) there are at least one interior point and two points in the border; or *ii*) there is not an interior point, so the procedure use two points located on the border. $Faces_k$ represents the set of no treated faces.
PartitionFace(F) selects the points of *F* to build the new faces. It takes one interior point and two probabilistic selected points of the border, or (if there is not interior point) only two probabilistic selected points of the border. The feasible points for a point are those visible and not adjacent to it. The selection is done according to Equation 1. Also, this process uses two additional procedures *SelectInteriorPoint*(*F*) and *SelectBorderPoint*(*F*), where the point selection is achieved according to one of the following criteria: *i*) at random; *ii*) the largest quantity of feasible points; or, *iii*) the lowest quantity of feasible points.

Algorithm 2 BuildSolutionk

$S_k \leftarrow \emptyset$ /* solution builded by *k*-ant */
while ($Faces_k \neq \emptyset$) **do**
    Let *F* be a face in $Faces_k$
    if *F* is Pseudo-triangle without interior points **then**
    $S_k \leftarrow S_k \cup \{F\}$ /* *F* is a new pseudo-triangle */
    $Faces_k \leftarrow Faces_k - \{F\}$
    else
    PartitionFace(*F*)
    end if
end while

4. Experimental Evaluation

For the whole experimental evaluation, we developed a generator of points, using different functions of random generation of CGAL Library [1]. The points in the plane are non collinear, uniform distributed, with coordinates in (0, 1000). The set of instances has been formed by 5 collections of 10 points sets of different cardinality: 40/80/120/160/200-collection respectively. We have the 40-collection with the point sets LD401, LD402,… ,LD410; and so on. We emphasize that these collections are not available in previous related researches, and con-tribute in this paper. The ACO-MWPT algorithms have been implemented in C programming language.

We show the initial experimental phase, that will allow us to decide which are the most suitable parameter settings. The analyzed collections correspond to LD401, LD402, LD403 and LD404. The parameter setting, are according to the Equation 1 and 2, corresponding to (α-β-ρ-*elit*-*criterion*), where α: 1; β: 1 and 5; ρ: 0.1, 0.25 and 0.5; . If *elit* is set to 1, the trail is update in an elitist way; in other case, the updating is done in a not elitist way. The criterion and are set to 1. So, there are twelve parameter setting, and for each were performed 30 runs by using different random seeds. The number of cycles, C, is 1000; the number of ants, K, is 50. We obtain average, median, best, and variance/standard deviation values, considering the objective function (weight); and quantity of pseudo-triangles. Then, we select only the four best parameter settings. The numerical work were done using the BACO parallel cluster, composed by 60 PCs, with a 3.0 GHz Pentium-4 processor each one and 90 PCs with a 2.4 GHz Core 2 Quad processor each one, under CONDOR batch queuing system.

Next, the results for this experimental study are resumed in Tables 1 to 6. In Tables 1 to 4, we show the results according to the four best parameter settings with respect to the smaller weights. The configuration (1-5-25-1) obtained the smallest weight for the LD402. See Table 2. Table 5 shows a comparison between the four best parameter settings for the obtained smaller weight values and the four best parameter settings for the obtained smaller average values. The values are similar. Better results were obtained by β: 5; *elit*: 1; ρ: 0.1, 0.25 and 0.5. We obtained better results giving more relevance to the heuristic information and updating the trails in an elitist way. We considerer necessary to observe the influence of parameters in general. Table 6 shows the influence percentage for each parameter with respect to the best found weights values. Likewise, we observe what happened with the best average values. Additionally, in Table 7, we show the influence percentage of every parameter for the best performance with respect to the average. We can observe that the percentages are similar. Figure 2 shows the boxplots of the weights obtained, for the 30 seeds for LD401, LD402 and LD403 for the four best configurations.

Figure 2, from a) to d), show the boxplots of the weights obtained for the 30 seeds for LD401, LD402, LD403 and LD404 for the four best configurations. The boxplots are not similar; for the LD402, the best weights obtained are outliers and has a good behavior. In the case of LD404, the boxplots are regular, but the best values are not approximate to the minimum.

In this sense, Tables 6 and 7 give a clearer idea over which are the suitable parameter settings.
With respect to the quantity of pseudo-triangles found in the most approximate pseudo-triangulation we can observe that not necessarily it is minor.

Table 1: *MWPT: Results for LD401.*

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-5-10-1	6557443,73	6607908,75	6115636,63	166770,70	51
1-5-25-1	6644026,13	6658654,75	6286985,04	166104,26	48
1-5-50-1	6669542,40	6713656,75	6320652,56	159069,87	49
1-5-50-0	6777518,93	6847951,25	6322956,39	158225,32	52

Table 2: *MWPT: Results for LD402.*

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-5-25-1	4748353,60	4757694,50	4442710,43	114885,64	49
1-5-10-0	4681136,53	4685804,75	4470550,00	69468,03	48
1-5-10-1	4707699,73	4747199,25	4490214,33	83905,98	50
1-5-25-0	4729018,67	4749318,25	4524206,34	77542,87	43

Table 3: *MWPT: Results for LD403.*

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-5-25-1	6069210,67	6071705,00	5684342,58	143063,20	49
1-5-10-1	5980440,00	6021063,00	5699513,63	136174,19	50
1-5-25-0	6075029,87	6118394,25	5744775,81	110439,30	45
1-5-50-0	6073308,80	6104511,25	5746463,24	121285,65	51

Table 4: *MWPT: Results for LD404.*

Par. Setting	Average	Median	Best	Std. Dev.	#PT
1-1-50-1	6236883,73	6258985,50	5627098,22	218860,42	48
1-5-10-1	6162888,00	6154961,25	5668910,18	166455,96	49
1-5-50-1	6229822,93	6237045,50	5869145,72	202030,98	50
1-5-50-0	6237883,20	6252135,50	5903381,03	139767,35	47

Table 5: *MWPT: Comparison between four best BEST and AVERAGE parameter settings.*

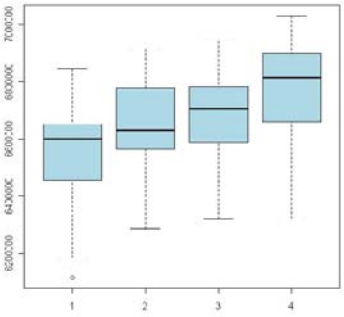
Par. Setting	Best	Par. Setting	Average
LD402 1-5-25-1	4442710,43	LD402 1-5-10-0	4681136,53
LD404 1-1-50-1	5627098,22	LD403 1-5-10-1	5980440,00
LD403 1-5-25-1	5684342,58	LD404 1-5-10-1	6162888,00
LD401 1-5-10-1	6115636,63	LD401 1-5-10-1	6557443,73

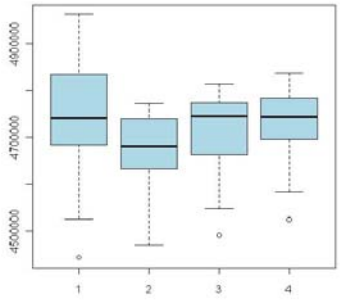
Table 6: *MWPT: Abstract Table for LD401, LD402, LD403, and LD404 w.r.t. best BEST values.*

α	β	ρ	*elit*
1 (100%)	5 (95%)	0.1 (31.25%)	1 (62.50%)
	1 (5%)	0.25 (31.25%)	0 (37.50%)
		0.5 (37.50%)	

Table 7: *MWPT: Abstract Table for LD401, LD402, LD403, and LD404 w.r.t. best AVERAGE values.*

α	β	ρ	*elit*
1 (100%)	5 (88%)	0.1 (37.50%)	1 (62.50%)
	1 (12%)	0.25 (25.00%)	0 (37.50%)
		0.5 (37.50%)	





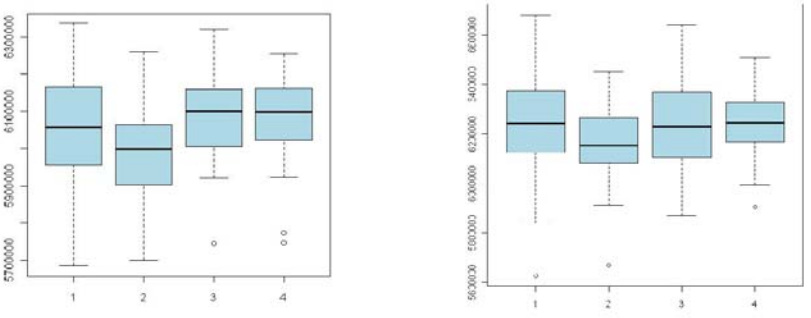


Figure 2: MWPT: Boxplots for a) LD401, b) LD402, c) LD403 and d) LD404.

5. Conclusion

In this work, we present the design of approximation algorithms for solving the Minimum Weight Pseudo-Triangulation problems for sets of points in the plane. The proposed ACO model for the MWPT is represented by an Ant System (AS), a particular instance of the class of ACO algorithms. We also detailed the generation of sets of points for the experimental evaluation, considering that is an other contribution. We show the initial experimental phase, where we have preliminary results that guide the future experimentation. So, from our analysis of these approximation algorithms, we obtained the suitable parameter setting. The future work will consist to continue the experimentation with the total collection of set of points and to compare the proposed algorithms against other strategies like Greedy or others metaheuristics. Finally, since the metaheuristics have proven to behave very well in solving this class of NP-hard problem, there are several directions for further research.

Acknowledgment

The authors would like to thank to Research Project Tecnologías Avanzadas de Bases de Datos 22/F614 financed by Universidad Nacional de San Luis, San Luis, Argentina; to the Project of Ministerio Ciencia e Innovación MTM2008-05043- Spain; and, to Instituto de Física Aplicada, Universidad Nacional de San Luis-CONICET, San Luis, Argentina, to allow us to use the cluster. Likewise, to the colleagues who contributed with opinions and technical advice.

References

1. Computational Geometry Algorithms Library (CGAL). http://www.cgal.org/
2. Dorzán, M., Gagliardi, E., Leguizamón, M. and Hernández Peñalver, G., Algoritmo ACO aplicado a la obtención aproximada de Triangulaciones de Peso Mínimo, XXXV Conferencia Latinoamericana de Informática (CLEI), 2009.
3. Dorzán, M., Gagliardi, E., Leguizamón, M., Taranilla, M. and Hernández Peñalver, G., Algoritmos ACO aplicados a problemas geométricos de optimización, XIII Encuentro de Geometría Computacional (EGC09), 2009.
4. Dorigo, M. and Stutzle, T., Ant Colony Optimization, Massachusetts Institute of Technology, 2004.
5. Garey, M. and Johnson, D., Computers and Intractability, W. H. Freeman and Company, 1979.
6. Gudmundsson, J. and Levcopoulos, C., Minimum weight pseudo-triangulations. Computational Geometry, Theory and applications, 38, 2007.
7. Mulzer, W. and Rote, G., Minimum weight triangulation is NP-hard. Proceedings of the 22nd Annual ACM Symp. on Computational Geometry, 2006.
8. Pocchiola, M. and Vegter, G., Pseudo-triangulations: theory and applications. Proceedings of the 12th Annual ACM Symposium on Computational Geometry, 1996.
9. Rote, G., Santos, F. and Streinu, I., Pseudo-triangulations-a survey. Surveys on Discrete and Computational Geometry-Twenty Years Later, Contemporary Mathematics, E. Goodman, J. Pach, R. Pollack, eds. American Mathematical Society, 2008.
10. Rote, G., Wang, C., Wang, L. and Yinfeng Xu, On constrained minimum pseudo-triangulations. Computing and Combinatorics, Springer-Verlag, Lecture Notes in Computer Science 2697, 2003.

IX

Distributed and Parallel Processing Workshop

A Network Failure-Tolerant P2P-VoD System

JAVIER BALLADINI, EDUARDO GROSCLAUDE, REMO SUPPI[1], EMILIO LUQUE[*]

Department of Computer Engineering, National University of Comahue, Argentina,
{jballadi, oso}@uncoma.edu.ar
[1]Department of Computer Architecture and Operating Systems,
Autonomous University of Barcelona, Spain, {remo.suppi, emilio.luque}@uab.es

Abstract. *Most Video-on-Demand (VoD) P2P systems lack a careful analysis of all the issues that may emerge when deploying their services over a network with a high probability of failures such as the Internet. As e result, they are not able to ensure an interruption-free video visualization service. To solve this problem, we propose a network failure tolerance architecture and scheme that can be easily adapted to already developed P2P-VoD systems. The solution uses techniques from: transmission management based on communication status, reception of data from multiple sources, suitable selection of server nodes, service migration mechanisms, and resource reservation. Based on the good results obtained with the previous proposal of a network failure-tolerant client-server VoD system, this work adapts and extends the functionality for VoD systems with P2P architectures.*

1. Introduction

With the current style of digital life, the entertainment and media industry has become an economically relevant sector. Video streaming has become one of the most popular Internet activities, benefited by the technology improvements of communications networks. The large number of customers and the high network bandwidth requirements of these services have driven a large number and variety of studies in the area.

Video streaming can be classified in: live streaming and video on demand (VoD). In the case of live streaming, servers broadcast live or TV programs, and users view these contents sequentially from the moment they access the service. Within live streaming, there is also interactive live streaming, such as Internet telephony and videoconferencing, with even greater time restrictions. Unlike live streaming, VoD allows the users to reproduce a video, selected from a large set of pre-stored videos, at any time and using the typical interactive commands of a DVD player.

Peer-to-peer (P2P) systems, where the nodes act as clients and servers at the same time, have been successful in distributing files to large numbers of users. P2P systems are also widely used for video distribution, including downloading videos in full before playing them and live video streaming (such as Coolstreaming). Recently, new systems to support VoD using P2P over the Internet have been designed.

In Internet, communication bandwidths fluctuate due to congestion. Congestion may be caused either by an increase in traffic or a physical failure such as links or routers being down. When a link is down, routing algorithms divert traffic to other links, which can cause congestion in the links that received the diverted traffic. It is normal that, despite the protection of IP routing, downed links and routers are followed by long instability periods in package routing, causing multiple rejections due to rerouting through wrong paths [1]. Therefore, for a VoD system to provide a high-quality service, that is, interruption-free reproduction and constant video quality[1], it should tolerate communication bandwidth fluctuations and temporarily inaccessible destinations from certain sources. Given a video and a certain communication between a server node and a client node, a network failure occurs when the communication bandwidth becomes insufficient to ensure a high-quality service (video delivery).

To avoid these network problems, various techniques can be used: transmission management based on communication status, reception of data from multiple sources, suitable selection of server nodes, service migration mechanisms, and resource reservation. However, current P2P-VoD systems, such as [2, 3], have deficient designs and do not tolerate these network failures.

Initially, our group worked with network failure tolerance for high quality VoD systems based on client-server architectures. The decision of using a simple architecture allowed limiting the problem, thus facilitating the design and verification of the techniques used. Based on experience and the good results obtained, we set out to adapt and extend the proposal to support P2P architectures. Thus, our current research line is focused on the design of an Internet P2P-VoD system that is network-failure tolerant and correctly manages communication degradation to provide a high quality service. In this paper, the design of a P2P-VoD architecture with a network failure tolerance scheme is described; this design can be implemented in most P2P-VoD systems (including the most recent), allowing a substantial increase in the benefits offered, which translates into a greater end user satisfaction.

The remaining sections of this article are organized as follows: in section 2, related work is discussed. In section 3, the architecture of the system is described and the basic concepts and results of the techniques proposed are discussed. Finally, in section 4, the conclusions obtained and future works are presented.

[1] Video quality is maintained while the service is provided, with no degradation in the quality of the displayed video.

2. Related work

Some of the techniques proposed in relation to failure tolerance in VoD systems are data recovery using redundancy of data, streaming data from multiple servers with transmission rate reassignment, and reducing transmission rate by degrading image quality. Using these techniques only, if servers went down or communications deteriorated considerably, the visualization of the multimedia content would still be interrupted. The solution is using service migration, which is a technique that allows varying the streaming source and thus avoid network failures. Service continuation mechanisms have been proposed for TCP connections, such as [4], that migrate the service by changing the streaming source in case of network problems. Even though this scheme is very relevant in the area of e-commerce, its failure detection mechanism does not take into account the temporary restrictions and the high bandwidth requirements of videos.

There are several articles on network failure detection for service migration in multimedia systems. Some articles, such as [5], propose failure detection techniques based on package delay, suitable for live streaming but inadequate for VoD; in VoD, the significant measurement is bandwidth and not package delay. In [6, 7], failure detection schemes are proposed for live streaming that are based on bandwidth. Clients receive transmissions from multiple server nodes and, using a client-driven protocol, each server delivers a part of the total transmission rate of the video. Rate assignment is dynamic and is based on the conditions of communication paths. Even though these articles do not propose service migration, their client-driven failure detection mechanism is not applicable to VoD, either. To avoid interruptions in visualization, a high-quality VoD system plans video data delivery to clients, prioritizing the delivery to those clients with a greater urgency to receive video data. Thus, a client cannot know if it is the communication that is deficient or if the server decided to prioritize the transmissions sent to other clients with greater urgency. The schemes implemented in VoD systems [8, 9] are not suitable either, because, as in the previous case, the clients are responsible for detecting network failures (by controlling reductions in data reception rate).

A similar situation occurs in papers such as [10], which use heartbeat mechanisms to provide tolerance to network failures in a VoD system. This scheme allows detecting a complete loss of communication; however, the detection of insufficient bandwidth is not detectable. Some papers on failure tolerance in VoD systems, such as [11], deal with nodes being down, but do not consider communication degradation.

For network failure tolerance, the selection of the best server nodes among a number of possible candidates is also important to serve a certain client node. However, the existing alternatives do not meet all the requirements of a network failure-tolerant P2P-VoD system. For instance, [3, 2] do not have a defined policy that takes into account communication status to prefer one peer over another. In [12], a scheme selecting those peers with the highest communication bandwidth is presented. However, it does not consider the

selection of diversified communication paths to prevent a network failure to affect many connections.

3. System architecture

Our VoD system assumes an Internet P2P architecture, where peers simultaneously receive content from multiple sources and stream content to multiple destinations. Source peers are peers that act only as servers and are the source of the content; the entire catalogue of videos is distributed among them. Limited-size peer collaboration groups that are interested in the same content and generally have very close reproduction points are formed. Among these, control connections and video data are established with the purpose of exchanging video segments. The downloads from multiple sources allows: balancing the load among server nodes, resisting more easily the event of a server node being down, and diversifying connection paths to tolerate network failures. In addition to these permanent connections, there are temporary ones that typically are brief transmissions dedicated to obtaining data segments (possibly less urgent) that are not present in their server nodes with permanent connections. The time required to establish these connections is less because path diversification does not have to be considered for selecting the server node. To avoid a control data management overload, a client node can have a maximum number of permanent connections (set at 5, although it could be different) and temporary connections.

Video data communications have a TCP-friendly congestion control, which ensures that our application can coexist with other Internet applications. This means that available communication bandwidths may vary significantly during a session. Our solution is based on resource reservation. Each client node establishes permanent connections and negotiates a minimum data transmission rate with its server nodes. When there is a deficient communication that prevents a server node to fulfill the rate agreed upon, the client node can renegotiate the rates with its server nodes or migrate the service, replacing the affected server node by a substitute one. Control communications between a client node and a server node are done by means of a TCP connection called *stream control channel*.

To adapt our proposal to existing VoD systems, no collaboration group creation, segment planning, or content replication and discovery strategy is imposed. The following paragraphs present the module architecture of peers and describe the failure detection and recovery technique, multi-source stream management and synchronization, and how connections are established within collaboration groups.

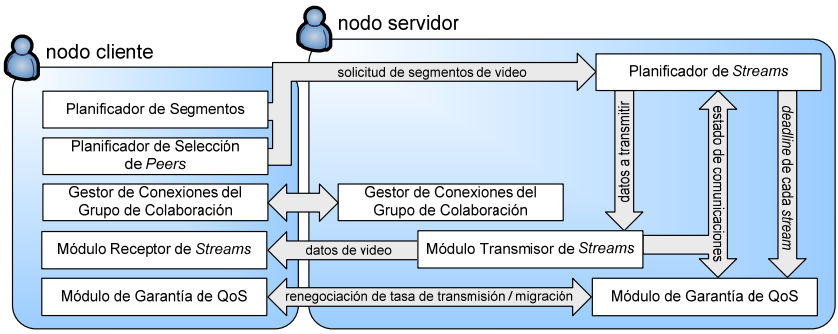


Figure 1: *Peer architecture: participating modules when acting as server and client*

3.1 *Peers*

The internal architecture of peers in their roles as server and client is shown in figure 1. Four modules implement the functionality of a server node: a stream transmission module (STM), a stream scheduler (SS), a quality of service assurance module (QoSAM), and a neighborhood connection manager (NCM). These modules interact with client node modules: a stream receiver module (SRM), a segment scheduler (SeS), a peer selection scheduler (PSS), and the counterpart of QoSAM and NCM. The NCM establishes the connections within a collaboration group. The SeS uses prefetching strategies (and possibly also batching and patching strategies) to determine the video segments that will be requested to given server nodes. These server nodes are selected by means of the PSS. Typically, the server node with less useful segments is selected as client node, those with permanent connections being preferred. This strategy allows not to overload those peers with many useful segments, such as source peers.

A prefetching technique provides video in advance to client nodes when communication paths are under-used, allowing these nodes to support low bandwidth times during communications [13]. When the traffic of data towards a client node decreases, the server node can use this extra bandwidth with other client nodes. The SS module (based on our previous work [14]) receives segment requests and carries out transmissions by means of a schedule that takes communication bandwidth into account and gives priority to those client nodes with a greater urgency for video data. This is done as an attempt to avoid interruptions during the visualization of the multimedia content. However, if the communication bandwidth with a client node is insufficient during a certain time, this node could consume all buffer data and cause an interruption in video reproduction due to buffer underflow.

To solve the buffer underflow problem and achieve an interruption-free reproduction service, the QoSAM component is included. This module is

responsible for determining if the communication bandwidth with a given client node is enough or not to ensure a continued data transfer at the agreed rate. When the QoSAM detects a network failure, it communicates with the client node, which can renegotiate the transmission rate or agree to migrate the service. The purpose of this service migration is changing the transmission source in an attempt to receive the multimedia contents through an alternative path that avoids the congestion or the total interruption of communications.
The STM (partially published in [15]) sends the SS and the QoSAM information about the bandwidth for each connection; it obtains this information from its congestion control protocol. With this information (and more), the SS schedules the delivery of the videos to the client nodes, and the QoSAM detects network failures in communications. The SS uses the video data transport offered by the STM and whose reception is managed by the SRM. The features of this data transport are: reliable, TCP-friendly, and supports late data selection (some of the packages sent to client nodes may be removed if they have not reached the cable yet). To support many client nodes/connections and high loads of traffic, the STM uses a minimum amount of resources per connection and uses the available network outgoing bandwidth to its maximum (so that this critical resource does not go to waste). This module was implemented in a general-purpose operating system (Linux) and, to be independent from specific kernel versions, it was fully developed within user space.

3.2 Failure detection and recovery

Failure detection is carried out by the QoSAM and uses two detection mechanisms that work together to provide tolerance in the case of network failures or server nodes that are down. The main detection mechanism is located at the server node and is responsible for detecting any type of network failure. Upon detection of a failure, it communicates with the client node. A protection mechanism, located in the client node, detects if the server node is down or if there is a total (or almost total) loss of communications (which would prevent the main mechanism in the server node to communicate failures to the client node).
This protection mechanism consists in sending, every given period of time and through the stream control channel, messages (heartbeat) to the other end asking if it is “live”. The main mechanism works as follows. Every peer serving another peer agrees to ensure a minimum communication bandwidth. Bandwidth samples, provided by the STM, are used to asses communication bandwidth. When this bandwidth is below the agreed minimum, an early warning is sent to the client node indicating that it should start planning its recovery. After a reasonable time, the server node either confirms the failure to the client node or sends a negative of the failure.
The failure recovery process is simultaneous with the failure detection process, since it begins when the client node receives the failure early

warning, and not when the failure has been confirmed. This overlapping produces a time gain that is essential to maintain service quality in content delivery. When an early warning is received, the client node looks for and reserves resources in an alternative server node in case the service needs to be migrated. If then a failure negative is received, the client node cancels the resource reservation in the alternative server node. If the failure is confirmed, the client node renegotiates a lower transmission rate with the server node that is affected by the failure and a higher one with the other server nodes. However, a renegotiation could only be considered if the new minimum rate ensured by the affected server node is greater than a certain threshold value (e.g., 10% of the maximum bit rate of the vide) and the remaining server nodes accept a renegotiation with a greater transmission rate. If an agreement is reached, the resource reservation at the alternative server node is cancelled; otherwise, if no agreement is reached, the service is migrated.

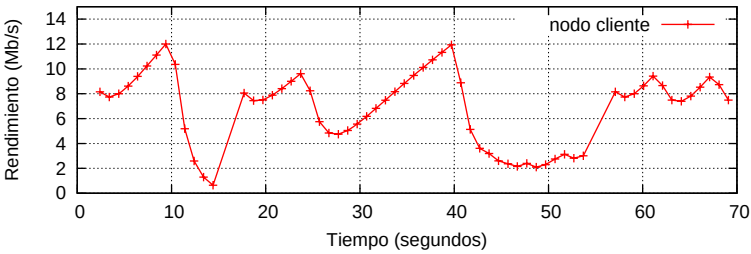


Figure 2: *Behavior of the network failure detection and recovery mechanism*

Experiment. To show the efficacy of the network failure detection and recovery techniques proposed, a simple experiment (due to space issues and for clarity's sake) was carried out using the Ns-2 network simulator. The experiment is configured as two server nodes and one client node, where the client node receives content from only one server node at a time. Figure 2 shows the performance curve assumed by the server nodes for the client node. At second 10, a link is down (for 20 seconds), which prevents the communication between the client node and its server node. The protection mechanism at the client node detects the failure and the service is migrated; note that the main mechanism of the server node was never able to inform the client node about this failure due to the total loss of communication. Recovery takes place at time 15.25. At time 40, the communication of the client node is affected by network congestion. The server node waits a reasonable time and, since the communication does not improve, it orders the migration of the service before the client suffers any interruptions in video visualization.

3.3 Multi-source stream management and synchronization

The peers, through the SeS, periodically determine the video segments to download from the server nodes, based on the adopted prefetching policy, such as: sequential, random, local-rarest, hybrid of sequential and local-rarest [3], hybrid of sequential and associations based on contents between different video segments [2], etc. To prevent wasting reserved communication bandwidth, the client node must deliver new segment requests before the server nodes finish transmitting the previous ones. Temporary connections should not be extremely brief to avoid a control overload (establishing the connection, bandwidth negotiation, etc); thus, if segment size is small, more than one segment at a time should be requested.

When a peer changes its reproduction point or its status (normal play, fast play, prefetch, and pause), it should send this information to its server nodes [2]. The SS module of a server node calculates the reproduction deadline for the following segment to transmit to a client node, based on the following equation: $deadline = tpmss(upr + ttdupr)$. In this equation, *tpmss* is the reproduction time associated with the first frame of the following video segment to transmit to the client node, *upr* is the last reproduction time reported, and *ttdupr* is the time elapsed since the last recorded reproduction time was determined. Based on the reproduction state and the deadline, the SS determines the urgency for data of its client nodes and schedules the delivery of the multimedia content (see experimental results in [14]). As transmissions finish, the SS updates the corresponding list of segments to transmit by removing those that were already delivered.

When a bandwidth renegotiation or a service migration is carried out because of the deterioration of the communication between a server node and a client node, the client node will redistribute its requests for (still undelivered) video segments to all its server nodes. The purpose of this new distribution of segment requests is redistributing the segments that were assigned to the server node with which communications are affected.

3.4 Establishing connections in collaboration groups

Within a collaboration group, the number of permanent parallel connections that a client node can have are limited to avoid overloads caused by control data management. Thus, once the collaboration group to which a given peer will belong is determined, the peer should select with which server nodes it will establish permanent connections to download video segments. This selection is carried out by the NCM. This module is also responsible for implementing an admission control for temporal connection requests received from client nodes. The admission control accepts or rejects requests based on

[2] Depending on the specific policy of each VoD system, certain status or reproduction point changes could cause the transfer of the peer to another group of collaborating peers.

a policy that considers the number of available connections and the deadline of the segments requested.
To establish the permanent connections, the following scheme is defined. The five peers with path diversification and maximum communication bandwidth are selected, non-source peers being preferred over source peers. Then, a transmission rate (less or equal to the one offered) is reserved at each server node so that the sum of the individual rates is equal to the maximum rate of the video. If all peers are non-source peers, a transmission rate that is proportional to the rate offered, with respect to the total rate offered by the entire group (of 5 peers), is reserved at each of them. On the other hand, if there are some source-peers, a 10% of the necessary rate is reserved at each source peer, and the rest is assigned to non-source peers, the value assigned to each one being proportional to the individual rate offered with respect to the rate offered by all non-source peers combined. If the demand is not covered, the rate assigned to each source peer is increased proportionally to their assignment. When possible, source peers should be replaced by non-source peers.
This scheme requires information regarding the available bandwidth for communications. This information is obtained by the STM during its use, which means it will be unknown at the beginning, but known when the system reaches a steady state.
Peers with different communication paths are preferred so that, in case of a network failure, the least possible number of connections is affected. The selection of different paths is carried out by means of the scheme presented in [16], which was designed to be implemented in a content distribution network (CDN). This technique uses ping commands to infer path structure through heuristics.

4. Conclusions and future work

We have analyzed the impact of implementing an Internet VoD system, were the service offered by the network cannot ensure quality of service. Our proposal is a network failure-tolerant P2P-VoD system that includes an architecture and mechanisms that offer an integral solution for managing communication degradation, ensuring system users a visualization with no interruptions or image quality degradation.
In this architecture, peers receive the multimedia content from multiple transmitting peers, reducing the impact of network failures balancing the load. Our solution proposes a novel scheme for failure detection that is located in the server nodes, which gives them the freedom to dynamically increase or decrease transmissions without the client nodes detecting failures due to low data reception rates. This allows planning content delivery to the clients to ensure service quality. Recovery in case of a network failure is based on the renegotiation of transmission rates, the migration of the services, and a new, suitable selection scheme of server nodes. Failure

detection overlaps failure recovery to save time and therefore reduce the probability of interruptions during video visualization.
We believe that our failure tolerance scheme can substantially improve the service of already built P2P-VoD systems, increasing user satisfaction. Future works are focused on finishing the implementation of a real prototype and a system simulation prototype for the system proposed and carrying out larger scale experiments.

References

1. Boutremans, C., Iannaccone, G., Diot, C., Impact of link failures on voip performance, in NOSSDAV '02, Proceedings of the 12th international workshop on Network and operating systems support for digital audio and video, New York, USA, ACM, 2002.
2. He, Y., Liu, Y., Vovo: Vcr-oriented video-on-demand in large-scale peer-to-peer networks, IEEE Trans. Parallel Distrib. Syst. 20(4), 2009.
3. Xiaoyuan Yang, P.R., Kangaroo, Video seeking in p2p systems, in Proc. of IPTPS 2009, Boston, MA, USA, 2009.
4. Sultan, F., Bohra, A., Iftode, L., Service continuations: An operating system mechanism for dynamic migration of internet service sessions, in Proc. Symposium in Reliable Distributed Systems (SRDS), Oct. 2003.
5. Karol, M., Krishnan, P., Li, J., Voip network failure detection and user notification. Computer Communications and Networks, ICCCN 2003. Proceedings. The 12th International Conference on, 20-22 Oct. 2003.
6. Nguyen, T., Zakhor, A., Multiple sender distributed video streaming. IEEE transactions on multimedia 6, 2004.
7. Magharei, N., Rejaie, R., Adaptive receiver-driven streaming from multiple senders. Multimedia Systems 11(6), 2006.
8. Guo, Y., Suh, K., Kurose, J., Towsley, D., P2cast: peer-to-peer patching scheme for vod service, in WWW '03: Proceedings of the 12th international conference on World Wide Web, New York, USA, ACM, 2003.
9. Do, T.T., Hua, K.A., Tantaoui, M.A., Robust video-on-demand streaming in peer-to-peer environments. Comput. Commun. 31(3), 2008.
10. Maharana, A., Rathna, G., Fault-tolerant video on demand in rserpool architecture. International Conference on Advanced Computing and Communications (ADCOM), 20-23 Dec. 2006.
11. Anker, T., Dolev, D., Keidar, I., Fault tolerant video on demand services. Proceedings of 19th IEEE International Conference on Distributed Computing Systems, 1999.
12. Kim, H., Kang, S., Yeom, H., Node selection for a fault-tolerant streaming service on a peer-to-peer network. Multimedia and Expo, IEEE International Conference on 2, 2003.
13. Antoniou, Z., Stavrakakis, I., An efficient deadline-credit-based transport scheme for prerecorded semisoft continuous media applications. IEEE/ACM Trans. Netw. 10(5), 2002.

14. Balladini, J., Souza, L., Suppi, R., Un planificador de canales lógicos para un servidor de VoD en internet, XII Congreso Argentino de Ciencias de la Computación (CACIC 2006), 2006.
15. Balladini, J., Souza, L., Suppi, R., A network scheduler for an adaptive VoD server, in E-Business and Telecommunication Networks, Communications in Computer and Information Science, Volume 9, Springer-Verlag Berlin Heidelberg, 2008.
16. Guo, M., Zhang, Q., Zhu, W., Selecting path-diversified servers in content distribution networks. Global Telecommunications Conference. IEEE GLOBECOM '03. 6, 2003.

Dynamic Scheduling in Heterogeneous Multiprocessor Architectures. Efficiency Analysis.

LAURA C. DE GIUSTI[1], MARCELO NAIOUF[1], FRANCO CHICHIZOLA[1], EMILIO LUQUE[2], ARMANDO E. DE GIUSTI[1]

[1]Instituto de Investigación en Informática LIDI (III-LIDI) – School of Computer Sciences – Universidad Nacional de La Plata. Argentina
[2]Universidad Autónoma de Barcelona (UAB) - Computer Architecture and Operating System Department (CAOS) Spain
{ldgiusti, mnaiouf, francoch}@lidi.info.unlp.edu.ar, emilio.luque@uab.es, degiusti@lidi.info.unlp.edu.ar

Abstract. *A MPAHA (Model for Parallel Algorithms on Heterogeneous Architectures) model that allows predicting parallel application performance running over heterogeneous architectures is presented. MPAHA considers the heterogeneity of processors and communications.*
From the results obtained with the MPAHA model, the AMTHA (Automatic Mapping Task on Heterogeneous Architectures) algorithm for task-to-processors assignment is presented and its implementation is analyzed.
DCS_AMTHA, a dynamic scheduling strategy for multiple applications on heterogeneous multiprocessor architectures, is defined, and experimental results focusing on global efficiency are presented.
Finally, current lines of research related with model extensions for clusters of multicores are mentioned.

Keywords: *Dynamic Scheduling-Parallel Algorithm Model-Distributed Architectures-Heterogeneity.*

1. Introduction

The problem of *automatic task-to-processor mapping* in heterogeneous architectures is highly complex [1]. This complexity can be briefly represented considering the two main elements relating the parallel application to the supporting architecture: each node processing capacity and the cost of inter-processor communications in time [2].
In Computer Science, models are used to describe real entities such as processing architectures and to obtain an "abstract" or a simplified version of the physical machine, capturing crucial characteristics and disregarding minor details of the implementation [3][4]. In the case of parallel systems, the most currently used architectures –due to their cost/performance relation– are heterogeneous clusters and multiclusters; for this reason, it is really important

to develop a model that fits the characteristics of these platforms. An essential element to be considered is the potential heterogeneity of processors and communications among them, which adds complexity to the modeling [5][6].
At present, there are different graph-based models to characterize the behavior of parallel applications in distributed architectures [7]. Among them, we can mention TIG (Task Interaction Graph), TPG (Task Precedence Graph), TTIG (Task Temporal Interaction Graph) [8], TTIGHA (Task Temporal Interaction Graph on Heterogeneous Architectures) [9] and MPAHA (Model on Parallel Algorithms on Heterogeneous Architectures) [10].
Once the graph modeling the application has been defined, the *scheduling* problem [11] is solved by an algorithm that establishes an automatic mechanism to carry out the task-to-processor assignment, searching for the optimization of some execution parameter (usually, time) [12]. Among the known mapping/scheduling algorithms, we consider AMTHA (Automatic Mapping Task on Heterogeneous Architectures), a mapping algorithm, to carry out the assignment of tasks of the application to the processors of the architecture [10]. This algorithm considers the heterogeneous characteristics of the architecture taken into account in the MPAHA (Model on Parallel Algorithms on Heterogeneous Architectures) model [10].
Usually, scheduling/mapping algorithms are used to obtain the best assignment of the processes that make up an application to the processors of the architecture in which it will be run. In this paper, the DCS_AMTHA (Dynamic Concurrent Scheduling) algorithm to carry out the scheduling of multiple parallel applications on heterogeneously distributed architectures (cluster) is defined. This algorithm is based on AMTHA and the goal is to optimize the efficiency achieved by the whole system.
In Section 2, the MPAHA model is briefly described. In Section 3, the AMTHA scheduling algorithm is dealt with. In Section 4, DCS_AMTHA scheduling algorithm is detailed. Section 5 describes the experimental work, and Section 6 presents results of speedup and efficiency of DCS_AMTHA for different processing architectures and working loads. Finally, Section 7 presents the conclusions and the future lines of work.

2. MPAHA Model

The MPAHA model is based on the construction of a directed graph *G (V, E)*, where:

- *V* is the set of nodes representing each of the tasks T_i of the parallel program.
- *E* is the set of edges representing each of the communications between the nodes of the graph.

2.1 Model parameters

Nowadays, most applications are formed by a set of tasks that perform different functions and that communicate among themselves to exchange information at any point. Each of these tasks can in turn be split in various blocks (subtasks) which do not communicate with other blocks from other tasks.

In the first parameter of the graph (V), each node represents a task T_i of the parallel program, including its subtasks (St_j) and the order in which they should be executed to perform the task. If there is a heterogeneous architecture available, the computation times for each of the composing processors should be taken into account. That is, node i ($V_i \in V$) stores the computation time at each of the different types of processor for each subtask of task T_i. Therefore, V_i *(s,p) = execution time of subtask s in processor type p.*

In the second parameter of the graph (E), the edges represent the communications exchanged between each pair of tasks. In this set, an edge A between two tasks T_i and T_j contains the communication volume (in bytes), the source subtask ($\in T_i$), and a target subtask ($\in T_j$). That is, $E_{i,j}$ *(o,d) = communication volume between the source subtask ($o \in T_i$) and the target subtask ($d \in T_i$).*

It should be noted that, given the heterogeneity of the interconnecting network, instead of representing the time required for the communication, the corresponding communication volume between two subtasks is represented. Fig. 1 shows an example of a graph generated with this model.

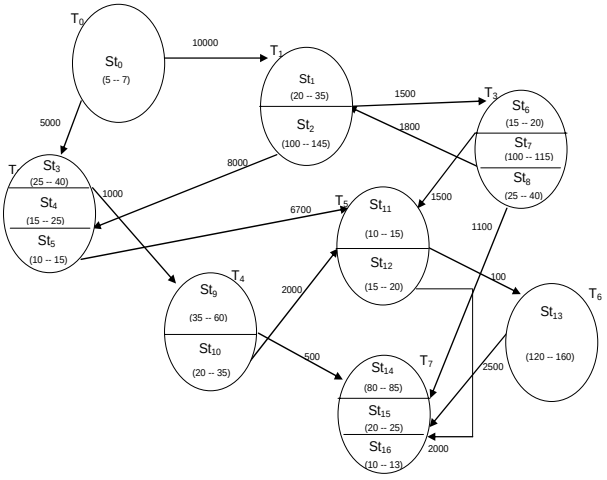


Figure 1: *Example of a graph generated by the model.*

3. AMTHA mapping algorithm

AMTHA is a static mapping algorithm that is applied to the graph generated by the MPAHA model. It allows determining the assignment of tasks to the processors of the architecture to be used to minimize the execution times of the application in that architecture. This algorithm must also provide the order in which subtasks (forming the task) assigned to each processor should be executed (*task scheduling*). AMTHA considers an architecture with a limited number of heterogeneous processors. As regards the interconnecting network, the algorithm also considers that bandwidth and transmission speed can be heterogeneous.
The AMTHA algorithm uses the values of graph G generated by the MPAHA model. These values are: the time required to compute a subtask in each type of processor, the communication volume with adjacent processors, and the task each subtask belongs to.
The AMTHA algorithm assigns one task at a time until all tasks have been assigned. Figure 2 shows the pseudo-code with the main steps of the algorithm:
When the execution of the algorithm ends, all the tasks have been assigned to one of the processors and the order in which the subtasks forming the tasks assigned to these processors will be executed has also been determined.

Calculate ***rank*** for each task.
Whereas (not all tasks have been assigned)
1. Select the next task ***t*** to assign.
2. Chose the processor ***p*** to which task ***t*** should be assigned.
3. Assign task ***t*** (selected in step 1) to processor ***p*** (selected in step 2).
4. Update the ***rank*** of the tasks involved in step 3.

Figure 2: *Pseudo-code with the basic steps of the AMTHA algorithm.*

The following paragraphs describe each of the three steps followed during the execution of the AMTHA algorithm.

3.1 Calculating the rank of a task

Given a graph *G*, the rank of a task *Rk(T)* is defined as the sum of the average times of the subtasks forming it and that are ready for execution (all predecessors have already been assigned to a processor and are already there). Equation 1 expresses this definition:

$$Rk(T) = \sum_{i \in L(T)} W_{avg}(St_i) \quad (1)$$

where:

L(T) is the set of subtasks that are ready for task *T*.

W_{avg} *(St)* is the average time of subtask *St*. The average time is calculated as shown in Equation 2.

$$W_{avg}(St_i) = \frac{\sum_{p \in P} V_{St_i}(type\ of\ \ processor\ \ p)}{\#P} \quad (2)$$

where *P* is the set of processors present in the architecture and *#P* is the number of processors forming this set.

3.2 Selecting the task to execute

After obtaining the *rank* of each application task, the task that maximizes it is selected. If there are two or more tasks that have the same maximum value, the algorithm breaks this tie by selecting the one that minimizes the total execution time average for the task. Equation 3 shows this calculation:

$$Tavg\ (T) = \sum_{i \in T}\ W_{avg}(St_i) \quad (3)$$

3.3 Selecting the processor

Selecting the processor involves choosing the computer within the architecture that minimizes the execution time when the selected task is assigned to that processor.

In order to understand how the time corresponding to processor *p* is calculated, the fact that each processor keeps a list of subtasks LU_p that were already assigned to it and that can be executed (all its predecessors are already placed), and another list that contains those subtasks that were assigned to *p* but whose execution is still pending LNU_p (some of their predecessors have not been placed yet) must be taken into account.

Therefore, to calculate which processor *p* will be selected, two possible situations are considered:

1. All subtasks of task *t* can be placed in *p* (that is, all its predecessors have been placed).
2. Some of the subtasks of *t* cannot be placed in *p* (this happens when some predecessor of this subtask has not been placed).

In the first case, the time *Tp* corresponding to processor *p* is given by the moment in which *p* finishes the execution of the last subtask of *t*. However, in the second case, the time *Tp* corresponding to processor *p* is given by the time when the last subtask of LU_p will finish plus the addition of all execution times in *p* for each of the subtasks on LNU_p.

3.4 Assigning the task to the selected processor

When assigning a task T to a processor p, there is an attempt to place each subtask St_k belonging to T to the processor at a moment when all the adjacent subtasks have already finished (including the predecessor subtask within T, if any). This can be a free interval between two subtasks that have already been placed in p, or an interval after them. If subtask St_k cannot be placed, it is added to the LNU_p list. Each time a subtask St_k is added to the LU list of one of the processors, an attempt is made to place all the predecessors belonging to the already assigned tasks.

