RedUNCI
RED DE UNIVERSIDADES CON CARRERAS EN INFORMÁTICA

Computer Science & Technology Series

XVII Argentine Congress of Computer Science Selected Papers

Armando De Giusti | Javier Díaz
(Eds.)

edulp
Editorial
de la Universidad
de La Plata

Computer Science & Technology Series

XVII Argentine Congress of Computer Science
Selected Papers

Computer Science & Technology Series

XVII Argentine Congress of Computer Science
Selected Papers

Armando De Giusti | Javier Díaz
(Eds.)

edulp
Editorial
de la Universidad
de La Plata

De Giusti, Armando
Computer science & technology series: XVII Argentine Congress of Computer Science Selected Papers / Armando De Giusti y Javier Díaz; edición literaria a cargo de Armando De Giusti y Javier Díaz. - 1a ed. - La Plata: Universidad Nacional de La Plata, 2012.
316 p.; 24x16 cm.

ISBN 978-950-34-0885-8

1. Ciencias de la Computación. 2. Informática. 3. Tecnología. I. Díaz, Javier II. De Giusti, Armando, ed. lit. III. Díaz, Javier, ed. lit.

Computer Science & Technology Series

XVII ARGENTINE CONGRESS OF COMPUTER SCIENCE
SELECTED PAPERS

Diagramación: Andrea López Osornio

Editorial de la Universidad Nacional de La Plata (Edulp)
Calle 47 N.º 380 / La Plata B1900AJP / Buenos Aires, Argentina
+ 54 221 4273992 / 427 4898
editorial@editorial.unlp.edu.ar
www.editorial.unlp.edu.ar

EDULP integra la Red de Editoriales Universitarias Nacionales (REUN)

Primera edición - 2012
ISBN Nº 978-950-34-0885-8
Queda hecho el depósito que marca la Ley 11.723

Topics

XII IntelligentAgents and SystemsWorkshop
Chairs Guillermo Leguizamón (UNSL) Alejandro García (UNS) Ana Casali (UNR)

XI Distributed and Parallel ProcessingWorkshop
Chairs Armando De Giusti (UNLP) Jorge Ardenghi (UNS) Maria Fabiana Piccoli (UNSL)

X InformationTechnologyApplied to EducationWorkshop
Chairs Cristina Madoz (UNLP) Sonia Rueda (UNS) Marcela Chiarani (UNSL) UrielCukierman (UTN)

IX Graphic Computation, Imagery andVisualizationWorkshop
Chairs Silvia Castro (UNS) Oscar Bría (INVAP) María Jose Abásolo (CIC-UNLP)

VIII Software EngineeringWorkshop
Chairs Patricia Pesado (UNLP) Elsa Estévez (United Nations)Alejandra Cechich (UNCOMA) Horacio Kuna (UNM)

VIII Database and Data MiningWorkshop
Chairs Hugo Alfonso (UNLPam) Rodolfo Bertone (UNLP) Olinda Gagliardi (UNSL)

VIArchitecture, Nets and Operating SystemsWorkshop
Chairs Javier Díaz (UNLP) Antonio Castro Lechtaller (IESE) Hugo Padovani (UM) Nelson Acosta (UNCPBA)

III Innovation in Software SystemsWorkshop
Chairs Pablo Fillottrani (UNS) Carlos Neil (UAI) Marcelo Estayno (UNLZ)

II Computer Science Theoretical Aspects Warkshop
Chairs Susana Esquivel (UNSL) Marcelo Falappa (UNS)

II Signal Processing and Real-Time Systems Workshop
Chairs DanielPandolfi (UNPA) Horacio Villagarcía Wanza (UNLP) Hugo Ramón (UNNOBA)

I ETHICOMP Latinoamérica
Chairs Simon Rogerson (Monfort University - ReinoUnido) Terrell Ward Bynum (Southern Connecticut State University - EE.UU.) William Fleischman (Villanova University - EE.UU.) Guillermo Feierherd (Universidad Nacional de La Patagonia SJB - Argentina) Mario Arias Olivera (UniversitatRoviraiVirgili - España)