3.5 Updating the rank value of pending tasks

The first action within this step consists in assigning *-1* as rank value to the task t that has been assigned to processor p. The reason for this is to prevent task t from being re-selected for assignment.
Also, the following situation is considered in this step: for each subtask St_k placed in step 3.4, the need to update the rank of the tasks to which successor subtasks St_{succ} of St_k belong is analyzed; that is, if all predecessors of St_{succ} are already placed, then the rank of the task St_{succ} belongs to is updated by increasing it by $W_{avg}(St_{succ})$.

4. Dynamic Concurrent Scheduling (DCS_AMTHA) Algorithm

Given a parallel application, the AMTHA algorithm generates an assignment of its tasks so as to achieve an efficient use of the heterogeneous architecture in which it will be executed [10]. As the multiprocessor architectures increase the number of processors, a way to obtain high efficiency is to increase the volume of parallel work, simultaneously dealing with multiple applications. In this work, DCS_AMTHA algorithm is developed in order to carry out the scheduling of multiple applications on a heterogeneously distributed architecture.
This algorithm allows overlapping two or more applications when using the architecture. Application A_i scheduling is carried out using the AMTHA algorithm; the tasks that make up A_i can be assigned to empty gaps generated by the assignment of applications prior to A_i. In this case, A_i time zero ($\mathbf{t_0}$=time on which execution starts) is the time on which A_i reaches the system (S_i).
The *DCS_AMTHA* algorithm allows the reassignment of tasks that are already assigned to a processor but not being run yet.
Application A_i is assigned through the AMTHA algorithm, considering $\mathbf{t_0}$ as the moment of reaching S_i. Those tasks belonging to previously assigned

applications ($A_0..A_{i-1}$) whose starting time in the scheduling occurs after de-assigning S_i, and together with the A_i tasks, are part of the new scheduling process.
Among the most outstanding features of this algorithm, the following are to be pointed out:

- At some moment, it requires process migration or reconfiguration of location of each application task. The generated communication overhead depends on the physical distribution of processors.
- It prioritizes to obtain *minimum system end time*, not regarding the order in which applications reach the system.

5. Experimental Work

In previous works, *DCS_AMTHA* was compared to two alternative algorithms for scheduling multiple applications (*SCS_AMTHA* and *SS_AMTHA*). As a result, it was concluded that the *DCS_AMTHA* algorithm yields best response times for the whole system, that is, the architecture is used in a more efficient way [13].
Based on these results, the behavior of the algorithm after scaling the architecture and/or the number of applications in the system is analyzed.
A set of applications was selected, in which each of them varies in terms of the number of application tasks, task size, number of subtasks making up a task, and communication volume among subtasks. In all the applications, the total computing time exceeded that of communications (coarse grained application). In this work, 150 synthetic applications were generated and placed on a queue *Q*. Each application has the following characteristics:

- The number of tasks in the application is between 25 and 50.
- The number of subtasks in each application task is between 5 and 10.
- The computation time for each subtask is between 100 and 650 seconds.
- The degree of interaction between the subtasks of different tasks is high.

To run the applications, a heterogeneous architecture formed by two clusters interconnected by a switch is used. The first one (*cluster 1*) is composed by 25 processors (Pentium IV 2.4Ghz, 1Gb RAM), and the second one (*cluster 2*) is composed by 25 processors (Celeron, 2 GHz, 128Mb RAM). The connection is made through a 100-Mbit Ethernet network. This architecture was chosen so that the clusters forming it are of different characteristics in terms of the processors' computing power.
Various tests were carried out, identified by two parameters: the computing power of the subset of processors used (*WPT*), and the number of applications used (*AN*). That is, the test *WPT - AN* uses the first *AN* applications of *Q* and runs them over an architecture whose total computing power is *WPT*. The applications of the experiment reach it at different times

(between 100 and 600 seconds difference between two successive applications), so that the following is observed: $S_i < S_{i+1} \quad \forall i \in [0..148]$

6. Results

To assess the behavior of the *dynamic concurrent scheduling* algorithm (*DCS_AMTHA*) when the architecture and/or the number of applications in the system are scaling, the *speedup and efficiency* parameters are analyzed (in this case, over heterogeneous architectures).
The speedup metrics is used to analyze the algorithm performance in the parallel architecture as indicated in Equation (4).

$$Speedup = \frac{SequentialTime}{ParallelTime}. \quad (4)$$

In heterogeneous architectures, the "*Sequential Time*" is given by the time of the best sequential algorithm executed in the machine with the greatest calculation power. In this multiple application case, the time each application requires is added. "*Parallel Time*" is the end time for the entire system.
To assess how good the speedup obtained is, efficiency is calculated. To this aim, the speedup obtained is compared with the total computing power (*WPT*) of the architecture upon which work is being carried out (which determines the theoretical speedup), as indicated in Equation (5).

$$Efficiency = \frac{Speedup}{WPT}. \quad (5)$$

The total computing power considers the relative calculation power of each machine with respect to the power of the most powerful machine. The *WPT* is calculated with Equation (6), where *#P* is the number of machines of the architecture used, and WP_i is the relative calculation power of machine *i* regarding the best machine power.

$$WPT = \sum_{i=1}^{\#P} WP_i. \quad (6)$$

Figure 3 shows, for the most representative tests, the speedup achieved by the whole system when scheduling was done with the DCS_AMTHA algorithm. As it can be observed in the graph, speedup increases as the number of applications *NA* increases, with a tendency towards stabilization from *NA*=120.
On the other hand, it can also be seen that speedup increases with computing power *WPT* up to a point where increasing this power does not yield a significant improvement ($WPT \approx 30$).

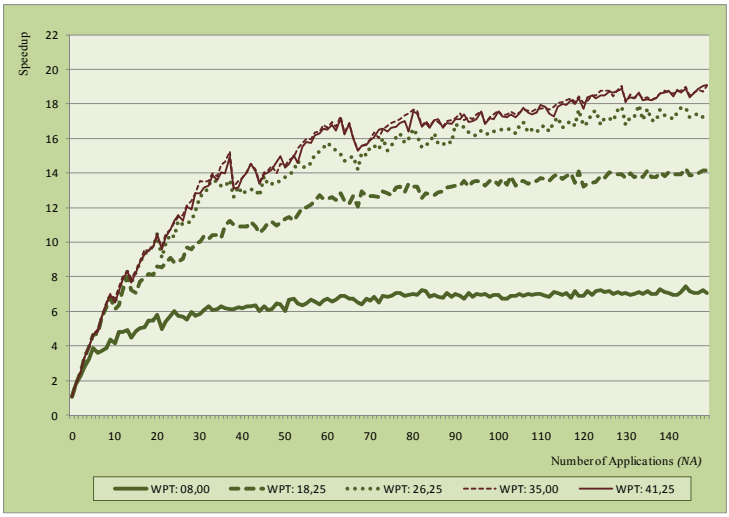


Figure 3: *Speedup obtained with each test.*

Figure 4 shows the efficiency achieved for the same tests represented in Fig. 3. The same as with speedup, efficiency increases with *NA*, and it stabilizes after *NA*=120. However, unlike speedup, efficiency decreases as the architecture grows (*WPT*).

By combining both results, the limit up to which *WPT* can be increased to achieve relevant improvements in response times can be inferred. For these particular tests, this limit is given by an architecture with *WPT* = 30.

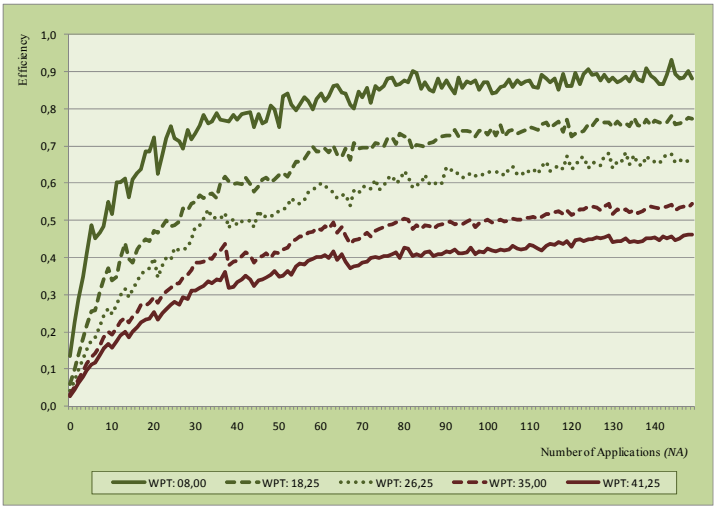


Figure 4: *Efficiency obtained with each test.*

7. Conclusions and Work Guidelines

DCS_AMTHA, a dynamic scheduling strategy for multiple applications on heterogeneous multiprocessor architectures, is defined. It is based on AMTHA task-to-processor assignment on heterogeneous architectures.
Experimental results for different processing configurations and different working loads are analyzed, focusing on the global efficiency obtained.
Within the future work lines, research will be carried out on the assessment of the scheduling algorithm presented (*DCS_AMTHA*) on clusters of multicore processors, taking into account the particular features of such architectures.

References

1. Grama, A., Gupta, A., Karypis, G., Kumar, V., "An Introduction to Parallel Computing. Design and Analysis of Algorithms", 2nd Edition, Pearson Addison Wesley, 2003.
2. Kalinov, A., Klimov, S., "Optimal Mapping of a Parallel Application Processes onto Heterogeneous Platform", Proc. of 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), IEEE CS Press, 2005.
3. Leopold, C., "Parallel and Distributed Computing. A survey of Models, Paradigms, and Approaches", Wiley, New York, 2001.
4. Attiya, H., Welch, J., "Distributed Computing: Fundamentals, Simulations, and Advanced Topics. 2nd Edition", Wiley-IEEE, New Jersey, 2004.
5. Topcuoglu, H., Hariri, S., Wu, M., "Performance-Effective and Low-Complexity Task Scheduling for Heterogeneous Computing", IEEE Transactions on Parallel and Distributed Systems, vol. 13, 2002.
6. Goldman, "Scalable Algorithms for Complete Exchange on Multi-Cluster Networks", CCGRID '02, IEEE/ACM, Berlin, 2002.
7. Roig, C., Ripoll, A., Senar, M. A., Guirado, F., Luque, E., "Modelling Message-Passing Programas for Static Mapping", Euromicro Workshop on Parallel and Distributed Processing (PDP'00), IEEE CS Press, USA, 1999.
8. Roig, C., Ripoll, A., Senar, M., Guirado, F., Luque, E., "Exploiting knowledge of temporal behavior in parallel programs for improving distributed mapping", EuroPar 2000, LNCS, vol. 1900, Springer, Heidelberg, 2000.
9. De Giusti, L., Chichizola, F., Naiouf, M., De Giusti, A., "Mapping Tasks to Processors in Heterogeneous Multiprocessor Architectures: The MATEHa Algorithm", International Conference of the Chilean Computer Science Society, IEEE CS Press, 2008.
10. De Giusti, L., "Mapping sobre Arquitecturas Heterogéneas", PhD Dissertation, Universidad Nacional de La Plata, 2008.
11. Cuenca, J., Gimenez, D., Martinez , J., "Heuristics for Work Distribution of a Homogeneous Parallel Dynamic Programming Scheme on Heterogeneous

Systems", Proceeding of the 3rd International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (HeteroPar'04), IEEE CS Press, 2004.
12. Cunha, J. C., Kacsuk, P., Winter, S., "Parallel Program development for cluster computing: methodology, tools and integrated environments", Nova Science Pub., New York, 2001.
13. De Giusti, L., Chichizola, F., Naiouf, M., De Giusti, A., "Scheduling Strategies in Heterogeneous Multiprocessor Architectures using AMTHA Algorithm (DCS_AMTHA, SCS_AMTHA, SS_AMTHA)", Technical Report, March 2009.

A Multipath Routing Method for Tolerating Permanent and Non-Permanent Faults*

GONZALO ZARZA[1], DIEGO LUGONES[1], DANIEL FRANCO[1], EMILIO LUQUE[1]

[1] Computer Architecture and Operating Systems Department,
Universitat Autònoma de Barcelona, Spain
{gonzalo.zarza, diego.lugones, daniel.franco, emilio.luque}@uab.es

Abstract. *The intensive and continuous use of high-performance computers for executing computationally intensive applications, coupled with the large number of elements that make them up, dramatically increase the likelihood of failures during their operation. As interconnection networks are critical parts of such systems, network faults have an extremely high impact because most routing algorithms are not designed to tolerate faults. In such algorithms, just a single fault may stall messages in the network, prevent the finalization of applications, or lead to deadlocked configurations.*
This work focuses on the problem of fault tolerance for high-speed interconnection networks from the design of a fault-tolerant routing method intended to treat a large number of dynamic faults (permanent and non-permanent). To accomplish this task our method takes advantage of communication path redundancy by means of a multipath routing approach. Experiments show that the method allows applications to finalize their execution in the presence of several faults, with an average performance value of 97% compared to the fault-free scenarios.

Keywords: *Interconnection networks, fault tolerance, adaptive routing.*

1. Introduction

High-performance computing systems have opened a trend in modeling life style and daily behavior of modern societies by means of applications and services such as molecular dynamics simulations, DNA sequencing, weather forecasting, and geological activity studies. Even a simple Google search is based on high-performance computer (HPC) systems [1].
The steady increase in complexity and number of components of HPC systems leads to significantly higher failure rates. Because of this, and to the long execution times of computationally intensive applications, various computer systems show a Mean Time Between Failures smaller than the

* Supported by the MEC-Spain under contract TIN2007-64974.

execution time of such applications. This means that at least one failure will probably occur during their execution.
Questions arise from the analysis of these situations such as: how do failures (and their duration) affect these HPC systems? Are such systems able to maintain their operation and performance standards in spite of failure occurrences? If they are not, what should the solution be? What are the best options to achieve fault tolerance and system service continuity?
Undoubtedly, system performance is closely tied to the robustness of the fault tolerance mechanisms of the network. For this reason, high-speed interconnection networks (HSINs) must avoid performance degradations and allow applications to finalize their executions even in the presence of multiple time-varying failures.
We focus this work on the problem of fault tolerance for HSINs because of their primary role as the linking element of HPC systems. There are three main approaches that could be chosen to achieve this goal: component redundancy, network reconfiguration, and fault-tolerant routing algorithms [2]. The component redundancy approach is often used in some systems but the high extra cost of redundant spare components is an important drawback. The second approach stops the network and reconfigures routing tables in case of a network fault in order to adapt them to the new topology after the failure. This approach is very flexible and powerful but at the expense of killing network performance. Routing algorithms designed for fault tolerance looks for alternative paths when a fault disables the original path used to communicate a pair of source-destination nodes. This last approach could be outlined as the most interesting and suitable option but, at the same time, the design of fault-tolerant routing algorithms implies great challenges.
For this reason, we focus our work in the problem of fault tolerance for HSINs by means of an adaptive routing approach. In this paper we present a method that exploits communication path redundancy through an adaptive multipath routing policy with the aim of solve a certain number of permanent and non-permanent link failures. The method is based on source-destination communication path information and consists of three phases. The first phase is responsible for on-line fault diagnosis and uses physical level monitoring at intermediate nodes along the source-destination path. If a message encounters a faulty link as it progresses towards its destination, the second phase immediately reroutes the message to the destination by an alternative path. In the third and last phase, the source node is notified about the link failure in order to disable the faulty path and also to establish new paths for the following messages to be sent to that destination. At a first stage, failures are considered and treated as non-permanent. If a failure persists over time, its status changes from non-permanent to permanent.
In a previous work, we have introduced a method capable of dealing with dynamic faults appearing at random during network operation [3]. The method allows the system to remain operational while measures are taken to circumvent the faulty components but it was not designed to treat transient and intermittent faults.
The main contribution of this work is the ability to treat permanent and non-permanent dynamic faults while treating network congestion caused by

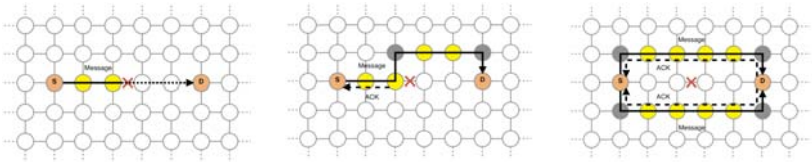
faults. Moreover, the method notifies the event of a failure only to sources nodes that try to send messages by faulty links. This action reduces the overall traffic overhead because it avoids the distribution of status-information over the entire network.
Experimental results show average performance over 97% for a set of test scenarios with several faults in a 1024 nodes bi-dimensional torus network for standard traffic patterns.
The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 describes the multipath routing method for tolerating dynamic faults, details its behavior and explains the treatment of non-permanent faults. Evaluation environment, test scenarios and results are presented in Section 4. Finally, some conclusions and future work are drawn in Section 5.

2. Related Work

Many research studies have been published in the field of fault tolerance for interconnection networks throughout the past few decades. The vast majority assumes the existence of diagnostic techniques, and focus on how the availability of information obtained from these diagnoses can be used to develop robust and reliable routing algorithms. This means that diagnoses problem is not addressed by those methods; static fault models are used which means that all the information about faults need to be known in advance; and they assume that there are mechanisms to correctly distribute this information to network nodes.
Some interesting work in the area of component redundancy is presented in [4]. On the side of the network reconfiguration approach, there are works based on deterministic routing methodologies for tori and meshes [5], and others that achieve error detection and recovery for *k*-ary *n*-cube topologies [6]. The latter proposal tolerates non-permanent faults but relies on table based routing strategies and packet injection is required to be temporarily stopped during a global reconfiguration phase.
Some routing strategies using static fault models have been proposed for different network topologies like *k*-ary *n*-tree [7] and direct networks [8]. The authors of the latter work suggest the use of intermediate nodes to circumvent faults but fault detection, information distribution, and checkpoints are assumed to be provided by the interconnection network. The use of intermediate nodes was first proposed by Valiant for the purpose of traffic balancing [9].
One of the only work capable of dealing with dynamic faults was proposed in [10] as an evolution of [11]. The method is based on a variation of the turn-model, needs five virtual channels to support fully adaptive routing, and packet drops are allowed under certain situations. In addition, status-information must be distributed through control messages and rerouting decisions must be taken based on such information. All these actions have an extra cost.

3. Multipath Fault-tolerant Routing

The multipath fault-tolerant routing method presented in this paper is largely based on the Distributed Routing Balancing (DRB) concept introduced in [12].
The current proposal differs from other works found in literature through its combined features. More precisely, it is able to support dynamic faults and at the same time to provide a multipath routing approach. In addition, the method introduces a novelty approach addressing system service continuity and performance degradation at the same time. To accomplish these tasks, we use a non-global scheme to distribute real-time path information in order to choose the best source-destination paths. Furthermore, our method does not require global reconfiguration or stopping packet injection at any time, using a limited number of virtual channels.
Conceptually, our proposal consists of three steps. In the first step, the failure in the original path is discovered when a message tries to use a faulty link, as could be seen in Fig. 1(a). In the second step, shown in Fig. 1(b), the message is rapidly rerouted to its destination through an alternative path to allow system service continuity. This action is intended to be a fast and temporary response to failures and may not be the optimal solution. For this reason, we include a third step that seeks to reconfigure new paths to improve performance and ease routing paths. In this step, shown in Fig. 1(c), the source node is notified about the discovery of a link failure in the path, in order to disable the faulty path and reconfigure new paths for the following messages. Once those new paths have been configured, their latency values are recorded and sent back to the source node (from the destination), in order to calculate the number of alternative paths that must be used according to network traffic conditions.



(a) Step 1 (b) Step 2 (c) Step 3

Figure 1: *Method behavior example.*

The configuration and use of simultaneous alternative paths between source and destination nodes allow the method to deal with link failures. Additionally, the use of such simultaneous paths provides path redundancy and allows performance improvements by means of communication balancing and distribution.
Alternative paths are created using intermediates nodes that can be used for two different purposes. One of these purposes is to allow source-destination path segmentation when encountering faults on the fly, in order to circumvent

dynamic faults as shown in Fig. 1(b). The second purpose of intermediate nodes is to be used as scattering and gathering areas from source and destination nodes when knowing the location of faults, as shown in Fig. 1(c). From these scattering/gathering areas, alternative paths are established based on intermediate nodes carefully chosen to ensure that they are not in the original path, using the available links in routers. The set of alternative paths between each source-destination pair is called *multipath* or *metapath* [12].
Intermediate nodes are chosen according to their distance to the nodes that have detected faults or to the source and destination nodes, as appropriate. The nodes of 1-hop distance are considered first, then nodes of 2-hop distance, etc. If necessary, e.g. if a link fails, *multipaths* could be expanded in order to include additional alternative paths. This case is shown in Fig. 1(c), where two alternative paths were included.
In this work, we consider only two intermediate nodes so that the path is divided in three segments: the first ranges from the source (*S*) to the first intermediate node (*In1*), the second between the two intermediate nodes, and the third from the second intermediate node (*In2*) to the destination (*D*). This segmented path is called a *multistep path* (MSP), and uses minimal static routing in each segment. When using *multistep paths* deadlock freedom becomes a key issue. In our proposal, having a separate virtual channel for each step ensures deadlock freedom. As we are considering two intermediate nodes, one extra virtual channel is used (if required) from *S* to *In1*, another from *In1* to *In2*, and a third one from *In2* to *D*. This way, each step defines a virtual network, and the packets change virtual network at each intermediate node. Although each virtual network relies on a different virtual channel, they all share the same adaptive channel(s). A total of 4 virtual channels are need.

3.1 Method Behavior

The behavior of the method, including all its functionalities, could be seen in Fig. 2. The behavior diagram shown in Fig. 2 consists of four main blocks: *Source endnode*; *Message routing*; *ACK routing*; and *Destination endnode*. Source and destination endnode blocks contain the actions implemented at the source and destination nodes, respectively; while message and ACK routing blocks represent actions carried out by the routers along the source-destination paths. Each block is composed by several elements (stage boxes and decision elements), where the colored elements represent the set of actions performed in the absence of failures and congestion, and the colorless correspond to the additional features for fault tolerance and congestion control.
When a source node injects a message in the interconnection network, it traverses a set of routers before reaching the destination node. Two monitoring actions are conducted at each router along the source-destination path: link state and traffic load monitoring. Link state monitoring is performed directly over router physical channels, while traffic load monitoring is accomplished by the router over the message. These actions are

represented in Fig. 2 by the two colored decision elements in the *Message routing* block.
If there are no faults in the source-destination path, each message registers and transports the accumulated latency information about the path it traverses (either the original or an alternative one), by means of the *Latency Accumulation* element in *Message routing* block. When the message reaches the destination node, the accumulated latency value is obtained from the packet and, if the path is fault-free, sent back to the source node by means of an ACK message (*ACK Injection (path latency)* element in the *Destination endnode* block) in order to notify the source node about the network traffic burden.

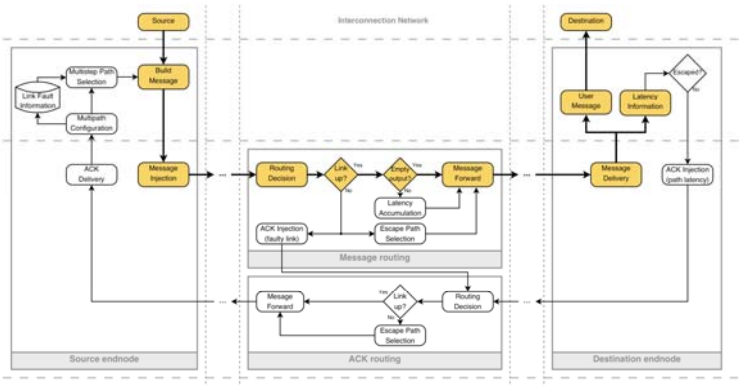


Figure 2: *Method behavior diagram.*

On the other hand, two actions are triggered if a message tries to use a faulty link while it traverses the source-destination path. Firstly, the message is rerouted to its destination through an escape path (*Escape Path Selection* element in the *Message routing* block). At the same time, the *ACK Injection (faulty link)* element sends back a special ACK message to the source node. This ACK message -sent by means of the *ACK routing* block- carries information about fault location (e.g. node and port identifiers) to avoid the use of the faulty path. These triggered actions were previously illustrated in Fig. 1(b).
These two kinds of ACK messages have higher priority in the routing unit, and their size is less than 1% of the data messages because they only transport control info: a latency value or failure information. Notice that only one of those ACK messages is sent for each data message, as appropriate.
Using the link fault information together with the set of collected latencies, the number of alternative paths needed for a specific source-destination pair is determined. From this action, performed at the *Multipath Configuration* element in the *Source endnode* block, the method avoids the use of faulty paths and fairly distributes the communication load over the multipath. The communication load distribution is accomplished by selecting the appropriate MSPs at the *Multistep Path Selection* element.

The outcome of this phase is then used by the source node to distribute the load among all the MSPs in base of their latency. The path with lower latency is most frequently used, and then messages are distributed over the MSPs according to their relative latency values.
The set of actions at node level of our proposal do not introduce a high overhead because they consist of simple comparisons and accumulations locally performed, and do not delay send/receive primitives. As shown in Fig. 2, each message is forwarded without any overhead when the output link is non-faulty. The escape path mechanism is invoked only when faults are detected, and latency accumulations are performed when messages are waiting in the queue. Hence, computing these operations is performed concurrently with packet delivery. Furthermore, interconnection networks usually are not designed to continuously operate at saturation points, thus small overheads could be tolerated to avoid faults (if necessary).
Our method relies on physical level information about links status. This information is already available on almost all modern network devices. Current devices test and control their ports and links by means of physical parameters such as potential difference, impedance, etc. For example, the *InfiniBand* architecture offers four link states: *LinkDown*, *LinkInitialize*, *LinkArm* and *LinkActive* [13]. Even the simplest Ethernet router makes available the link state information.

3.2 Non-permanent Faults

Our method considers faults as non-permanent at a first stage to prevent the misuse of resources. If faults persist over time, their status is changed to permanent.
In order to achieve this functionality, information about location and status of faults is stored in the *Link Fault Information* element (*Source endnode* block). From this information, source nodes are be able to use fault-free paths which otherwise would incorrectly appear as faulty (due to non-permanent faults).
Information about faults is obtained from the ACK messages sent back to source nodes from those routers that have detected the faults. The node and port identifiers obtained from these ACK message are used for storing and indexing the fault information in the *Link Fault Information* element. Each entry is composed by these two identifiers and two additional numbers used to manage the fault status: the *stage number* and the *attempt number*.
The information in the *Link Fault Information* element is updated each time a fault ACK message arrives to the source node. If the ACK carries information about a new fault, a new entry is included. In turn, if there is already an entry for the fault, its *stage number* is increased. A fault is only considered as permanent if its *stage number* is greater than or equal to three. Notice that a fault is considered to be permanent only after receiving at least three different notifications about the fault.
On the other hand, the *attempt number* has been included to treat differences in the duration of non-permanent faults. In fact, it is intended to act as a timer to delay the use of a path that has been notified as faulty. This seeks to reduce the possibility of considering a non-permanent fault as a permanent one.

As stated above, the *Multistep Path Selection* element must select the MSPs before the injection of each new message. There is where the information stored in the *Link Fault Information* element is used.
When selecting the MSP, the *Multistep Path Selection* element looks for entries corresponding to the links of the source-destination path. Notice that a link can be identified by means of the ID of the router and port to which it is connected.
The path is fault-free and can be selected if there are no entries for all the links along the path. On the other hand, if there are faults along the path, it may or may not be used depending on faults status. If at least one entry corresponds to a permanent fault, the path cannot be used and an alternative path should be selected. If any entry corresponds to a permanent fault, the *attempt number* must be considered for the selection. The path can be selected only if the *attempt number* of every link along the path is below ten. Otherwise, the *attempt number* is increased for all the links along the path and the path cannot be used. By means of this action, the *attempt number* eventually will be equal to ten and the path would be selected.
If a path containing a non-permanent fault is selected and used, two situations may arise. In the best case, the fault will have disappeared and a latency value will be received from the destination node. In this case, every entry for that path will be removed from the *Link Fault Information* element and this action would improve the resources utilization. In the worst case, the message will be rerouted to its destination and a new fault notification will be received.

4. Performance Evaluation

This section describes the test scenarios used to evaluate our proposal and provides the explanation of experimental results.
The simulation environment is provided by the commercial modeling and simulation tool OPNET Modeler [14]. This tool gives support for modeling communication networks, and allows the injection of faults in model components. The whole actions and functionalities of our proposal have been modeled using this tool.
Experimentation is based on bi-dimensional torus chosen mainly due to its current popularity and multiple alternative paths between nodes. The network was modeled based on interconnection elements, connected among them through links; and endnodes that provide the interface to connect processing nodes to the network.
The simulations were conducted for a 1024 nodes network arranged in a 32x32 torus topology. We have assumed Virtual Cut-Through flow control and several standard package sizes with a constant packet injection rate. Link bandwidth was set to 1 Gbps, and the size of routers buffers to 2 MB.
In order to evaluate the behavior and also to measure performance, the experiments were conducted using standard communication patterns with up to 60 simultaneous randomly injected link failures. Permanents faults were used in order to validate the method against worst-case scenarios. Standard communication patterns were used due to their application in computational intensive scientific applications [15].

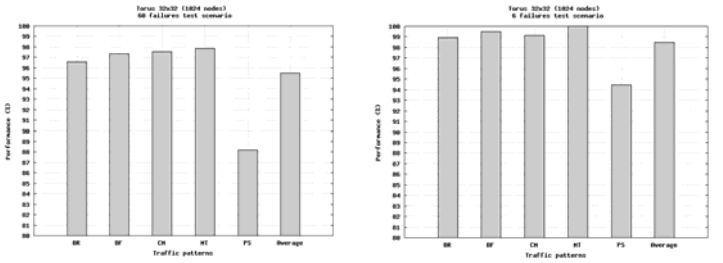
These patterns are: *Bit Reversal (BR), Perfect Shuffle (PS), Butterfly (BF), Matrix Transpose (MT),* and *Complement (CM).*

In order to evaluate our proposal, as a first step, a set of fault-free scenarios was simulated several times to get average latency values in the absence of failures. Later, faults were injected in those scenarios to measure the average latency values of each approach. Up to 60 network links were simultaneously failed during the evaluation of test scenarios. Finally, performance degradation was measured as the difference between latency values of faulty and fault-free scenarios.

The vast majority of previous work in literature assume static failures, therefore, it is not possible to make direct performance comparisons against them. One of the only proposals dealing with dynamic faults is [10] and achieves an average throughput of 85.5% using the *Uniform* traffic pattern and 88.8% using *Permutation* traffic, both in the presence of 7 random link failures and allowing packet drops.

The performance results for the standard traffic patterns in the 60-failures test scenario are shown in Fig. 3(a). In order to compare the ratio between performance and the number of failures of our proposal, results of the 6-failures test scenario were also included in Fig. 3(b).

As shown in Figs. 3(a) and 3(b), our method obtains very promising performance values. An important point to emphasize is the fact that with a linear increase of 10 times the number of faults, the average performance degradation value is just about 3%. In the worst case the performance is about 88% for the *Perfect Shuffle* pattern with 60 simultaneous faults, and 100% in the best case for *Matrix transpose* pattern with 6 simultaneous faults. Performance values are even better if we consider the average values, obtaining a 97% performance value in the worst case (60 faults).



(a) 60 failures (b) 6 failures

Figure 3: *Results of test scenarios*

5. Conclusions

In this paper, we have proposed a multipath fault-tolerant routing method designed to deal with the problem of fault tolerance for high-speed

interconnection networks. Our method is able to deal with dynamic link failures using a limited number of virtual channels and without requiring network reconfigurations or stopping packet injection at any time. Our proposal needs few additional hardware resources and is able to treat intermittent, transient and permanent link failures maximizing resources utilization. Unlike other fault-tolerant approaches, our method does not degrade at all the system performance in the absence of faults.

Evaluation results show an average performance value above 97% for several test scenarios ranging from 1 up to 60 simultaneous link failures using standard communication patterns. From these results we may conclude that our method is capable of reroute messages to their destinations through fault-free paths with negligible performance degradations even in the presence of a large number of faults.

We are currently working on improving the method to tolerate a larger number of link failures. In addition, future work includes the enlargement of current fault models to address information losses caused by failures of network devices.

References

1. Barroso, L., Dean, J., Holzle, U., Web search for a planet: The Google cluster architecture, Micro, IEEE 23(2), March-April 2003.
2. Abd-El-Barr, M., Design and analysis of reliable and fault-tolerant computer systems, Imperial College Press, London, UK, 2007.
3. Zarza, G., Lugones, D., Franco, D., Luque, E., A multipath fault-tolerant routing method for high-speed interconnection networks, in 15th International European Conference on Parallel and Distributed Computing (EuroPar), 2009.
4. Sem-Jacobsen, F., Skeie, T., Lysne, O., et al., Siamese-twin: A dynamically fault-tolerant fat-tree. In: International Parallel and Distributed Processing Symposium (IPDPS 2005), IEEE Computer Society, April 2005.
5. Mejia, A., Flich, J., Duato, J., Reinemo, S. A., Skeie, T., Segment-based routing: an efficient fault-tolerant routing algorithm for meshes and tori, in International Parallel and Distributed Processing Symposium (IPDPS), IEEE Computer Society, 2006.
6. Puente, V., Gregorio, J. A., Immucube: Scalable fault-tolerant routing for k-ary n-cube networks, IEEE Transactions on Parallel and Distributed Systems 18(6), 2007.
7. Gómez, C., Gómez, M. E., López, P., Duato, J., An efficient fault-tolerant routing methodology for fat-tree interconnection networks, in ISPA, Volume 4742 of Lecture Notes in Computer Science, Springer, 2007.
8. Gómez, M. E., Nordbotten, N. A., Flich, J., Lopez, P., Robles, A., Duato, J., Skeie, T., Lysne, O., A routing methodology for achieving fault tolerance in direct networks. IEEE Transactions on Computers 55(4), 2006.

9. Valiant, L. G., Brebner, G. J., Universal schemes for parallel communication, in STOC ‘81: Proceedings of the thirteenth annual ACM symposium on Theory of Computing, New York, USA, ACM, 1981.
10. Nordbotten, N. A., Skeie, T., A routing methodology for dynamic fault tolerance in meshes and tori, in International Conference on High Performance Computing (HiPC), LNCS 4873, Springer-Verlag, 2007.
11. Skeie, T., Handling multiple faults in wormhole mesh networks, in Euro-Par ‘98: Proceedings of the 4th International Euro-Par Conference on Parallel Processing, London, UK, Springer-Verlag, 1998.
12. Franco, D., Garcés, I., Luque, E., Distributed routing balancing for interconnection network communication, in HIPC ’98, 5th International Conference On High Performance Computing, 1998.
13. InfiniBand Trade Association: InfiniBand architecture specification: release 1.2, Volume 1, InfiniBand Trade Association, Portland, OR, 2004.
14. OPNET Technologies: Opnet modeler accelerating network R&D, http://www.opnet.com.
15. Duato, J., Yalamanchili, S., Ni, L. M., 9, in, Interconnection networks. An Engineering Approach, Morgan Kaufmann, 2003.

Could be improved the efficiency of SPMD applications in heterogeneous environments?*

RONAL MURESANO, DOLORES REXACHS, EMILIO LUQUE

Universitat Autònoma de Barcelona
Computer Architecture and Operating System Department (CAOS)
Barcelona, SPAIN
rmuresanog@caos.uab.es
{dolores.rexachs, emilio.luque}@uab.es

Abstract: *The goal of this work is to execute SPMD applications efficiently on heterogeneous environments. Applications used to test our work have been designed with message-passing interface to communicate and are developed to be executed in a single core cluster. However, this paper presents a novel methodology to execute SPMD applications efficiently on heterogeneous architectures as Multicore cluster. These applications were selected because they present high communication volumes and synchronism. These two elements generate challenges for programmer when they wish to execute SPMD applications efficiently. Hence, our main goal is to achieve an improvement in execution time while the efficiency is maintained over a threshold defined by the programmer taking into consideration the communications heterogeneities present on Multicore cluster. This objective is achieved through mapping and scheduling strategies defined in our methodology. Finally, the results obtained show an improvement around 40% in the best case of efficiency in the SPMD applications tested, when our methodology is applied.*

1. Introduction

Currently, the parallel applications are designed to execute complex computational problems and this execution can take a long execution time to be finished. However the actual trend in high performance computing (HPC) is to execute application faster and efficiently. For this reason, parallel computing has included new techniques in order to improve the application performance. One of these techniques is to group a set of cluster in an architecture called Multicluster and other is to include node which integrate a number of PEs to execute the application in a Multicore architecture. Both architectures generate new challenges for programmer when they wish to improve the performance metrics. These challenges are due to the different communications path in which the communication processes can be established. These communication imbalances generate issues that have to be managed in order to improve the parallel application metrics.
A first approach of this work is to include the execution of SPMD application on multicore clusters. These kinds of clusters allow us to execute applications in an

environment with more computational power, in order to obtain a faster execution. However, multicore clusters add high heterogeneities in communication paths and these heterogeneities generate communication issues when message-passing applications are developed to be executed within a single core cluster, and these applications are executed on heterogeneous cluster.
Then, this work is focused on managing the communication heterogeneities in order to improve the performance metrics of parallel applications. We are mainly centered on execution time, speedup and efficiency. These metrics are affected by the number of cores included in the parallel environment. Hence, we have to administrate the workload in order to determine the adequate number of cores and number of tasks necessary for executing these applications efficiently, taking into consideration the environment. The SPMD applications were selected due to they have a repetitive behavior and high communication volume.
The main objective is to improve the execution time when a heterogeneous environment is used, while the efficiency is maintained over a threshold defined by programmer. The multicore cluster presents a set of heterogeneities due its different communications paths, number of PEs per chip, shared cache, etc. Then our goal is to manage the speeds and bandwidths of the communications paths with the aim of executing the SPMD applications efficiently. This process is realized through a novel methodology which is divided by three main phases: characterization, mapping and scheduling.
The characterization phase allows us to characterize the parallel machine according to the SPMD applications. This phase lets us to find the communication time of SPMD task in each communication link. Also, this phase enables us to find the computation time of the task into the environment. This phase gathers the inputs necessary to calculate the task distribution model in next phase.
The objective of mapping phase is to distribute the tasks between the PEs according to the communications latency and the communications numbers presented by each process. This process enables us to create two different kinds of tasks, the internal tasks in which the communication is realized in the same PEs, and the edge tasks in which these tasks have to communication with other tasks located in other core. The number of tasks was determined through a model in which the numbers of tasks are calculated depending on the overlapping strategy between internal computation and edge communications time.
Finally, the scheduling phase allows us to develop an overlapping strategy with the aim of eliminating the communication inefficiencies present on multicore cluster. Also, this phase lets us to design a tasks priority strategy in order to execute the tasks with the objective of overlapping the internal tasks with the edge communications.
The work is structured as follows: section 2 defines the problem. Related work is described in section 3. The methodology description is shown in section 4. Section 5 describes the implementation. Next, section 6 describes the performance evaluation, and finally conclusions are in section 7.

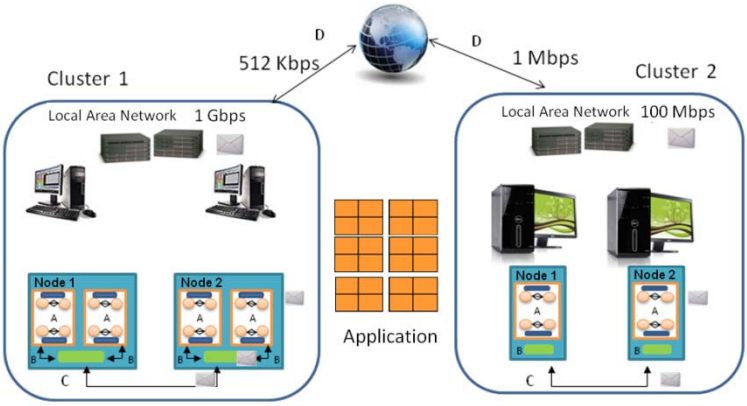


Figure 1: *Multicluster with multicore Node*

2. Problem Formulation

Evolution in the parallel programming field has allowed that scientific applications can be programmed with more complexity and accuracy. These precisions require high computational power and clusters are generally limited by the number of nodes. Such limitations originate application scalability and performance issues. The programmers have to find suitable solutions that will improve the application performance metrics. Also, there are many computer centers within organizations and universities, which have computational power to execute parallel applications. However, these centers are usually underutilized and the resources are in an idle state.

In order to benefit from such computational cluster capacities and to execute applications faster, some of these computational centers can be combined for creating a cluster architecture called Multicluster (Fig. 1). However, to use this kind of architecture, the programmer must consider computational environment heterogeneity. A Multicluster environment has different computational and interconnection network architectures, and both elements have to be managed if performance wants to be improved.

To execute parallel applications in this environment is a challenge due to the workload allocation for each Processing Element (PE), and the number of tasks that will be assigned to each core (Multicore node), node, or cluster. Another heterogeneity that Multicluster environment can present is the multicore node (Dual or Quad Core). A multicore node adds more level of complexity to the Multicluster, due to its different internal communication levels, which programmers have to deal with if they wish to find strategies for mapping.

A Multicluster has different types of communications, some of them through network links such as: local area network (LAN) or wide area network (WAN); and others by internal processor buses like core-to-core communication through cache memory or communication between chip

processors via main memory. All these communications have different speeds and bandwidths and they represent a challenge when the programmer wants to manage them for efficient application execution. The heterogeneity present in a Multicluster with multicore nodes can generate that performance metrics such as efficiency, speedup and execution time worsen. An inadequate mapping strategy could decrease the effectiveness of the parallel application in Multicluster environments. One more element to consider in this environment is that applications are designed under parallel programming paradigms such as, Master/Worker, SPMD, etc., each of them having a different communication patterns and execution models. The programmer has to evaluate if the execution of these parallel paradigms can improve application performance in a multicluster environment.

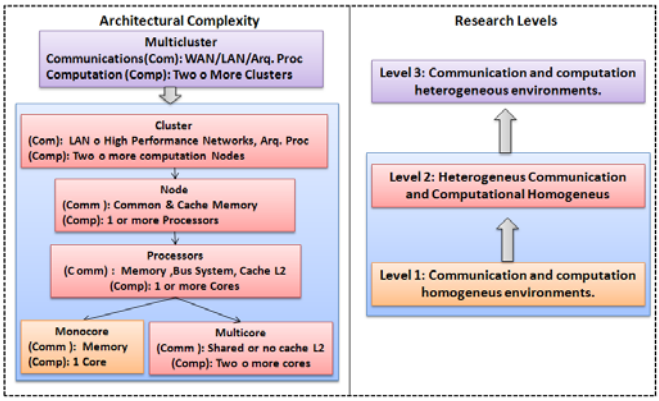


Figure 2: *Levels and Complexity Architecture*

A methodology to migrate a master-worker parallel application from its original cluster to a multicluster environment was developed by Argollo [1] using a Master/Worker (M/W) paradigm. The proposed methodology targets are to decrease the execution time in the multicluster environment guaranteeing a pre-established threshold level of efficiency. Unlike the M/W paradigm, the behavior of SPMD is to execute the same program in all PE, but with a different set of data task. These applications are synchronized and they have a high communication volume in each iteration, making the execution of SPMD applications on a heterogeneous communication environment a challenge.

From the above problem, we plan to develop a methodology for SPMD applications in heterogeneous communication systems, considering an efficient execution. The objective is to execute in the shortest execution time possible, maintaining the efficiency level over a threshold value defined by the programmer. Moreover, our work considers an SPMD application which permits us to set up a number of tasks greater than the PE present. For this reason, tasks must be allocated between PEs, through some important key such as: number of communications between PEs, communication volume

and links involved in the communication process. To solve the problem, our research has been divided in layers (Fig 2) which allows us to define a Multicluster architecture. This shows the heterogeneity between different network links and buses, and also illustrates the computational hierarchy between PEs. Additionally, the complexity levels let us to identify the computational and communication parameters which are present in this environment.

Dividing the problem through the different complexity levels allow us to give a solution in levels, which are able to resolve the inherent complexity of heterogeneous environments present in a multicluster with multicore node. To analyze the influence of communications, the first step is to make a characterization of the environment including different bandwidths for each level and size of computation.

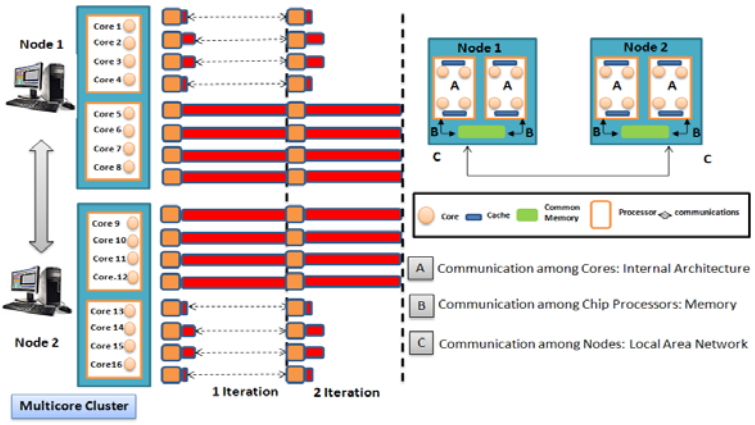


Figure 3: *Multicore cluster and SPMD tasks assigment.*

Then, obtaining the computation and communication time by each task in the PE, we develop the mapping and scheduling strategies in order to obtain the minimum execution time while the efficiency is maintained.

This work presents a proposal for level 1 and 2 (Fig 2), the level 3 is currently under development. At level 1 and 2 the heterogeneities is presented in communications paths, and we try to manage them with the aim of maintaining the efficiency parameters. For this reason, we propose a methodology to evaluate the computational and communication parameters of SPMD applications on Multicore clusters. In order to develop the mapping strategy, we have evaluated the environment heterogeneity. This evaluation allows us to assign a set of tasks to each PE. The Mapping strategy intends to manage the workload imbalance caused by different communication link latencies. Otherwise, workload imbalance would certainly decrease the application performance. Once the mapping is finished, a scheduling strategy is considered. The scheduling is based on an overlapping strategy in which the

internal computation and edge communication are overlapped. This process is made considering the architecture hierarchy and the communication link latencies.

The problem is shown in figure 3, where, tasks are assigned on multicore nodes and are executed with an SPMD application. Each task has a similar execution time but the communication process can be different due to the latencies of different communication links. For example, if a task has a communication with another task but in different node, the communication is made through LAN and this link has more latency than communications which are made through internal processor architecture. The slower communication limits the time defined for each iteration, then, tasks that finishes before of this time must wait until the iteration ends (Fig. 3). This idle time could generate an inefficient time to the execution and decrease the performance metrics in application. Then, our methodology establishes a suitable solution to improve the efficiency and speedup.

3. Related Works

The conceptual study has been divided in three main aspects related to characterization, mapping, and scheduling for SPMD applications on Multicore nodes.

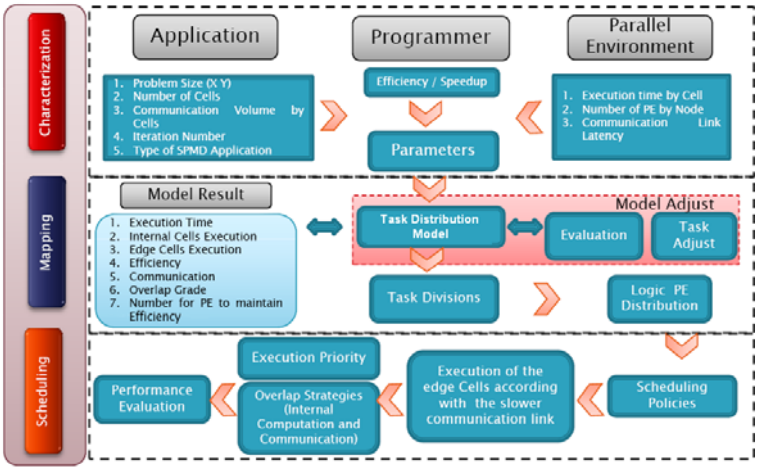


Figure 4: *Methodology for efficient execution of SPMD applications.*

Then, exits works oriented to characterize and study the effects of communications links on multicore architecture [2], and multicluster in [3] where the communications delays and bandwidths are evaluated with the aim to obtain an improvement of the efficiency within the environment. Also, these works

present different communications levels and the way to administrate them in order to manage the troubles generated by communication parameters and even more when these communications are different. An important key in a heterogeneous environment is to manage the workload properly, due an incorrect distribution could generate inefficiency in the system when a SPMD application is executed [4].
In mapping topic, Virkram [5] has studied a suitable strategy which permits us to improve some performance metrics in SPMD applications. However, this work is mainly focused on seeking the best speedup, obtaining the lower execution time. This mapping tries to search the maximum number of nodes which application need to execute without evaluate the efficiency level. Additionally, different kind of mapping are studied some are statics [6] and others dynamics [7]. The statics mapping are focused on homogeneous architecture of single core node, obtaining different manner to distribute the workload between nodes. These distributions are by rows, columns, and blocks or through acyclic blocks, etc. On the contrary, dynamics mapping presents their distribution based on the computational power of the core inside environment.
Additionally, scheduling strategies have been studied [8] to be developed for large-scale architectures which use heterogeneous distributed systems for SPMD tasks, the objective of these scheduling are to minimize the execution time of SPMD tasks, but they do not use an overlapping strategy to minimize the inefficiency generated by communication links. In order to obtain a better performance metrics for SPMD application, the evaluation have been made in a multicore cluster [9] and we can appreciate the system degradation when are added more PE to system.

4. Methodology

The methodology developed is composed by three phases: characterization, mapping and scheduling of SPMD tasks and are detailed below (Fig. 4).

4.1. Characterization Phase

The main function of this phase is to determine the parameters which will be included in tasks distribution model. The characterization is made through a testing of the environment where communication and computation values are determined. This phase is divided in three types of inputs. One of them is the application parameters. The application parameters offer to our methodology information related with some application characteristics such as problem size, number of cells, iteration number etc. Additionally, this phase determine the application behavior within the application. Another element included is the parallel environment in which are evaluated the computational and communication time of a task inside the communication environment. The latencies and bandwidths are evaluated with the aim of obtaining the

characteristics of the environment. Finally, the programmer establishes the efficiency desired in order to calculate the set of task necessary to achieve our goal.

4.2. Mapping Phase

The objective of mapping strategy is to determine the number of tasks to manage the computational idle time generated by communication paths. Then, we have to evaluate the slower communication path in order to assign the set of tasks to each core present. This mapping process is made through a core affinity, in which we can select which process will execute the set of tasks calculated. Also, the mapping let us to cover the idle time generated by communication links as is shown in figure 5. Once the model is calculated, we could determine some model result such as: execution time, overlap grade, number of PEs necessaries according to the efficiency defined, internal execution time, and communication time, etc. These values are estimated with the analytical model defined in [10].

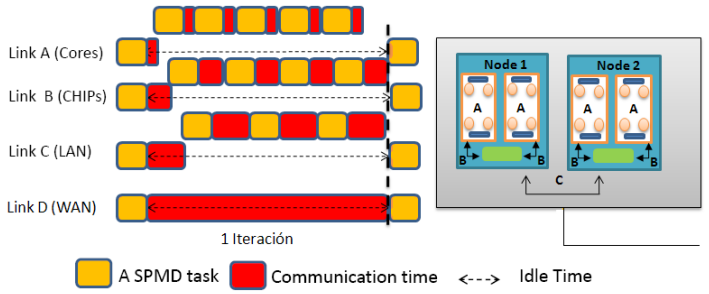


Figure 5: *Communications managing through a mapping strategy.*

4.3. Scheduling Phase

The objective of the scheduling phase is to establish the tasks priorities executions. The tasks are divided in two types, the internal and the edge tasks. The scheduling process assigns priorities to tasks, where the highest priorities are established for tasks that have communications through the slower links. The objective to assign priorities is to overlap the internal computation and edge communication. The priorities are assigned in the follow way: firstly, tasks with two external communications are selected with the priority 1, then tasks with one external communication will be assigned the priority 2 and the other internal tasks which will be overlapped with the edge communications and are assigned with the priority 3 (Fig 6).

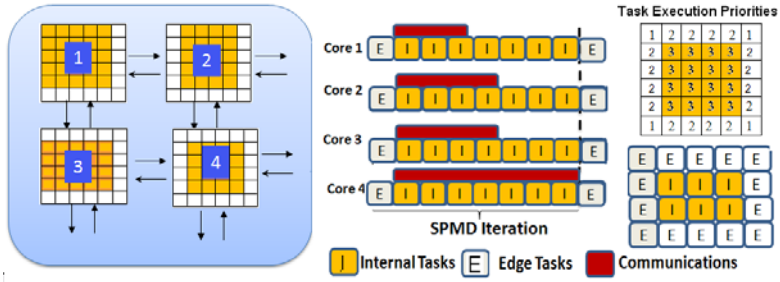


Figure 6: *Scheduling process priorities*

The priorities allow us to organize the way to execute the tasks of SPMD applications, and also permit us to manage the communication delays. The priorities let us to execute the internal tasks time while the edge communication is performed. This strategy gives possibilities to overlaps and administrates the inefficiencies generated through the communications.

5. Implementation

To implement our methodology, we develop a framework in C where the characterization, mapping and scheduling are included. This allow us to execute the application including our program module, where these modules determine the characteristics within environment and we develop the mapping with a tools to solve linear inequalities in order to obtain values for X(i) and Y(i) which are the number of SPMD task that will be assigned to each PEs.
To develop the mapping, we have to distribute the task among the PEs. For this, we design an affinity core process in which we can know where the process is executed. To realize this core affinity process, we have developed a logical topology of the process that executes the SPMD application. The topology allows us to identify each process among PEs and our tool is designed to assign the process into the PEs desired.

6. Performance Evaluation

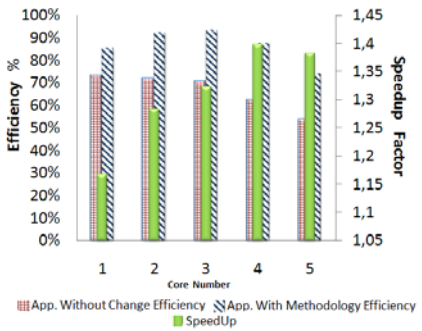
Our experiments were conducted on a multicore cluster DELL with 4 nodes, each node has 2 Quadcore Intel Xeon E5430 of 2.66 Ghz processors, and 6 MB of cache L2 shared by each two core and RAM memory of 12 GB by blade and we use a heat transfer, wave equation and Laplace application.
The first step is to evaluate the communications paths, in this cluster there are four communication types which are defined as, intercore, intercore without cache, interchip and internode, each of them has a different communication behavior. The values are obtained through a ping-pong tool which allows us

to determine the different communication links. The differences present between each communications link are approximately one and half order of magnitude in some cases. This characterization have been tested with different packet sizes because, we noticed that communications do not have a linear relationship with the packet size. Also, the computational task time is obtained in order to evaluate the relationship between computation and communication of a task.

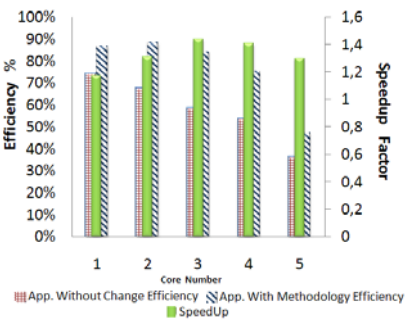
Table 1: *Task Distribution Model Evaluation*

Application	Comp time	Comm time	Effic	Problem Size	Xi	Yi	Tejek	Cores
Heat Transfer	0,13msec	5,8 msec	85%	10000x6000	2500	1410	45,72 Seg	16
Laplace Equ.	0.07msec	4,8 msec	85%	8000x2000	1950	980	28,16 Seg	8
Wave Equ	0.02msec	4,9 msec	85%	17280	2300	------ --	530 msec	7

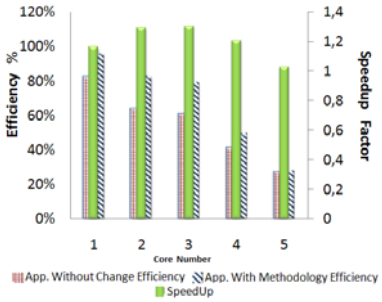
Once finished the characterization, we assigned the set of tasks to each PEs according to a tasks distribution model defined in [10]. The table 1 shows an example of results obtained of the task distribution model in which we can determine the number of tasks necessaries in order to maintain the efficiency desired. Then, we have to evaluate the effectiveness of the mapping and scheduling strategies with the aim of achieving an efficient execution.
Figures 7a, 7b and 7c show how could be the improvement between the original application and when we apply our methodology. The result shows an improvement around 40% in best case, and allows us to evaluate the effectiveness of mapping and scheduling over the SPMD applications tested. The numbers of PEs used to test are fixed and we can observe how the efficiency is improved and maintained while the PEs number is below by PEs number calculated by our methodology (Table 1).



(a) Heat Transfer Application



(b) Laplace Equation



(c) Wave Equation

Figure 7: *Efficiency and Speedup in different SPMD applications.*

As regards to speedup, we observe that maximum speedup is located near to the efficiency threshold which we defined to calculate our task distribution model. Moreover, speedup was calculated considering a reference value of the parallel time of the application without apply our methodology and divided by the parallel time of the application with our proposed changes. Finally, the experiment reported in this section made possible the analysis of our methodology with different SPMD applications. We have achieved through our methodology to maintain the efficiency in a heterogeneous communication environment as has been demonstrated in this section.

7. Conclusion and Future works

This work allows us to show how a SPMD application can be executed efficiently in a heterogeneous environment. The efficiency is maintained due the mapping and scheduling strategies, where in both cases, we try to manage the communication latency. Our methodologies through mapping enable to improve the execution time while the efficiency is managed. Also, we can set the amount of tasks necessaries each PE according to the value of the slower path. Finally, the execution order permits us to develop an overlapping strategy between internal computation and edge communications, allowing us to control the inefficiencies of communications links.
Some important future lines consist of generalizing the methodology to include other scientific computation applications, and the selection of the optimal PEs number in order to realize an efficient execution in a heterogeneous environment as a multicluster environment.

References

1.Argollo, E., Rexachs, D. and Luque, E., "Tuning application in a multi-cluster environment", Lecture Notes in Computer Science, vol. 4128, 2006.

2. Trahay, F., Brunet, E., Denis, A. and Namyst, R., "A multithreaded communication engine for multicore architectures", Parallel and Distributed Processing, IPDPS 2008, IEEE International Symposium on, 2008. http://dx.doi.org/10.1109/IPDPS.2008.4536139

3. Plaat, A., Bal, H. E. and Hofman, R. F. H., "Sensitivity of parallel applications to large differences in bandwidth and latency in two-layer interconnects", HPCA '99, Proceedings of the 5th International Symposium on High Performance Computer Architecture, 1999.

4. Pastor, L. and Bosque, J. L., "An efficiency and scalability model for heterogeneous clusters", in Proceedings of the 3rd IEEE International Conference on Cluster Computing, 2001.

5. Vikram, K., and Vasudevan, V., "Mapping data-parallel tasks onto partially reconfigurable hybrid processor architectures", IEEE Transactions on Very Large Scale Integration Systems, vol. 14, No. 9, 2006.

6. Guirado, F., Ripoll, A., Roig, C., Yuan, X. and Luque, E., "Predicting the best mapping for efficient exploitation of task and data parallelism", Lecture notes in computer science, 2003.

7. Sanyal, S. and Das, S., "Match: Mapping data-parallel tasks on a heterogeneous computing platform using the cross-entropy heuristic", 19th IEEE International Parallel and Distributed Processing Symposium, 2005.

8. Panshenskov, M. and Vakhitov, A., "Adaptive scheduling of parallel computations for spmd tasks", ICCSA 2007, vol. 4706/2007, 2007.

9. Pinto, L., Tomazella, L. and Dantas, R., "An experimental study on how to build efficient multi-core clusters for high performance computing", 11th International Conference on Computational Science and Engineering, 2008.

10. Muresano, R., "Aplicaciones single program multiple data (spmd) en ambientes distribuidos", Master's thesis, Universitat Autònoma de Barcelona, 2008.

VIII

Information Technology Applied to Education Workshop

Problem Based Learning and Software Simulation Tools: A Case of Study in Computer Science First Year Students

JAVIER GIACOMANTONE[1], TATIANA TARUTINA[2]

[1] Instituto de Investigación en Informática (III-LIDI),
Facultad de Informática – Universidad Nacional de La Plata
jog@lidi.info.unlp.edu.ar
[2] Departamento de Física, Facultad de Ciencias Exactas
Universidad Nacional de La Plata
tarutina@fisica.unlp.edu.ar

Abstract. *In this work an adapted problem based learning strategy is presented for the particular case of first year computer science university students. A pilot study was conducted to determine the validity of the proposed alternatives in one particular undergraduate course, Computer Organization. The main problems detected in first year students involve deficiencies in the learning process, resulting in a lack of critical thinking and problem solving skills that are essential in Computer Science related disciplines. The main objective of testing different models, through problem based oriented learning alternatives and intensive use of simulation software, was to assess how students develop self learning, problem analysis, problem solving and communication skills.*

Keywords: *Problem based learning, software simulation, cooperative learning, computer organization.*

1. Introduction

Graduates of computer science programs are expected to face scientific and technological advances and to be able to solve problems in scientific research or industry. It means that self-learning skills, creative thinking, cooperative work, and communication skills are needed [1]. Their acquisition is a gradual process that is possible to achieve trough a self-directed learning and lifelong learning process. It is a challenge for a teacher not only to develop their own subject material but to promote and guide the student to develop these fundamental abilities.

The task of including new learning schemes in the first year of a graduate university program in computer science faces a few common problems [2] [3], normally related to the large number of students, the lack of some basic skills expected to be developed in secondary school and heterogeneity in the basic background knowledge. This paper presents an alternative learning scheme and results of a pilot pedagogical experience that has been carried out in a first year course, Computer Organization. The main objective of this

work was to test the influence of different learning strategies and the use of software simulation tools, in a more deep approach to learning, where students have more control and responsibility over the learning process [4]. Computer Organization is a course in the first year of a Computer Science degree program at University of La Plata, Argentina. It covers fundamental topics such as number systems and number representation, digital circuits, Von Neuman model, basic concepts of digital memories and an introduction to assembly programming language. This introductory course aims to provide strong foundations to allow the students to understand, and effectively solve problems in a field that has a high rate of technology change.
A traditional learning strategy (TLS) consists of three hour theoretical lectures and three hours of problem solving a week during a semester. In this work an alternative thread is proposed oriented to problem based learning (PBL) [5], [6] and intensive use of software simulation tools [7], called modified new learning strategy (MNLS). A pilot experience was carried out in two stages. The first stage evaluated the results of TLS and a new learning strategy (NLS) that produced valuable information for the second stage approach (MNLS).
This paper is organized as follows: Section 2 briefly describes the main ideas behind PBL approaches. Section 3 presents the particular case of study. In section 4 results are presented. Finally conclusions and future works are given in section 5.

2. PBL and Principal Constrains

Important developments in Computer Science and engineering education in recent years has been oriented to Problem-based learning strategy (PBL) [8], [2], [1], [9]. PBL was first applied in the Medicine School at McMaster University (Canada) as an innovative educative proposal. Although it was successfully adopted by other prestigious medical schools like Harvard Medical School, the particularities and necessary adaptations to engineering and computer science programs remains an active research area.
A traditional simplified teaching scheme involves theoretical knowledge first taught to the students followed by practical lectures explaining how to solve problems applying the previously learnt theoretical concepts. Finally the teacher sets an exam to
test the basic knowledge and skills acquired by the students. The main characteristics of TLS are to set the teacher as the transmitter of linear and rational knowledge and the student as a passive receiver defining a structure environment of individual learning. The assessment responsibility resides entirely on the lecturer.
PBL can be defined as a learning environment in which the problem solving process involves searching for information and discovering the new knowledge necessary to tackle the problem [10]. It has been shown in the literature that PBL assists to gain skills in problem solving and lifelong learning abilities in contrast to short term surface learning. In a PBL approach, small groups of

students work collaboratively to solve a particular problem, with no previous preparation, with the student being the center of the learning process, constructing knowledge as an active participant in a flexible and cooperative environment. The teacher guides and facilitates the whole learning scheme. The assessment is now shared among the student, the group and of course the teacher. PBL promotes self-learning, developing problem-solving skills, cooperative learning, and improving oral and written communication.
Certain constrains or boundary conditions should be addressed before any attempt to apply PBL, an adapted PBL scheme or a mixture method between TLS and PBL could be applied to a particular course. One of the important constrains is the number of students, as PBL requieres small groups, and therefore, problems or projects need to be designed carefully involving many resources. It has been pointed out that open end problems are recommended from the beginning, reinforcing the main characteristics of PBL. Nevertheless if PBL is to be applied to a first year undergraduate level [11], no technical background or particular skill should be assumed and a work example strategy [12] and progressive difficulty tasks [13] would be more convenient.

3. Case of Study: Computer Organization

Computer Organization is a course held in the first semester of the first year of the Computer Science program. This article describes results based on a pilot experience developed to test two different learning strategies.
The first and second stages of this study were based on a set of selected topics of the course syllabus where students presented mayor difficulties to model and solve problems. The first stage of the experiment was carried out with two groups of 30 students. The first group was named Traditional Learning Strategy group (TLS) and was the control group. The TLS group assisted a 3 hour formal lecture once a week and another 3 hour solving problems class a week. These students learned theoretical concepts and received instruction how to understand and solve specific related problems.
The second group of 30 students was divided into five teams and followed another course thread named new learning strategy (NLS). The NLS group was faced with more general problems and was challenged to work collaboratively in each of the five teams to solve them. In order to find a solution to the proposed problem they needed to build up the necessary body of theoretical concepts, search bibliography and organize the work among them. Finally each group had to present their results. The teacher guided the work, but the learning was centered in the student. The problems in TLS test how previously learnt theoretical concepts are applied to them. The problems in NLS help to develop the necessary skills and build the required background knowledge.
The NLS group worked collaboratively in teams on three tasks, with the report written at the end of each one. The main steps of NLS oriented to PBL were: to define the problem to be solved in each task, to discuss with the group the different ideas of each member about previous knowledge

necessary to solve the problem. When an attempt to explain the task and the specific problem is done the team had to specify, search and study all the new knowledge necessary to solve the problem. A possible solution is a result of repeating this process where two fundamental steps are feedback and brainstorming, a term used to define the discussion and exchange of ideas and hypothesis within the group. The tasks were carefully prepared not only to study the fundamental topics of interest in Computer Organization but also to use intensively the software simulation tools both to explore possible solutions and to grasp theoretical concepts.

The second stage also had two groups of 30 students but a different or modified new learning strategy (MNLS) was adopted based on the results of the first stage. Two main modifications were included, a worked example approach and a more gradual transition difficulty level between tasks [14]. A project was included as an additional task integrating previous concepts rather than an adapted Project Based Learning strategy (PjBL). Finally in the second stage a rotation team membership was included to balance the leadership tendency. In table 1 the characteristics of the groups on stage 1 and stage 2 are summarized.

The traditional learning thread suggested two optional software simulation tools to be used by the students. NLS and MNLS groups extensively used a digital circuit design software, Digital Works demo version, written by D. J. Barker at the University of Teesside and a didactic graphical simulator, called MSX88, based on Intel 8086 family [15]. Recent research has shown the convenience of an integrated software simulation tool for Computer Organization and Architecture [7]. Further research is necessary to establish the particular needs of a similar tool for Computer Organization under a particular learning scheme like MNLS.

An ideal design of these pedagogical alternatives should consider independent groups each one following a particular thread such as TLS, NLS or MNLS, and particular resources assigned to each one. In the first phase consisting of stages one and two, the main objectives were to gather information in order to, in a future second phase, improve the learning approach under more controlled experimental conditions.

Table 1: *Group Characteristics*

Computer Organization				
Phase 1	Stage 1		Stage 2	
Group	TLS	NLS	TLS	MNLS
Students	30	30	30	30
Teams	-	5	-	5
Tasks	-	3	-	3 + Proj.
Ind. Eval.	1	2	1	2

4. Learning Results and Student Feedback

The assessment scheme had three parts: independent evaluation, student feedback and comparative tests. First, continuous and progressive evaluation was carried out with NLS and MNLS student groups. It was continuous because the evaluation was carried out at the end of each different proposed task, and progressive because the later tasks and the final project had more weight than the earlier ones. The main objective of progressive test was to determine how efficiently the accumulative skills were acquired. Second, students were asked to complete an anonymous questionnaire to summarize their opinion of the learning experience. Finally, an exam with the same set of problems was presented both to NLS, MNLS and to the control groups. The tests were carefully designed not to alter the normal schedule of the course and not to overload the students subject to the experience with activities. The purpose was to evaluate certain learning strategy adapted to the particular case of study and to a particular restrictive set of students.
Table 2 presents partial results for the five teams of students and three tasks of stage 1. Stage 2 included a final project and an individual evaluation was carried out in both stages.

Table 2: *Evaluation results on stage 1 and stage 2*

Teams	Stage 1 (NLS)			Stage 2 (MNLS)				A-I-T NLS	A-I-T MNLS
	Task-1	Task-2	Task-3	Task-1	Task-2	Task-3	P		
T1	B	C	B	B	B	A	B	4	5
T2	D	C	B	C	B	B	B	2	5
T3	B	B	A	A	B	A	A	6	6
T4	C	B	A	B	D	B	B	4	4
T5	C	C	B	A	A	B	B	5	5
Approved Individual Test								70%	83%
Fail Individual Test								30%	17%

The first stage provided positive results and feedback. The motivation and self-confidence gained solving the first tasks encouraged them to take responsibility in their own learning process. The first stage also revealed that the process of identifying the problem from a given situation was probably the most difficult, suggesting a work guided example orientation on similar situations for stage 2. Stage 1 revealed latent problems, like the heterogeneity of the background knowledge and basic mathematical skills. Students of NLS and MNLS extensively used the library facilities comparing to TLS ones. Natural leadership in some of the teams positively influenced on the teamwork, meanwhile other teams seemed to be more balanced and cooperative. In some particular cases the teacher guidance was necessary to

balance the intervention of more outspoken students as well as to facilitate the discussion.

Table 3: *Average marks from students opinion. Table scale from 0 to 5 is used.*

	Stage 1	Stage 2
Open end designs	2	2,5
Simulation tools	4	4
Collaborative working	3,8	4
Report presentation	3	3
Worked example orientation	-	4,5
Final integrating project	-	3
Overall learning experience	3	3,5

The information on students learning experience came from the teacher constructive communication with each group and from an anonymous questionnaire completed by the students. Moreover, the list of five statements was presented to the students of stage 1 that they had to mark from 0 to 5 indicating their agreement about the contribution to the learning experience from negative to positive respectively. Students of stage 2 had two more statements that correspond to additional activities that were included in the second stage. The results are given in table 3.
Individual evaluation of each student to solve particular problems was carried out among TLS, NLS and MNLS and the results are summarized in table 4 indicating the percentage of students of each group that obtained a mark on a scale from 1 to 10 with 4 being the minimum mark needed to approve the exam. The MNLS group obtained the better grades than TLS in both stages. A detailed analysis of each of the exercises and the obtained results revealed that in TLS groups good results were correlated with similar problems to the ones taught in the traditional lecture scheme but failed when a new situation that needs the same background knowledge were presented. The students of MNLS group were able to identify this uncorrelated type of problem and model a solution based on their background knowledge.

5. Conclusions and Future Work

This paper discusses a PBL approach particularly adapted to study Computer Organization in a first year Computer Science degree. Comparing results with non-PBL strategy reveals that average grades were higher than in the

traditional learning group, especially in stage 2 greatly improved by the feedback of stage 1.
First year students lack of skills and background knowledge to solve both complex and open end problems, has to be particularly considered in order to apply any PBL related method. In stage 2 the problems presented to the student were prepared with increasing level of difficulty and using worked examples.

Table 4: *Results of individual evaluation gathered by groups. The scale indicates that students with the mark below 4 didn't pass the individual exam.*

	Stage 1		Stage 2	
Scale	TLS	NLS	TLS	MNLS
10 9	2%	3%	1%	5%
8 7	20%	21%	22%	24%
6 5 4	40%	37%	42%	47%
3 2 1 0	38%	39%	35%	24%

The presented case of study indicates that PBL can be successfully applied to first year students improving basic problem model and solving skills. It has been observed that students in the MNLS group were motivated and developed self-driven learning skills. Future work involves evaluating the proposed method for larger number of students, improving the progressive complexity criteria on tasks and including simulation tools for the particular need of Computer Organization students, like number system representation software simulation and educational simplified architecture software simulator. Further research should consider working with independent groups for TLS and MNLS not only for the preselected topics but for the complete course syllabus.

References

1. Costa, L. R. J., Honkala, M., Lehtovuori, A., Applying the Problem-Based Learning Aprpoach to Teach Elementary Circuit Analysis, IEEE Transactions on Education, vol. 50, No 1, 2007.

2. García Famoso, M., Problem-based learning: a case of study in computer science. Proceedings of the Third International Conference on Multimedia and Information Technologies in Education, Spain, 2005.
3. Dattatreya, G. R., A Systematic Approach to Teaching Binary Arithmetic in a First Course, IEEE Transactions on Education, vol. 36, No 1, 1993.
4. Waters, R., McCraken, M., Assessment and evaluation in problem based learning. Proceedings of Frontiers in Education, vol. 2, 1997.
5. Schmidt, H. G., Problem-based learning: Rationale and description, Medical Education, vol. 17, 1983.
6. Ditcher, A. K., Effective teaching and learning in higher education, with particular reference to undergraduate education of professional engineers, International Journal of Engineering Education, vol. 17, No. 1, 2001.
7. García, M. I., Rodríguez, S., Pérez, A., García A., p88110: A Graphical Simulator for Computer for Computer Architecture and Organization Courses, IEEE Transactions on Education, vol. 52, No. 2, 2009.
8. Linge, N., Parsons, D., Problem-Based Learning as an Effective Tool for Teaching Computer Network Design, IEEE Transactions on Education, vol. 49, No. 1, 2006.
9. Montero, E., González, M. J., Student Engagement in a Structured Problem-Based Approach to Learning: A First Year Electronic Engineering Study Module on Heat Transfer, IEEE Transactions on Education, vol. 52, No. 2, 2009.
10. Schmidt, H. G., Foundations of problem-based learning: some explanatory notes, Medical Education, vol. 27, 1993.
11. Noor, M., Implementing a problem based learning for undergraduates course. A first experience, The Learning Conference, Institute of Education, University of London, UK, 2003.
12. Sweller, J., The worked example effect and human cognition, Learning and Instruction, vol. 16, No. 2, 2006.
13. Merill, M. D., A Task-Centered Instructional Strategy. Journal of Research on Technology in Education, vol. 40, No. 1, 2007.
14. Sweller, J., Cognitive load during problem solving: Effects on learning, Cognitive Science, vol. 12, No. 2, 1988.
15. Martínez, R. D., MSX88: Una Herramienta para la Enseñanza de la Estructura y Funcionamiento de los Ordenadores, I Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica, 1994.

A two LOM Aplication Profiles: for Learning Objects and for Information Objetcs

ALFONSO VICENTE[1], REGINA MOTZ[1]

[1] Instituto de Computación,Universidad de la República
Julio Herrera y Reissig 565, 11300, Montevideo, Uruguay
{avicente, rmotz}@fing.edu.uy

Abstract. *Learning Objects are the central concept of the current paradigm of E-Learning, but curiously, there is still a widespread confusion about how much to include, how big should be, or what is the "correct" granularity for a Learning Object. Because of this, different works use the same term to different things. This paper attempts to differentiate the concepts of Learning Objects and Information Objects, and analyze the potential for achieving adaptivity in the two levels. Particularly, studying the design of two LOM application profiles for exploit the specifics of each level of granularity.*

Keywords: *Learning Objects (LOs), Information Objects (IOs), Learning Object Metadata (LOM).*

1. Introduction

Learning Objects (hereafter LOs) are the central concept of the current paradigm of E-Learning. They were conceived as building blocks, which we can build a lesson, a unit, or a course. LOs are for the E-Learning what the objects are for the Object-Oriented Programming paradigm. They allow benefits in terms of reuse, economy and distributed development; because it is possible the widespread use of LO repositories, and the automation of the search, selection and use of LOs.

A traditionally metaphor for LOs was the LEGO blocks[1] [1]. As Wiley notes, the main problem with this metaphor is that any piece can be combined with any other in almost any way. This generates a LEGO-type thinking, which can conduce to the idea of "open a box of learning objects and have fun assembling them". In [2] Wiley proposes to differentiate between several types of LOs, some of these composed of other LOs. However, all objects are called LOs, whatever their granularity.

In [3] CISCO Systems presents its Learning Object Strategy, with a hierarchy of objects, locating the LOs (in CISCO's terminology RLOs, meaning Reusable Learning Objects) into the hierarchy, following the Learnativity's

[1] Coined by Wayne Hodgins while watching their children play, in 1990.

content ecosystem (see Figure 1). In this work, the term RLO is an object with a specific level of granularity.
This kind of disagreement in terminology (different works using the same term to different things) has caused much confusion in the literature. However, there is consensus among most authors in consider a LO should be centered in an objective.

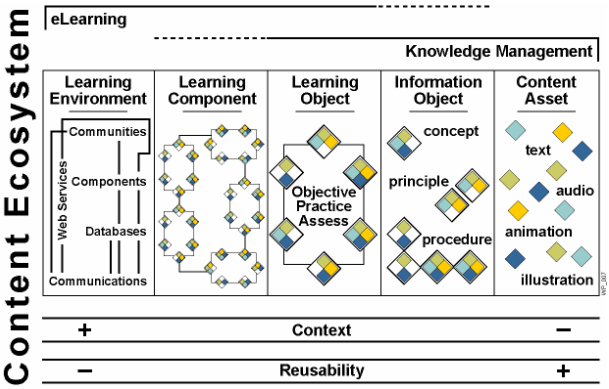


Figure 1: *A 2003 image of the Learnativity's content ecosystem of Wayne Hodgins, from [3]*

In those days, the adaptivity goal appears on the scene, causing a movement of attention toward more fine-grained objects. As far as I know, the only approach is to reduce the granularity of LOs, and in the worst case, this may cause LOs to lose their objective.
In respect to education, there are two important trends in E-Learning: one focused in pedagogy (e.g. instructional design, educational modeling) and one focused in adaptivity; and not many works emphasize the two trends. Adaptivity relies on reusability, and in the LO context, the statement "smaller is better" apply. But for the instructional design defenders it is clear that a LO should not be so small, in order to perform its pedagogical function.
One solution might be to begin to differentiate between LOs and IOs in the practice. This might be supported by the use of some specific metadata to each kind of object.
The most widespread metadata standard to describe educational content is LOM [4]. Some authors, like [7, 8], have already extended LOM, but have made only one extension to fit the needs of their LOs.
The aim of this paper is to suggest the convenience of differentiate the concepts of LOs and IOs in the practice, and hence, have different metadata formats to describe them. This is a new alternative way to explore.
The next section provides background on the most widespread metadata standards for educational content, as well as a vision (rather subjective) of current and future trends. Section 3 explains how adaptivity relies on reuse,

and finally, on metadata. Section 4 summarizes the most related works. Section 5 introduces the two LOM applications profiles, one for LOs and another for IOs. Finally, Section 6 presents some conclusions and future work.

2. A few words about standards

To achieve the benefits of a LO strategy, standards must exist. The two most outstanding standards are LOM [4] and SCORM [5].

LOM (an IEEE LTSC standard basically unchanged from 2002) means Learning Object Metadata. It is a conceptual schema that let to describe educational content (but not only LOs) through an element hierarchy grouped in nine categories. An element can be simple or compounded, and the simple elements has a data type and a domain, typically a predefined vocabulary or a reference to another standard. There has been many criticisms of the generality of LOM [6, 7]; the IEEE recognizes LOM is generic, and describe the way to extend it, through application profiles. It's interesting to see LOM as an uncoupled standard, in the sense that each object has its own metadata, and that it is all that sets the standard. A few projects [8] have developed their own uncoupled platforms based on repositories of objects described by LOM, in most cases extending LOM through their own application profile. The roadmap of LOM evolution (as suggested by Erik Duval to the LTSC-LOM list in 26 June 2009) includes finishing the corrigenda process in order to resolve a small number of minor inconsistencies, then work in the DCAM and RDF binding, and finally "discuss what other items people want to work on".

SCORM (an ADL standard in constant development since 2001) means Shareable Content Object Reference Model. It was born to take the best from the early efforts, specifications and standards. SCORM integrates several existing standards, including LOM for descriptive metadata. In SCORM's terminology, a LO is a Shareable Content Object (SCO). In October 2001, SCORM 1.2 was the first real release of the standard. SCORM 2004 was a significantly improvement of the standard: eliminating ambiguities, making SCORM conformant with IEEE standards (including LOM), supporting ECMAScript (JavaScript), and adding optional features for sequencing and navigation. SCORM is not only about the metadata of the objects, but also about the packaging, sequencing and communication with the LMS. Nowadays, while ADL [9] will continue to develop SCORM 2004, LETSI [10] is working in a SCORM successor, called SCORM 2.0, because today's requirements go beyond the SCORM's original design scope. SCORM 2.0 has a modular architecture and goes in the direction of actual trends, like Web Services.

It is still unclear, but we can expect the imminent RDF binding of LOM together with SCORM 2.0 may contribute to a Semantic Web Services approach, to allow the exploit of common semantic and the delivery of learning "as a service". The future is promising, but the technology will not solve the conceptual issues.

3. Why to use LOM Metadata?

The use of standard metadata as LOM, provides a consensus vocabulary in order to made explicit some intrinsic concepts. The usefulness of this approach is that LOM is an excellent model to illustrate LOM's properties.
A simple adaptivity feature can choose to use a high quality or a low quality image, depending on the bandwidth. A more complex adaptivity feature can choose between first present the basic definitions or the key concepts, based in the learner's cognitive style. In any case, the adaptive features rely, at the end, on metadata. There should be metadata that describe the size of the images, and the instructional type of the objects. The richer the metadata, the greater the opportunities to achieve adaptivity. Noting the content ecosystem, we see that adaptivity could be achieved at several levels of hierarchy. For example, to choose between images of high or low resolution, we need metadata in the level of content assets. According to cognitive style, we could organize the IOs within a LO, in order to have first a definition or an example, and we need the instructional type of the IOs. According to the skills needed and the time available for the student, we could offer a sequential or a discover approach through LOs, and in the first case we need the suggested flow.
Adaptivity can be achieved by the use of LOM (or a LOM application profile) and algorithms that exploit these metadata. In the Activemath project [6, 8], a LOM application profile was used, with advanced Artificial Intelligence techniques to select and order the LOs.
The SCORM 2004 adopters have to face the fact that SCO is the minimum unit of interaction, but can achieve personalization introducing show-nothing SCOs [11], which allow executing instructional algorithms that decide what will be the next SCO. SCORM 2.0 changes direction and propose a modular approach that includes the consideration of specialized orchestration services.

4. Related work

In 1999 and 2000, Wiley [1, 2] argues that the LEGO metaphor generates a "LEGO-type thinking", in which the blocks can be assembled in any manner, and by anyone. Because of this way of thinking, some people generate educational content combining blocks without care about the absence of an instructional theory. The atoms metaphor is presented, and it is obvious that the atoms need to be assembled in certain structures prescribed by their own internal structure, and the assembler should have some training. Beyond the metaphors, the criticism is about treating Los like components of a knowledge management system and the author suggest the term Information Object would be appropriate in this case. Also, he presents a taxonomy that differentiates between five types of LOs. The Wiley's work is an early proposal of differentiate types of LOs, with one designed to support instructional strategies.

In 2003, a CISCO Systems whitepaper [3] identify Reusable Learning Objects (RLOs) and Reusable Information Objects (RIOs) in its strategy, depicted in see Figure 2). An RLO consist of an overview, a set of RIOs, a summary, a practice. The CISCO's view, maps the terms "lesson" for a RLO and "topic" for a RIO (however, in the RLO's definition it says that many RLOs can be combined to form a lesson). A RIO is classified based on their instructional purpose: concept, fact, process, principle or procedure.

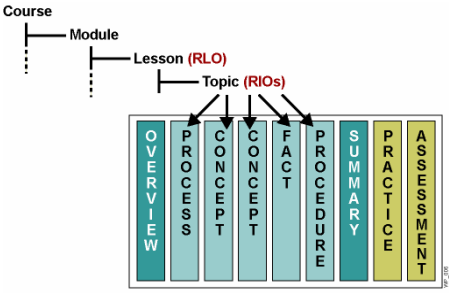


Figure 2: *The hierarchy of CISCO's Learning Object Strategy, from [3]*

In 2006, Roberts and Blackmon [11], tell the story of evolution of the grain size of SCOs in SCORM adopters. In the beginning, some course designers had one SCO per course. So, the SCORM adoption only assured the inter-LMS interoperability. The goal of reuse, led the grain size move from the course level to the learning objective level. The equation SCO = LO has been usual, but nowadays, the authors says that the goal of adaptivity can be reached with even more fine-grained SCOs and SCORM sequencing rules. What is clear is that the SCO's shrink cause the loss of context, and the need of more relationships. Hopefully, seems to be the trend support the shrinking of the SCOs: one of the enhancements of the SCORM's 4th edition is the possibility to share additional objective data and learner tracking information between SCOs. The equation SCO = IO may be the usual in the future.

In the "old-days", accordingly to the Wayne Hodgins content ecosystem, the fine-grained objects appear more related to Knowledge Management than to E-Learning. But in 2006, Wayne Hodgins says in an interview [12] that there is a meta-trend of "getting small", applied to E-Learning standards. The standards are "taken down to the smallest possible unit size and made to be interoperable", and he also warns us to "be prepared to see this trend continue every downward on the smallness scale as today's standards are themselves broken down into smaller individual components".

5. A two LOM application profile

For this work, the hierarchy presented in the content ecosystem of Wayne Hodgins is adopted (see Figure 1). A Content Asset is basically a file. An Information Object is a set of one or more Content Assets that can be identified with an instructional type (e.g. definition, motivational example). A Learning Object is a set of one or more Information Objects which is focused on an atomic objective (e.g. “know the concept of entity”, “motivate about the need of attributes in relations”). A combination of Learning Objects, where each has its own objective, is a Learning Component, although the Learning Component may have an objective of higher granularity (e.g. “mastering the relational model”). Finally, the Learning Environment includes not only the objects but also people and technology.
The adaptivity goal causes a movement of attention toward more fine-grained objects. Instead of simply focus our attention in the selected level of granularity, calling LOs to these objects, we can distinguish between LOs and IOs. Adaptivity can take place (at least) at the LO level and at the IO level. We advocate the convenience of particularly differentiate LOs and IOs, because these two intermediate levels of granularity allows the best possibilities for reuse.
Beyond the granularity issue, there are other conflicts between pedagogy and adaptivity. An instructional design imposes some structure for LOs, and the freedom degree of an adaptivity algorithm should not allow break these structure. There is an apparent dichotomy between instructional design demanding structure and adaptivity demanding freedom.
We argue this dichotomy can be reconciled: depending on a student's cognitive style, the system may choose a LO with the appropriate instructional design for the cognitive style. In this way we achieve adaptivity in terms of cognitive style and LOs with an appropriate instructional design.

We consider the following kinds of reuse for this work:

- **Redeploy**: reuse content “as is”, like redeploy a SCORM course in a LMS, or reuse an entire Learning Component
- **Rearrange**: reorder LOs within a Learning Component, like choose to see first a LO to “motivate about attributes in relations” or directly see a LO to “know about attributes in relations”. Here, we need metadata in LOs.
- **Rewrite:** borrow assets from IOs to create new IOs, like changing an image format based on the browser’s support, or the image quality based on the session bandwidth.

LOM is not enough to describe rich metadata, and we want different metadata at each level. Because of this, We propose a LOM application profile for Learning Objects (LOM-LO) and a LOM application profile for Information Objects (LOM-IO).

The application profiles add new elements to capture metadata not considered in LOM, define some elements as Required (R) or Forbidden (F) to ensure minimal descriptions and prohibit descriptions that do not apply, define constant values for some required elements, and create (or extend) some vocabularies.
The following tables describe the conceptual schema of the LOM-LO and LOM-IO application profiles.

Table 1: *The General category.*

	Element	LOM-LO	LOM-IO
	1. General	R	R
	1.1. Identifier	R	R
	1.1.1. Catalog	R	R
	1.1.2. Entry	R	R
	1.2. Title	R	R
	1.3. Language	R, “es-UY”	R, “es-UY”
	1.4. Description		
	1.5. Keyword		
	1.6. Coverage		
	1.7. Structure	F	F
	1.8. Aggregation Level	F	F
new	1.9. Object Type	R, “Learning Object”	R, “Information Object”

In the *General* category, we can highlight the prohibition of the *Aggregation Level* element, and the inclusion of a new element *Object Type* to differentiate between LOs and IOs. This is an improvement in terms of shared semantic, and can be used as a first point of access. Another important point is that we can not see the *Structure* as a *General* attribute, but an *Educational* attribute (see *Instructional Theory* on Table 5).

Table 2: *The Life Cycle category.*

Element	LOM-LO	LOM-IO
2. Life Cycle	R	R
2.1. Version	R	R
2.2. Status	R	R
2.3. Contribute	R (author)	R (author)
2.3.1. Role	R (author), “author”	R (author), “author”
2.3.2. Entity	R (author)	R (author)
2.3.3. Date	R (author)	R (author)

In the *Life Cycle* category, *Version*, *Status* and at least one *Contribute* element with the role author are required. Here there is no difference between LOs and IOs.

Table 3: *The Meta-Metadata category.*

Element	LOM-LO	LOM-IO
3. Meta-Metadata	R	R
3.1. Identifier		
3.1.1. Catalog		
3.1.2. Entry		
3.2. Contribute	R (creator)	R (creator)
3.2.1. Role	R (creator), “creator”	R (creator), “creator”
3.2.2. Entity	R (creator)	R (creator)
3.2.3. Date	R (creator)	R (creator)
3.3. Metadata Schema	R, “LOM-LO”	R, “LOM-IO”
3.4. Language	R, “es-UY”	R, “es-UY”

In the *Meta-Metadata* category, at least one *Contribute* element with the role creator, *Metadata Schema* and *Language* are required. While the classification of the object can be made through the *Object Type* element of the *General* category, the *Metadata Schema* can be useful to identify the specific application profile being used.

Table 4: *The Technical category.*

Element	LOM-LO	LOM-IO
4. Technical	R	R
4.1. Format		R
4.2. Size		R
4.3. Location		
4.4. Requirement		
4.4.1. OrComposite		
4.4.1.1. Type		
4.4.1.2. Name		
4.4.1.3. Minimum Version		
4.4.1.4. Maximum Version		
4.5. Installation Remarks		
4.6. Other Platform Requirements		
4.7. Duration		

The only distinction between LOs and IOs in this category is the requirement of the *Format* element for IOs. One LO can be materialized and have a format or may consist only of metadata (like a view over the IOs).

Table 5: The Educational category.

	Element	LOM-LO	LOM-IO
	5. Educational	R	R
	5.1. Interactivity Type	R	R
	5.2. Learning Resource Type	F	F
	5.3. Interactivity Level	R	R
	5.4. Semantic Density	R	R
	5.5. Intended End User Role		
	5.6. Context		
	5.7. Typical Age Range		
	5.8. Difficulty	R	R
	5.9. Typical Learning Time		
	5.10. Description		
	5.11. Language	R, “es-UY”	R, “es-UY”
new	5.12. Media Type	F	R
new	5.13. Instructional Type	R	R
new	5.14. Instructional Theory	R	F

In the Educational category, we can highlight the replacement of the controversial element *Learning Resource Type*, for the elements *Media Type* and *Instructional Type*, and the inclusion of the new element *Instructional Theory*, only for LOs. Examples of vocabularies for these elements are showed below:

- Media Type = {text, diagram, figure, graph, slide, table}
- Instructional Type = {exercise, example, simulation, question, questionnaire, exam, index, experiment, problem statement, self assessment, lecture}
- Instructional Theory = {sequential, learning by doing, learning by example, exploration}The *Instructional Theory* element replaces, with advantages, the *Structure* element in the *General* category.

For the *Rights* category we define all elements, except *Description*, as required.

Table 6: *The Relation category.*

Element	LOM-LO	LOM-IO
7. Relation	R	
7.1. Kind	R (has part), “haspart”	
7.2. Resource	R	
7.2.1. Identifier	R	
7.2.1.1. Catalog	R (URI), “URI”	
7.2.1.2. Entry	R	
7.2.2. Description		

The *Relation* category, with at least one haspart relation required, allows us to describe the composition of a LO in terms of the IOs which compose it. Extensions are not presented for categories Annotation and Classification.

6. Conclusion and Future Work

The search of adaptivity has led to a trend to decrease the granularity of LOs. However, the consideration of instructional design associated with LOs, implies a limit to this trend. In this work, we present a two LOM application profiles to manage the difference between LOs and IOs. This approach is based on the distinction between LOs and IOs, and focuses on the capture of metadata at these two levels.

Because LOM may remain current, and conceptually unchanged, for a while longer, the way to adapt it will be through application profiles. Two LOM application profiles are proposed, that exploits the particularities of each level of granularity.

Thisl work requires a proper empirical validation, and this is the most direct future work. However, there are have a couple of interesting issues to address.

Many of the descriptions can easily be generated automatically, as the size of the IOs. Some others may also be generated automatically, but not as easily, as the haspart relationships within an authoring tool. In other cases where the metadata exists, it might be interesting to validate it. The automatic generation and validation of the metadata are an interesting issue to investigate.

Another interesting issue is the attempt of reconciliation between pedagogy and adaptivity, through a mapping between cognitive styles and instructional design. In technology terms, a system may search and choose, or automatically build, a LO with the appropriate instructional design for the cognitive style.

In either of these two issues, we need to work with more educational people in our teams that help us with their pedagogical knowledge.

Acknowledgments

We want to thank Ximena Otegui, for her advice on educational issues; and to Patricia Orecchia, Silvia Motta and Lilian Sapelli, for their corrections and suggestions on English writing.

This work was supported by Comisión Sectorial de Enseñanza (CSE-UdelaR), JARDIN and SoLiTe projects.

References

1. Wiley, D. A., The Post-LEGO Learning Object (1999). Retrieved July 30, 2009, from http://opencontent.org/docs/post-lego.pdf.
2. Wiley, D. A., Connecting learning objects to instructional design theory, 2000. Retrieved July 30, 2009, from http://reusability.org/read/chapters/wiley.doc.
3. Cisco Systems: Reusable Learning Object Strategy: Designing and Developing Learning Objects for Multiple Learning Approaches, 2003. Retrieved July 30, 2009, from http://www.e-novalia.com/materiales/RLOW__07_03.pdf.

4. LOM Standard, http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf, last accessed July 30, 2009.
5. SCORM Standard, http://www.adlnet.gov/technologies/scorm/default.aspx, last accessed July 30, 2009.
6. Ullrich, C., The Learning-Resource-Type is Dead, Long Live the Learning-Resource-Type! 2005. Retrieved July 30, 2009, from http://www.carstenullrich.net/pubs/Ullrich-LearningResource-LOLD-2005.pdf.
7. Canabal, M., Sarasa, A., Sacristán, J.C., LOM-ES: Un perfil de aplicación de LOM, 2008. Retrieved July 30, 2009, from http://www.educaplus.org/documentos/lom-es_v1.pdf.
8. ActiveMath, http://www.activemath.org, last accessed July 30, 2009.
9. Advanced Distributed Learning, http://www.adlnet.org, last accessed July 30, 2009.
10. The International Federation for Learning, Education, and Training Systems Interoperability, https://letsi.org, last accessed July 30, 2009.
11. Roberts, E. J., Blackmon, W.H.: SCO Sighs: Why ADL Won't Say How Big SCOs Should Be, in Interservice/Industry Training, Simulation, and Education Conference (2006).
12. Interview with Wayne Hodgins, 2006. Retrieved July 30, 2009, http://www.profetic.org/spip.php?article7949.

ZinjaI: An Integrated Development Environment for a first programming course with C++

PABLO NOVAR[1], HORACIO LOYARTE[1]

[1]Universidad Nacional del Litoral, Facultad de Ingeniería y Cs. Hídricas
Departamento de Informática. Santa Fe, Argentina

Abstract. *Most of the students in Argentinian universities tends to experience huge adaptation problem over their first year. It is the main cause for very high indexes of abandonment. In the case of computing/informatics systems careers, in the first programming course, the students must learn a series of concepts related to computing algorithms abstraction, programming language syntax and the real implementation of programs using C++. It is a known fact that this is a cryptic language for the beginner programmer, and usually the very complex Integrated Development Environments (IDE) existing today are not designed to solve this particular issue. Instead, the software seem to be an additional handicap. ZinjaI is a new IDE for writing C++ programs developed with student's needs in mind, with powerful features for making design, edition, debugging and logic tracing of programs simpler tasks. The utilization of this tool in several first year cohorts seems to make a significant improvement for the learning process.*

Keywords: *Integrated Development Environment, programming teaching, C++.*

1. Introduction

Argentinian universities have minimal admission requirements for degree careers and the students pay no fee for their studies. In addition, middle school is going through a crisis. All these factors produce that meaningful number of applicants for coursing a degree career have serious difficulties for successfully finish the first year subjects of its curriculum. These difficulties are increased in the engineering careers, like the case of study (*Ingeniería Informática*, in *Universidad Nacional del Litoral*, Santa Fe, Argentina).
This career has an initial enrollment of about 300 students, and the drop out in the first year is approximately 50%. Through surveys and assessment process analysis it was possible to detect the subject *Fundamentos de*

1 The National Universities are public and free in Argentina.

Programación (Programming Fundamentals) as one of the most difficult ones for students. This subject develops computational algorithm concepts and the students have to solve problems creating computer programs using the standard language ANSI C++.
It was observed that professional C++ programming environment in addition to some cryptic characteristics of the language, tends to confuse and slow down the learning process of programming concepts for inexpert students. The language syntax, the programming environment, the error messages, the English language (the students are Spanish speakers), etc., constitute additional obstacles to basic difficulty when learning the concepts, the logic and basic structures of computational algorithmic: the main objective of the subject.
As a result of this analysis, it was proposed the development of new IDE, ZinajI[1], aimed to learning/teaching needs, with features for facilitating edition, debugging and testing of C++ programs, and contributing in defective way to learning programming basis in general and C++ language in particular.

2. IDE for teaching of programming

The professional IDEs provide important features for accelerating production of complex code to developers. It is common to find in most of them: automatic indentation, syntax highlighting, integrated debugging features for facilitating erroneous logic detection, trace and breakpoints for code analysis, compilation and execution a through menu commands and many other characteristics.
Besides these features for developing programs, in the learning processes, an introductory course of C++ programming requires other characteristics that clash with professional IDE design[2, 3]:

- The programming interfaces must be clean, simple and intuitive. The professional IDEs have generally complex interfaces flooded with tools and commands that the student will not take advantage. These kinds of interfaces constitutes an obstacle and distraction for learning.

- The setup must be simple and the software installed must have a minimal resource requirement for running in obsolete PCs. It cannot demand high hardware requirements to students who are starting to learn how to program.

- The IDE must provide to the user several levels of helps and assistance in order to improve and smooth the learning process (i.e.: early error detection). That means that the IDE must selects the necessary information for each context and avoid another distracting information (i.e.: compilation parameters, idiomatic barriers, etc.).

The powerful features of professional IDEs demand the presence of many command menus and other elements. Most of them will never be used by a beginner programmer and only lead to confusion.

3. Principal features of ZinjaI

ZinjaI was developed having all the topics introduced in the previous section in mind in order to provide a more suitable environment for students. Some its main features are:

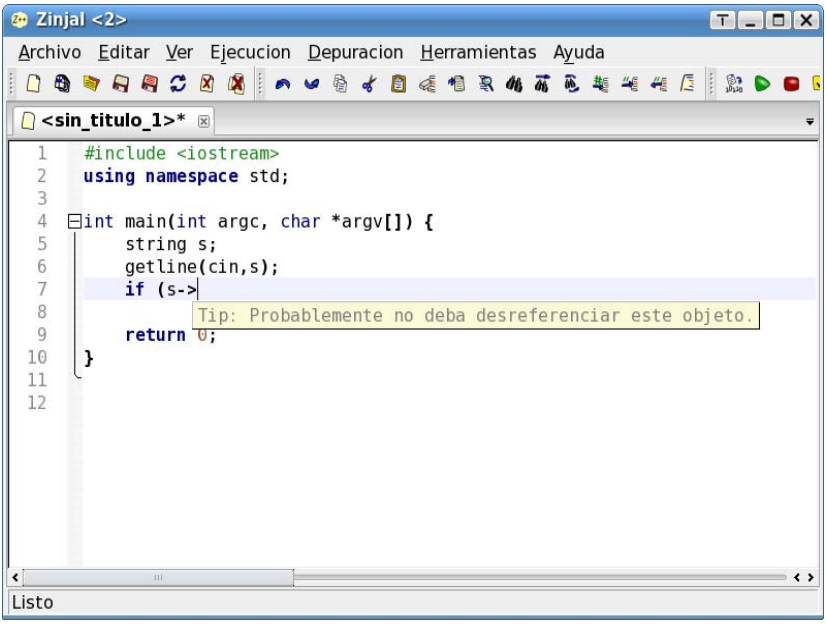


Figure 1: *ZinjaI 's basic interface. It shows emerging message wich warns the user about potential errors detected by the auto-completion system.*

- Easy distribution: it is a free and open source software. The whole system (IDE, compiler, debugger, etc.) is deployed in an easy setup package (from 8 to 35 Mb, depending on the platform and version). The software is also prepared to run without any installation process at all.

- Portability: the system can run in both Microsoft Windows (from Windows 98 to latest official release, Windows Vista) and GNU/Linux (in any modern distribution), adapting projects between platforms in a transparent way for the user.

- The initial interface is very simple and clearly intuitive. By each edition tab the system proposes the initial code of a C++ program, so the user can start writing his solution right inside the main function (like shows Fig. 1) through a set of predefined templates or through the new file wizard, and test his program with a single click. So, the system allows the rapid development of C++ programs without need to create, configure and customize projects.

- It has several edition facilities, like syntax highlight, intelligent and automatic indentation, advanced search and replacement, folding and expansion of logical code blocks and some special commands for C++ like automatic header file directive inclusion, management of source's comments, context sensitive auto-complete system, emerging help for function, etc.

- ZinjaI presents a complete help system (IDE documentation, tutorials, advanced features, etc) and an integrated Spanish quick help about standard C++ language.

- The system parses and improves compiler output: errors and warnings are organized into a tree shape, restructuring some lines o discarding others, that result in an easier reading and interpretation.

- It also has a Project mode for management of multiple advanced execution and compilation profiles. The fact that new ZinjaI's releases are developed with the old ones show its capability for complex projects handling.

- Debugging system includes inspections management, hierarchical gdb objects exploration, breakpoints (basic breakpoints, conditional breakpoints and watchpoints with full scopes awareness), backtracing, step by step execution, especial table layouts for classes, vectors and matrices, with parsing and reformulation of debugger expressions in order to improve the quality of information presented.

- Very specific student aimed features such as generating and visualizing flow diagrams for selected pieces of code (Fig.2), sharing source and other text files through a LAN network for facilitating the teacher's job, automatic class hierarchy representation, etc.

- Finally, it integrates external tools without adding complexity to basic interface and most common tasks. In addition to student aimed tools, there is a set of advanced components for demanding users: documentation generation through Doxygen, visual interface design through wxFormsBuilder, profiled execution through gprof, source and text files comparison and merging, building scripts generation, etc.

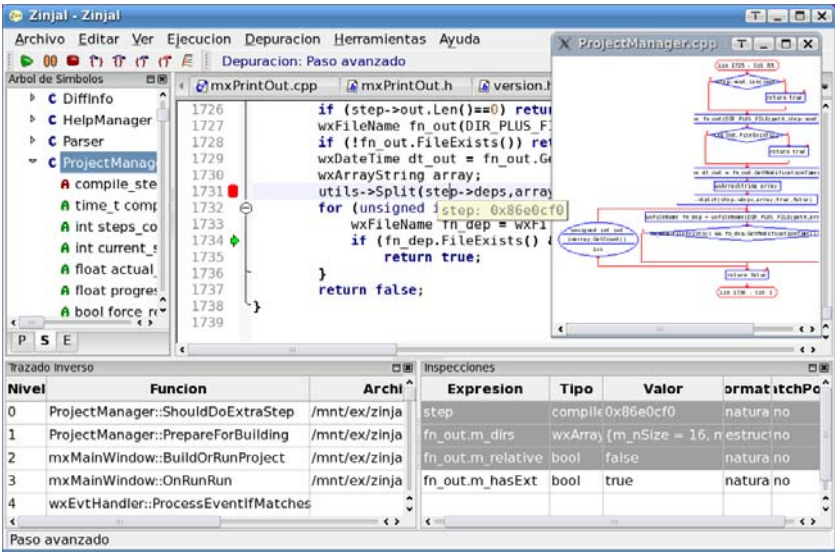


Fig. 2. *Project mode interface in a debugging session. It shows some additional tools in panels, and flow chart representation of the analyzed piece of code.*

The help system of ZinjaI describes in great detail all of its features. Of course, a beginner student in first contacts with ZinjaI can work without knowing the advanced ones. The student can incorporate these features while getting fluency in design and development of C++ programming for solving problems.

4. Development of ZinjaI

The software has been built employing a set of completely free and portable tools and libraries. Development was done on a GNU/Linux platform using the standard ANSI/ISO C++ programming language, object oriented programming paradigm, and wxWidgets library for presenting visual components. For compiling and debugging ZinjaI relies on GNU tools (GCC and GDB). Also, some code taken from other free projects was adapted to implement some features (i.e.: the parser that ZinjaI internally uses was taken from the RedHat's Source Navigator project). The main reasons for choosing wxWidgets over other powerful alternatives such as QT, GTK+, FLTK, etc. includes: its object oriented interface, its very high portability and its deep integration with the host operative system, and the fact that it provides a whole framework that also simplify process management with input/output redirection, sockets and other networking components, files and string manipulation, and more.

5. Impact and results in classroom application of ZinjaI

To test the tools acceptance level and main aspects to guide its development an survey was performed. Then, to verify its influence in the educational process the survey was applied in two parallel courses with a significant number of students: one of them (in FICH-UNL[2], where the subject is called *Fundamentos de Programación*, Fundamental Programming, 91 students) was employing the proposed development; the other one (in FI-UNER[3], where the subject is called *Computación I*, 83 students) continued in the traditional teaching mode and without the tool. The comparison gets an special relevance due to:

- very similar topics were taught in both classes
- same teacher in charge of them
- similar number of students
- both courses are in the first year of their respective careers

Analyzing the collected information, several observations can be done:

1. The tool has achieved an important acceptance level in its first intervention: the very first ZinjaI's version was deployed in the middle of course time and more than 75% of the students has adopted it as his main working platform. It must be said that the every student can choose whatever software he wants to carry on the practices, considering that the university labs provide several free and commercial alternatives. Asking for the reasons to the students that choose not to work in ZinjaI the most frequent one is that they where already very familiar with other specific software.

- The tool is user friendly: when the students where interrogated about their reasons to work with it, the main answers where the teacher's recommendation (53%), the ease of use (40%) and the fact that its interface and help is in their home language (36%). In average, the grade of additional difficulty perceived by the students working with ZinjaI when they start programming and change from pseudocode to C++ is very similar. However, when they were asked about the comfort and help level that the software provides the difference is bigger in benefit of ZinjaI (about 3 points in a 1-10 scale for both aspects).

- Some of the strongest system feature where not explicitly appreciated by the students: between the main reasons for choosing

2 FICH-UNL: Facultad de Ingeniería y Ciencias Hídricas, Universidad Nacional del Litoral. Santa Fe, Argentina.

3 FI-UNER: Facultad de Ingeniería, Universidad Nacional de Entre Ríos. Entre Ríos, Argentina.

that software, “the amount of different features” (considering the specific ones aimed to classroom work) or “low requirements” (as an example, it only takes about 11MB in memory when running in its initial mode) has showed low percentages (around 14% each).

- Students believes that ZinjaI has a positive impact in the learning process: in contrast to the results exposed in the previous two items, when the students where asked about how they saw the software influence, more than 72% said that the tool contributed to the progress they made, 22% said that it only let them work faster and in a more comfortable way but without a real influence in their academic performance, and only 7% expressed that there was no difference when comparing to other tools and less than 2% that it has a negative impact. Even if those answers are loaded with a high level of subjectivity and very conditioned by the students lack of experience, the big number of individuals in the samples provides some extra credibility to those numbers.

Looking at these results it can be observed that the introduction of the new software into the learning process has made a positive difference for the student's experience. However, the original main design feature was supposed to be a new debugging system conceived to create in the students the habit of taking advantage of debugging as a process not only for finding and fixing their own bugs, but also for analyzing right programs in order to investigate their behaviors and get a better understanding of many theoretical and practical topics introduced in the course [4-9]. In the presented work, the students only have had access to a limited and basic debugging system. This feature was still under heavy development, and that's one reason why it's impact is expected to be actually bigger with the new versions. It also must be said that in those courses there was another tool being tested that had significant influence in the comparison (PseInt[10,11], a pseudo-code interpreter employed in the four first weeks to introduce the most basic and general logical aspects before getting contact with a real programming language), reason why the student's qualifications can't be taken as a direct indicator of how ZinjaI affects the final results.

6. Conclusions and further work

The proposed development has showed a positive impact in the learning process of FICH-UNL students. This fact is supported by partial results extracted from surveys analysis and comparisons between the two selected group of students. The feedback level from academic community (both teachers and students) is determinant and essential to lead the development of new features. The software presented is now in a noticeable more mature and stable state comparing to the releases used for this study. The authors pretend to continue evaluating the impact in the following cohort and improve the software achieving a better integration between the tools usage and the learning/teaching experience.

References

1. Zinja, I., Integrated development environment. Available at http://zinjai.sourceforge.net/.
2. Reis, C., Cartwright, R., Tamimng a Professional IDE for the Classroom, SIGCSE'04, March 3-7, 2004, Norfolk, Virginia, USA, 2004.
3. Moroni, N., "Entornos Para el Aprendizaje de la Programación".
4. Valles, M., Técnicas cualitativas de investigación social, Síntesis, Madrid, 2000.
5. Cross, J. H. et al., "Using the Debugger as an Integral Part of Teaching CS1", 32nd ASEE/IEEE Frotiers in education Conference, Noviembre 2002.
6. Ko, A. J., "Preserving Non-Programmers' Motivation with Error-Prevention and Debugging Support Tools", 2003.
7. Chmiel, R. and Loui, M. C., "An Integrated Approach to Instruction in Debugging Computer Programs", 33rd ASEE/IEEE Frotiers in education Conference, Noviembre, 2003.
8. Nagvajara, P. and Taskin, B., "Design-For-Debug: A Vital Aspect in Education", Internacional Conference on Microelectronic Systems Education, 2007.
9. Gallego, C. M. et al., "Depuración Estructural: Acercando la teoría a la práctica de la Programación".
10. Loyarte, H. and Novara, P., "Desarrollo de un Intérprete de Pseudocódigo para la Enseñanza de Algorítmica Computacional", I Congreso de Tecnología en Educación y Educación en Tecnología, TE&ET, La Plata, 2006.
11. PseInt, Spanish pseudocode interpreter. Available at http://pseint.sourceforge.net/.

Virtual characters as study guides. Evolution towards a virtual collaborative learning environment

GONZALEZ ALEJANDRO[1], MADOZ CRISTINA[1], GORGA GLADYS[1], DE GIUSTI ARMANDO[1]

[1]Institute of Research in Computer Science III-LIDI. School of Computer Science. National University of La Plata. Argentina
{agonzalez, cmadoz, ggorga, degiusti}@lidi.info.unlp.edu.ar

Abstract. *An analysis of the characteristics of the use of virtual characters in hypermedia study materials is carried out. The results of using characters in study materials for the introductory course to Computer Science majors are presented.*
Instructor and student characters act as companion buddies during the study, providing indications, guidance, and eliciting questions.
The research lines required for transferring these characters to a virtual environment that is "intelligent" and collaborative are described.

Key words: *multimedia, hypermedia, virtual characters, collaborative virtual environments.*

1. Introduction

Our teaching experience indicates that, when starting any university major, and particularly in the case of Computer Science, students have to face new situations that they need to overcome in order to be able to succeed in the university environment. These situations are strongly related with cultural, social, and emotional aspects of new students, as well as aspects related to the incorporation of information technology in a different academic environment.
Taking this into account, a proposal aimed at incorporating and using ICTs in the learning process is analyzed.
A working methodology favoring student motivation during the initial stage of their majors is established, promoting innovative educational proposals to assist them during the learning process. In this paper, the results of a magister thesis in Information Technologies Applied to Education are presented [10]. Within this context, educational contents and materials of the introductory course have been reviewed, and a new proposal emphasizing the introduction of information technology resources to be used during the learning process in order to promote and facilitate the incorporation of the new concepts and contents that students will learn during this introductory course has been produced [6] [12].
In a general context, the proposed contents are considered to be suitable to successfully level all students; however, it is clear that students will have to make

a considerable effort to incorporate these contents due to their insufficient previous training and the brief duration of the course.
For this reason, an additional hypermedia web resource is incorporated to the proposal so that students can chose when and where to study the topics included in the introductory course. Thus, by combining a high interactivity level with the contents, the possibility of analyzing various situations based on eliciting questions, having more time to assimilate the concepts, and the option of completing systematic evaluations, students will be immersed in a favorable environment to stimulate satisfactory learning situations.
This new educational space that combines technological aspects and multimedia features will allow the incorporation not only of new knowledge, but also of skills, aptitudes and attitudes that will play a role in the development of the students' critical and reflective thinking [7].

2. Theoretical framework

Cognitive learning theories emphasize the acquisition of knowledge and internal mental structures. These theories are mainly aimed at the conceptualization of the learning processes of students, and they analyze how the information is received, organized, stored and located. Through these theories, knowledge acquisition is described as a mental activity that involves internal codification and structuring on the part of students. In this context, students are seen as a very active participant in the learning process.
Additionally, the “situational learning” [11] considers social interaction as an essential component of the learning process. In this context, knowledge is derived from the activity, the environment, and the culture to which students belong. A significant concept in situational learning is that of "authentically activated", that is, the activity is defined by a community of practice and not by an academic analysis of contents. The purpose here is not that of recovering intact knowledge structures, but rather, that of providing students the means for them to be able to create novel understandings that are situationally specific by "assembling" previous knowledge, obtained from different sources relevant to the problem at hand. Constructivists highlight the flexible use of previous knowledge more than the ability to remember pre-produced ways of thinking. There seems to be agreement among the various perspectives of constructivism [8]. These agreements are based on:
a) The learning process is (or it should be) an active meaning construction process rather than an information acquisition one.
b) The instruction process is a support or mediation process for this meaning construction process that goes beyond the communication or transmission of information. There is also agreement in considering that, as proposed by Jerome Bruner, knowledge is not in disciplinar contents, but in the constructive (or co-constructive) activity of the person on the content area, the same as in any given socio-educational context.
Rogoff and Hernández, among others, have established significant distinctions among the main constructivist psycho-educational paradigms that become

instructional approaches. They mention, among others, a model called "instructional model of experts and beginners", where the actions of the educational agent are emphasized: experts are in charge of modeling and promoting certain knowledge on beginners. In this sense, the experts in the model would help favoring and bringing beginners closer to knowledge by means of the mechanisms normally used for their tasks.

3. Creation of virtual characters

The incorporation of virtual characters to study materials is developed by means of a "multimedia script" process. The initial idea of the script is answering these questions: "what is the purpose?" and "who is it aimed at?". Ideas then become texts, images, and sounds. To this end, it is advisable to have a work routine to structure contents [9]. This work routine for creating scripts will have the following phases: contents, narration, icons, sounds, and technical aspects.

Script contents are given by the textual material that will be used in the various sequences and the way in which this material is inter-related through a conceptual hierarchy that will go from general to specific.

The narrated script establishes the way in which information will be presented. It is equivalent to what is known as literary script, indicating point of view and style.

The iconic script shows the images that are available: graphs, photographs, figures, charts, video or animation images, and at which time of the narration these will be used.

The sound script is developed synchronically with the narrative script. Sound records must be sequential, and this sequence will be indicated by means of an order number.

The technical script is produced by computer science professionals as they become acquainted with the idea of the educator. It consists in defining the bases for development, methodology, programs to use, presentation formats, screen design, effects to be used in each section, etc.

At present, various resources that allow designing through a prototype technique are used. It can be of disposable or evolutive nature. In the case of disposable designs, a slide or screen presentation product is normally used to simulate and present the contents script. This product is then passed on to the programmer, who uses it to extract the information and build the final product in a different format and with a different software application. In the case of evolutive designs, all work is done with a software application that allows generating the final study material.

Bou Bouzá [2] takes movie script elaboration principles and applies them to multimedia scripts. Movie scripts, the same as television scripts, have three elements: discourse, performance and message.

In multimedia applications, the discourse element of movie scripts is equivalent to the information to transmit, that is, what we want to tell. The performance element is present as performance components. The message or background

can be found behind the plot or in the conclusion drawn from the story we are being told, or the information provided.
In the design of educational applications, Bou Bouzá establishes two types of possible scenes: those related to the educational strategy, called "educational loop", and those used as strategy companions or support, called "narrative loop". The term "loop" refers to the repetition of a succession of similar elements. The narrative loop is used in this context to accompany the educational process, for instance, by describing concepts or asking questions that guide students in their learning process.
It is desirable that applications make users apply their own experience to what they are learning through educational loops; in this way, the narrative function acts as a reinforcement of the educational function. For example, a series of practical exercises would be part of the educational loop, and it would be accompanied by narrative loops to introduce concepts or present the corresponding final conclusions.

4. Developing characters

The creation of virtual characters is tackled taking Bou Bouzá's script generation concepts as a starting point, and adding Rib Davis' characterization suggestions. The experts/beginners model is used for the construction of a cognitive model.
Rib Davis, the same as Bou Bouzá, works on movie scripts and presents the characteristics required for building characters, which is useful for the development of multimedia characters.
Scripts are filled with characters. The character creation process can be seen as "a compilation of individual fragments taken from here and there", not randomly chosen, but selected with the intention of creating with them characters that are both credible and appropriate for a specific script [15].
Characters are defined from a set of "personal features", in addition to a scenario, a history and one or more objectives. According to Rib, the ingredients needed to build a character are those that result from an individual and that make each person different from the rest. Even though Rib describes how to create characters for theater, cinema and literature, the elements used for creating these can be adapted to the creation of virtual characters within a multimedia script.
In order to create a character, three basic aspects have to be considered:
a. How is the character when it is born (due to genetics and its environment)?
This is called "Birth features" and they include: gender, race, social class, family background, and name.
b. How does the character change and evolve through learning and experience?
This aspect considers features related to education, aptitudes, family, sexuality, background history.
c. How is the character now?
This aspect includes age, appearance, friends and foes, vision of the world, beliefs, personality, use of language, commonly used expressions.

Virtual characters are built taking into account that they are going to accompany students through the multimedia play, pretending to be helpers or as guidance for the learning process. The expert/beginner model refers to the desired cognitive characteristics for each character.
In the case of experts, unlike beginners, it can be said that [3][4]:
1. They are aware of the most significant characteristics and patterns of information.
2. They have acquired a great deal of knowledge, and this knowledge is organized and available in such a way that it shows a deep understanding of their object of study.
3. The expert knowledge is not reduced to a set of facts or propositions, but it reflects application contexts, that is, it is "conditional" or it is subject to a set of circumstances.
4. They can retrieve, with little effort, the most relevant aspects of their knowledge.
5. They have a thorough knowledge of the discipline or area of knowledge; however, this is no guarantee that they are able to teach others.
6. They have various flexibility levels in their approach to new situations.
Experts are able to faster automate the operations involved in information processing, which allows them focusing on higher-level processes such as analysis and synthesis. Thus, they know the context, they operate, and they know how to successfully “move” in it [17].

5. Experience carried out. Characters for the EPA module

The contribution of this paper consists in the creation of three virtual characters for a module of the introductory course to Computer Science majors at the UNLP.
The introductory course is given on a classroom basis and has a duration of 6 weeks including diagnostic tests. The module selected is that of “Expression of Problems and Algorithms” (EPA). In this module, students tackle problem resolution and algorithm development topics using a basic programming language called Visual DaVinci, which contains a set of instructions focused on teaching control structures and modularization in algorithm expression [5].
In EPA, students must put into practice their cognitive skills to solve problems. They usually face obstacles during the learning process, based on which different teaching strategies are sought.
A study of the mistakes made in diagnostic tests in 2006 and 2007 was carried out, and the most common difficulties for problem resolution were identified. In theoretical and practical classes, as well as in surveys carried out by the faculty in previous years, it was noted that students had difficulty in reaching the abstraction level necessary to solve problems that have to be solved with computers.
Based on the data obtained, students indicate that they have difficulties when solving one specific problem presented in the diagnostic test. Sixty percent of students indicate that they do not know how to correctly break down the

problem into sub-problems and, if they are able to do it, they are not able to correctly communicate the modules. The most common problems are related to problem interpretation and contents related to modularization and parameter conversion concepts.

These issues that present difficulties for most students are selected for this experience, so as to favor learning strategies with hypermedia material including exercises accompanied by tutorial features.

To design the material, the cognitive skills required to solve problems are analyzed. In order to clearly present the necessary steps, a multimedia script is developed introducing the use of characters for the different conflictive situations, so as to facilitate the task and provide guidance aimed at bridging the gap between the beginner and expert knowledge.

The incorporation of characters is aimed at generating a strategy that helps students make the transition from high school to university by improving the understanding of the topics to learn. To this end, the characters are conceived from various study areas that provide a structure of knowledge areas; students are familiar with this type of organization because high-school subjects are grouped in study areas.

The story presented by the three characters guides the learning process, and the discourse used is based on this guidance and usual doubts students have when learning the first steps of algorithm generation. In this case, three characters are used: two tutors and one student.

"Tutor" characters help the student and provide guidance from their respective areas of knowledge. The "student" character is a possible student model for this material. For "expert tutors," various knowledge areas were analyzed through the cognitive characteristics of renowned experts in each possible area [14]. Thus, two characters were selected, and their characteristics analyzed and adapted to each of the personality stereotypes.

The selected study areas are Computer Science and Philosophy. These two areas allowed creating characters based on their "expert" characteristics. The personalities selected for the tutors were Socrates and Ada Byron King, since they both have interesting expertise levels to help beginner students in the resolution of problems with a computer.

In the case of the third character, a student with learning difficulties was selected.

The character "Ada" is named after the original personality, but, instead of reflecting its original time of existence, it is a contemporary female with no reference to social class. The real educational characteristics are also kept and it is presented with a successful and creative intelligence. Ada knows that she can apply her cognitive strategies to the resolution of problems that are not only of a scientific nature. She is fun and addresses students in a nice manner.

"Soca", the second character, is a pseudonym for Socrates. This is a male character and his features are those corresponding to his time; however, his appearance is more related to Native American features. He uses his communication skills and resorts to questions to provide cognitive clues to guide students.

Edu, the student character, is short for Eduardo. This is a male character. He represents a student that is new to the university environment, originally from

out of town, who decides to move to this city and study at the National University of La Plata. He is likeable and is at a loss with the new topics he must learn. He asks for help, both to his tutors and the real students that are using the study material.

Figure 1: *Characters of the EPA module: Ada, Soca and Edu*

6. Results obtained

The material was used during the two weeks that preceded the diagnostic tests of the introductory course to Computer Science majors. A total of 24 students participated.

The material was designed following the evolutive prototype method.

A classroom workshop was prepared to share the experience of using the material, and to review its structure based on the use of the students.

The students used the material at home, and there was a classroom workshop for those students who used the study material. To assess the material, a survey covering several items was conducted.

One of these items was related to the essential features that identify each character and their relevant participation in the material. It included statements that should be graded in the following scale: I strongly agree, I agree, I agree a little, I do not agree. In the case of virtual tutors, their strengths as study guides are ascertained. In the case of the student character, the statements were aimed at determining how closely real students identified with this virtual model.

In the case of Soca, the statement was: “With the help of Soca I was able to ask myself more questions before I started writing the solution to the problem in Visual DaVinci.” In this case, one of the highest agreement rates was recorded: 70% "I agree", and 25% "I strongly agree". Both categories add up to 95%. Soca is fundamental for the methodology on which the material is based – asking questions and helping students be able to ask themselves questions to analyze the solution before writing it down.

As regards Ada, the statement was: “With the help of Ada I was able to easily understand the main aspects that could be solved and the data required”; here,

there was a 75% rate of agreements, one disagreement, and 25% of the students could not answer this question. If these results are combined with the age variable, interesting facts can be noted. The oldest student (47 years old) is the person who disagrees. The students who did not answer are all between 17 and 20 years old, and belong to the same group that could not answer other questions of the survey.
The least agreement rate corresponded to Edu. The statement was: "I identified myself with the character Edu". This question had the lowest percentage of agreements – 45% (30% strongly agree and 15% agree). The youngest students answered in the 45% group, whereas older students either agreed or disagreed. The 47-year-old student strongly disagreed. The total number of disagreements added to 40%, which is to be expected due to the average age of the group of students who took the survey and the features of Edu.
One of the first conclusions mentioned, the use of characters was appealing to all participating students.
The use of various study areas forces students to review the material and adapt it to the new discipline in order to be able to present it. The use of characters makes the presentation of the various topics more appealing and facilitates the introduction of the study areas.
The recreation of expert tutors from different disciplines took into consideration the analysis of the cognitive processes required to facilitate an approach between beginners and experts. This allowed generating the features of the characters Ada and Soca.
Students identified the character Edu as their equal or peer. It requires a greater interaction degree to allow molding the character to various preferences.
Among the modifications that can be introduced, it would be interesting that students could create their own student avatars o "being able to build themselves," as students frequently mentioned. This creation refers not only to the physical appearance and gender of Edu, but also to its personality and knowledge level. In the case of "personality", different profiles presenting Edu as disoriented as regards the topics, a know-it-all, hasty in his or her answers or with difficulties to reason, can be generated.

7. Work proposal: Evolution of the characters

The idea is providing all characters with a certain "intelligence" so that they become "virtual intelligent pedagogic agents." A pedagogic agent can be defined as an intelligent agent who observes the learning process of the students and makes decisions regarding how to better help each particular student. This agent can act as: tutor, apprentice, or assistant. It can help small groups of students who are collaboratively working in their learning process [1][18].
Students would move within a collaborative virtual learning environment which would be focused on what is being communicated and the assistance provided to the students so that they can learn together. This must be tackled with a suitable methodology that should include training activities for

teamwork that promote the development of cognitive strategies for the group task specialized in a study or expertise area [16].
The work carried out with the characters and their level of expertise allows establishing possible reasoning strategies for pedagogic agents. The type of virtual activities to develop needs to be reformulated so as to favor the distributed learning process and integrate virtual pedagogic agents in one environment.

8. Conclusions and future lines of work

The creation of the script for each character is a creative process that allows incorporating various multimedia discourse elements, ranging from voiceover, the introduction of the characters, their story and personality, to the presentation of procedures, cognitive clues and pieces of advice for the learning process. For the incorporation of each discourse, the task being introduced was considered, as well as the required cognitive processes, and a development of the software prototype that incorporated the hypermedia technology that was suitable for the topic.
Tutor characters are accepted as study guides. In principle, both profiles seem to be necessary for the resolution of problems with a computer – one of the profiles to teach how to reflect on the problems, and the other profile to guide students on the details of the concept they must learn.
In this context, students valued the interactivity that can be achieved with the object of study, which is evident from the numerous comments and answers to the survey question asking them to mention some characteristic that they found useful to understand how to solve problems with a computer – several students explicitly mentioned the advantages offered by the "interactive top-down" method. They also mentioned that being able to browse through the links of the diagram allows them to clearly see how the program works. Through the links, they can have a general view, as well as a view of the parts and the detail currently selected.
The development of interactive images to structure the reasoning process required for the development of an algorithm is also important – 60% of the surveyed students stated that being able to access the "interactive top-down" design helped them understand "how to build the solution to a problem and, from there, obtain the code or program."
This work has been designed considering aspects of the teaching-learning process. The proposal of assisting students with collaboration-based virtual learning environments implies achieving a suitable specification of requirements for the pedagogic agents, as well as researching tools and methods for the design and development of intelligent virtual environments.
From a pedagogic standpoint, instruction structures must be conceived and designed to allow collaborative work in the learning domain taking pedagogic agents into account.

References

1. Aguilar, R. and De Antonio, A., Agentes Pedagógicos Virtuales Inteligentes. Una estrategia para Entrenamiento de Equipos, in R. Rivera, M. Ostos, H. Andrade y O. Gutiérrez (Eds.), Avances en las Tecnologías de la Información, 4º CISC, August 2004, Veracruz, 2004.
2. Bou Bouzá, G., El guión multimedia, Grupo Anaya, Madrid, 1997.
3. Bransford, J. D., Brown A. L. and Cocking R., How People Learn: Brain, Mind, Experience, and School, The National Academy Press, Washington D.C., 1999.
4. Brown, J. S., Collins, A. and Duguid, P., Situated cognition and the culture of learning, Educational Researcher, 18, 1989.
5. Champredonde, R. and De Giusti., Design and Implementation of The Visual Da Vinci Language, Thesis defense, School of Computer Science, UNLP, 1997.
6. De Giusti, Madoz, Gorga G., Análisis del proceso de articulación para Alumnos de Informática, utilizando herramientas de Educación a Distancia, XII Congreso Argentino de Ciencias de la Computación, CACIC 2006, San Luis, 2006.
7. De Giusti, González, Gorga Madoz, Sanz, El caso de Algoritmos, Datos y Programas. Posibilidades de uso de un entorno virtual de enseñanza y aprendizaje según el perfil de los alumnos, III Congreso de Tecnología en Educación y Educación en Tecnología, TE&ET'08, Bahía Blanca, 2008.
8. Diaz Barriga, F., Principios de diseño instruccional de entornos de aprendizaje apoyados con TIC: un marco de referencia sociocultural y situado, Tecnología y Comunicación Educativas, Nº 41, 2005.
9. Galán Fajardo, E., El guión didáctico para materiales multimedia, Espéculo, Revista de estudios literarios, Universidad Complutense de Madrid, 2006. Last visited on June, 2008 at http://www.ucm.es/info/especulo/numero34/guionmu.html.
10. Gonzalez, A., TICs en el proceso de articulación entre la Escuela Media y la Universidad. Personajes virtuales como herramientas de un entorno de aprendizaje multimedia, Magister Thesis on Information Technologies Applied to Education, presented in December, 2008 at the School of Computer Science of the UNLP, 2008.
11. Lave, J. and Wenger, E., Communities of Practice: Learning, Meaning, and Identity: Cambridge University Press, 1998. Last visited on July 2008 at http://www.learning-theories.com/communities-of-practice-lave-and-wenger.html.
12. Madoz C., Gorga G., Análisis del proceso de articulación para Alumnos de Informática, utilizando herramientas de Educación a Distancia, TE&ET, Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología, vol. 1, No 1, 2006.
13. Malbrán, M. del C., Indagaciones en la mente del experto. Programa de Incentivos, UNLP, Proyecto H462, 2005.
14. Mayer, R., The Cambridge Handbook of Multimedia Learning, Cambridge University Press, United States of America, 2007.

15. Rib, D., Escribir guiones: desarrollo de personajes, Paidós, Barcelona, 2004.
16. Rickel, J. and Johnson, W. L., Virtual Humans for Team Training in Virtual Reality, in Proceedings of the 9th World Conference on AIED, 18, 1999.
17. Sternberg R., Metaphors of mind: Conceptions of the nature of the intelligence, New York, Cambridge University Press, 1990.
18. Vizcaino, A., Enhancing Collaborative Learning Using a Simulated Student Agent, Doctoral Thesis, Universidad de Castilla-La Mancha, 2002.

VII

Graphic Computation, Imagery and Visualization Workshop

Coastal Monitoring and Feature Estimation with Small Format Cameras: Application to the Shoreline of Monte Hermoso, Argentina

NATALIA REVOLLO[1, 2], CLAUDIO DELRIEUX[3], GERARDO PERILLO[1, 4], MARINA CIPOLLETTI[1,3]

[1] Instituto Argentino de Oceanografía, *CONICET*
Camino a la Carrindaga km 7 Complejo CCT Edificio E1 B8000FWB,
Bahía Blanca Argentina iado@criba.edu.ar
[2] Facultad de Ingeniería, Universidad Nacional de Jujuy, *UNJU*
Gorriti 237 San Salvador de Jujuy
Jujuy Argentina
[3] Instituto de Investigaciones en Ingeniería Eléctrica
Dpto. de Ing. Eléctrica y de Computadoras, UNS-CONICET
Avenida Alem 1253 B8000CPB
Bahía Blanca Argentina iiieuns@uns.edu.ar
[4] Departamento de Geología, UNS
San Juan 670, Bahía Blanca Argentina geologia@criba.edu.ar

Abstract. *Image and video processing of natural phenomena is one of the preferred non-invasive monitoring techniques for environmental studies that is, however, limited through the high cost of the required equipment and the limited access and precision of the processing algorithms. In this work we propose a low cost methodology for environmental studies using unexpensive off-the-shelf hardware and simple yet powerful processing algorithms. The images are taken using small format RGB cameras and processed in standard laptop equipments using open source libraries and processing algorithms specifically developed in general purpose programming languages. We applied this methodology to the coastal monitoring the shoreline of Monte Hermoso, Argentina, aimed at establishing accurate measurements of specific coastal features, for instance the coastal length. The experimental results show that our proposed unsupervised processing algorithm obtains results with a very high level of accuracy.*

Keywords: *Image and video processing. Environmental monitoring. Reprojection, segmentation, and feature extraction algorithms.*

1. Introduction

Image and video processing of natural phenomena, together with remote sensing, are among the preferred non-invasive monitoring techniques for obtaining qualitative and quantitative information that may be used for environmental, economic, and social decision making, and establishing policies. Remote sensing imagery (*i.e.*, satellite or airborn), however, is not versatile, often not precise enough, and may have very high operational costs.

Monitoring using computer vision, on the other hand, may overcome these limitations when two conditions are met: if they are based on affordable, easily replaceable off-the-shelf equipment, and if the processing algorithms are reliable.
The long term goal of our study is to establish an accurate model of the sea-wave-land interaction in the shoreline of Monte Hermoso, Argentina. There are other worldwide projects that study the shoreline dynamics, for instance ARGUS (Holman and Stanley, 2007), INDIA (Morris et al., 2001), HORS (Takewaka et al., 2002), CAM-ERA project, KOSTA project and HORUS project (Andrés Osorio et al., 2007). These research projects present results of shoreline dynamics, but unfortunately most of the image processing algorithms that they implement are not well documented.
This work presents itself as the first results of beach shoreline dynamics research employing low cost small format cameras and image processing in Argentina. Determination of several coastal features and their dynamics (i.e., the perimeter value of the polygonal that represents the shoreline) allows the study of several geomorphologic processes, and the elaboration of a geophysical predictive model that may serve in several decision making instances.
Images are acquired from a static vintage point in a building. The processing pipeline requires very little tuning to be fully unsupervised. First, images are rectified to a zenithal plane, and linearly georeferenced using four GPS reference points. After this step we can guarantee that the raster scale is even enough for a further vectorization and quantification process.
Then the image is binarized with respect to proximity to a prototype in color space. The purpose of this binarization is to classify every pixel in the raster as either sea or land. The binarized image is then processed with a set of morphology filters to even out noise or spurious misclassified pixels.
Finally, we applied the usual linear length extraction methods to the border between the classified areas in the raster. We argue that these methods, though popular, incur in a significant, systematic error in excess, which is also resolution independent. For this reason, we applied also a super-resolution variant of the Marching Squares algorithm to measure the linear length of the coastal shoreline. These length estimations were tested against the length of a polyline drawn by hand that visually fits the shoreline tightly (and therefore considered here as the gold standard). Results show that our unsupervised method is able to provide a very precise and accurate estimation of the coastal length.

2. Methodology

2.1. Image Rectification Stage

Data Acquisition. The digital images used in the rectification process were taken with unexpensive small format digital RGB camera from the beach of Monte Hermoso in Buenos Aires Province, Argentina. The camera was located in a fixed position at a height of 30 m on top of a building, having a

panoramic view of the beach. The obtained images have a 1293x1142 resolution.

Camera Model. In oblique images, the scale of the raster varies along the position, making difficult the measurement of geometric variables under study. For this reason, a camera model is used to relate points in the image to their corresponding geographic coordinates. A two-dimensional projective transformation was used to transform the oblique projection plane to a zenithal plane (Lerma, 2002). This situation is characteristic in the rectification of photographic images (see Fig. 1).

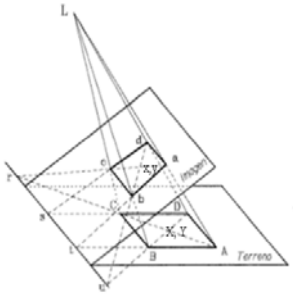


Figure 1: *Two-dimensional projective transformation of two nonparallel planes. The X and Y coordinates represent the corresponding geographic coordinates of a point (x , y) that is projected in the oblique plane of a digital image.*

The expressions of the two-dimensional projective transformation (1) express the projection between two nonparallel planes from eight parameters.

$$[A]=\begin{vmatrix} a_1 b_1 c_1 \\ a_2 b_2 c_3 \\ a_3 b_3 c_3 \end{vmatrix} \neq 0 \qquad x=\frac{a_1 X + b_1 Y + c_1}{a_3 X + b_3 Y + 1} \qquad y=\frac{a_2 X + b_2 Y + c_2}{a_3 X + b_3 Y + 1} \qquad \begin{bmatrix} x \\ y \end{bmatrix} = [A] \begin{bmatrix} X \\ Y \end{bmatrix} \tag{1}$$

Image Rectification. An oblique image rectification using two-dimensional projective transformation needs four homologous points. In this task four control points in the oblique image and four real land points were needed. These points allow calculating the projection matrix. Four control points (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) were determined in the oblique image. These points limit the geographic area of interest (shown in Fig. 2). The geographic coordinates of the control points in the image were obtained

using a Global Positioning System. For each control point the coordinates (x, y) in pixel on the image was determined using a GIS software, whose values are shown in Table 1.

Figure 2: *Control Points ((x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)) whose geographic coordinates were obtained with a Global Positioning System and are related with the pixels coordinates in the image.*

Table 1: *Pixel Coordinates of the control points in the image.*

Points	*x*	*y*
Point1 (x_1, y_1)	367	448
Point2 (x_2, y_2)	507	407
Point3 (x_3, y_3)	1043	610
Point4 (x_4, y_4)	637	866

The GPS-established coordinates are shown in Table 2, and were positioned on an information layer in Google Earth (Fig. 3). The geographic distance between these points was obtained, and the pixel coordinates in the image of the projected real points were determined are shown in Table 2. The pixel values of the four control points on the oblique image and the four geographic points allow to calculate the projection matrix and then the projective transformation for the rectification.

Figure 3: *GPS-established control points taken in the image. These geographic coordinates determine the parameter of the projection matrix.*

Table 2: *Geographic coordinates of the control points in the image.*

Points	*Latitude*	*Longitude*
Point1 (X_1, Y_1)	38º 59’ 21,27’’	61º 16’ 36,27’’
Point2 (X_2, Y_2)	38º 59’ 28,65’’	61º 16’ 37,39’’
Point3 (X_3, Y_3)	38º 59’ 27,40’’	61º 17’ 19,82’’
Point4 (X_4, Y_4)	38º 59’ 21,27’’	61º 17’ 18,88’’

Image rectification was implemented in a standard general-purpose programming language, using the Free Vision Library OpenCV, that implements a variety of tools and functions used in the image processing and computer vision in real time. The rectification procedure takes the oblique image and the matrix parameters, and produces as output the rectified image that will be used for later processing (see Fig. 4).

2.2. Image Segmentation Stage

The significant part of the rectified image including the shoreline area was clipped into a new image (see Fig. 5). In this image, further processing will be applied to extract the shoreline, and to compute an adequate estimation of its length.

Image Classification. The purpose of this processing is to determine the likelihood of a pixel being either sea or land. A prototype of sand color was determined (Table 3), and the distance in color space to this prototype was established for every pixel in the image. The resulting pseudocolored image can be seen in Fig. 6. After this classification step, the likelihood of a pixel as being sea is proportional to the distance to the sand color prototype (and

therefore to the grey level in Fig. 6). This secondary, “distance” image can be binarized against a threshold, producing the desired classification.

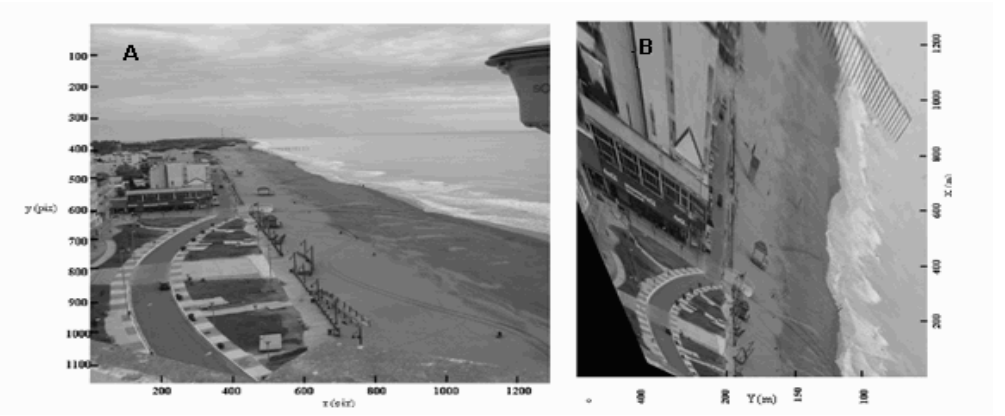


Figure 4: *a) Oblique image of Monte Hermoso where known geographic coordinates, and b) previous image after rectification (shown offscale).*

Table 3: *Prototype pixel Coordinates and RGB values.*

Prototype Pixel(Position 93,839)	*value*
Red Component	99
Green Component	93
Blue Component	85

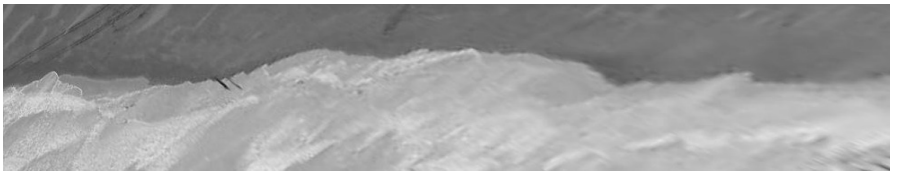


Figure 5: *A clip of the rectified image containing the area of interest.*

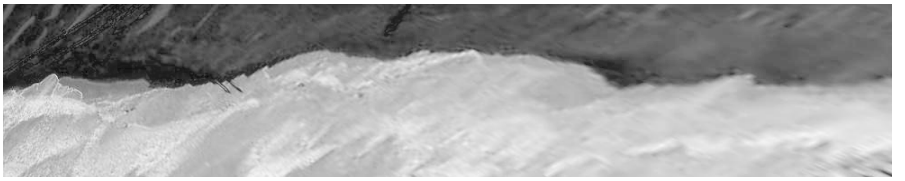


Figure 6: *Pseudocolor image where the gray level is related to the distance in color space of the pixel color to a prototype.*

Usual threshold setting techniques are based on a histogram analysis of the relative frequency distribution of distances. The criterion of minimizing together the amount of false negatives and positives gives rise to the minimum distance classifier, and the criterion of minimizing the conditional risk of misclassification gives rise to the Bayesian classifier.
In our images, however, neither of these criteria meets fully the desired result of a cleanly segmented shoreline. For this reason, we set an empirically tuned threshold value that produced the best results in the following processing steps in the pipeline (see Fig. 7), *i.e.,* removing the undesired misclassified areas that are visible in the uppermost right part of the image (false positives), and the ones that are also seen in the lowermost left part (false negatives).

Figure 7: *Images binarized with different threshold values. Empirically, the middle one performs best in the further processing steps.*

Noise and misclassified areas removal. Noise and misclassified areas in the binarized image obtained in the previous step (Fig. 7(b)) was removed by means of morphological filtering (dilation and erosion). The resulting image (Fig. 8) is clean enough for the border extraction and measurement step, and therefore is taken as the correctly classified image.

Figure 8: *Correctly classified image after noise and misclassified pixel removal.*

Border Extraction. The extraction of the polygonal that represents the shoreline was computed using several methods. The two most popular ones (implemented in almost all GIS software) are the outermost edge detection method, and the chain code method. The outermost edge detection method regards every edge between pixels classified in the interior area and the exterior area of the classified image as part of the border between these two areas. The chain code method surrounds the classified area with a polyline, every line of which may be oriented in any of the eight principal angles.

It is easy to see that, though often used, these methods incur in a rather high systematic error in excess, which even worse is resolution independent. For this reason, we devised a super-resolution method based on the marching squares algorithm, but that takes into account the classification distance of the pixels prior to binarization, therefore being able to do a much finer segmentation of the border (see Fig. 9).

Supervised Measurement using a (GIS) Software. Also, a supervised measurement of the shoreline was made with a GIS software tool. The software employed was GvSig, this is a free software developed by the Generalitat Valenciana. The rectified image was loaded in the system and then a new vectorial layer of the shoreline was digitalized by hand. The length of the resulting polyline is then taken as the gold standard against which the accuracy and precision of our unsupervised methods are tested.

3. Results and Discussion

The lengths of the shoreline as measured with all the methods implemented in this paper are shown in Table 4. As predicted, both the outermost borders, and the chain code methods commit an unacceptable error in excess. The super-resolution marching squares algorithm, however, is only about 2.5% above the supervised measurement. This may be attributed to the fact that the tendency in user assisted vectorization is always to obliterate tiny details, and therefore it is most likely that the actual length of the shoreline is slightly above the reported by the supervised method.

Table 4: *Length obtained using different algorithms of measurement*

Algorithm	*Length*
Manual segmentation	1193
Outermost borders	1558
Chain Code	1322
Super-resolution Marching Squares	1225

The behaviour of the four methods can be appreciated in detail in Fig. 9, where we superimposed the four different versions of the border over a close-up of a clip of the rectified image containing part of the shoreline.

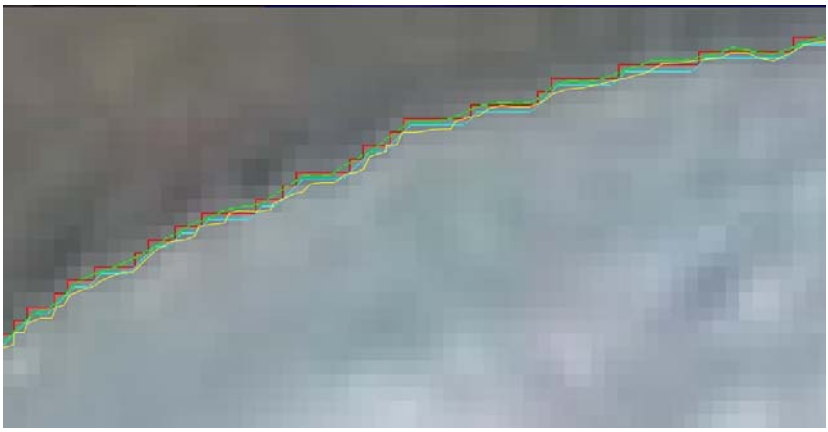


Figure 9: *The borders obtained with the four methods superimposed over a close-up of the original image: outermost borders (red), chain code (green), super-resolution marching squares (cyan), and supervised (yellow).*

4. Conclusion and further work

Image and video processing of natural phenomena using unexpensive equipment posses a significant challenge in environmental studies. In this work we implemented a coastal monitoring and feature estimation system using unexpensive, small format RGB cameras. The methodology was applied to measure shoreline lengths in the area of Monte Hermoso, Argentina.

Our processing framework acquires the images from a static vintage point in a building. The images are then rectified using a reprojection transformation to a zenithal plane, and then they are georeferenced using the projection of four locations whose GPS coordinates are known. Then the pixels in the rectified image are classified in the RGB color space with respect to proximity to a prototype. Finally, the length of the border is estimated using three different algorithms.

The experimental results were tested against a supervised estimation, and show that our proposed unsupervised processing algorithm obtains highly accurate results, arguably as good as the supervised ones.

The processing pipeline requires only a small set of externally fixed parameters (the position and projection of four geographic coordinates, the color prototype against which the classification is performed, and the empirically selected threshold value for binarization). These parameters are

robust enough along whole sequences of images, and therefore the entire process is amenable for processing video sequences lasting several minutes.
Several research lines are opened by these results. Among them the most important one is to develop a portable, low-cost, easy to operate monitoring station that is able to acquire and process video sequences along several hours. For this purpose, some strategies for having adaptive tuning of the processing parameters will be required, specifically for making the classification independent to illumination and weather conditions. Also important is a drastic optimization of the execution time of the algorithms, in order to have an adequate frame per second processing rate. Finally, the implementation of these algorithms in an embedded system that might be able to connect to a repository server through GSM connectivity, will enable outstanding real-time environmental monitoring capabilities.

References

1. González, C. R. and Woods, R. E., "Tratamiento Digital de Imágenes", Addison-Wesley/Díaz de Santos, 1988.
2. Browne, M., Blumenstein, M., Tomlinson, M., "An Intelligent System for Remote Monitoring and Prediction of Beach Safety Proceeding Artificial Intelligence and Applications", R. Lane 453, 2005.
3. Chu, D., Popa, L., Tavakoli, A., Hellerstein, J., Levis, P., Shenker, S. and Stoica, I., "The Design and Implementation of a Declarative Sensor Network System", Proceedings of the Fifth ACM Conference on Embedded Networked Sensor Systems (SenSys), 2007.
4. Lerma García, J. L., "Fotogrametría Moderna: Analítica y Digital", Editorial Universidad Politécnica de Valencia, 2002.
5. Uunk, L., "Automated collection of intertidal beach bathymetries from Argus video images", MSc Thesis, 2006.
6. Osorio, A. F., Pérez, J. C., Ortiz, C. and Medina, R., "Técnicas basadas en imagines de video para cuantificar variables ambientales en zonas costeras", Avances en recursos hidráulicos, No 16, 2007.
7. Intel Corporation "Open Source Computer Vision Library" (OpenCV), http://www.intel.com/technology/computing/opencv/.
8. García León, J., Cuartero Sáez, A., "Comparación de los procesos de rectificación y ortoproyección mediante fotogrametría terrestre Digital", Universidad de Extremadura, España.
9. Holman, R., Stanley, J., and Ozkan-Haller, T., "Applying Video Sensor Networks to Nearshore Environment Monitoring", Pervasive Computing 2, 2003.

VI

Software Engineering Workshop

Assessing e-Governance Maturity through Municipal Websites - Measurement Framework and Survey Results

ROCÍO RODRÍGUEZ[1], ELSA ESTEVEZ[2, 3], DANIEL GIULIANELLI[1], PABLO VERA[1]

[1] Universidad Nacional de La Matanza. Departamento de Ingeniería e Investigaciones Tecnológicas. Instituto de Investigación y Desarrollo.
Florencio Varela 1903, San Justo, Provincia de Buenos Aires, Argentina
{rrodri, dgiulian, pablovera}@unlam.edu.ar

[2] Universidad Nacional del Sur, Dpto. de Ciencias e Ing. de la Computación
Laboratorio de I&D en Ingeniería de Software y Sistemas de Información (LISSI)
Av. Alem 1253, Bahía Blanca, Argentina.

[3] United Nations University, International Institute for Software Technology
P.O. Box 3058, Macao SAR, China
elsa@iist.unu.edu

Abstract. *The paper presents a measurement framework for assessing the e-Governance maturity level of countries through the analysis of municipal websites. The paper also introduces the results of a survey carried out to apply and validate the framework. Applied to municipal websites of different countries, the framework considers websites content and design. For each country, the sample included three websites of local governments belonging to regions with low, medium and high population, respectively. The country measure was calculated based on the average obtained by the municipal websites adjusted by a correction factor based on the compliance of general features. The numerical values obtained by countries allow comparing their degree of e-Governance maturity and ranking them accordingly. The contribution of this paper is to present a novel approach for assessing e-Governance maturity of countries based on analyzing how electronic public services are delivered through municipal websites to citizens living in different populated areas.*

Keywords: *e-Governance, Accessibility, Friendship, Navigability, Usability.*

1. Introduction

One way to define Electronic Governance (e-Governance) is through its objectives. e-Governance objectives according to the Argentinean government agency ONTI [8] (National Office of Information Technology) are the following: "to provide better services to citizens, to improve efficiency and effectiveness in public administration, to reduce costs, and to increase transparency and participation for a more integrated and developed

society". In practical terms, it means providing accessible and useful electronic public services, and moreover, empowering citizens through participation.

The difference between e-Government and e-Governance is that the former concentrates on the electronic delivery of public services, while the latter also considers active citizen participation in government decision-making processes. In order to promote citizen participation, governments need to facilitate access to information and enable knowledge acquisition by citizens. In turn, these initiatives contribute to increase transparency and at the end to deliver better governance.

In order to promote citizen participation, governments deliver various types of services through their websites, like e-mails to contact government officials, surveys assessing citizen opinion about service delivery, forums for citizens to raise opinions on different issues, like policies, environment, etc. However, delivering such services through government websites is not enough. In addition, services and information should be accessible easily, intuitively and fast.

Based on the above premises, the measurement framework presented in this paper was defined. The framework includes metrics for assessing websites design and content considering the following features:

1) *Information* – websites should include informative text enabling users to acquire knowledge about the institution or the services provided by it.
2) *Functionality* – services offered through the website, such as tax payment, state of debts, consulting administrative procedures, etc.
3) *Truthfulness* – quality of information published on the website. Government websites should provide real, relevant and up to date information.
4) *Citizen Participation* – offered services which increase the degree of interaction between government and citizens. Assessing two-way interaction services motivates government to advance from the informational stage, where government simply publishes information online (one-way interaction) and citizens passively consumes such information.
5) *Friendship* – assesses the user-friendliness of websites. Government websites should be friendly to anybody who visits them, regardless the user literacy or expertise. The language used by government websites should be simple [4].
6) *Usability* – measures user efforts for interacting, learning how to navigate, or accessing content and services offered through the website [7].
7) *Accessibility* – measures the degree in which a website can be accessed by people, despite the limitations of individuals or usage context.
8) *Navigability* – assesses user efforts for browsing the website pages.

It can be noticed that all metrics associated with web design, can be used to evaluate any website, not necessarily government websites. However, those metrics measuring which and how services are provided are key for assessing

the maturity of e-Governance. In addition, since most public services are delivered by local governments, the research team decided to assess municipal websites.
The rest of this paper is structured as follows. Section 2 explains the metrics used by the framework. Section 3 introduces how the framework results can be used for ranking countries. Section 4 outlines the methodology used for conducting the survey, while Section 5 presents the results. Finally, Section 6 draws some conclusions.

2. Origin of Metrics

The framework includes three types of metrics: i) those published by international organizations or national governments, ii) those defined by researchers and practitioners, extracted from the literature, and iii) those proposed by the authors who participated in the research team. The origin of metrics is explained below.

1) *Standards* – some metrics were extracted from the standards adopted by the World Wide Web Consortium (W3C) [16], while others were derived from recommendations published by ONTI [8] in Argentina. The latter includes metrics related to website content and design [16], [19].
2) *Academic and Government Publications* – after reviewing the existing literature, metrics for assessing municipal websites were extracted from publications from: Spain [5], United States [11], New Zealand [6], Chile [2] and Australia [1].
3) *Proposed by the Research Team* – new metrics related to web design and web development, particularly targeted to measure features of municipal websites were proposed by the research team.

Figure 1 shows the composition of metrics according to their origins.

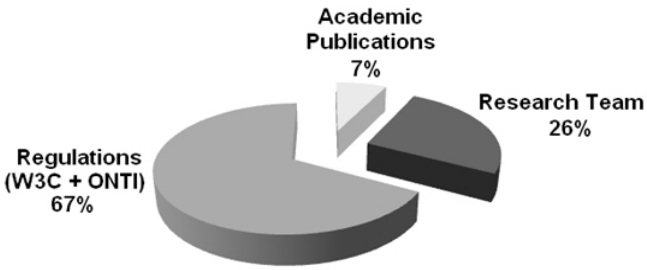


Figure 1: *Origin of Framework Metrics*

3. Measurement Framework

The framework includes 152 metrics grouped according to the eight features presented in the Introduction: 1) Information, 2) Functionality, 3) Truthfulness, 4) Participation, 5) Friendship, 6) Usability, 7) Accessibility and 8) Navigability. Each metric can be considered by more than one feature. For example, the metric "The main menu is maintained in the rest of the pages" is considered by Friendship, Usability and Navigability, since it affects all of them. However, the same metric may influence each feature in a different degree. Therefore, a weight value was defined for measuring how the metric affects the feature. The possible values are: high (5 points), medium (3 points) and low (1 point). Intermediate values are also used in order to achieve greater accuracy. It is possible that a metric does not influence a feature at all. In such case, no weight value is assigned.

The complete list of metrics defined by the framework along with the weight values assigned for each feature is available in [19].

The procedure for applying the framework is explained as follows. First, an initial value is calculated for each website by adding the weights of all features of the satisfied metrics. For example, suppose a website only satisfies the four metrics shown in Table 1 – i) the website does not contain private advertisement, ii) the website does not use frames, iii) all features are available without leaving the site, and iv) the website provides information about possible transports that can be used to reach the municipality. The columns of Table 1 correspond to the eight features considered by the framework: Friendliness (FR), Navigability (NA), Usability (US), Accessibility (AC), Information (IN), Truthfulness (TR), Functionality (FU), and Participation (PA). The included values are the weight defined by the framework for the feature/metric. Therefore, adding all the weights of individual metrics, results in 24 (5 + 5 + 10 + 4 = 24). The initial value for this website is defined as 24.

Table 1. Example for calculating the score for a municipal website

	FR	NA	US	AC	IN	TR	FU	PA	Total
It has no private advertisements						5			**5**
Do not use frames				5					**5**
All features are available without leaving the site	3	4		3					**10**
Transport information to reach the municipality					4				**4**
Total by Feature	3	4		8	4	5			**24**
TOTAL SCORE									**24**

The Framework also enables to calculate the total amount of points by columns (see Table 1) showing the total score obtained by the website for each feature.

To facilitate the calculation of scores by country, a software tool was developed that allows recording the compliance of metrics for each site. The tool automatically adds the weights for each feature/metric producing the final score for the website.
Figure 2 shows the procedure for calculating the final value for a country.

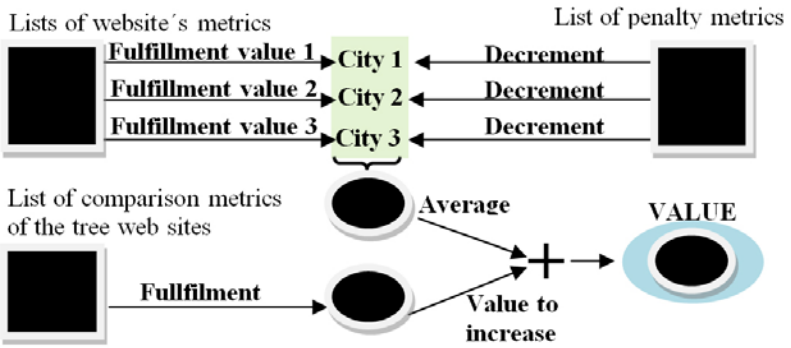


Figure 2: *Procedure for determining the country value*

First, three municipal websites are assessed and their score is calculated, as explained above, obtaining the *Fulfillment values 1*, *2* and *3*. All fulfilled metrics contribute with a positive value, while unfulfilled metrics give no value. However, the framework defines that unfulfilling a set of metrics (*List of penalty metrics*) at the same time causes a penalty, since in such case the quality of the site dramatically decreases. The penalty is expressed by a *Decrement value*. After calculating the score for each of the three websites and considering the decrement of the penalties, an average is calculated. At the end, a final value is calculated adding or subtracting points to the average, based on comparing the results of metrics of the three websites.

Following, we illustrate and explain some of the penalty metrics.

1) *Penalty Aim* - difficulty to reach site content
   Penalty Metrics List: a) website has a sitemap; b) website offers search services
   Justification - In case both functionalities are missing, the only way to find given information is by opening all pages linked from menus and links. This may discourage users for using the site.
   Penalty Value: 5 points.

2) *Penalty Aim* - links without the proper signaling
   Penalty Metrics List: a) links are underlined; b) links are highlighted when passing the mouse over them; c) links are highlighted with a hand icon when passing the mouse over them.

Justification - If links are not highlighted, users can not distinguish links from regular text.
Penalty Value: 5 points.

3) *Penalty Aim* - difficulty to find the website URL
Penalty Metrics List: a) website has a URL related to the organization, and b) website is referred by main search engines, like Google, Yahoo and Alta Vista.
Justification - users may have difficulties in remembering URLs which are not related to organizations and if they are not referred by main search engines.
Penalty Value: 5 points.

4) *Penalty Aim* - difficulty to identify the organization through the page banner
Penalty Metrics List: a) banner of the main page includes the organization name, and b) banner of the main page includes the organization logo
Justification - banner is the first thing a user sees when opening a page. A banner not related to the institution can mislead users.
Penalty Value: 3 points.

5) *Penalty Aim* - difficulty to return to the website home page
Penalty Metrics List: a) website provides a visible link to the home page; b) browser back button is enabled
Justification - not having a visible option to return back can confuse users, who may have difficulties for navigating through the site.
Penalty Value: 5 points.

6) *Penalty Aim* - inconsistent design of website pages

Penalty Metrics List: a) all website pages place menus in the same position and menu options are consistent; and b) font types are used consistently through website pages
Justification - If pages of the same website follow different designs, user may be disappointed while navigating the site or may experience difficulties to learn how to navigate through it.
Penalty Value: 3 points.

7) *Penalty Aim* - the website does not facilitate communications with users
Penalty Metrics List: a) website provides the municipality address; b) website provides phone numbers; c) website provides e-mail address or contact form.
Justification - providing contact details facilitates communication with users
Penalty Value: 4 points.

8) *Penalty Aim* - facilitating two-way interactions with citizens

Penalty Metrics List: a) website offers a chat; b) website offers a forum; c) surveys are conducted through the website; and d) website manages complaints.
Justification - enabling chats in government websites enable to initiate dialogue with citizens; while forums and surveys enable citizens to express their opinions.
Penalty Value: 4 points.

Finally, the value obtained by calculating the score of the three municipal websites can be increased in case the three websites fulfills a set of pre-defined metric. Such metrics are called comparative metrics, and some of them are explained below.

9) *Comparative Aim* - municipal websites domains follow predictable naming
Justification - choosing domain names following a standard makes it easier for users to remember them.
Comparative Added Value: 3 points.

10) *Comparative Aim* - consistent country-wide municipal websites
Justification - if municipal websites follow national standard conventions for web design, users can easily apply the knowledge learnt while navigating one website to other government websites. In addition, a national, consistent look and feel is promoted.
Comparative Added Value: 5 points.

The following section explains the survey carried out for applying the framework.

4. Survey

4.1. Methodology

From the 152 metrics of the framework, there are a set of metrics that are measured manually by simple website inspection. For example: links highlighted when passing the mouse over them, website has music, etc. In addition to manual assessment, there are various software tools that enable measuring some other metrics, such as W3C validators [14][15][16], Xenu software - offering a report of broken links, weight and image resolution, etc. These tools avoid manual inspection of websites for measuring for example, if the weight or resolution of website images exceeds the metric bound. Based on our experience, the results of using the mentioned tools have shown 100% reliability. Finally, other metrics were inspected analyzing the source code of the web pages, i.e. usage of tables for schematization, use of relative units, use of frames, etc.
Prior to analyzing the websites, detailed guidelines were specified for carrying out website inspection. In particular, a procedure was defined explaining how

to inspect each of the 152 metrics, with emphasis on those metrics who might have different interpretation, so that the measurement process is independent of the evaluator point of view [1].

4.2. Selected Countries

After defining the measurement framework and the guidelines for its application, a list of websites was selected. The methodology for selecting websites follows.

- a country is randomly chosen
- information of the capital city of the selected country is seek determining the geographical region where is located
- the more recent official census of the country is analyzed to determine if the capital city is located in a high, low or medium density region.
- the capital city of a selected country is always part of the survey. To complete de survey two more geographical region are taken (eg. if capital city is located in a high population density region, lower and medium density regions of the country are chosen)
- the most important city of each of the two selected regions are selected.
- municipal websites of each of the three selected cities are inspected.

This methodology ensures equal selection criteria for all countries.
Table 2 shows selected cities for each country (capital city is remark in bold).

Table 2: *Countries and cities of the survey*

Country	Selected City
ARGENTINA	**Ciudad Autónoma de Buenos Aires**
	San Juan
	Ushuaia (Tierra del Fuego)
AUSTRALIA	New South Wales
	Western
	Camberra
BOLIVIA	**La Paz**
	Chuquisaca (Sucre)
	Veni (Santa Ana del Yacuma)
CHILE	**Santiago de Chile**
	Rancagua (Cachapoal)
	Coyhaique
COLOMBIA	**Bogotá**
	Agua Chica (Cesar)
	Cumaribo (Vichada)
COSTA RICA	**San Jose**
	Heredia
UNITED STATES	California
	Kentucky
	Columbia
FRANCIA	**París** (Ile-de-Francie)
	Picardie (Amiens)
	Corse (Ajaccio)
LUXEMBURG	**Tirana**
	Diekirich
	Vianden
MEXICO	**México DC**
	San Luis Potosi
	Colima
NIGERIA	Kano
	Ondo
	Abuja
PERU	**Lima**
	Callao

	Guanacaste (Cañas)		Moquegua
ECUADOR	**Quito**	PUERTO RICO	**San Juan**
	Santa Elena		Camuy
	Galapagos (Santa Cruz)		Vieques
SPAIN	**Madrid**	VENENEZUELA	**Caracas**
	Albacete		Alberto Adriani (Merida)
	Teruel		Atabaco (Amazonas)

5. Survey Results

Applying the measurement framework enables to obtain a numeric value for each country. Such value indicates the country e-Governance maturity level, assessed through municipal websites. Based on the defined framework, the maximum score a country can obtain is 1183. Figure 3 shows the final scores obtained by the surveyed countries.

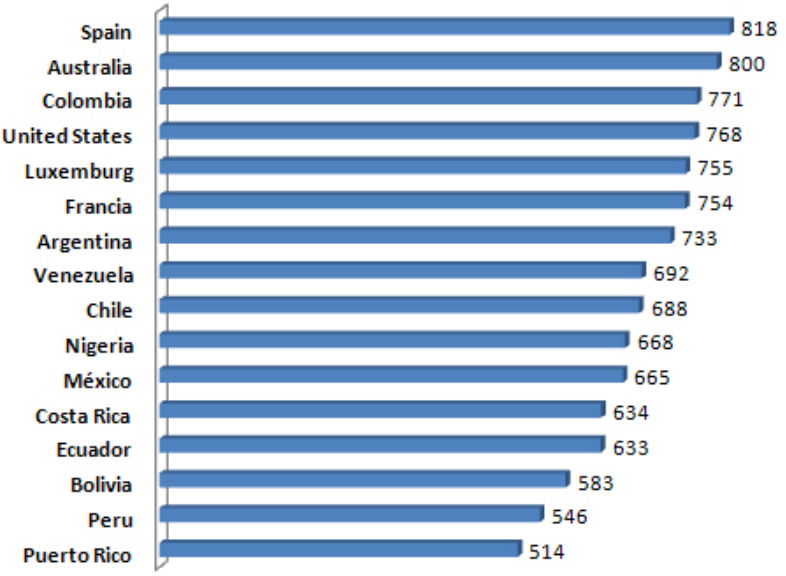


Figure 3: *e-Governance Ranking*

While the best positioned countries shown in Figure 3 have obtained a score that represents almost 70% of the maximum value, the three countries within the lowest positions does not reach 50% of the maximum score.

Additionally, it is possible to calculate the percentage of fulfillment for each country (country score / maximum score * 100), and also to consider the percentage of fulfillment for those metrics assessing content and those assessing web design. As mentioned, the maximum score of e-Governance is

1183 (969 points belongs to design metrics while 206 belongs to content metrics). Table 3 shows the percentages obtained by the surveyed countries. The table lists countries in descendent order according to the overall e-Governance percentage. The highest percentage achieved in each category is shown shaded.

An interesting feature shown by the table is that the fulfillment of design metrics is greater than the fulfillment of content metrics, excepting México which percentages are almost equal - 56.55 and 56.80.

Table 3. *Percentages reached by each country*

Country	e-Governance	Design	Content	Country	e-Governance	Design	Content
Spain	69.15	73.37	51.94	Chile	58.16	59.75	52.91
Australia	67.62	71.21	54.37	Nigeria	56.47	60.78	37.86
Colombia	65.17	70.90	41.26	Mexico	56.21	56.55	56.80
United States	64.92	66.56	60.19	Costa Rica	53.59	57.79	35.92
Luxemburg	63.82	69.25	41.75	Ecuador	53.51	56.76	40.29
Francia	63.74	65.94	56.31	Bolivia	49.28	52.53	35.92
Argentina	61.96	65.12	49.51	Perú	46.15	48.19	39.32
Venezuela	58.50	63.78	36.41	Puerto Rico	43.45	47.27	26.21

Finally, the information shown in Table 3 is graphically depicted in Figure 4. The line at the top shows the fulfillment of e-Governance, the one at the bottom the fulfillment of content; while the one in the middle reflects the fulfillment of web design metrics. It is clearly depicted that the fulfillment of design metrics is greater than the fulfillment of content metrics, almost in all cases.

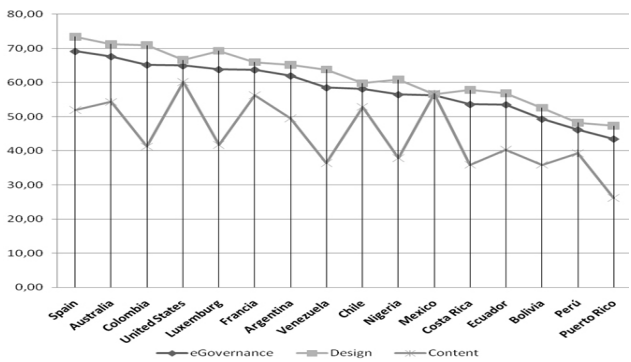


Figure 4: *Percentages of e-Governance, Design and Content Fulfillment by Country*

6. Conclusions and Future Work

The paper presented an extended version of a measurement framework for assessing country e-Governance maturity level based on analysis of municipal websites. The novel approach of the framework is considering a holistic approach for ranking countries based on how electronic public services are offered by local governments located in different populated areas. The values obtained by the municipal websites are adjusted with values representing more accurate the country-wide situation.

A survey comprising 16 countries was conducted to show the applicability of the framework. Survey results show that municipal websites better fulfill design metrics than content metrics. From the surveyed countries, only 6 reach at least 50% of the maximum score defined for content metrics. This highlights the weak implementation of contents provided in municipal websites.

Future research lines include extending the framework to define different levels of maturity, specifying guidelines for government websites. To achieve this aim, existing e-Governance models will be analyzed to determine their strength and weaknesses.

References

1. Department of the Premier and Cabinet, Guidelines for State Government Websites, Australia, 2007. Available at http://www.egov.dpc.wa.gov.au/documents/WebGuidelinesVersion2.1_final.doc.
2. e-Government Software AG Class, Alianza Sumaq. Estudio sitio Web Muinicipales: eGovernment en Chile, Chile, 2006. Available at http://www.cetiuc.cl/wp-content/uploads/2007/01/presentacion-estudio-municipalidades.pdf.
3. Krug, S., Don't Make Me Think: A Common Sense Approach to Web Usability, New Riders, California, USA, 2000.
4. Marcos Mora, M., Rovira Fontanals, C., Evaluación de la usabilidad en sistemas de información web municipales: metodología de análisis y desarrollo, Spain, 2005. Available at http://www.semanticaweb.net/archives/2005_evaluacion-municipales-isko.pdf.
5. New Zealand, NZ Government Web Standards and Recommendations, Nueva Zelanda, 2007. Available at http://www.e.govt.nz/standards/web-guidelines/web-standards-v1.0
6. Nielsen, J., Loranger, H., Prioritizing Web Usability, New Riders, California, USA, 2006,
7. ONTI, Contents, Available at http://www.sgp.gov.ar/contenidos/onti/onti.html.
8. ONTI, Estándares para Sitios y Portales de Internet para la Administración Pública Nacional (ETAP), Argentina, 2005. Available at http://www.sgp.gov.ar/contenidos/onti/productos/pnge/docs/etap_sitios_web_oficiales.pdf.

9. ONTI, Plan Nacional de Gobierno Electrónico: Decreto 378/2005. Available at http://www.sgp.gov.ar/contenidos/onti/productos/pnge/docs/pnge_decreto_378_2005.pdf.
10. USA, Requirements and Best Practices Checklist for Government Web Managers, 2005, Available at http://www.usa.gov/webcontent/reqs_bestpractices/checklist/long.pdf.
11. Rodríguez, R., Countries list, 2007, Argentina. Available at http://www.investigamos.com.ar/ge/listado_paises.pdf.
12. Rodríguez, R., Vera, P., Trigueros, A., Metrics List, 2008, Argentina. Available at http://www.investigamos.com.ar/ge/listado_metricas.pdf.
13. UNESCO, Overview on E-governance, 2002. Available at http://portal.unesco.org/ci/en/files/6532/10391876090Overview_on_e-governance_working_paper.doc/Overview%2Bon%2Be-governance%2Bworking%2Bpaper.doc.
14. W3C, CSS. Available at http://jigsaw.w3.org/css-validator/.
15. W3C, Guiando la web hacia su máximo potencial. Available at http://www.w3c.es/.
16. W3C, HTML 4.01 Transitional. Available at http://validator.w3.org.
17. W3C, Links Checker. Available at http://validator.w3.org/checklink.

Facing Communication Challenges in Global Software Development

GABRIELA N. ARANDA[1], AURORA VIZCAÍNO[2], MARIO PIATTINI[2]

[1] GIISCo Research Group, Universidad Nacional del Comahue
Faculty of Informatics, Buenos Aires 1400 - 8300 Neuquén, Argentina
{garanda, acechich}@uncoma.edu.ar
[2] ALARCOS Research Group, Information Systems and Technologies, Department
UCLM-INDRA Research and Development Institute, Escuela de Informática, Universidad de Castilla-La Mancha, Paseo de la Universidad 4 - 13071 Ciudad Real, Spain.
aurora.vizcaino@uclm.es, mario.piattini@uclm.es

Abstract. *The main challenges during global software development projects are related to the lack of face-to-face communication. Since stakeholders' satisfaction is crucial as a factor that can influence a team performance, we have focused our research on the need of people feeling comfortable with the technology they use. In this article we introduce an approach that proposes a way of choosing the most suitable technology for a given group of people, taking advantage of information about stakeholders' cognitive characteristics, and we present preliminary results of an experiment we have carried out to validate our proposal.*

1. Introduction

Communication is a common problem in Global Software Development, as well as the time difference between different sites and the cultural diversity of stakeholders [4,7]. In such scenario, groupware tools become the main channel for communication, then analyzing their impact on stakeholders' perception and performance is an interesting focus for research.

One of the most common ways of classifying groupware is according to their synchronous or asynchronous characteristics (depending on if the users have to work at the same time or not) [8]. According to GSD literature, both categories are important, because asynchronous collaboration allows team members to construct ideas individually and contribute to the collective activity of the group for later discussion (especially when groups are distributed across time zones), but also real time collaboration and discussions are necessary components of group dynamic to give stakeholders the chance of having instant feedback [11]. However, is also true that sometimes people are keener on one kind of collaboration than the other. So, as communication among people involves aspects of human processing mechanisms that are analyzed by the cognitive sciences, we decided to look for references into the Cognitive Informatics, an interdisciplinary research area that applies concepts from psychology

and other cognitive sciences to improve processes in engineering disciplines like software engineering [16]. After analyzing varied psychological issues, we set our interest in using some techniques called Learning Style Models (LSMs), which may be useful to select groupware tools and elicitation techniques according to the cognitive style of stakeholders [14]. Most of related works using LSMs in informatics concern only educational purposes [3], however there are a few related works that use psychological techniques to solve communicational problems in Software Engineering. A work in that direction is [15] where cognitive styles are used as a mechanism for software inspection team construction. By means of a controlled experiment, this work proves that heterogeneous software inspection teams have better performance than homogeneous ones, where the heterogeneity concept is analyzed according to the cognitive style of participants. In our approach we choose a different perspective, and even when we also used the concept of cognitive styles to classify people, our approach is different because, as we have explained previously, we do not try to say which people seem to be more suitable to work together. Instead, our goal is choosing the best strategies to improve communication for an already given group of people. Having this in mind, we will give an introduction to some basic concepts about cognitive informatics and learning styles models, and we will introduce a methodology, based on concepts from fuzzy logic, to select groupware tools and requirement elicitation techniques. The last sections will compare results from two different surveys we have carried out in order to get examples to validate our methodology and we will present some conclusions and guidelines for future work.

2. Cognitive aspects of communication

Cognitive Informatics relates cognitive sciences and informatics by using cognitive theories to investigate and look for solutions to software engineering problems [6]. Doing so we can use concepts from cognitive psychology (concerning the way people attend and gain information), to improve the requirement elicitation process. Cognitive styles are a part of cognitive psychology theories that classify people's preferences about perception, judgment and processing of information [15], and try to explain differences in human behaviour. Similarly, learning styles models (LSMs) classify people according to a set of behavioural characteristics that concern the ways people receive and process information, while their goal is improving the way people learn a given task. Considering that elicitation is about learning the needs of the users [12], and also an scenario where users and clients also learn from analysts and developers (for instance, they learn how to use a software prototype or new vocabulary), we can say that during the elicitation process everybody "learns" from others. Then, even when LSMs have been discussed in the context of analyzing relationships between instructors and students, we propose taking advantage of LSMs by adapting

it to virtual teams that deal with distributed elicitation processes. The model we have chosen, after studying different LSMs, is called the Felder-Silverman (F-S) Model. According to our analysis, the F-S model is the most complete because it covers the categories defined by the most famous LSMs (like the Myers-Briggs Indicator Type, the Kolb model, the Herrmann Brain Dominance Instrument, etc.) and, additionally, the F-S model has been widely and successfully used with educational purposes in engineering fields [10]. The F-S Model introduces four categories (Perception, Input, Processing and Understanding), each of them further decomposed into two subcategories (Sensing/Intuitive; Visual/Verbal; Active/Reflective; Sequential/Global). 0 shows a summary of the characteristics for each subcategory [9].

Table 1: *Felder and Silverman categories and subcategories*

Category	Opposite Subcategories	
Processing	*Active* people tend to retain information by doing something active with it (discussing, applying it or explaining it to others).	*Reflective* people prefer to think about the information quietly first.
Perception	*Sensing* people prefer learning facts and solving problems by well-established methods.	*Intuitive* people prefer discovering possibilities and relationships, and dislike repetition.
Input	*Visual* people remember best what they see (such as pictures, diagrams, flow charts, time lines, films, and demonstrations).	*Verbal* people get more out of words, and written and spoken explanations.
Understanding	*Sequential* people tend to gain understanding in linear steps, with each step following logically from the previous one.	*Global* people tend to work in large jumps, absorbing material almost randomly without seeing connections, and then suddenly "getting it".

Classification into the different categories is obtained by filling a multiple-choice test, available on the WWW, which returns a rank for each subcategory. Depending on the circumstances, people may fit into one category or the other, being for instance, sometimes active and sometimes reflective; so preference for one category is measured as strong, moderate, or mild. A sample result is shown in 0.

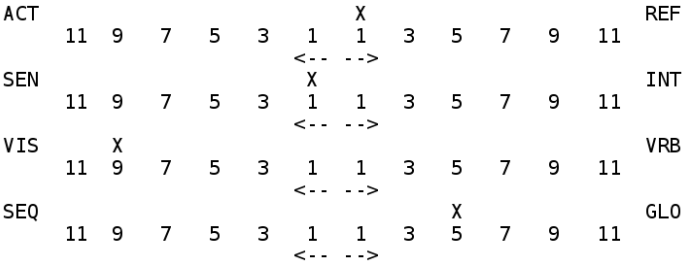


Figure 1: *Sample F-S test results for a stakeholder*

Numbers 9-11 mean a strong preference, 5-7 moderate, and 1-3 slight, therefore, the stakeholder in the example in 0 has a slight preference for the reflexive and sensitive subcategories, moderate for the global subcategory, while his preference for the visual subcategory is strong. According with their authors, people with a mild preference are balanced on the two dimensions of that scale. People with a moderated preference for one dimension are supposed to learn more easily in a teaching environment, which favours that dimension. Finally, people with a strong preference for one dimension of the scale may have difficulty learning in an environment, which does not support that preference. With the goal of making everybody feel comfortable in the virtual environment, we propose choosing groupware tools and elicitation techniques more according to their learning styles, as we explain in the next section.

To work easily with the F-S test results, we have decided to express it as a 4-tuple, in the same order they are returned in the results' web page. To clearly identify the opposite subcategories, we have chosen using a negative sign for the categories that appear on the left side (active, sensing, visual, and sequential) and with a positive sign for the others (reflective, intuitive, verbal, and global). Doing so, the sample stakeholder's learning style presented in 0, would be expressed like (1, -1, -9, 5). This convention will be used in the rest of the paper.

3. Supporting personal preferences in global software development

In order to support personal preferences when selecting technologies for virtual teams, we propose a methodology that uses concepts from fuzzy logic and fuzzy sets [1], to obtain rules from a set of representative examples, in the way of patterns of behaviour.

The methodology is divided into two stages: the first one (Stage 1) is independent of any project and comprehends phases 1 to 4, and the second one is dependent of a given project and covers phases 5 and 6, as it is shown in 0.

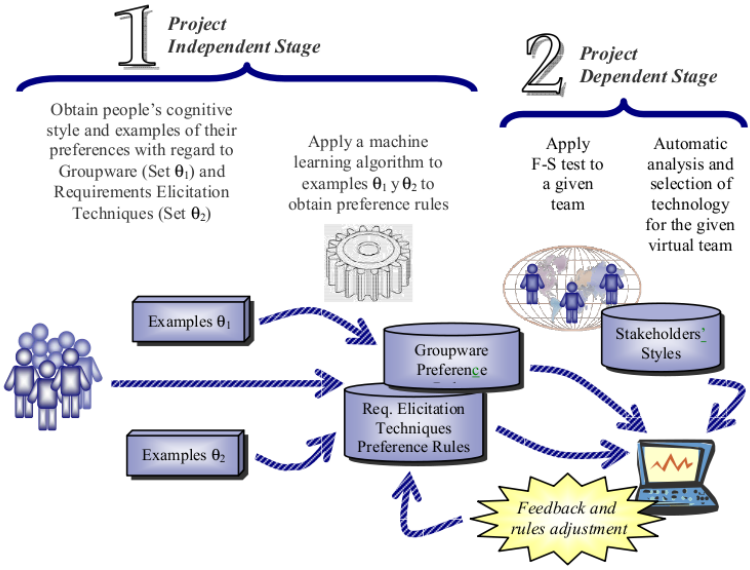


Figure 2: *Phases to define and analyze personal preferences to choose appropriate technology in Virtual Teams*

Phases 1 to 3 are about looking for a set of examples (which are real data about preferences of stakeholders in their daily use of groupware tools and requirements elicitation techniques), and analyzing them to discover their relationship with classifications in the F-S model. To do so, we have used the machine learning algorithm proposed in [5] to turn each example into an initial rule and iteratively we found a finite set of fuzzy rules that reproduce the input-output system's behaviour, which has been presented in [2]. For instance, one of them is:

If X_1 in {VAc,SRe,VRe} and X_2 in {SSe,MIn,VIn} and X_4 not in {SGl,MGl} then Email

which can be interpreted as: "If a user has a strong preference for the Active subcategory or a slight or strong preference for the Reflective subcategory, and a slight preference for the Sequential subcategory or moderated or strong preference for the Intuitive subcategory and his preference for the Global subcategory is not slight or moderate, he would prefer using Email (no matter which preference would be for the Visual-Verbal category)". As we mentioned before, phases 1 to 4 constitute the project independent part, then, our methodology has the characteristic that the example and preference rule databases can be improved along surveys and applied on more and more GSD projects. The remaining phases (Stage 2) consist of the application of our methodology to a specific GSD project during a requirement elicitation process, so that it is called the project dependent stage. In this stage, we obtain the personal preferences of every person who will work in a given virtual team, by asking him to fill the learning style test (Phase 5). This information is stored in a database that can be accessed every time a group of people needs to communicate to each other. Later, the technology selection process itself is done.

To do so, the personal preferences of a set of stakeholders that need to communicate to carry out a given task are studied and confronted, by means of an automatic tool, to choose and suggest the most appropriated set of technology (Phase 6). As we have explained in [2] such strategies must take into account other factors besides cognitive profiles of stakeholders, like time difference between sites, the degree of sharing of a common language, and the current situation at the requirement elicitation process.

4. Experiment design and execution

In order to validate certain aspects of our proposal we have carried out a controlled experiment with the participation of post-graduate computer science students from the University of Castilla-La Mancha (Spain) and the University of Comahue (Argentina). We chose to apply our experiment in the requirements elicitation process, since communication and knowledge sharing are crucial for stakeholders' (client, users, and analyst) common understanding. We divided 24 people into 8 teams, and attempted to simulate global development teams. The teams were therefore formed of three people. Two members played the role of analysts and the other played the role of client. The 'client' had to describe to the 'analysts' the requirements of a software product that the analysts would supposedly have to implement. The analysts then had to use the information obtained from the client's explanations to write a software requirements specification report. As the team members were geographically distributed they had to use a groupware tool to communicate. As our intention was to compare the teams that used our proposal and the teams that did not, we divided the teams into two groups. Half of them (denominated as Group 1) used the best groupware tool according to our preference rules, and the rest (Group 0) used a different (less suitable) groupware tool. The teams were randomly assigned to one of the two groups and our set of rules was applied to find the most suitable tool for each team. Later, the teams in Group 0 were assigned a different tool, as is shown in the fourth column of 0.

Table 2: *Assigned groupware tools*

Team	Suitable GW tool	Assigned GW tool	Suitability
G1	IM	Email	-
G2	Audio	IM	-
G3	Audio	Audio	+
G4	IM	IM	+
G5	IM	Email	-
G6	IM	IM	+
G7	Audio	IM	-
G8	Audio	Audio	+

Additionally, we ensured that the remaining variables were fixed for all the treatments. Therefore, requirements elicitation techniques were reduced to interviews and use case models for all the teams, and more experienced people were assigned first to avoid them being in the same team. As there were 3 people in each team, we chose to have two analysts and one user per team, as we considered that such a distribution would give us the opportunity to analyze not only the user-analyst relationship, but also the analyst-analyst relationship. We avoided educational differences by assigning the same roles to people from the same country, so Spanish students played the role of analysts and Argentinean students played the role of users. Finally, we ensured that each team had the same challenges to overcome: they had a time difference of 4 hours, they had the same difference in timetables, the cultural difference was the same (low according to the Hofstede model [13]) and they had the same idiomatic differences as regards pronunciation and vocabulary. Team members were able to communicate freely for a week, but only by using the groupware tool assigned, and after that time each team gave us the requirements specification that the analysts had written with the user's approval. Finally, on receiving the requirements specification, we asked the team members to fill in a post-experiment questionnaire in order to obtain their personal opinion of the requirements elicitation process. To do so, stakeholders where asked to rank their satisfaction through the use of a scale of 0-4 (0=very bad, 1=bad, 2=acceptable, 3=good, 4=very good).

5. Analyzing stakeholders' satisfaction about communication

Analysing the data collected by means of the post-experiment questionnaire, we obtained that, with regard to stakeholders' satisfaction with communication during the experiment, most people in Group 1 ranked their satisfaction as 4="very good", while most people in Group 0 ranked their satisfaction as 3="good" (0). This difference between both groups would indicate that: Stakeholders' satisfaction with communication seems to be better in groups that used the most suitable groupware tool according to our set of preference rules.

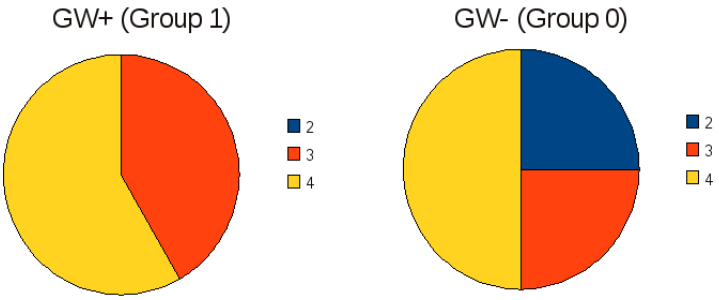


Figure 3: *Stakeholders' satisfaction about communication in both groups*

In a second step, we specifically analyzed if stakeholders satisfaction about communication was different considering the cultural or language differences that could happen in a team. To do so we included additional questions to differentiate stakeholders' satisfaction about communication with the member of their own country and with the member of a different country.
Regarding the question about communication with the member of the same country, it could only be answered for Spanish people, who have a Spanish partner in each team; therefore only 16 answers could be compared. Then, concerning to stakeholders' satisfaction with communication with members from the same country, we obtained that more people in Group 1 ranked their satisfaction as 4="very good", while most people in Group 0 ranked their satisfaction as 3="good" and 2="acceptable" (0). This difference between both groups would indicate that, stakeholders' satisfaction with communication with members from their own country seems to be better in groups that used the most suitable groupware tool according to our set of preference rules.

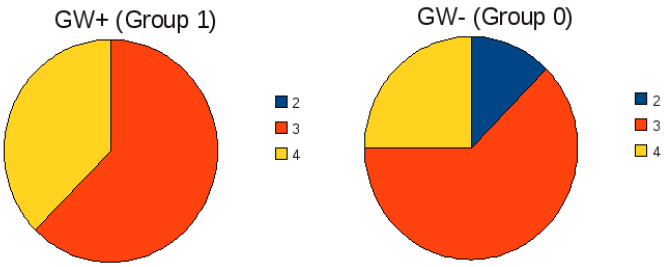


Figure 4: *Stakeholders' satisfaction about communication with members from their own country, in both groups*

Similarly, when analyzing stakeholders' satisfaction with communication with members from a different country, we obtained that more people in Group 1 ranked their satisfaction as 4="very good", as well as most people in Group 0 ranked their satisfaction as 3="good" and 2="acceptable" (0). This difference between both groups would indicate that, stakeholders' satisfaction with communication with members from different countries seems also to be better in groups that used the most suitable groupware tool according to our set of preference rules.

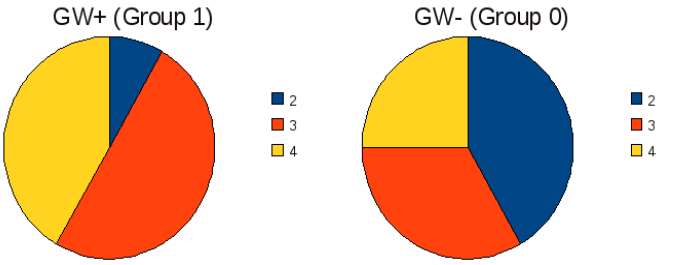


Figure 5: *Stakeholders' satisfaction about communication with members from a different country, in both groups.*

As a conclusion, our preliminary results show that our proposal seems to improve stakeholders' satisfaction with regard to communication with the rest of the group when using the groupware tool deemed to be suitable for them according to our technology selection approach. Furthermore, stakeholders satisfaction about communication was analyzed considering the possible effects of cultural and language differences between people from different countries, and the results showed that: First, for both groups (0 and 1), stakeholders' satisfaction about communication with team members from the same country was better than satisfaction about communication with team members from a different country (which is understandable, because people from a same country share a lot of information, customs, etc). And second, when considering stakeholders' satisfaction about communication with team members located in a different country, we observed the same difference between groups 0 and 1 that we have observed previously, which means that stakeholders' satisfaction about communication seems to be higher when groupware tools were chosen according to our technology selection approach, no matter if stakeholders are from the same country or not.
Bearing this in mind, our current work is focused on analyzing other factors like the quality of software specifications from the point of view of external reviewers (which would consider the product quality), as well as the quality of communication (by means of qualitative research techniques to analyze text and conversations recorded during the experiment).

6. Conclusions and Future Work

In order to save costs, many organisations have adopted a distributed structure for software development, which is called global software development (GSD). In such environments, software development projects are affected by many factors which complicate communication and knowledge exchange. Bearing this in mind, in this paper we propose a methodology for groupware tools selection which focuses on cognitive style models, by using the Felder and Silverman (F-S) learning style model. This proposal has been applied in a controlled experiment, and some of its preliminary results are shown here. We believe that this experiment could be seen as a first step in a series of experiments, which must be repeated in order to contrast the results obtained in different scenarios. However, the preliminary results seem to support our hypothesis indicating the influence of cognitive profiles in the election of groupware tools.

Acknowledgements

This work has been funded by the PEGASO/MAGO project (Ministerio de Ciencia e Innovación MICINN and Fondos FEDER, TIN2009-13718-C02-01). It is also supported by MEVALHE (HITO-09-126) and ENGLOBAS (PII2I09-

0147-8235), funded by Consejería de Educación y Ciencia (Junta de Comunidades de Castilla-La Mancha), and co-funded by Fondos FEDER, as well as MELISA (PAC08-0142-3315), Junta de Comunidades de Castilla-La Mancha, Consejería de Educación y Ciencia in Spain. Also by the 04/E072 project de la Universidad Nacional del Comahue, from Argentina.

References

1. Aranda, G., Cechich, A., Vizcaíno, A. and Castro-Schez, J. J., Using fuzzy sets to analyse personal preferences on groupware tools, in X Congreso Argentino de Ciencias de la Computación, CACIC 2004, San Justo, Argentina, October, 2004.
2. Aranda, G., Vizcaíno, A., Cechich, A., Piattini, M. and Castro-Schez, J. J., Cognitive-Based Rules as a Means to Select Suitable Groupware Tools, in 5th IEEE International Conference on Cognitive Informatics (ICCI'06), Beijing, China, July, 2006.
3. Blank, G. D., Roy, S., Sahasrabudhe, S., Pottenger, W. M. and Kessler, G.D., Adapting Multimedia for Diverse Student Learning Styles, Journal of Computing in Small Colleges, 18), 2003.
4. Carmel, E. and Agarwal, R., Tactical Approaches for Alleviating Distance in Global Software Development, IEEE Software, 18, 2001.
5. Castro, J. L., Castro-Schez, J. J. and Zurita, J. M., Learning Maximal Structure Rules in Fuzzy Logic for Knowledge Acquisition in Expert Systems. Fuzzy Sets and Systems, 101, 1999.
6. Chiew, V. and Wang, Y., From Cognitive Psychology to Cognitive Informatics, in Second IEEE International Conference on Cognitive Informatics, ICCI'03, London, UK, August, 2003.
7. Damian, D. and Zowghi, D., The impact of stakeholders geographical distribution on managing requirements in a multi-site organization, in IEEE Joint International Conference on Requirements Engineering, RE'02, Essen, Germany, September, 2002.
8. Ellis, C. A., Gibbs, S. J. and Rein, G. L., Groupware: Some Issues and Experiences, Communications of ACM, 34, 1991.
9. Felder, R. and Silverman, L., Learning and Teaching Styles in Engineering Education, Engineering Education, 78, 1988.
10. Felder, R. and Spurlin, J., Applications, Reliability and Validity of the Index of Learning Styles, International Journal of Engineering Education, 21, 2005.
11. Herlea, D. and Greenberg, S., Using a Groupware Space for Distributed Requirements Engineering, in 7th IEEE Int'l Workshop on Coordinating Distributed Software Development Projects, Stanford, California, USA, June, 1998.
12. Hickey, A. M. and Davis, A., Elicitation Technique Selection: How do experts do it?, in International Joint Conference on Requirements Engineering (RE03), Los Alamitos, California, IEEE Computer Society Press, September, 2003.

13. Hofstede, G., Cultures and Organizations, Software of the Mind: Intercultural Cooperation and its Importance for Survival, 1st ed., New York, McGraw-Hill, 1996.
14. Martín, A., Martínez, C., Martínez Carod, N., Aranda, G. and Cechich, A., Classifying Groupware Tools to Improve Communication in Geographically Distributed Elicitation, in IX Congreso Argentino de Ciencias de la Computación, CACIC 2003, La Plata, Argentina, October, 2003.
15. Miller, J. and Yin, Z., A Cognitive-Based Mechanism for Constructing Software Inspection Teams, IEEE Transactions on Software Engineering, 30(11), 2004.
16. Wang, Y., On the Cognitive Informatics Foundations of Software Engineering, in Third IEEE International Conference on Cognitive Informatics, ICCI'04, Victoria, Canada, August, 2004.

Towards Scaling Up DynAlloy Analysis using Predicate Abstraction

RODRIGO ARIÑO[1], RENZO DEGIOVANNI[1], RAUL FERVARI[1], PABLO PONZIO[1], NAZARENO AGUIRRE[1]

[1]Departamento de Computación, FCEFQyN, Universidad Nacional de Río Cuarto, Ruta 36 Km. 601, Río Cuarto (5800), Argentina.
{rodrigoarino, renzo.degiovanni}@gmail.com,
{rfervari, pponzio, naguirre}@dc.exa.unrc.edu.ar

Abstract. *DynAlloy is an extension to the Alloy specification language suitable for modeling properties of executions of software systems. DynAlloy provides fully automated support for verifying properties of programs, in the style of the Alloy Analyzer, i.e., by exhaustively searching for counterexamples of properties in bounded scenarios (bounded domains and iterations of programs). But, as for other automated analysis techniques, the so called state explosion problem makes the analysis feasible only for small bounds. In this paper, we take advantage of an abstraction technique known as predicate abstraction, for scaling up the analysis of DynAlloy specifications. The implementation of predicate abstraction we present enables us to substantially increase the domain and iteration bounds in some case studies, and its use is fully automated. Our implementation is relatively efficient, exploiting the reuse of already calculated abstractions when these are available, and an "on the fly" check of traces when searching for counterexamples. We introduce the implementation of the technique, and some preliminary experimental results with case studies, to illustrate the benefits of the technique.*

1. Introduction

With the well established success of model checking, computer scientists and an increasing number of practitioners have increased their interest in the construction of automated tools for the verification of software systems. Despite some breakthrough advances in automated analysis techniques, such as the use of symbolic representations in model checking, algorithms for Automated verification essentially work by performing an exhaustive exploration of the state space of programs, and thus they can fail to terminate in a reasonable amount of time, even for simple models/programs and properties. Automated SAT based analyses, such as those associated with the Alloy Analyzer and bounded model checkers, are no exception to this situation, and therefore techniques for tackling the so called *state explosion problem* are necessary. Alloy[13] is a specification language of increasing popularity in the last few years. Alloy is based on an extension of first order logic. Its main features are its simplicity and ease of use, and the availability of an automated

tool -the Alloy Analyzer- for simulating and finding violations of properties of Alloy specifications. As models of specifications (models in the sense of mathematical logic) are potentially infinite, the tool all the possible instances up to a bound on the data domains given by the user, and checks if they satisfy the desired property. If an instance that violates the property is found, it is showed to the user as a counterexample. If, on the other hand, no counterexample is found, one cannot in principle guarantee the validity of the property being checked, since there might exist counterexamples to the property for bigger domains. However, in the absence of counterexamples, it definitely allows us to gain more confidence about the validity of the property.
While Alloy is useful for modeling static properties of systems, it has problems when dealing with the dynamics of software systems. To overcome these problems an extension to the language was proposed, called DynAlloy[8]. Based on dynamic logic, DynAlloy provides a simple way to specify properties of executions. DynAlloy also has an automated tool which allows users to find of specifications, by translating DynAlloy specifications into Alloy.
The usefulness of Alloy is justified by the small scope hypothesis[13], which asserts that most errors have counterexamples of small size. Some research supporting this hypothesis has been carried out[14]. Unfortunately, the analysis of Alloy and DynAlloy specifications do not scale up well, and analysis is only possible using small bounds. The reason for this problem is strongly related to the state explosion problem mentioned, and is due to the analyses being based on a reduction to boolean satisfiability, which is a well known NP problem.
In order to overcome the scalability issues associated with automated analysis techniques, many approaches have been proposed: abstraction[2,5,11,6], symbolic execution[12,7], static analysis, etc. We are interested in a predicate abstraction technique that allows us, given a set of predicates over the state space of a model, to automatically construct an abstract model whose states are boolean valuations of the predicates over concrete states. As the abstractions we construct are conservative, the properties that can be verified on the abstract model also hold in the concrete model. Therefore, as the abstract model is usually simpler -in the sense that it has less states than the concrete model- less time and memory is required to complete the verification tasks.
A problem associated with abstraction is that we often construct abstract models that are too coarse to verify a certain property. For this reason, automatic algorithms for refining abstractions have been developed[6,1]. They are based on the observation that when trying to verify a property an abstract model two different kinds of counterexamples may appear: real counterexamples that when concretized perform a violation of the property on the concrete model, or spurious counterexamples, i.e., undesired behaviours introduced by the loss of information resulting from the abstraction.
Given a spurious counterexample, the information it provides can be used to construct new abstraction predicates. The purpose of the new predicates is to refine the current abstract model, eliminating the spurious counterexample. The result of this may be that the new refined model can be used to verify a property for which the original abstraction failed.
In[10] we introduced a fully automatic predicate abstraction algorithm for DynAlloy specifications, inspired by Graf and Saidi's work[11]. The

algorithm traverses the abstract state space of the DynAlloy program in a depth first order. At each step an abstract state is constructed using only the previous abstract state and the specification of the action currently executed. A key aspect of our algorithm is that previously constructed abstract states can be reused in the construction of new abstract states. As we use Alloy to automatically construct the abstractions, this has a big impact on the algorithm's run time performance. Now, we present a tool that fully implements that algorithm. This tool was de veloped with extensibility in mind, making it easy to add new abstraction algo rithms, or modifications of the current one. This tool implements an adaptation of Das and Dill's technique[6] in order to deal with spurious counterexamples and refine abstractions.

2. The Alloy and DynAlloy Specification Languages

Alloy is a specification language based on relational logic, an extension of first order logic designed to support relational operators: relational image, closures, transpose, etc. Alloy syntax is OO-like and simple, it has only a few reserved keywords, making the language easy to learn and use for people with basic mathematical training. The language was designed with automatic tool support in mind, so it was developed together with an automatic tool -the Alloy Analyzer- which allows users to simulate models and search for counterexamples of properties that the model is intended to satisfy. Given an Alloy specification, a property, and bounds over the data domains of the Alloy model, the Analyzer constructs a propositional formula -representing the specification and the negation of the property- that is passed as input to an off-the-shelf SAT solver. If the SAT solver finds a model, a counterexample to the property is constructed based on this model. Otherwise the property does not have counterexamples on the given bounds.
Alloy is a model-oriented language, designed to specify properties of software systems. To model state change in Alloy it is necessary to introduce identifiers for states before and after the action is executed; this is a common practice in the Z language[15], on which Alloy was inspired. Although Alloy allows us to describe simple (single action) state change rather easily, it does not provide an adequate way to specify properties of more sophisticated executions of systems[8]. One way to simulate executions in Alloy is to manually introduce a notion of execution trace as a signature, which depends on the specification's signatures and operations and, therefore, must be defined on a per model (or per program) basis. This problem was addressed by an extension of Alloy -the DynAlloy language. DynAlloy introduces the notion of atomic action to model state change, and operations to compose these actions, whose semantics was inspired by dynamic logic. For the sake of space, we will introduce the most relevant parts of the DynAlloy language by means of examples. The state of a system is specified in Alloy and Dynalloy using signatures. Signatures define sets of elements, and relations between them. As an example, we define the structure of linked lists of integers:

```
one sig Null {}   sig Node {          sig List {
                    next: Node+Null,     head:Node+Null,
                    value: Int          }
                    }
```

In the above definitions *Null* is a singleton set, representing the null reference, present in most programming languages. *Node* is the universe of nodes that may belong to a linked list, *next* is a binary relation that relates each node to its successor and *value* assigns to each node the integer value stored in it. Finally, list is the set of all possible lists, and *head* associates a head node (or a null reference) with each list.

DynAlloy atomic actions are useful for describing operations over the signatures of the model. They must be defined in terms of their corresponding pre and post conditions. For example, an operation that deletes the element at the head of a list can be defined as follows:

```
action removeFirst[l: List] {
    pre { l.head != NullValue }
    post { l'.head = l.head.next }
}
```

removeFirst can only be executed if the list's head is non null. Note the similarity of the dot expressions (like *l.head*) with that of object oriented languages. The postcondition introduces the variable *l'* to denote the state after the execution of the action. In this case, it states that the successor of the head of the list becomes the new head. In this way, we can think of the postcondition of an action as a relation between pre and post states.

DynAlloy provides three operators for composing atomic actions: sequential composition (;), non deterministic choice (+) and bounded iteration (*). Combining atomic actions and tests (assertions) using these operators we form DynAlloy programs. Programs are also annotated with pre and postconditions, to specify intended properties of programs, and then search for counterexamples. For example:

```
assertCorrectness removeAll[l:List]{
  pre = {}
  prg = {([l.head=NullValue]?;removeFirst(l))*;
          [l.head=NullValue]?}
  post = { l'.head = NullValue } }
```

Assertion *removeAll* can be thought of as the specification corresponding to the program while (head(l) != NullValue) do removeFirst(l). Its post-condition asserts that when the program finishes the list is empty, which should obviously be valid. Given a DynAlloy specification, the DynAlloy automatically constructs an equivalent Alloy model which allows one to "execute" the program using the Alloy Analyzer. Bounds on the number of iterations to be executed and on the number of elements of signatures are

necessary to perform the translation; the user must provide them. Executions of the program that violate the specification are shown to the user by the analyzer. It is worth noting that experiments showed that the analysis of dynamic properties is more efficient using DynAlloy (and its translation into Alloy) than the traditional Alloy approach, explicitly defining trace signatures in Alloy[9].

3. A Predicate Abstraction Algorithm for DynAlloy

Cousot introduced the idea of Abstract Interpretation[4], which consists of interpreting computations of programs over simpler (abstract) domains that encode less information about the computations, but are usually easier to construct and explore. The method is based on the definition of two functions: $\alpha: \wp(S) \to S^A$, $\gamma: S^A \to \wp(S)$. α maps sets of concrete states to abstract states, and is called the abstraction function; γ associates with each abstract state the set of concrete states it represents. α and γ must conform a so called Galois Connection, that is, $\alpha(\gamma(s^A)) = s^A$ and $\phi \Rightarrow \gamma(\alpha(\phi))$; this ensures that abstract states represent over approximations of sets of concrete states. In addition, the sets of initial states must be included in the concretization of the initial abstract state, and applying an abstract transition τ_i^A to an abstract state s^A must result in a set of states containing $\tau_i(\gamma(s^A))$. This requirement ensures that each concrete trace has a corresponding trace on the abstract model. The advantage of Abstract Interpretation is that the (safety) properties we verify on the abstract domain are also valid for the concrete domain. This technique is very powerful, allowing one, for instance, to apply model checking algorithms to infinite state systems, and to analyze systems with complex state spaces over which model checking algorithms would fail to terminate in a reasonable amount of time.

Two decades later, Graf and Saidi introduced Predicate Abstraction[11] as a way to automate the construction of abstract domains, given a set $\phi_1, \ldots, \phi_l$ of predicates over the state of the program that must be provided by the user.

Based on this work we developed a predicate abstraction algorithm for DynAlloy [10]. Our concrete state space is composed of the DynAlloy signatures defining the state space of the program. Given abstraction predicates $\varphi_1(s), \ldots, \varphi_l(s)$, our abstract state space consists of the set of monomials over boolean $B_1, \ldots, B_l$. A monomial is a conjunction of B_i's and $\neg B_i$'s, where each of the variables ap pear at most once. The Boolean constant false is also considered as a monomial. Note that, to obtain the concrete set of states represented by an abstract state (monomial) a replacement of each B_i for the corresponding φ_i suffices. Hence, the concretization function is defined as:

$$\gamma(m) = m[B_1/\phi_1, \ldots, B_l/\phi_l]$$

To explain how to perform one step in the abstract execution of the program we first need to introduce the abstraction function:

$$\alpha(\psi(s)) = (\bigwedge\{B_i \mid all\ s : \psi(s) \Rightarrow \phi_i(s)) \wedge (\bigwedge\{\neg B_i \mid all\ s : \psi(s) \Rightarrow \neg\phi_i(s))$$

Thus, to automatically construct the abstract monomial corresponding to ψ we use the Alloy Analyzer to check whether ψ implies each of $\varphi_i$or $\neg\varphi_i$ for all concrete states *s*. Since we use the Alloy Analyzer for performing these checks, the answer is not necessarily absolute (we assume that a formula is valid if the analyzer is unable to find bounded counterexamples for it). The bounds needed to run the checks must be provided by the user. To perform an abstract execution we first have to calculate the abstract state corresponding to the initial concrete states. We do this by directly applying α to the precondition of the DynAlloy program. Next, we have to calculate the abstract transitions and apply them in the order prescribed by the program. Thus, given an abstract state s^A and an abstract action τ_i^A, the idea to abstractly execute τ_i^A starting at s^A is to apply τ_i to each state of $\gamma(s^A)$, and then abstract away the result. Now, since DynAlloy postconditions are not predicates over the system state, but instead they relate action's initial states with final states, we need, in order to apply the abstraction function, to write a predicate for the strongest postcondition (SP) of τ_i with re spect to $\gamma(s^A)$. That is, assuming τ_i has precondition $pre_{\tau_i}(s)$ and postcondition $post_{\tau_i}(s, s')$, $SP(\tau_i, \gamma(q)) = \forall s : pre_{\tau_i}(s) \wedge \gamma(q)(s) \wedge post_{\tau_i}(s, s')$. Now α can be applied to $SP(\tau_i, \gamma(q))$ to obtain the desired abstract successor.

For example, if we consider the abstraction predicates φ_0= NoLoops(l) and φ_1 = NullHead(l), an abstract execution of the *RemoveFirst* action (defined in section 2) starting at the monomial $B_0 \wedge \neg B_1$ will finish in the abstract state B_0.

It is interesting to note that our algorithm does not construct the complete abstract state space as the technique showed in[11] does, since this is -as some experiments have shown- very expensive. Instead, it constructs the abstractions "on demand", that is, the algorithm visits the program control tree in a depth first search manner, starting by abstracting the initial states, and then applying abstract transitions in the mentioned order. The DFS algorithm executes loops up to a bound given by the user. The postcondition of the DynAlloy program (concrete property to verify) is always included as an abstraction predicate (say φ_p). Therefore, as the procedure executes over approximations of concrete traces, we can ensure that if all the abstract traces end up in a monomial with B_p set then the concrete program satisfies the intended property (up to the bound for loops unrolling). If this is not the case, the algorithm will stop when it explores the first trace that does not satisfy the above requirement. These kind of traces are called abstract counterexamples, that is, they are traces that when concretized may or may not violate the concrete property. A concretized abstract counterexample that violates the concrete postcondition is called a real counterexample. Otherwise, it is a spurious counterexample: an abstract trace that

does not have any concrete counterpart, produced by the loss of information on the abstraction process. Spurious counterexamples must be eliminated from the abstract model if we want to continue the abstract verification. Furthermore, they can be useful to refine abstractions, as it is exploited by the so called *counter example guided abstraction refinement* techniques.

An important feature of the algorithm is that (part of) the abstractions produced can be reused. As it builds the abstractions using the Alloy Analyzer, which is based on an NP-Complete algorithm, reusing as much information as possible is imperative to improve the running time of the algorithm. The idea behind reuse is that, if we have previously built an abstract state $s^{A'}$ applying an action τ_i to another state s^A, then the Boolean variables appearing at $s^{A'}$ will be set in the same way on each state obtained by applying τ_i to any consistent abstract state $s^{A''}$ that is stronger than s^A. This is due to the way we calculate the abstractions (by calculating logical implications). Hence, the algorithm stores the result of each execution of an abstract transition, and uses this information when possible at the following uses of the same action in the DFS execution.

Due to a lack of space, we will leave out the counterexample guided refinement process, and concentrate on the process for the construction of the abstraction (and checking its suitability).

4. Some Notes on the Implementation

In this section, we briefly describe the implementation of a tool automating the above described abstraction technique. The tool consists of two main modules, the abstraction module, that implements the abstraction algorithm, and the abstraction refinement module, that implements predicate discovery based on the spurious counterexamples returned by the abstraction phase.

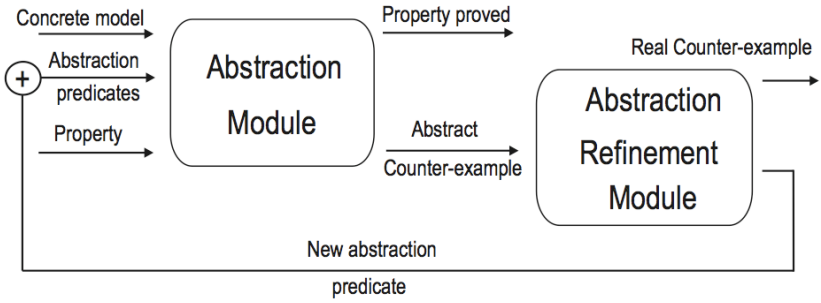


Figure 1: *Interaction of main tool's modules.*

The interaction of these modules is depicted in figure 1. The abstraction module takes a concrete model and abstraction predicates as inputs, and it abstractly executes the given program. As a result, either the abstract model satisfies the specification, or an abstract counterexample trace is found. In the first case, the

user is reported about the validity of the property and the whole execution ends. Otherwise, the counterexample found is passed as an input to the abstraction refinement module. If the counterexample is real, it is shown to the user and the process finishes; otherwise, the module uses the abstract trace to generate a new abstraction predicate. The new predicate is then added to the abstraction predicates and the abstraction process is restarted.
The tool is implemented using the Java programming language and it interacts with the latest versions of the Alloy Analyzer and DynAlloy Translator, to perform certain tasks. As we mentioned, the tool was developed with extensibility in mind, so that new abstraction algorithms and new ways to perform predicate discovery can be incorporated straightforwardly. A diagram illustrating the design of the abstraction module is shown in figure 2.

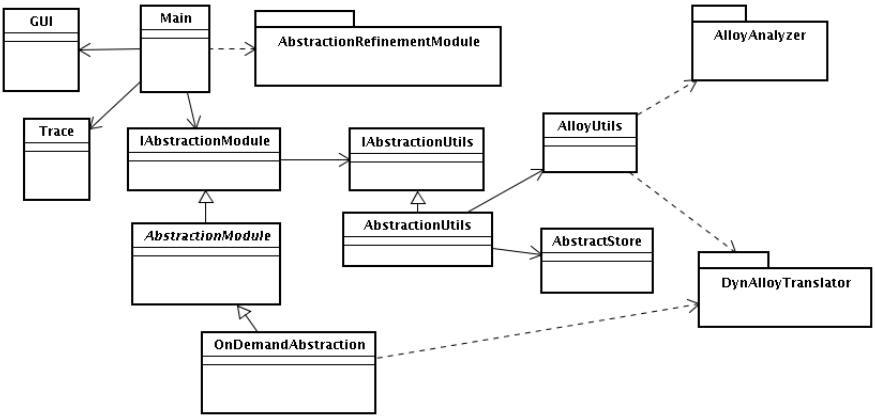


Figure 2. *Diagram for the abstraction module.*

The main components of this module are the following:

- Main: retrieves the information from the GUI and it is responsible of executing the abstraction and the predicate discovery algorithms. It oordinates the interaction between the Abstraction Module and the Abstraction Refinement Module.
- IAbstractionModule: Interface that defines common operations to abstraction algorithms.
- AbstractionModule: Abstract class that contains attributes and implementations of common operations to abstraction algorithms for DynAlloy specifications. It contains attributes used to store input predicates, abstraction bounds, number of times loops should be executed, etc.
- OnDemandAbstraction: Implementation of the predicate abstraction algorithm discussed in section 3.
- IAbstractionUtils: Defines the signatures of the abstraction and concretization functions.

- AbstractStore: Stores the results of previously executed transitions, and provides methods to easily access this information, that will be used to avoid unnecessary calls to the Alloy Analyzer to improve the efficiency of the abstraction.
- AlloyUtils: Defines operations needed by the abstractor whose implementation is based on the Alloy Analyzer and the DynAlloy Translator source code.

5. Some Experimental Results

In this section, we present some experimental results using the presented tool. The experiments were carried out on an Intel Core 2 Duo of 2 Ghz, with 2GB of RAM, running GNU/Linux 2.6. The version of the Alloy Analyzer employed was 4.1.8. We experimented mainly with two properties over the model of linked lists introduced before; these properties are: (P1) that the deletion method preserves the acyclicity of linked lists, and (P2) that no occurrences of the elements to be deleted belong to the list, before the current position of the cursor used for the implementation of deletion.
Without using abstraction, the Alloy Analyzer was able to verify P1 in 18 minutes and 36 seconds, for 20 loop unrolls and 21 as a scope for domains, and P2 in 27 minutes, for 15 loop unrolls and 16 as a scope for domains. The Alloy Analyzer exhausted the available memory for 24 loop unrolls and 25 as a scope for domains for P1, and 16 loop unrolls and 17 as domain scope for P2, and crashed after several hours running. Using abstraction via the presented tool, we were able to verify P1 in 4 minutes and 18 seconds for 20 loop unrolls and 21 as scope for domains (with 52428513 less calls to the Alloy Analyzer, thanks to reuse of abstraction calculations), and in 57 mins. and 26 seconds, for 25 loop unrolls and 26 as scope for domains (and a total of 1677721308 less calls to the Alloy Analyzer). For P2, we were able to verify the property for 20 loop unrolls and 21 as scope for domains in 4 mins. and 23 seconds, while it took the tool mins and 18 seconds, for 25 loop unrolls and 26 as domain scope (the savings in calls to the Alloy Analyzer were similar to those of P1) The results are promising, but we were unable to achieve the performance we expected. We are currently optimizing several parts of the abstraction tasks, aiming at scaling up DynAlloy analysis for about an order of magnitude.

6. Conclusions

We presented an implementation of a predicate abstraction technique for DynAlloy specifications, developed for scaling up the analysis of DynAlloy specifications. This tool enabled us to substantially increase the domain and iteration bounds in some case studies, and our implementation allows us to apply it automatically. For the sake of efficiency, our implementation exploits the reuse of already calculated abstractions when these are available, and an “on the fly” check of traces when looking for counterexamples. The preliminary results of

some experiments we carried out are promising, showing that the technique is worthy. However, many improvements still need to be developed. First, there are several opportunities for saving space in the representation of the concrete and abstract computation trees, and we are currently moving to a control graph representation. Second, in its current form the tool can only handle DynAlloy specifications originating from code (where atomic actions are reversible); other more abstract DynAlloy models fail to be abstracted, due to technical reasons having to do with the calculation of weakest preconditions of traces when calculating or refining abstractions. We plan to enhance our tool so that these more abstract DynAlloy models are also handled.

References

1. Ball, T., Cook, B., Das, S. and Rajamani, S., Refining Approximations in Software Predicate Abstraction, in Proc. of International Conference on Tools and Algorithms for the Construction and Analysis of Systems, LNCS, Springer, 2004.
2. Clarke, E., Grumberg, O. and Long, D., Model checking and abstraction, ACM Transactions on Programming Languages and Systems, 16(5), ACM Press, 1994.
3. Clarke, E., Kroening, D., Sharygina, N. and Yorav, K., Predicate Abstraction of ANSI C Programs using SAT, Technical Report CMU-CS-03-186, Carnegie Mellon University, 2003.
4. Cousot, P., Cousot, R., Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fix points, Proc. of 6th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, ACM Press, 1977.
5. Cousot, P., Abstract interpretation, ACM Comp. Surveys, 28(2), ACM Press, 1996.
6. Das, S. and Dill, D., Counterexample Based Predicate Discovery in Predicate Abstraction, in Proc. of International Conference on Formal Methods in Computer Aided Design, Portland, USA, LNCS, Springer, 2002.
7. Dennis, G., Chang, F., Jackson, D., Modular Verification of Code with SAT, Proc. of the ACM/SIGSOFT Int. Symposium on Software Testing and Analysis, 2006.
8. Frias, M., Galeotti, J. P., López, C., Pombo and N. Aguirre, DynAlloy: upgrading Alloy with actions, in Proc. of the 27th International Conference on Software Engineering ICSE 2005, ACM Press, 2005.
9. Frias, M., Galeotti, J. P., López, C., Pombo and N. Aguirre, Efficient Analysis of DynAlloy Specifications, in ACM Transactions on Software Engineering and Methodology (TOSEM), ACM Press, 2007.
10. Aguirre, N., Frias, M., Ponzio, P., Cardiff, B., Galeotti, J. P. and Regis, G., Towards Abstraction for DynAlloy Specifications, in Proc. of the 10th International Conference on Formal Engineering Methods, LNCS, Springer, 2008.

11. Graf, S. and Saïdi, H., Construction of abstract state graphs with PVS, in Proc. of 9th International Conference on Computer Aided Verification, Haifa, Israel, LNCS 1254, Springer, 1997.
12. Khurshid, S., Pasareanu, C., Visser, W., Generalized Symbolic Execution for Model Checking and Testing, in Proc. of the 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, LNCS, Springer, 2003.
13. Jackson, D., Software Abstractions, The MIT Press, 2006.
14. Andoni, A., Daniliuc, D., Khurshid, S. and Marinov, D., Evaluating the "Small Scope Hypothesis", Technical Report MIT-LCS-TR-921, MIT CSAIL, 2003.
15. Woodcock, J., Davies, J., Using Z: Spec., Refinement and Proof, Prentice-Hall, 1996.

VI

Database and Data Mining Workshop

Dynamic Selection of Suitable Pivots for Similarity Search in Metric Spaces

CLAUDIA DECO[1], MARIANO SALVETTI[1], NORA REYES[2], CRISTINA BENDER[1]

[1] Facultad de Ciencias Exactas, Ingenieria y Agrimensura,
Universidad Nacional de Rosario, (2000) Rosario, Argentina
deco@fceia.unr.edu.ar, salvettimariano@hotmail.com, bender@fceia.unr.edu.ar
[2] Departamento de Informática,
Universidad Nacional de San Luis, (5700), San Luis, Argentina
nreyes@unsl.edu.ar

Abstract. *This paper presents a data structure based on Sparse Spatial Selection (SSS) for similarity searching. An algorithm that tries periodically to adjust pivots to the use of database index is presented. This index is dynamic. In this way, it is possible to improve the amount of discriminations done by the pivots. So, the primary objective of indexes is achieved: to reduce the number of distance function evaluations, as it is showed in the experimentation.*

Keywords: *Metric databases, dynamic index, Sparse Spatial Selection.*

1. Introduction

The digital age creates a growing interest in finding information in large repositories of unstructured data that contain textual data, multimedia, photographs, 3D objects and strings of DNA, among others. In this case, it is more useful a similarity search than an exact search. Similarity search is usually an expensive operation.

The similarity search problem can be formalized through the concept of metric space. Given a set of objects and a distance function between them, which measures how different they are, the objective is to retrieve those objects which are similar to a given one. An index is a structure that allows fast access to objects, and accelerates the retrieval. There are several types of indexes proposed for metric spaces that have differences, such as how they are explored, or how they store the information.

This paper presents an improvement to Sparse Spatial Selection (SSS). This improvement consists on implementing new policies of incoming and outgoing pivots, in order to that the index suits to searches, to dynamic collections, and to secondary memory.

The rest of the paper is structured as follows: Section 2 presents basic concepts. Section 3 describes the problem of pivots selection. Section 4 presents the proposed method, and Section 5 shows experimental results. Finally, conclusions and future lines of work are presented.

2. Basic Concepts

A *metric space* (*X, d*) consists of a universe of valid objects *X* and a *distance function* $d:X\times X\rightarrow\Re^{+}$ defined among them. This function satisfies the properties: strictly positiveness $d(x,y)>0$, symmetry $d(x,y)=(y,x)$, reflexivity $d(x,x)=0$ and triangular inequality $d(x,y)\leq d(x,z)+d(z,y)$. A finite subset *DB* of *X*, with $|DB|=n$, is the set of elements where searches are performed. The definition of the distance function depends on the type of objects. In a vector space, *d* may be a function of Minkowski family: $L_s((x_1, ..., x_k),(y_1, ..., y_k))=(\sum |x_i-y_i|^s)^{1/s}$. Some examples are: Manhattan distance ($p=1$), Euclidean distance ($p=2$), and Chebychev distance ($p=\infty$).

In metric databases queries of interest can be: range search and *k*-nearest neighbors search. In the first, given a query *q* and a radius *r*, objects that are at a distance less than *r* are retrieved: $\{u\in DB / d(u,q)\leq r\}$. In *k nearest neighbors* search, the *k* objects closest to *q* are retrieved, that is: $A\subseteq DB$ such that $|A|=k$ and $\forall u\in A, v\in DB-A, d(q,u)\leq d(q,v)$. The basic way of implementing these operations is to compare each object in the collection with the query. The problem is that, in general, the evaluation of the distance function has a very high computational cost, so searching in this manner is not efficient when the collection has a large number of elements. Thus, the main goal of most search methods in metric spaces is to reduce the number of distance function evaluations. Building an index, and using the triangular inequality, objects can be discarded without comparing them with the query. There are two types of search methods: *clustering-based* and *pivots-based* [1]. The first one splits the metric space into a set of equivalence regions, each of them represented by a *cluster center*. During searches, whole regions are discarded depending on the distance from the cluster center to the query. *Pivot-based* algorithms select a set of objects in the collection as *pivots*. An index is built by computing distances from each object in the database to each pivot. During the search, distances from the query *q* to each pivot are computed, and then some objects of the collection can be discarded using the triangular inequality and the distances precomputed during the index building phase. Some pivot-based methods are: *Burkhard-Keller-Tree* [2], *Fixed-Queries Tree* [3], *Fixed-Height FQT* [3], *Fixed-Queries Array* [4], *Vantage Point Tree* [5], *Approximating and Eliminating Search Algorithm* [6], *Linear AESA* [7] y *SSS* [8,9].

This paper presents an improving to the Sparse Spatial Selection (SSS) method, implementing new policies for selecting incoming and outgoing pivots from the index. The proposed method is dynamic, because the collection can be initially empty, or can increase or decrease with time. Also, this proposal generates a number of pivots based on the intrinsic dimensionality of the space.

3. Related Work on Pivots Selection

Pivots selection affects the efficiency of the search method in the metric space, and the location of each pivot with respect to the others determines the ability to exclude elements of the index without directly comparing them with the query. Most search pivots-based methods select pivots randomly. Also, there are no guidelines to determine the optimal number of pivots, parameter which depends on the specific collection. Several heuristics have been proposed for the selection of pivots, as if the distance function is continuous or discrete. In [7] pivots are objects that maximize the sum of distances among them. In [10] a criterion for comparing the efficiency of two sets of pivots of the same size is presented. Several selection strategies based on an efficiency criterion to determine whether a given set of pivots is more efficient than another are also presented. The conclusion is that good pivots are objects far away among them and to the rest of the objects, although this does not ensure that they are always good pivots.

In [8] the Sparse Spatial Selection (SSS) which dynamically selects a set of pivots well distributed throughout the metric space is presented. It is based on the idea that, if pivots are dispersed in the space, they will be able to discard more objects during the search. To achieve this, when an object is inserted into the database, it is selected as a new pivot if it is far enough from the other pivots. A pivot is considered to be far enough from another pivot if it is at a distance greater than or equal to $M \times \alpha$. M is the maximum distance between any two objects. α is a constant parameter that influences the number of selected pivots and its takes optimal experimental values around 0.4.

In all of the analyzed techniques for selecting pivots, the number of pivots must be fixed in advance. In [10] experimental results show that the optimal number of pivots depends on the metric space, and this number has great importance in the method efficiency. Because of this, SSS is important in order to adjust the number of pivots as well as possible. To improve SSS, we propose that the index suits to searches, after the index was adapted to the metric space.

4. Proposed Method

We present a new indexing and similarity searching method based on dynamic selection of pivots. The proposed method is *dynamic*, because it could be applied to an initially empty database that grows over time. The method is *adaptive,* because it is not necessary to preset the number of pivots to be used because the algorithm selects pivots as necessary to self-adapt it to space complexity.

In the construction of the index, SSS is applied to select the initial pivots. Then, as time passes and searches are performed, we apply new policies for selecting pivots in order to eliminate those least discriminating pivots from the index, and to select objects as candidate pivots to put them into the index.

In this way, we can adapt dynamically the index to searches performed during a given time.

4.1. Initial construction of the index and growth of the collection

Let *(X,d)* be a metric space, where $DB \subset X$ is the database. Let M be the maximum distance between objects ($M=\max\{d(x,y) \mid x,y \in X\}$), and α a value between 0.35 and 0.40 [8]. The collection of elements is initially empty.
The first object x_1 inserted into the database, is the first pivot p_1. When the second (or new) object is inserted in the database, its distance to all pivots that are already in the index is calculated. If these distances are all greater than or equal to $M \times \alpha$, this object is added to the set of pivots. That is, an object of the collection will be a pivot if it is more than a fraction of the maximum distance of all pivots. Thus, the set of pivots does not have to be selected randomly, because pivots are chosen as the database grows. Then, distances from the new pivot against to all database objects are calculated and stored. Also, the value of M is updated. The pseudo code is:

```
Input: (X,d), DB ,M , α
Output: Pivots
0   Pivots ← {x1}
1   FOR ALL xi ∈ DB DO
2           IF(∀p ∈ Pivots, d(xi, p) >= M×α)
3               THEN Pivots ← Pivots ∪ {xi}
4           END IF
5           Recalculate the value of M, and update.
6   END FOR ALL
```

Pivots that were selected for the initial index are far apart (over $M \times \alpha$). For many authors, this is a desirable feature of the set of pivots. The number of pivots depends on the initial dimensionality of the metric space. When the construction begins, the number of pivots should grow rapidly in the index, but this number will be stabilized when the database grows.

4.2. Exchange of Pivots in the Index

Given a query *(q,r)*, distances of q toward all pivots is calculated. Applying triangular inequality, all elements $x_i \in DB$ such that $|d(x_i,p_j)-d(p_j,q)|>r$ for some pivot p_j are discarded. Not discarded objects form the list of candidates, $\{u_1,u_2,\ldots u_n\} \subset DB$, and should be compared directly with the query. From this list of candidates, the statistics on the elements of the database that are part of search results is increased in a unit. If $\max_{1 \le j \le k} |d(q,p_j)-d(e,p_j)|>r$, then the element e is outside the query range. So, the pivot p_j discriminates the object $e \in DB$. With searches, statistics of discrimination of each pivot are stored. These statistics are calculated when search results are obtained.

Selection policy for the Outgoing Pivot. When a pivot is a "bad pivot"? In a query, at most n elements (i.e., all elements of the database) can be eliminated. This elimination would have split these n discriminations between k pivots. And in a query, it is possible to know which pivot discriminates each element $e \in DB$.
Taking into account the objects discriminated by the various pivots, and a set of B queries, we define the percentage of discrimination for a pivot p_i as $[\%Disc(p_{ij}]=Disc(p_{ij})/(B \times n)$, where $Disc(p_i)$ is the amount of items that pivot p_i discriminates and $(B \times n)$ represents the total of possible discriminations. Then, p_i is a "bad pivot" when $[\%Disc(p_{ij}] < 1/k$ where $1/k$ is an experimental threshold, which is proposed as a constant that depends on the number of pivots in the index. If $[\%Disc(p_{ij}] < 1/k$, we say that p_i is very little relevant to discriminate, at least in light of the B searches made to the database. Then, it is selected as a victim, and it could be replaced in a future. So, the selection policy of a victim pivot within the index is to choose the "less discriminating pivot". After B searches the pivot with the lower percentage of discrimination is determined. If it is less than a threshold of tolerance with value T, it is replaced. Then, $T = 1 / (1.1 \times k)$ represents a 10% of tolerance, used to stabilize the algorithm. This parameter has been evaluated experimentally. Recalling that k is the current number of pivots in the index, this constant of tolerance is used in the next situation:

```
0   IF ( min_{1≤j≤k} discrimine [j] < 1 / (1.1 × k) ) THEN {
1           OutgoingPivot ← GetPivot();
2           ChangePivot();
3           GenerateIndex();
4   }
```

In line 0, it is evaluated whether the proportion of j-th discrimination is less than the tolerance threshold. If so, this pivot is defined as the "least discriminating pivot" leaving it available for pivots exchanging (line 2). When a pivot is replaced, a whole column of the distance matrix between the incoming pivot, which is returned from *GetPivot*() method, and all elements of the database are recalculated (*GenerateIndex*() method). The complexity of changing a pivot is $n \times \Theta(distance\ function)$. If discrimination percentage is not less than T, nothing is done.

Selection policy for the Incoming Pivot. To choose which object becomes a pivot, the policy is to propose "the candidate pivot". The approach is similar than to select a victim pivot. The idea is to use statistical data of database elements obtained from queries. An object $e \in DB$ will be a candidate pivot if it is most of the times in the list of candidates, because an element that was often part of the list of candidates is difficult to discriminate with the current pivots. This implies that if this element is selected as pivot, in future searches it will improve the percentage of discrimination around the region that surrounds it. Another benefit is that it adapts automatically pivots to the region where the majority of searches are made. This transforms the index into a dynamic structure, achieving its main objective: to reduce distance

computations in searches. Then, the element that most often was a candidate will be taken as the incoming pivot.
The following pseudo code shows the implementation of the policy for selecting incoming pivot (*GetPivot*() method). In this method the account of times that *i*-th element was part of the list of candidates is used. Thus, the element that most often was taken as a candidate is chosen as the candidate pivot to enter to the index.

```
        Input: DB, Stats
        Output: Stats, an array with the time they get the items
in the database in the list of candidates.
        0 Candidate = NULL; maxCurrentStats = 0;
        1 FOREACH ( e on Database)) DO
        2     IF (Stats(e) > maxCurrentStats)
        4           THEN Candidate = e;
        5         maxCurrentStats = Stats(e);
        6     END IF
        7 END FOREACH
        8 RETURN Candidate
```

In line 0, variable values are initialized. In line 2 the statistical value of each element *e* is compared with the current maximum value, which represents the element that most often was the candidate and which will be taken as incoming pivot. If the statistic value of *e* exceeds the current one, in line 5 *e* is selected as the future incoming pivot, and the iteration continues.

5. Experimental Results

For experimentation several sets of synthetic random points in vector spaces of dimension 8, 10, 12 and 14 are used. The Euclidean distance function is used. The number of distance evaluations required to answer a query using the proposed policies, how the index adjusted itself as queries are processed, and how database size affects the index performance, was analyzed. The database contains 100,000 objects, and range query retrieved the 0.02% elements from the database.

Number of Pivots in the Index. Our proposal creates a dynamic amount of pivots, depending on the space dimensionality, and not on the number of objects in the database. For experiments α=0.5 was set, in order to achieve a uniform and distant distribution of pivots in the space. This value of α was chosen from experimental results showed later (Figures 5 and 6). Table 1 and Figure 1 show the number of pivots depending on the collection size, as the number of elements is growing.

Table 1: *Number of pivots selected in vector spaces of dimension 8, 10, 12 and 14.*

dim	*n*, size of the collection (x 10^3)									
	10	**20**	**30**	**40**	**50**	**60**	**70**	**80**	**90**	**100**
8	11	12	12	12	13	13	14	14	15	16
10	13	18	20	20	21	21	21	21	21	22
12	25	28	28	31	32	34	34	34	34	34
14	38	47	53	60	60	61	63	66	69	69

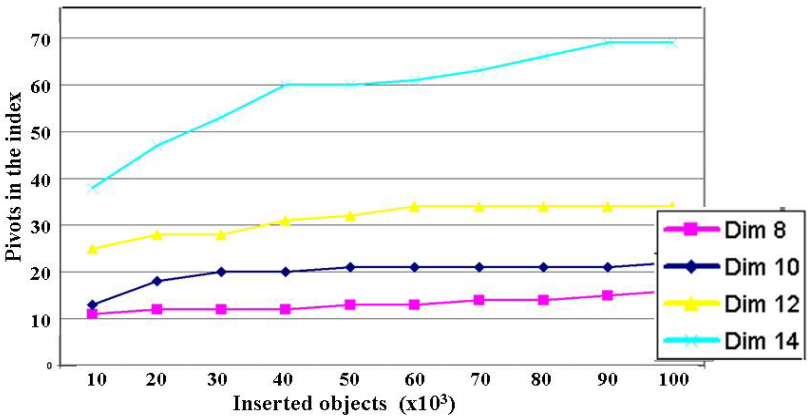


Figure 1: *Number of pivots selected in vector spaces of dimension 8, 10, 12 and 14.*

As it is noted, the number of pivots grows very quickly with the insertions of the first objects in the collection, and then continues to grow but in a slower degree until it get to stabilize. So, with few elements inserted, number of pivots depends on number of elements in the database. Already with a considerable amount of elements inside, the set of current pivots covers all the metric space. In addition, number of pivots in the index increases as the size of the space increases. So, the number of pivots in the index depends on the complexity of the collection of objects.

Search efficiency. To analyze the efficiency of the index for searching, a database with 10,000 objects, 1,000 queries and dimensionalities 8, 10, 12 and 14, are used. 20 periods were run, and information from all periods was averaged. For each dimension, the amount of distance functions evaluated (#DE), the amount of pivots used in the index (#P), and the amount of discriminations carried out (#DR), were recorded, assessing the proposed method against SSS.

***Table 2**: Efficiency in synthetic metric spaces with different methods.*

	***dim* = 8**			***dim* = 10**			***dim* = 12**			***dim* = 14**		
Method	**#P**	**#DE**	**#DR**	**#P**	**#DE**	**#DR**	**#P**	**#DE**	**#DR**	**#P**	**#DE**	**#DR**
SSS	17	17994	6141634	24	26391	6393097	34	35721	6623490	60	62976	6915951
Proposal	15	15737	6394202	23	24380	6657667	33	34686	6717067	44	45683	7025895

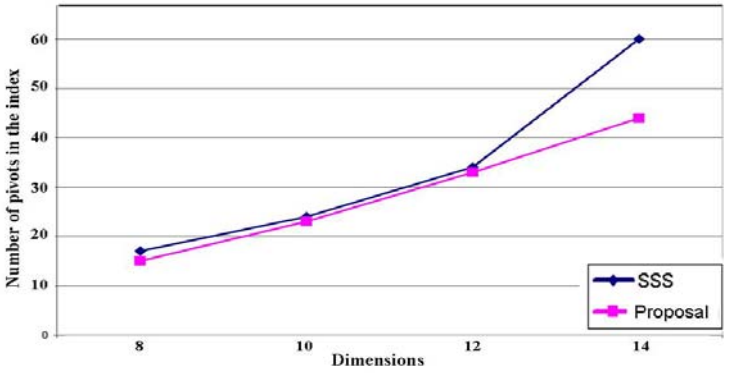


***Figure 2:** Number of pivots, depending on dimensions 8, 10, 12 and 14.*

Table 2 shows that the number of pivots used with our proposal is always lower than in the implementation of SSS, highlighting a marked difference in *dim*=14, with 16 pivots less. In the remaining dimensions, the difference is little, but it remains at most 2 pivots less in our favour. In Figure 2, it can be noted that our proposal has a minor number of pivots. This is an important result, because the strategy of pivots selection of SSS presents a similar efficiency to other more complexes methods and the number of pivots that it selects is close to the optimal number for other strategies.

Figure 3 shows the amount of distance evaluations. The number of evaluations for our proposal always remains below, and, in general, with a uniform linear growth when the size increases. Except in *dim*=14, where SSS shows a slight growth with the amount of reviews, with a difference of about 17,000 reviews, in other dimensions the difference never exceeds 3,000.

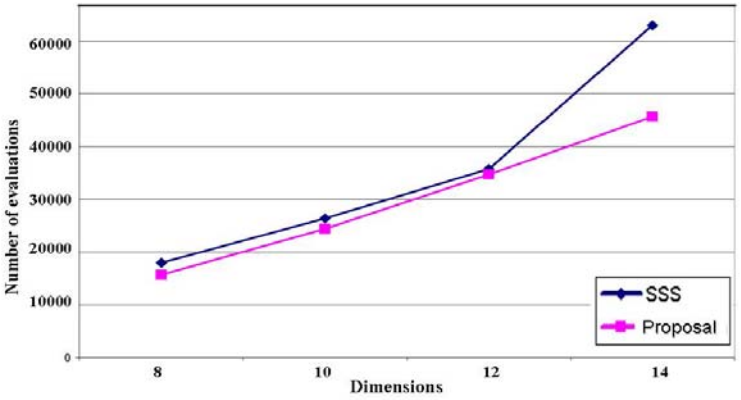


Figure 3: *Distance functions evaluated, depending on dimensions 8, 10, 12 and 14.*

As results exposed in [8], the number of evaluations of the distance function in SSS is always around to the best result obtained with pivot selection techniques and strategies proposed in [10]. In conclusion, the proposal here presented, makes a number of distance evaluations similar to the best results obtained in previous works, even using a smaller number of pivots, which clearly implies space savings.
Figure 4 shows that the use of our proposal in searches obtains a greater number of discriminations by pivots, in all dimensions where it was experienced.

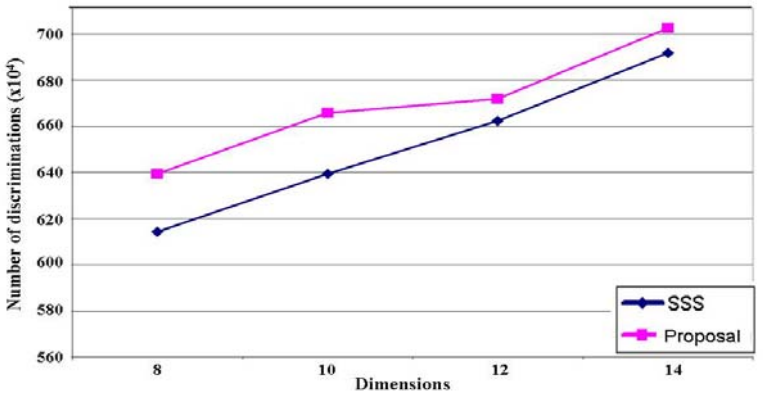


Figure 4: *Discrimination carried out, depending of dimension 8, 10, 12 and 14.*

In the first two dimensions, there is a major difference between numbers of discriminations made. A great performance with the selected pivots with our selection policies, in contrast with pivots selected using pure SSS, is showed. This is because, with the time, the proposal makes an adjustment of pivots, and they make better discrimination, reducing the amount of computations of

the distance function at query time. Thus, it shows that both selection policies of pivots (incoming and outgoing) are good, and that maintains the dynamism of the index.

About parameter α. The value of parameter α determines the number of pivots. Values between 0.35 and 0.40, depending on the dimensionality of the collection, are recommended [8]. Here we decided to use values of α from 0.32 to 0.54, in order to evaluate the number of pivots in the index, since a rise in α represents a reduction in the number of pivots and this is noted better in spaces of higher dimension. In dimensions 2, 8, 10, 12 and 14, α varies in the range shown above.

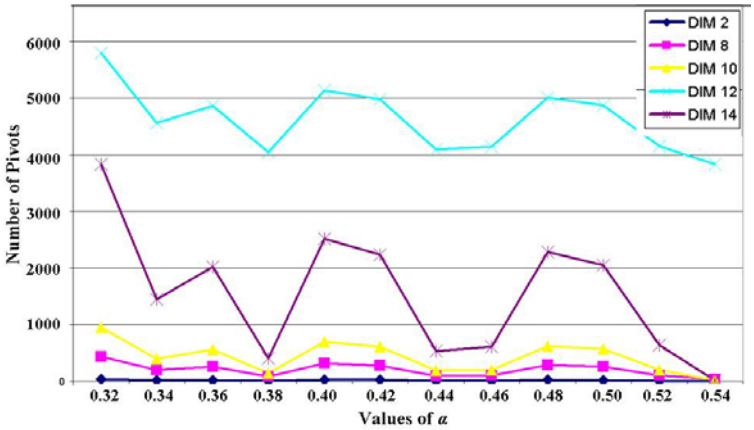


Figure 5: *Number of pivots selected by the parameter α .*

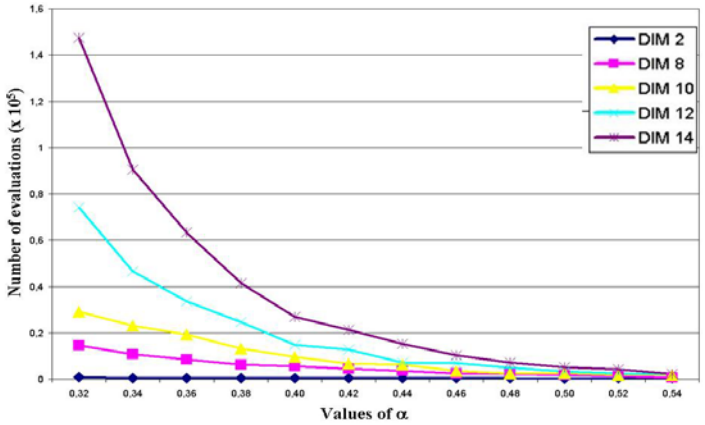


Figure 6: *Evaluations of the distance function according to the parameter α .*

As shown in Figure 5 for all dimensions, the number of pivots varies with α, with some local maximum and minimum and large amplitude in greater

dimensions. Then the observed value 0.50 decreased the number of pivots, as expected. Figure 6 shows the number of distance evaluations, varying α, in order to analyze the impact of our proposal in searches. As seen, in all dimensions there is a consistent behaviour, which gradually begins to decrease when α increases. This is because when distance between pivots increases, the required distance function evaluations decrease. This value has a uniform behaviour because values of 20 periods were averaged. The early periods have largest number of assessments and with the passing of periods, pivots were adapting to searches. In addition, it achieves more discrimination when α increases, because with the passing of periods, in our proposal, pivots are adjusted and searches are improved, discriminating more elements and decreasing the computations of distance functions.

5. Conclusions

This paper presents a new indexing and similarity search method based on dynamic selection of pivots. One of its most important features is that it uses SSS for the initial selection of pivots, because it is an adaptive strategy that chooses pivots that are well distributed in the space to achieve greater efficiency. In addition, two new selection policies of pivots are added, in order to the index suites itself to searches when it is adapted to the metric space. The proposed structure automatically adjusts to the region where most of searches are made, to reduce the amount of distance computations during searches. This is done using the policy of *'the most candidate*' for the incoming pivot selection, and the policy of *'the least discriminates'* for the outgoing pivot selection. Future work is to evaluate the performance of this proposal with real metric spaces, such as collections of texts or images. Moreover, from the results of experiments, to implement algorithms that work with indexes in secondary memory can be proposed. An improvement to selection policies would be to use a data warehouse for training the index with historical search data.

References

1. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J. L., Searching in Metric Spaces, ACM Computing Surveys, 33(3), 2001.
2. Burkhard, W. A., Keller, R. M., Some approaches to best-match file searching, Communications of the ACM, 16(4), 1973.
3. Baeza-Yates, R. A., Cunto, W., Manber, U., Wu, S., Proximity matching using fixed-queries trees, in M. Crochemore and D. Gusfield (editors), Proc. of the 5th Annual Symposium on Combinatorial Pattern Matching, LNCS 807, 1994.

4. Chávez, E., Navarro, G., Marroquin, A., Fixed queries array: a fast and economical data structure for proximity searching, Multimedia Tools and Applications (MTAP), 14(2), 2001.
5. Yianilos, P., Excluded middle vantage point forests for nearest neighbor search, in 6th DIMACS Implementation Challenge: Near Neighbour searches ALENEX'99, 1999.
6. Vidal, E., An algorithm for finding nearest neighbor in (approximately) constant average time, Pattern Recognition Letters 4, 1986.
7. Micó, L., Oncina, J., Vidal, R. E., A new version of the nearest neighbor approximating and eliminating search (AESA) with linear pre-processing time and memory requirements, in Pattern Recognition Letters, 1994.
8. Pedreira, O., Brisaboa, N. R., Spatial Selection of Sparse Pivots for similarity search in metris Spaces, in 33nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'07), LNCS vol: 4362, Springer, 2007.
9. Pedreira, O., Fariña, A., Brisaboa, N. R. and Reyes, N., Similarity search using sparse pivots for efficient multimedia information retrieval, in The Second IEEE International Workshop on Multimedia Information Processing and Retrieval, 2006.
10. Bustos, B., Navarro, G., Chávez, E., Pivot selection techniques for proximity search in metric spaces, in XXI Conference of the Chilean Computer Science Society, IEEE Computer Science Press, 2001.

IV

Architecture, Nets and Operating Systems Workshop

Quality of Service and Availability in a Full Mesh WAN using IP/MPLS. Case Study: The Network at the Department of Justice in Argentina

ANTONIO CASTRO LECHTALER [2,3], PATRICIA CROTTI [3], RUBÉN JORGE FUSARIO [2,3], CARLOS GARCÍA GARINO[1], JORGE GARCÍA GUIBOUT[1].

[1] Instituto Tecnológico Universitario, Universidad Nacional de Cuyo, Mendoza;
[2] Escuela Superior Técnica-IESE;
[3] Universidad Tecnológica Nacional, Buenos Aires, Argentina.
{Antonio Castro Lechtaler acastro@iese.edu.ar, Rubén Fusario rfusario@speddy.com.ar, Patricia Crotti pcrotti2002@yahoo.com.ar, Carlos García Garino cgarcia@itu.uncu.edu.ar, Jorge García Guibout jgarcia@itu.uncu.edu.ar}

Abstract. *At the Argentine Department of Justice (Ministerio Público Fiscal de la República Argentina), a project is developed to set up a WAN with national coverage, quality of service, and IP/MPLS/VPN links. The project focuses on quality of service, a direct result from the selected technology. In previous work, studies explored these issues. Hence, the project enabled the implementation of the examined theory. The advantages of MPLS technology are evaluated to determine quality standards in VoIP, videoconference, mission critical applications, e-mail, web access, and others. Other aspects of LANs are analyzed to obtain an availability level compatible to the required needs. The Project include practical recommendations to adequate the LAN to the desired availability levels.*

1. Design and Implementation of the Network at the Argentine Department of Justice (MPF).[1] Background

The Project to organize communications adequately at the Argentine Department of Justice (MPF) began in 2006. Its purpose is to link through a WAN all of its offices, taking into account that the District Attorney's Offices and their administration are nodes of this Department.

The MPF is an institution incorporated in the National Constitution[2] in its last reform. Organically, it consists of the Department of Justice and the Office of the Attorney General.

[1] MPF stands for Ministerio Público Fiscal (Argentine Department of Justice)

[2] The 1994 Reform established that the Department of Justice become one of the authorities of the Argentine Government – along with the Executive, Legislative, and Judiciary Powers– as an independent entity with functional autonomy and financial autarchy (art. 120).

The MFP is divided in 16 jurisdictions distributed along the entire country: Bahía Blanca, Buenos Aires, Comodoro Rivadavia, Córdoba, Corrientes, General Roca, La Plata, Mar del Plata, Mendoza, Paraná, Posadas, Resistencia, Rosario, San Martín, Santa Fe, and Tucumán.
The Project is carried out with its own personnel, from the Department of Computer Engineering and Communications under the Department's Technological Accessibility and Upgrade Plan
Its main objective consists of linking 350 District Attorney Offices and other dependencies of the MPF around the country with its own network, providing VoIP, data and video transmission, and institutional applications with an adequate level of information security.
After a preliminary survey carried out in 2007, the following assumptions are established for the network design:

- Ensure that the chosen technologies could remain current in the long run.
- Use Category 6 LAN Ethernet Networks.
- Divide the development areas in two: AMBA[3] and the rest of the territory.
- Implement LANs with local companies.
- Select computer technicians by jurisdiction for maintenance and follow-up.
- Implement a Network Operations Center: Secure Room.
- Provide the LANs with analogous equipment: switches, servers and UPS.
- Use a public transmission network for the WAN, working with IP/MPLS links; thus ensuring quality of service.
- Monitor and control the network in a centralized manner.
- Implement an effective and centralized security policy.
- Minimize maintenance costs and decrease reparation times.
- Maximize reliability and network availability.

The Project involves hiring specialized human resources and technical personnel as well as the purchase of technology, hardware, and the contracting of services from telecommunications vendors through open bids and current purchase regulation.
In 2008, the MPF Network Implementation Plan was launched. By March 2009, the Plan consists of:

- 350 linked D.A. Offices.
- 130 links through IP/MPLS/VPN networks.
- 3.500 network terminals.
- 130 LANs.
- 17 Videoconference equipment sets.
- 150 Servers.
- VoIP service in all D.A. Offices.

[3] Área Múltiple Buenos Aires (AMBA): Consists of the City of Buenos Aires and a significant part of its suburban area, called Gran Buenos Aires.

- Central Operations Control Room and "storage" system: located in Avenida de Mayo 760, including a Backup Processing Center.
- Massive storage system with a backup processing center.
- User support for institutional applications, network, and telephone services.
- Institutional applications for the management of the Judiciary Power in Criminal Law.

2. MPF Network Architecture

MPF WAN[4] architecture is based on protocols IP-MPLS [1] [2][5].

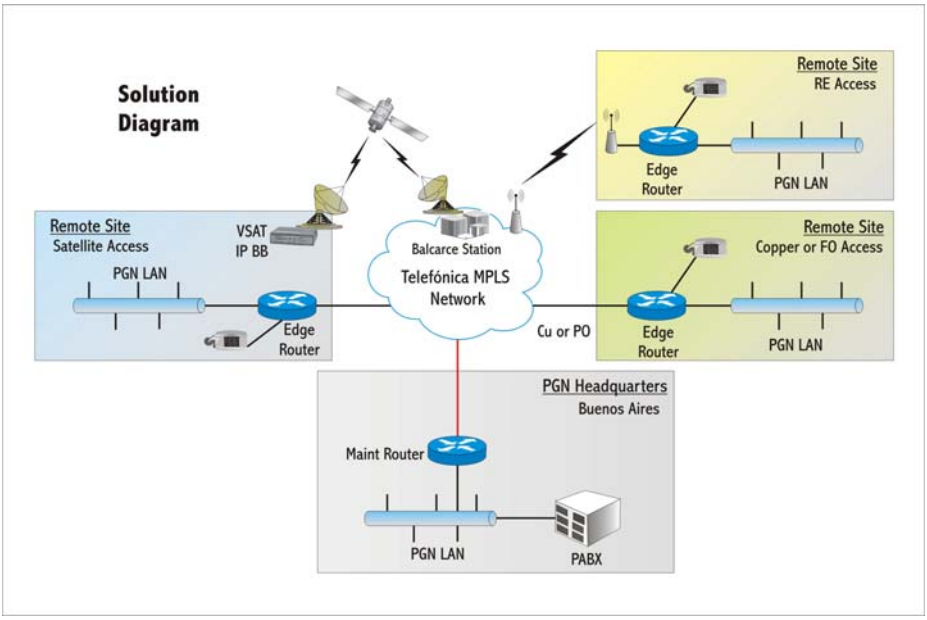


Figure 1: *MPF Network Architecture*

LAN is designed with protocols 802.3 for 100 Mbps, with structured wiring following norm ANSI/EIA/TIA 568, category 6. In the future, this design enables upgrades, up to 1 Gbps. Figure 1 shows the network architecture.

Internet connection is centralized in the MPF Central Office,[6] and is distributed through the MPF network to 130 destinations.

The network enables VoIP communication among all MPF offices using three voice channels per office.

4 Wide Area Network.
5 Internet Protocol - Multiprotocol Label Switching.
6 Located in Avenida de Mayo 760, Buenos Aires.

Data and voice transmission is full mesh for the network nodes, as well as videoconference among the head offices and the sixteen Prosecutor offices.
The network has a structure of terrestrial physical fiber optic links, copper and/or digital microwave, linking the different nodes and following transmission speeds ranging from 512 Kbps to 4 Mbps. The link between the Buenos Aires central node with the IP/MPLS network of the Internet provider is 100 Mbps.
The basis for this architecture relies on MPLS [3] performance, granting:

- An adequate security level.
- Bandwidth on demand.
- Access to the network from any place with Internet access.

Furthermore, the architecture also offers the following convenient features:

- "Full mesh" architecture.
- Quality of Service: Using MPLS the network controls the quality of service.
- Voice, data, video, and mission critical application transmission capabilities.
- Cost reduction in telephone services, videoconferences, and others.

Among the main services provided, we can mention:

- Multimedia: voice and video, and video streaming.
- VoIP.
- Reliable safeguarding of MPF information.
- Secure Access to internet through firewalls.
- Access to databases and applications from other government offices.
- Videoconference
- Institutional electronic mail

Given the required characteristics for the network traffic, the services provided are:

- Multimedia Traffic. Minimum delay and jitter. In addition, delayed packets are deleted.
- Videoconference Traffic. Similar to multimedia traffic but with less priority.
- High Priority Traffic. Sensitive to Time Out, for critical applications.
- Normal Priority Traffic. For the transmission of files, databases, and systems in general.
- Low Priority Traffic. Used in electronic mail and internet.

3. Features of the Network Level of Service with Internet Access

Service provides Internet access with a transmission speed of 40 Mbps and upgrading capabilities. Transmission is digital, with no tolerance for analog lines.
Internet access is through the Central Office. From that point, service is granted to all other offices through the WAN of the MPF.
Total bandwidth is distributed between domestic and international access, assigning it dynamically to ensure a "Committed Information Rate" (CIR %) for both cases, according to the following ratios:

$$CIR_{NAC}(\%) \geq CIR_{NAC(\text{min}imum)} = \frac{BW_{NAC(\text{min}imum)}}{BW_{TOTAL}};$$

$$CIR_{INT}(\%) \geq CIR_{INT(\text{min}imum)} = \frac{BW_{INT(\text{min}imum)}}{BW_{TOTAL}}$$

$BW_{TOTAL:}$ Transmission speed of each link

$BW_{NAC(minimum)}$ and $BW_{INT(minimum)}$: Desired Domestic and International bandwidth. Their sum never exceeds the total bandwidth (BW_{TOTAL}).

The Network guarantees a $CIR_{NAC(minimum)}$ (%) = 95% and $CIR_{INT(minimum)}$ (%) = 95%.

The network also complies with the following features to provide an adequate quality of service:

- Minimum link availability of 99,7 % –measured annually– and 99,5 % if measured per trimester.
- BER = 10^{-7}.

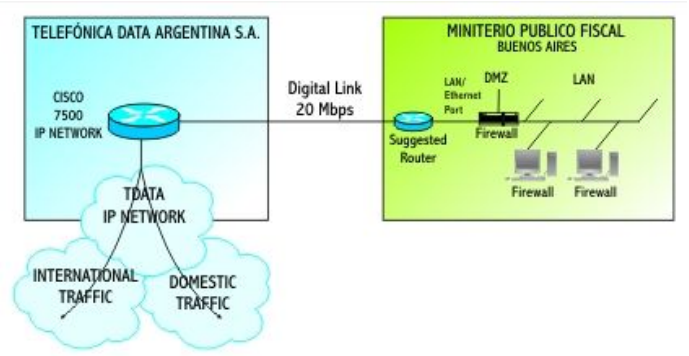


Figure 2

- Average Minimum Time between Failures – MTmBF, maximum per month: 30 hours.
- Minimum Time between Failures – TmBF, maximum per month: 15 hours.
- Maximum Time for Restoration of Service – TMRS, less than 2 hours per month.

Parameters used in the estimation:

- Average Minimum Time between Failures- MTmBF:
  Constant defining the average tolerance between two successive failures. MTBF > MTmBF must hold
  Where:
  Minimum Time between Failures– MTBF is defined as

$$MTBF = \frac{\sum_{1}^{n} TBF_i}{n}$$

n = occurrence of failures in a month.

TBFi = Time elapsed between failure (i) and failure (i-1).

- Time between failures - TBF:
  Constant defining the time between two consecutive failures. TBFi > TmBF must hold, where:

$$TBF_i = \left(FT_i - FT_{(i-1)}\right)$$

- Minimum Time between failures (TmBF).
  Constant defining the minimum tolerance between two consecutive failures.
- Maximum Time for Restoration of Service – TMRS.
  Constant defining the maximum tolerance for the restoration of service.

4. Performance Trials

4.1. Connectivity Test

Round trip times between a single station connected to the router at the Central Office and the sites detailed in Table 1 are tested. The resulting times are under 700 [ms] for international sites and 300 [ms] for domestic sites.

List of Sites	
uc.cache.nlanr.net	www.cisco.com
ns.uu.net	www.vend.org
www.fedworld.gov	www.presidencia.gov.ar
www.mre.gov.br	www.ibge.gov.br
www.sebrae.com.br	www.presidencia.cl

Table 1

Tests run through ICMP (PING) with packets 1024-byte long, sent in three separate opportunities, ten packets each.

4.2. Bandwidth Test

The following test is required:
The sum of FTP transmission rates between a single station connected to the access router at the Central Office and the sites detailed in Table 2.

List of Sites	
ftp.netscape.com	ftp.oracle.com
ftp.sco.com	ftp.freebsd.org
ftp.openbsd.org	ftp.sun.com
ftp.hp.com	ftp.cheyenne.com
ftp.conexion.com	

Table 2

The sum should are not less than 90% of the nominal available bandwidth in the channel setup by the provider, transferring files of at least 7 MB.

4.3 Level of Service

The following features are considered to determine Quality of Service [4] [5]:

- Guaranteed Delivery.
- Transmission Error Recovery.
- Congestion Control.
- Guaranteed bandwidth.
- Flow control.

These basic principles vary when facing the need to implement mail, chat, VoIP, and multimedia. The latter require other control parameters. Thus, the following features are considered:

- Delay: Period of time in transit through a path, where the amount of jumps involved becomes a significant parameter.
- Jitter: Difference in the delay through a path, originated by network instability.
- Packet Loss: Should be restricted.
- Priority: To ensure further the adequate delivery of critical traffic.
- Availability: Percentage of time in which service is operative.

The MPLS Network has three levels for quality of service:

- Level 1: Best Effort type for Internet and e-mail traffic.
- Level 2: Reliable data traffic, for MPF institutional applications.
- Level 3: Multimedia, voice, and video traffic.

Guaranteed values in the following parameters are determined to achieve the desired quality of service:

- Packet Loss.
- Delay or Latency.
- Jitter.

Based on these criteria, the parameters listed below are established:

- WAN IP/MPLS/VPN link availability of 99,5 % measured annually and 99.2% monthly.
- BER = 10^{-7}
- Average Minimum Time between Failures – MTmBF, per month: 40 hours.
- Minimum Time between Failures – TmBF, per month: 30 hours.
- Maximum Time for Restoration of Service – TMRS, per month: less than 4 hours in AMBA and less than 6 hours in the rest of the country.

Real time reporting is available to accomplish effective control. The system reports:

- Services (use of service graphs, traffic)
- Failures: list of failures, beginning date and time, final date and time, affected service (link), date and time of failure notice, failure origin, comments.
- Line use: line use percentage in bps, incoming and outgoing frames, compared to the total available bandwidth.
- Availability: Percentage of time in service, disaggregated per service.

  tTS = Total time of Service.
  tSE = Total time of Effective Service.
  tTI = *tTS* - *tSE* (Total Unavailability Time).

$$\text{Availability}(\%) = \frac{\text{tSE}}{\text{tTS}} * 100$$

5. Concluding Remarks

Based on the study of the network architecture developed at the Argentine Ministry of Justice, the following key features are determined for network quality of service:

- Full Mesh design in the architecture, IP / VPN / MPLS, enabling the delivery of a wide range of service based on IP, in theoretical range of 56 Kbps to 40 Gbps.
- Precise parameter definition to require the most adequate service for the variety of traffic and formats transmitted in the network, and to guarantee service availability to all offices around the country.
- Quality of Traffic ensured to critical mission applications with a WAN packet loss of less than 1%, called “Gold Traffic” by the company providing the services.
- Best Effort non priority data traffic, called “Bronze Traffic” by the service provider, with a packet loss level higher than 1%. The services provided on this quality level are internet surfing and electronic mail.
- Multimedia Traffic with two main services: VoIP and Videoconferencing. The services use the traffic called “Multimedia” by the provider which should guarantee isochrone signal transmission. Thus, packet loss, jitter, and delay need consideration. A packet loss lower than 0.5% is provided. Domestic traffic maximum delay is 200 mseg and jitter less than 30 seconds. International Traffic has a delay up to 300 mseg, and maintains the same value for jitter as the domestic case.
- Quality of service complemented with an appropriate definition of security policies, following standards and norms and their implementation. These policies are available to all network users to better visualize and understand the network.
- Central control room infrastructure with modern security features and equipment protection to permanently guarantee operations at the data processing center of the MPF.
- Availability of the network adequate to the operative needs of the organizations. TMBF availability qualifies the level of failure of the equipment and associated software, The TMRS expresses the average time for each failure reparation.

Based on quality of service, the network at the MPF ensures:

- Efficient and reliable transmission of voice, video, and data.
- MPF data transmission and safe storage.
- Resource optimization based on savings in communications, personnel, and information costs.
- Implementation of management systems which optimize MPF operations in real time.

Finally, this network is the practical implementation supported in previous work developed by this Research Group [6] [7] which focuses on achievable advantages in class and quality of service

6. Acknowledgements

The financial support provided by Agencia Nacional para la Promoción Científica y Tecnológica and CITEFA (Project PICTO 11-0821, Préstamo BID 1726 OC-AR) is gratefully acknowledged and the contribution of Doctor Adrián Marchisio, Secretary of Institutional Coordination to the MPF.

References

1. Chávez, E., Navarro, G., Baeza-Yates, R., Marroquín, J. L., Searching in Metric Spaces, ACM Computing Surveys, 33(3), 2001.
2. Burkhard, W. A., Keller, R. M., Some approaches to best-match file searching, Communications of the ACM, 16(4), 1973.
3. Baeza-Yates, R. A., Cunto, W., Manber, U., Wu, S., Proximity matching using fixed-queries trees, in M. Crochemore and D. Gusfield (editors), Proc. of the 5th Annual Symposium on Combinatorial Pattern Matching, LNCS 807, 1994.
4. Chávez, E., Navarro, G., Marroquin, A., Fixed queries array: a fast and economical data structure for proximity searching, Multimedia Tools and Applications (MTAP), 14(2), 2001.
5. Yianilos, P., Excluded middle vantage point forests for nearest neighbor search, in 6th DIMACS Implementation Challenge: Near Neighbour searches ALENEX'99, 1999.
6. Vidal, E., An algorithm for finding nearest neighbor in (approximately) constant average time, Pattern Recognition Letters 4, 1986.
7. Micó, L., Oncina, J., Vidal, R. E., A new version of the nearest neighbor approximating and eliminating search (AESA) with linear pre-processing time and memory requirements, in Pattern Recognition Letters, 1994.
8. Pedreira, O., Brisaboa, N. R., Spatial Selection of Sparse Pivots for similarity search in metris Spaces, in 33nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'07), LNCS vol: 4362, Springer, 2007.
9. Pedreira, O., Fariña, A., Brisaboa, N. R. and Reyes, N., Similarity search using sparse pivots for efficient multimedia information retrieval, in The Second IEEE International Workshop on Multimedia Information Processing and Retrieval, 2006.
10. Bustos, B., Navarro, G., Chávez, E., Pivot selection techniques for proximity search in metric spaces, in XXI Conference of the Chilean Computer Science Society, IEEE Computer Science Press, 2001.

A CIM Framework for Standard-Based System Monitoring Using Nagios Plug-ins

MARCELO LORENZATI[1], MIRIAM ESTELA[1], RODOLFO KOHN[1]

[1] Intel Software Argentina, Software and Services Group,
Corrientes 122, Piso 2, Garden Center,
5000 Córdoba, Argentina
{marcelo.lorenzati, miriam.estela, rodolfo.kohn}@intel.com

Abstract. *The Common Information Model is a widely-accepted industry standard to model distributed system objects as well as their behaviors and interactions to realize system management tasks. It is endorsed by the Distributed Management Task Force and appears as the preferred manageability solution to deal with the ever increasing heterogeneity characterizing today's datacenters. However, a number of enterprise-class system management products, like Nagios, are not compliant with this standard. Nagios is among the top open source monitoring tools with the power of a large community of developers producing plug-ins to manage a variety of enterprise systems. As part of the endeavor to accelerate CIM adoption, an extension framework, called Plugin Extension for CIM, has been developed in order to expose Nagios and other third-party plug-ins thru CIM, thus enhancing the capabilities of standard-based system management tools by the transparent use of the extensive variety of existing plug-ins. This paper describes the developed framework as well as its acceptance within the open source manageability community.*

Keywords: *Manageability Standards; Monitoring; CIM; Nagios; WBEM; Plug-ins.*

1. Introduction

The ever increasing datacenter complexity seems to continue raising system management costs during the next years, dramatically incrementing their weight in the whole of IT expenditures. IT infrastructures need to support dynamic business processes relying on distributed data and distributed applications usually exposed as web services to the Internet and often subject to extremely variable and unpredictable workloads. These business processes are part of services that have to be delivered within the limits imposed by service level agreements and violations turn into monetary penalties to companies. While carrying out a number of tasks as change management, patching, system fault diagnosis and healing, or intrusion prevention and detection, IT professionals have to assure services scale out and are executed as expected without affecting availability and reliability [1]. Furthermore, a datacenter infrastructure includes tens or thousands of hardware and software resources of different, and sometimes incompatible, technologies and vendors

thus reducing or completely eliminating interoperability between devices, applications, and networks. This heterogeneity is one of the most significant factors adding to distributed systems management complexity.

In order to deal with the growing heterogeneity and achieve real integration of management information, the Distributed Management Task Force [2] embraced the Web-Based Enterprise Management (WBEM) [3] specification that includes the Common Information Model (CIM) [4] and various interoperable Internet technologies for distributed systems management [5]. CIM is an industry-accepted standard specification to model management information in distributed systems and to provide standard mechanisms, through CIM profiles, to realize a number of essential management tasks.

Notwithstanding, there are various renown system management products providing limited support of CIM if any, both in the open source and non-open source worlds. Among them, Nagios [6] is one of the most widely-used open source monitoring tools in the IT industry. Its power mainly lies on both its plug-in-based architecture and the huge community of developers devoted to the creation of a vast number of plug-ins providing a broad range of management capabilities. However, it relies on a proprietary approach to expose these manageability plug-ins to the core monitoring engine.

As an effort to contribute to the adoption of CIM and enhance the potential of standard-based monitoring tools, we developed a CIM extension framework that allows a CIM broker to expose Nagios, and other third-party, plug-ins through a CIM interface. Thus standard-based monitoring tools can transparently benefit from thousands of new manageability features in a matter of seconds by configuring and deploying already existing plug-ins that instrument the system and generate alarms whenever certain measured values are outside the specified ranges, usually because of an error state.

The framework consists of a CIM extension schema, a configuration script, and a generic plug-in indication provider. The script is run at the beginning to configure the parameters used to execute each plug-in that is to be run in the monitored system. The indication provider is the software that executes the requested plug-in with the appropriate parameters whenever a CIM client subscribes a listener to receive the corresponding indications. The provider parses a filter embedded in the indication subscription, based on this filter reads the relevant plug-ins and their configuration parameters, and executes each plug-in with certain frequency defined among the parameters. As in the case of Nagios, a plug-in can be written in any language including scripting languages. When a plug-in detects a problem, the indication provider converts the output describing a warning or critical alarm into a CIM indication that is routed to the listener specified in the subscription. The proposed extension schema defines a set of classes necessary to carry out the mentioned activities.

The complete framework is called “Plugin Extension for CIM”. It is currently offered as a downloadable manageability package, called sblim-cmpi-pec, belonging to the open source project Standards Based Linux Instrumentation (SBLIM) [12]. This is an umbrella project run by IBM that encompasses a collection of system management tools to enable the implementation of WBEM in Linux. SBLIM packages are shipped within various Linux distributions such as SuSE.

Other efforts have been carried out in the industry to integrate the best of existing system management tools with WBEM standards. As an example, it is possible to mention the work done by Novell Inc. to expose statistical data obtained by Ganglia Monitoring System about data center systems thru the CIM statistical model. As of today, this seems to be the only project that integrates Nagios plug-ins to be exposed thru CIM and intends to enable Nagios engine to use CIM.
This paper describes the whole framework developed, its advantages for datacenter management, its contribution to industry standards adoption, and future work that includes more features and follows the converse way permitting Nagios core engine the transparent use of CIM-based functionalities. Section II provides a brief introduction to the main technologies involved, section III describes the framework implementation, section IV explains further work necessary, and finally the conclusions are stated emphasizing the contributions to the industry.

2. Technology description

2.1. The Common Information Model (CIM)

The Common Information Model is a standard to represent distributed system management information and define the objects and interactions required to realize specific system management tasks.
CIM is an object oriented model described with UML that includes classes, properties, methods, indications, associations and allowing subclassing for extension, but enforcing basic object hierarchy, abstraction and encapsulation. The syntax language to define the elements of CIM in a text format is Managed Object Format (MOF) which is based on Interface Definition Language (IDL).
This model provides a common definition of management information for systems, networks, applications and services, and allows for vendor extensions. The CIM's common definitions enable vendors to exchange semantically rich management information between systems throughout the network.
The CIM is made of two parts. The first one is the CIM Specification, which defines the language (syntax and rules) and the proper methodology for describing the management data. It also defines the CIM meta schema, its elements and the rules of each one of its elements.
The second part is the CIM Schema, which provides the classes and associations and its attributes as a complete Framework. It is comprised by the Core Model, Common Models and the extended Schema.
The Core Model defines all the concepts that are applicable to all the areas of management, and the Common Models are various models that define concepts that although they are common between a certain area of management, they are still independent of the implementation, technology or manufacturer.

Finally, the Extension schema expresses the technology specific concepts represented by classes, methods, attributes and associations, extending the common and core models.
Figure 1 shows part of the Core Model and Systems Model and two classes extending the schema.

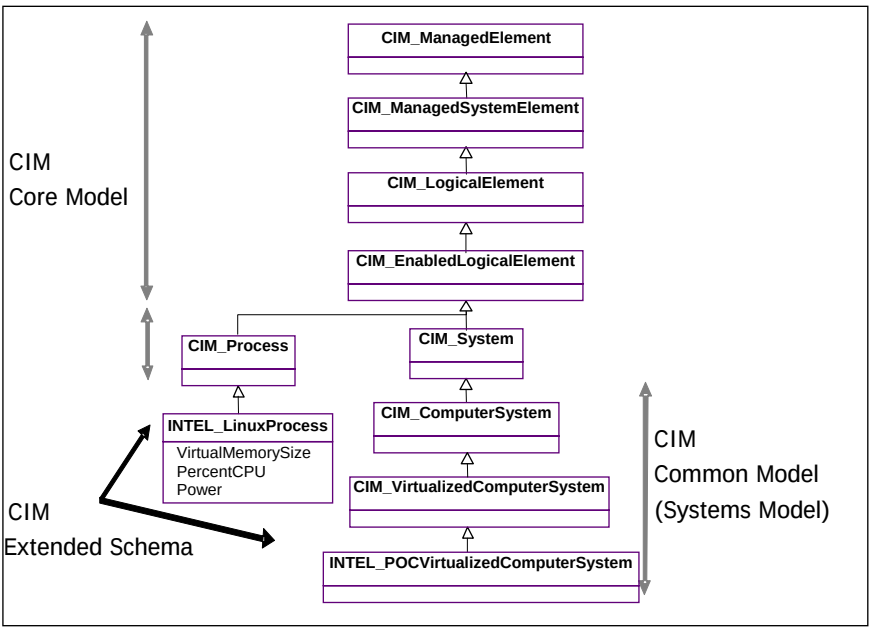


Figure 1: *A view of the Core Model and System Model hierarchies with two classes extending the CIM schema.*

Figure 2 shows a CIM server consisting of a CIM Object Manager (CIMOM) or CIM Broker that interfaces with a client through a CIM-XML adapter and an indication handler. It receives requests from a client and has an interface with the providers to perform the required actions. There are various CIM broker implementations like SFCB [7], openWBEM[9], OpenPegasus[10], and Microsoft's WMI [11].
CIM Providers instrument the system for manageability. There are different types of providers: Instance, Association, Property, Method and Indication Provider. The CIMOM invokes providers in order to realize management tasks on the CIM models.
The Common Manageability Programming Interface (CMPI) is a standard interface used between the CIM broker and a provider, so that a CMPI-compliant provider is portable through different CIM brokers.

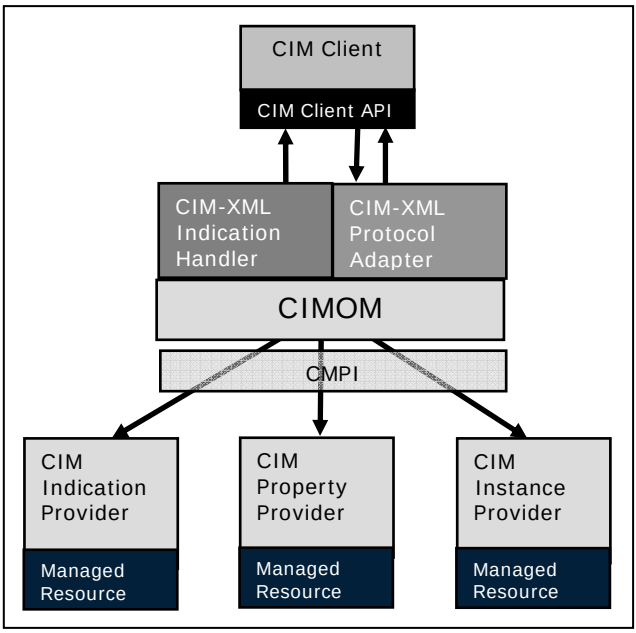


Figure 2: *A CIM Server implementation.*

CIM defines the concept of indications in order to report changes in the state of the environment. For any event in the system, a CIM indication may be generated by the corresponding CIM Indication provider developed to monitor a specific aspect of the system.
When a client needs to receive notifications for specific types of event, it has to send a subscription for the corresponding indications specifying the location of a listener that is to receive any indication generated from the mentioned subscription. This subscription may also include a filter in order to refine the scope of the events that need to be notified.
There are 3 main types of indications: CIM_InstIndication, CIM_ClassIndication, CIM_ProcessIndication.
The indication provider developed to execute Nagios plug-ins generates a sub-type of process indications called alert indications.

2.3. Nagios

Nagios is a widely used open-source monitoring tool for host systems and network services. It is built following a server/agents architecture. Usually, on a network, a Nagios server is running on a local host, and plug-ins are running on all the remote hosts that need to be monitored. These plug-ins send information to the server, which displays it in through a Graphical User Interface (GUI). It allows administrators to manage complex infrastructures

by monitoring the overall status trough a web browser interface and subscribe to alerts notifications trough email, instant message and SMS when problems happens.
Two basic monitoring entities are defined by Nagios: the host that is the physical or virtual asset to be monitored; and the services that are the real monitored instances or processes running on a given host. Nagios call-back mechanism is one-way, informational-only.
The Core building block is formed by 3 parts:

- The scheduler is the server part of Nagios. At regular interval, the scheduler checks the plug-ins, and according to their results, it performs some actions.
- The Nagios GUI: it is displayed in web pages..
- Plug-ins: small programs (in Perl, C, java, python among others) that check a service and return a result to the Nagios server.

When a plug-in is executed, it generates data sent to the core engine which processes them and runs event-handlers notifications depending on the case. The plug-in returns a value and a small line of text Plug-in output according to Nagios specifications. The plug-in must return one of a set of possible values (ok, warning, critical) and it must print information text to the standard output. Because the plug-ins are configurable by the user, their output depends on the parameters, options and the check that they carry out.

3. Exposing Nagios plug-ins through CIM

3.1. Framework Description

Plugin Extension for CIM (PEC) consists of a CIM extension schema, a configuration script, and a generic plug-in indication provider.
Each Nagios plug-in is executed passing specific options and arguments to obtain certain functionality. The options and arguments are edited in a configuration file to be read by the configuration script which populates the PEC schema in the CIM database. This configuration script uses the CIM client library SFCC [8].
The PEC indication provider is the software that, when the appropriate CIM subscription arrives, executes the requested plug-ins with the appropriate options and parameters read from the database, and sends the corresponding indication depending on the plug-in output.
The provider executes each plug-in in a different thread. A thread executes the plug-in as a forked child process, at a configured frequency. The plug-in's output is read through an unnamed pipe. As a new process is forked to run a plug-in, plug-ins can be developed in any programming or scripting language.
The provider converts plug-in WARNING and CRITICAL output into CIM indications, with the appropriate severity levels, that are routed to the listener specified in the subscription.

The CIM extension schema defines a set of classes and associations necessary to carry out the mentioned tasks.

3.2. Extended PEC schema

In the proposed solution, the extended model consists mainly of four classes as shown in Fig. 3.

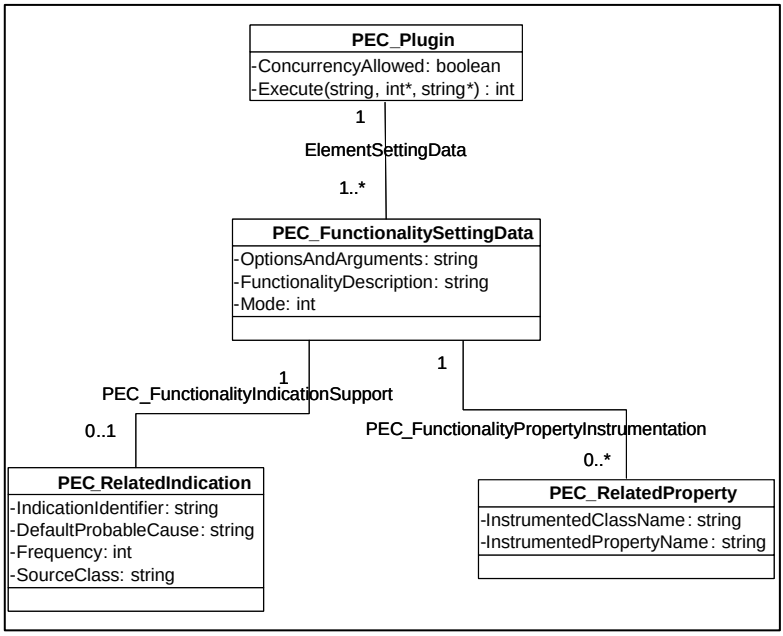


Figure 3: *Extended PEC model.*

The class PEC_Plugin represents the plug-in in the system, where its main method is Execute. Each plug-in could have one or more functionalities depending on the options and parameters configured in PEC_FunctionalitySettingData. The plug-in can be used to monitor the system through the PEC indication provider or to obtain a specific property value for another object. In the first case the functionality is associated to an indication defined in PEC_RelatedIndication. In the second case, the functionality is associated to a class property specified in PEC_RelatedProperty. The associations between these classes are represented by ElementSettingData, PEC_RelatedIndication, and PEC_FunctionalityPropertyInstrumentation. The usage of plug-ins to monitor the system and generate indications is the one implemented at the moment of writing this paper. The functionality to populate properties has not been implemented yet.

Figure 4 shows how these classes are connected to the CIM core and common models. PEC_Plugin actually inherits from PLUGIN_Executable which in turn inherits from CIM_SoftwareElement in the CIM Application model. PEC_FunctionalitySettingData inherits from CIM_SettingData in the core model and the other classes inherit from CIM_ManagedElement in the core model. Both figures 3 and 4 depict the attributes and methods of each class and association.

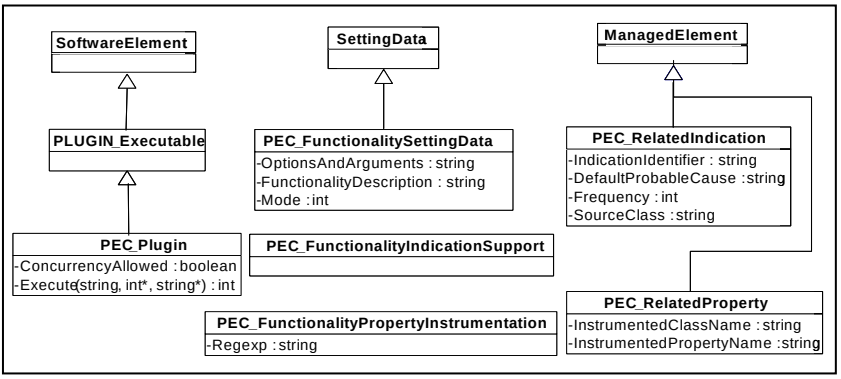


Figure 4: *Class hierarchy showing how the schema extends by inheritance the core and common models.*

3.3. CIM-based Monitoring with Nagios plug-ins

The indication provider is registered to the CIM Object Manager through the compilation and installation of the PEC project with the GNU autotools (.configure make and make install). For our tests, the CIM Object Manager SFCB (Small Footprint CIM Broker) was used but since our plug-in has been developed according to the Common Manageability Programming Interface specifications, it is portable through any other CIM Object Manager.
The Nagios plug-ins to execute have to be deployed in the monitored system, usually storing them in a specific directory. The provider will require configuring the path to the Nagios plug-in by setting an environmental variable (PEC_PLUGIN_PATH) in order to execute the selected plug-ins.
The CIM repository has to be populated as explained above through the configuration file and scripts. The scripts connect to the CIM broker through the library SFCC.
Once the repository has been populated the indication provider can be executed when the appropriate subscription is received by SFCB.
In order to subscribe to indications, a client reference tool has been developed. This client subscribes for indication passing a specific filter. These filters can include logical operators as AND, OR and NOT, also allows to apply arithmetic operators like ‘>’, ‘<’ or ‘=’.

The filter can only include the properties IndicationIdentifier and SourceClass that can be found in the class CIM_AlertIndication and PEC_RelatedIndication.
When a subscription is received the indication provider gets the appropriate plug-ins according to the filter received in the subscription. Then it reads the configuration data for each plug-in and executes it with the obtained options and arguments to realize the corresponding functionality.
Whenever the plug-in is run it can generate a WARNING or CRITICAL output depending on the monitored system state and the options passed for execution. In this case, the indication provider sends the configured indication number to the subscribed listener.

4. Future Work

At this moment the components to monitor a system, using Nagios plug-ins with CIM-based tools, have been developed. The package can be downloaded from SBLIM web site. Further work is necessary to implement the components to populate CIM class properties using third-party plug-ins. This would involve creating the appropriate libraries that would need to be linked into the corresponding CIM class providers which are to call the corresponding function wrapping a plug-in whenever the appropriate request is received.
Additionally, this work should be extended to enable a Nagios engine to use CIM-XML or other WBEM protocol, such as WS-Management, to access CIM elements. This work would have advantages like the capacity to take CIM-based actions as a response to an alarm, to seamlessly reuse CIM providers for monitoring, and using standard web-service interfaces that would have benefits such easier network boundaries traversal.

5. Conclusions

This paper describes a work that permits extending the capabilities of system management tools that are compliant with WBEM industry standards defined by the Distributed Management Task Force. Using the “Plugin Extension for CIM” package, a standard-based tool can transparently use available Nagios plug-ins and other third-party plug-ins, developed by a huge open-source community, to obtain thousands of new monitoring functionalities with no much effort. Thus, these system management applications can incorporate the monitoring of a large number of server aspects, including hardware and software, in a matter of seconds by just configuring the plug-in usage in a configuration file. After the configuration phase is concluded, these plug-ins can be seamlessly used and indications can be generated when the measured system state is not within the configured range.

As a result of this work, it is easy to combine the benefits of the huge diversity of monitoring capabilities provided by Nagios plug-ins with the advantages of using a standard interface like CIM that allows overcoming the complexity generated by heterogeneous distributed systems.
The "Plugin Extension for CIM" framework has been donated to the open source project SBLIM, run by IBM. It can be downloaded from Sourceforge as a package called sblim-cmpi-pec [12].

References

1. Kephart, J. O. and Chess, D., "The Vision of Autonomic Computing", Computer, vol. 36, No 1, 2003.
2. DMTF, www.dmtf.org.
3. DMTF, Web-Based Enterprise Management (WBEM) standards, Retrieved January 12, 2009 from http://www.dmtf.org/standards/wbem.
4. DMTF, Common Information Model (CIM), Retrieved January 12, 2009 from http://www.dmtf.org/standards/cim.
5. Westerinen, A. and Bumpus, W., "The Continuing Evolution of Distributed Systems Management", IEICI Transaction on Information and Systems, vol. E86-D, No. 11, 2003.
6. Nagios project home page, Retrieved May 5, 2009 from http://www.nagios.org.
7. SBLIM SFCB project home page, Retrieved May 5, 2009 from http://sblim.wiki.sourceforge.net/Sfcb.
8. SBLIM SFCC project home page, Retrieved May 5, 2009 from http://sblim.wiki.sourceforge.net/Sfcc.
9. OpenWBEM project home page, Retrieved May 5, 2009 from http://openwbem.sourceforge.net/.
10. OpenPegasus project home page, Retrieved May 5, 2009 from http://www.openpegasus.org.
11. Microsoft home page, Retrieved May 5, 2009 from http://www.microsoft.com.
12. SBLIM project home page, Retreived May 5, 2009 from http://sblim.wiki.sourceforge.net

I

Innovation in Software Systems Workshop

Biometric identification in electronic voting systems

EDUARDO IBAÑEZ[1], NICOLÁS GALDÁMEZ[2], CESAR ESTREBOU[1], ARIEL PASINI[3], FRANCO CHICHIZOLA[3], ISMAEL RODRÍGUEZ[1], PATRICIA PESADO[4]

[1]Full-Time Assistant in charge of assignments – UNLP
[2]Semi-Full-Time Graduate Assistant Professor – UNLP
[3]Full-Time Associate Professor – UNLP
[4]Head Professor – UNLP. Researcher of CIC Bs. As.
Institute of Research in Computer Science III-LIDI – School of Computer Science – UNLP
{eibanez, ngaldamez, cesarest, apasini, francoch, ismael, ppesado}@lidi.info.unlp.edu.ar

Abstract. *An extension of previous developments of electronic voting and e-government systems carried out at the School of Computer Science of the UNLP is presented, where a digital fingerprint recognition feature is added to the existing system used for faculty elections at this School.*
The characteristics and performance of the biometric recognition system are analyzed, as well as the modification of the on-site electronic voting system used in La Plata (hardware and software) and the adaptation to an Internet voting system that can be used at Regional Centers.
Finally, the generalizations of the use of the technology developed for e-government are discussed and current research and development lines are mentioned.

Keywords: *Biometric recognition, fingerprints, E-government, E-Democracy, Electronic voting, Voting systems, Distance voting, Internet.*

1. Introduction

1.1. E-government and electronic voting

The electronic government, or e-government, arena includes activities aimed at speeding up information management, thus allowing greater control and auditability. Government information systems present a set of distinctive characteristics (for example, they must be very reliable, they are distributed systems, they must respond in real time, etc.) that make their development and administration different from those of traditional systems [1][2].
Electronic democracy, through electronic voting, offers citizens the possibility of continuously participating in political decisions. This participatory form appears in the 1960's, when researchers realized the civic potential that the new electronic technology had. It becomes more relevant as technology evolves and the digital gap is reduced, and is finally massively incorporated to everyday life.

In these last few years, the fast expansion of technology in communication devices, such as mobile phones, PDAs (Personal Digital Assistant), portable computers with mobile connection to the Internet, allow the community to participate in political decisions from any place [3].

1.2. Biometric recognition

A biometric system can be used for the verification (identity certification) or identification of a person (establishing the person's identity). Each technique has its specific characteristics, but they all have two stages: training (digital fingerprints are recorded for all individuals that are entered into the system) and use (the information stored is used to verify the identity or identify individuals).
Unquestionably, the most widely used biometric system is that of recognition through fingerprints: various characteristics from different angles and sectors of the fingers are extracted and stored. Fingerprints do not change (through natural processes) throughout the life span of a person, but they can be altered by wounds, humidity, scars or dirt. Various low-cost devices to allow a general use of fingerprints have been developed [4][5][6].

1.3. Biometric recognition in electronic voting systems

One important aspect that has to be taken into account when using e-voting technology (and e-government in general) is personal identification of voters. The irrefutable identification of voters in real time is a complex goal of biometric recognition [7]. These techniques resort to physiological (face, fingerprints, iris and retina, among others) or behavioral (signature and voice) characteristics of people in order to identify them [8][9].
In Section 5, the incorporation of voter recognition through fingerprints and its integration to the electronic voting system developed at the School of Computer Science of the UNLP is analyzed.

2. Contribution

The extensions to the electronic voting system developed at the School of Computer Science of the UNLP to incorporate a biometric recognition feature (through fingerprints) for voter identification are analyzed, and the necessary hardware and software modifications are considered.
System reliability and response times are studied, including the case or regional centers where the votes cast are transmitted through the Internet.
Finally, e-government applications where the developed technology can be used, beyond the electronic voting application, are presented.

3. Electronic voting systems. Previous experience

III-LIDI has been working on the area of electronic voting since 2003, with several specific experiences, among which the following can be mentioned:

- Software development for electronic voting for faculty and graduate elections in 2003.
- Development of an integral electronic ballot box for the company TESUR for the national elections [10], in accordance with all requirements of the National Electoral Law [14] and reconfigurable to other types of elections (year 2004).
- Base software development for the control of peripherals in the electronic voting machines used in Capital Federal, Argentina (year 2006).
- Development of an integral electronic voting system for student elections at the School of Computer Science since 2007, including industrial, electronic, and software design [11].
- Auditing of electronic voting systems used in the Province of Río Negro, Argentina, and the equipment developed by ALTEC SE. (2007 and 2008)
- Development of remote voting record technology and equipment for the elections carried out in various locations and connected through the Internet with e central counting station (tested for regional centers of the School of Computer Science in 2008) [12].
- Currently, work is being done regarding the evolution of the EV machines developed by ALTEC SE for the Province of Río Negro and the multi-purpose positions developed by III-LIDI for the UNLP, which can be adapted to various e-government applications.

3.1. General description of the electronic voting system developed in 2007

Electronic voting is not just the act of casting the traditional vote using electronic devices. It also provides tools that allow speeding up the operations carried out on the day elections take place, such as voter identity record and verification, vote counting, and the transmission of results to the corresponding organization. Some electronic operations can be combined, whereas others can be done manually, such as manual identity verification and electronic generation of results. Considering that, in general, votes translate into political power, accuracy and quantification quality are aspects that should be particularly considered. Also, there are many electoral security and reliability issues that can be strengthened with technology [13].

In the case of elections for political authorities, the National Constitution and the laws governing the issue [14] (electoral or referendum laws) establish 4 essential requirements or characteristics for voting [1]: *Universal* (all citizens that fulfill a set of conditions must be able to vote, and those who do not fulfill such conditions must not be able to vote), *Equal* (all citizens belonging to the electoral universe must be able to vote only once and all votes have the same

value: one citizen, one vote), *Secret* (the identity of citizens cannot be linked, in any way, to the vote they cast) and *Mandatory* (all enabled citizens have the obligation to vote; compliance with this depends on electoral scope).
Based on the set of conditions mentioned above, a voting structure that has, for each election precinct, a computing equipment (for precinct board members) connected to a voting machine was developed. One of these machines is shown in Fig. 1. Precinct board members are in charge of identifying voters (through the presentation of an identity card with a picture or student picture ID) from the electronic electoral register stored in the equipment. This identification allows transmitting the corresponding authorization to vote to the voting machine. After the vote is cast, the voting machine transmits a signal to the machine used by the precinct board members so that they can continue with the process. Each voting machine has an LCD touch screen, a CPU, a UPS, a thermal printer, a storage ballot box for printed votes, a device that allows viewing the vote and that automatically drops votes into the ballot box, and 2 flash memories where vote counts will be stored. The equipment used for this election does not have an Internet connection.

Figure 1: *Electronic ballot box used in 2007- 2008*

After voting days are over (in the case of the UNLP, elections take place during 3 running days, which adds to the complexity of voting system management), votes are counted in a different machine. Results are transferred through the removable memory devices mentioned above (2 per session, one of them used

as backup) which are stored in a wax-sealed envelope at the closing of each session.

3.2. Electronic voting through the Internet at the regional center

The distance at which the regional center is located complicates the transportation of electronic ballot boxes and the safe transfer of votes to the headquarters of the School of Computer Science. For this reason, as well as the reduced student population at the regional center, it was decided not to use the machines described in 3.1., and instead developing a voting system that can be used through the Internet.
This new ballot box has two pieces of equipment (*Local voting equipment* and *Printing equipment*) connected over a VPN to ensure the integrity of the information sent during the electoral process.
The *Printing equipment* is located at the headquarters, and is in charge of printing the voting vouchers that will be stored in a physical ballot box, as well as updating (in two flash memories) the corresponding counters for the votes cast from the *Local voting equipment.* Additionally, on the screen of this equipment, a notice indicating the reception of a vote from the *Local voting equipment* is shown. This *Printing equipment* is located at one of the two enabled election precincts in the headquarters.
Vote counting was done using the same system used for the ballot boxes located at the headquarters.

4. Election model at the UNLP as from 2010

In 2009, the UNLP has modified its bylaws and as a result, authority elections from this year on have to combine 5 different representations in each School: Professors, Assistants in charge of assignments, Graduates/Auxiliary, Students and Non-teaching staff. Each of these "classes" of voters has different ways of expressing majorities and minorities, which means that, for the application of a general system, programming aspects should be adapted and made more flexible.
Of these 5 groups, we will focus on student elections because this is the most complex group (due to the number of registered voters, voter categories, representation of the majority group and up to 2 minority groups, among other particular characteristics).

4.1. Characteristics to consider for student elections at the UNLP

Student elections at the UNLP take place on an yearly basis and have a duration of 3 days. Ballot boxes are changed daily; at the end of each day, they are wax-sealed and locked in a storage space. After the three voting days are over, the votes in all ballot boxes are counted.

Students vote two classes of authorities: student members (voters are those who comply with regularity conditions), and/or student government association (any student in the electoral register may vote). The electoral register indicates which type of votes students are enabled to cast: STUDENT GOVERNMENT ASSOCIATION ONLY or FACULTY AND STUDENT GOVERNMENT ASSOCIATION. The electoral register is divided in electoral precincts. Ballot papers are divided in two sectors (student faculty authorities – student government authorities), and students can vote sections from different ballots. Those students who appear in more than one electoral register (because they are enrolled in more than one Academic Unit) can vote for the students government association at each Academic Unit to which they belong, but they must chose at the University (if they comply with regularity conditions) one Academic Unit where they will vote for student faculty members.
Electoral authorities are an Electoral Board, and, for each precinct, a President belonging to the Faculty, a graduate and a student.
The School of Computer Science has two regional centers, one at 500 km and the other at 200 km from the headquarters. The students at each regional center should have the possibility of voting under the same conditions as students at the headquarters, in both regional centers, elections last only one day.

5. Incorporation of biometric recognition to the electronic voting system

5.1. Schematic model

Governments are very interested in modernizing their information systems. In general, the use of technology to provide access to management services to citizens, as well as the mechanisms that allow a direct participation of citizens in decision making, requires a verification procedure of the people who have access to sensitive information or who are enabled to perform specific operations.
The use of biometric recognition techniques such as fingerprints, even if it is not the only option, can provide a solution to many e-government needs. This process is accompanied by a decreased cost of equipment and an increased accessibility to high-speed communication systems [15]. For these reasons, identity validation through fingerprint recognition for voting procedures is analyzed.
Currently, voters go to electoral precincts with their picture ID and precinct board members compare the picture on the ID card with the holder of the ID. In many cases, ID cards are worn or the picture is not very clear, leaving the final decision to the judgment of the president of the precinct board. Biometric techniques offer an alternative and accurate identity verification method. To use this technique in electronic ballot boxes with precinct board members that verify the identity of voters, a fingerprint sensor can be added

to the equipment in order to avoid the use of ID cards. Thus, the system would identify voters directly through their fingerprints. Once the voter is identified, the process proceeds just as with the old electronic voting system. Figure 2 shows a diagram of this system.
To identify voters during elections, fingerprints must already have been incorporated to the electronic register of voters.

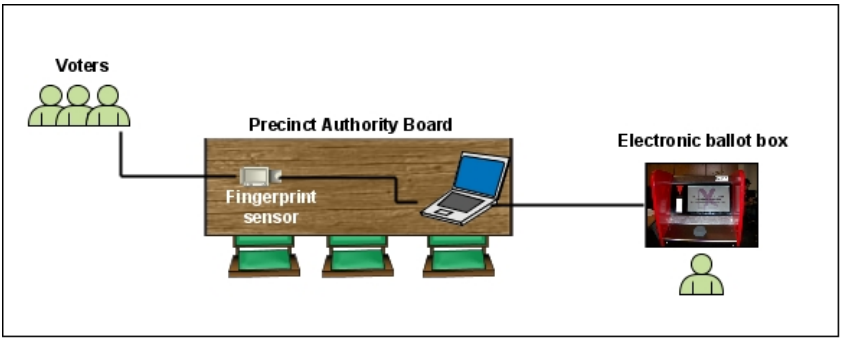


Figure 2: *Diagram of electronic voting with biometric recognition.*

5.2. Performance analysis

In previous work [16], the reliability and performance of biometric recognition in e-government applications has been analyzed. The algorithm used in this case for fingerprint recognition processes between 5,000 and 14,000 fingerprints per second. If the database is pre-sorted, processing time could improve significantly, reaching a rate of 15,000 to 70,000 fingerprints/second. In this case, the probability of finding the matching print within the first records searched is very high.

5.3. Changes in use cases

The same as with the voting structure, the interaction of actors with the system is slightly affected. Namely, the way in which voters are identified changes: instead of entering the ID number of each voter to enable the electronic ballot box, voters are identified through their fingerprints. Therefore, the electoral register is not searched for in terms of ID number, but in terms of fingerprints.

5.4. Considerations in case of error

If some fingerprint cannot be identified (due to finger alterations, cuts, etc.) the President of the precinct board will have the possibility of manually entering the ID number of the voter.

5.5. Benefits

Among the advantages of using biometric recognition, the high degree of certainty in voter identification can be mentioned, which means that the final decision is not left to the judgment of the President of the precinct board when ID cards are worn or otherwise damaged.
It should also be noted that this fingerprint biometric technique is one of the most widely used technologies and that various low-cost devices have been developed.

5.6. Extension to regional centers and Internet voting

The voting scheme will be replicated in regional centers, where the precinct authorities board will have a device to identify voters and an electronic ballot box.
The *Printing equipment* will remain at the headquarters, where vouchers will be printed.

6. Conclusions and future lines of work

An extension of previous developments of electronic voting and e-government carried out at the School of Computer Science of the UNLP has been presented, where a digital fingerprint recognition feature is added to the existing system used for faculty elections at this School.
The characteristics and performance of the biometric recognition system are analyzed, as well as the modification of the on-site electronic voting system used in La Plata (hardware and software) and the adaptation to an Internet voting system that can be used at Regional Centers.
Some of the current and future lines of work are auditing and certification of e-government systems, particularly electronic voting systems, and extension of the developed technology to other e-government applications requiring secure identification.

References

1. De Giusti, A., Feierherd, G., Pesado, P., Depetris, B., "Una aproximación a los requerimientos del software de voto electrónico de Argentina", Congreso Argentino de Ciencias de la Computación, 2004.
2. Tula, M., "Voto Electrónico", Ariel Ciencias Políticas, 2005.
3. Cantijoch Cunill, M., "El voto electrónico ¿Un temor justificado?", Revista TEXTOS de la CiberSociedad, 7. http://www.cibersociedad.net, 2005.
4. Arsaute, G. A., Tutores: Nasisi Óscar Herminio M. M. "Reconocimiento de características en huellas dactilares para la identificación humana", Universidad Nacional de San Juan, Facultad de Ingeniería. Instituto de Automática, 1997.
5. Beavan, C., "Huellas dactilares. Los orígenes de la dactiloscopía", Alba. 1990.
6. Arrieta, A., Marín, J., Sánchez, L. G., Romero, L., Sánchez, L. A., Batista, V., "Gestión y Reconocimiento Óptico de los Puntos Característicos de Imágenes de Huellas Dactilares", Universidad de Salamanca.
7. Reid, P., "Biometrics for Network Security", Prentice Hall, 2004.
8. Chirillo, J. et al., "Implementing Biometric Security", Wiley Publishing, 2003.
9. Woodward, J. D. Jr. et al., "Biometrics", McGraw-Hill Osborne Media.
10. Barbieri, A., Pasini, A., Estrebou, C., "Análisis de Urnas Electrónicas", Reporte Técnico III-LIDI, Facultad de Informática, UNLP, 2004.
11. Pesado, P., De Giusti, A., Pasini, A., Estrebou, C., "Voto Informatizado en la Facultad de Informática UNLP", Reporte Técnico III-LIDI, Facultad de Informática, UNLP, December, 2007.
12. Pesado, P., Pasini, A., Ibañez, E., Galdámez, N., Chichizola, F., Rodríguez, I., Estrebou, C., De Giusti, A., "Voto electrónico sobre internet", Congreso Argentino de Ciencias de la Computación, 2008.
13. Pesado, P., Feierherg, G., Pasini, A., "Especificación de requerimientos para sistemas de voto electrónico", Congreso Argentino de Ciencias de la Computación, 2005.
14. National Electoral Code, Decree 2135/83 Law 19,945 / 20,175 / 22,838 / 22,864 and as amended.
15. Srinivasan, V. S., Murthy, N. N., "Detection of singularity point in fingerprint images. Pattern Recognition", Vol 25, 1992.
16. Carri, J. I., Pasini, A., Pesado, P., De Giusti, A., "Reconocimiento biométrico en aplicaciones de E-Government. Análisis de confiabilidad/tiempo de respuesta", Congreso Argentino de Ciencias de la Computación, 2007.
17. Lee, H. C., Gaensslen, R. E. "Advances in fingerprint technology", Elsevier, New York, 1991.

Esta publicación se terminó de imprimir
en el mes de octubre de 2010,
en la ciudad de La Plata,
Buenos Aires,
Argentina.