Honour Commitee

Secretary of University Policies
Dr. Alberto Dibbern

UNLP President
Dr. Fernando Tauber

CONICET President
Dra. Marta Rovira

CIC President
Ing. Agr. Carlos Gerónimo Gianella

School of Computer Sciences (UNLP) Dean
Lic. Javier Díaz

Scientific Committee

Abásolo, María José (Argentina)
Acosta, Nelson (Argentina)
Alba Torres, Enrique (España)
Alfonso, Hugo (Argentina)
Ardenghi, Jorge (Argentina)
Arias Oliva, Mario (España)
Bertone, Rodolfo (Argentina)
Bría, Oscar (Argentina)
Brisaboa, Nieves (España)
Bynum, Terrell Ward (EEUU)
Cabero, Julio (España)
Cancela, Héctor (Uruguay)
Casali, Ana (Argentina)
Castro Lechtaller, Antonio (Argentina)
Castro, Silvia (Argentina)
Cechich, Alejandra (Argentina)
Chiarani, Marcela (Argentina)
Coello Coello, Carlos (México)
Collazos Ordóñez, César Alberto (Colombia)
Cukierman, Uriel (Argentina)
Diaz, Javier (Argentina)
Dix,Juerguen (Alemania)
Doallo, Ramón (España)
Esquivel, Susana (Argentina)
Estayno, Marcelo (Argentina)
Estevez, Elsa (Naciones Unidas)
Falappa, Marcelo (Argentina)
Fillottrani, Pablo (Argentina)
Fleischman, William (EEUU)
Gagliardi, Olinda (Argentina)
García, Alejandro (Argentina)
Gröller, Eduard (Austria)
Hernández, Gregorio (España)
Janowski, Tomasz (Naciones Unidas)
Kuna, Horacio (Argentina)
Leguizamón, Guillermo(Argentina)
Loui, Ronald Prescott (EEUU)
Luque, Emilio (España)
Madoz, Cristina (Argentina)
Manresa-Yee, Cristina (España)
Marín, Mauricio (Chile)
Marquez, María Eugenia (Argentina)
Naiouf, Marcelo (Argentina)
Navarro Martín, Antonio (España)
Neil, Carlos (Argentina)
Olivas Varela, José Ángel (España)
Padovani, Hugo (Argentina)
Pandolfi, Daniel (Argentina)
Pesado, Patricia (Argentina)
Piattini, Mario (España)
Piccoli, María Fabiana (Argentina)
Printista, Marcela (Argentina)
Puppo, Enrico (Italia)
Ramón, Hugo (Argentina)
Rogerson, Simon (Reino Unido)
Rossi, Gustavo (Argentina)
Rueda, Sonia (Argentina)
Santos, Juan Miguel (Argentina)

Sanz, Cecilia (Argentina)
Steinmetz, Ralf (Alemania)
Suppi, Remo (España)
Tarouco, Liane (Brasil)
Tirado, Francisco (España)
Utreras, Florencio (Chile)
Vendrell, Eduardo (España)
Villagarcia Wanza, Horacio (Argentina)
Vizcaino, Aurora (España)
Zamarro, Jose Miguel (España)

Organizing Commitee

Universidad Nacional de La Plata -
Facultad de Informática
Argentina

President
Pesado, Patricia

Naiouf, Marcelo
Lanzarini, Laura
Tinetti, Fernando
Sanz, Cecilia
Bertone, Rodolfo
Madoz, Cristina
Gorga, Gladys
Boracchia, Marcos
Villagarcía Wanza, Horacio
Thomas, Pablo
De Giusti, Laura
Esponda, Silvia
Romero, Fernando
Chichizola, Franco
González, Alejandro
Giacomantone, Javier
Pasini, Ariel
Corbalán, Leonardo
Cristina, Federico
Ibañez, Edurado
Marrero, Luciano
Dapoto, Sebastián
Delia, Lisandro
Montezanti, Diego
Pousa, Adrián
Iglesias, Luciano
Hasperué, Waldo
Encinas, Diego
Galdámez, Nicolás
Artola, Verónica
Rodríguez, Ismael Pablo
Estrebou, César
Martorelli, Sabrina
Pettoruti, José Enrique
Moralejo, Lucrecia
Sanz, Victoria
Frati, Fernando Emmanuel
Guisen, Andrea
Leibovich, Fabiana
Albanesi, Bernarda
Caseres, Germán
Villa Monte, Augusto
Maulini, Juan
Rucci, Enzo
Ronchetti,Franco
Rodríguez Eguren, Pablo Sebastián
Gallo, Silvana
Panci, Fernando
Sanchez, Mariano
Lorenti, Luciano
Valderrama, Cristina
Jacquemain, Eliana
Pizarro, Alejandra
Otero, Natalia
Mongou, Lourdes
Mieres, Deborah
Folegoto, Lucas
Blesa, Fernanda

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, Signal Processing, Real time Systems and Ethics in Computer Science).

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

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

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

CACIC 2011 in La Plata

CACIC'11 was the seventeenth Congress in the CACIC series. It was organized by the School of Computer Science of the University of La Plata.

The Congress included 11 Workshops with 148 accepted papers, 3 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.cacic2011.edu.ar/).

CACIC 2011 was organized following the traditional Congress format, with 11 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 281 submissions. An average of 2.5 review reports were collected for each paper, for a grand total of 702 review reports that involved about 400 different reviewers.

A total of 148 full papers, involving 393 authors and 77 Universities, were accepted and 25 of them were selected for this book.

Acknowledgments

CACIC 2011 was made possible due to the support of many individuals and organizations. The School of Computer Science of the University of La Plata, RedUNCI, the Secretary of University Policies, the University of La Plata, the National Agency of Scientific and Technological Advancement, CIC and CONICET were the main institutional sponsors.

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

Ing. Armando De Giusti
RedUNCI

Table of Contents

XII

Intelligent Agents and Systems Workshop

Using Possibilistic Defeasible Logic Programming for Reasoning with Inconsistent Ontologies

SERGIO A. GÓMEZ[1], CARLOS I. CHESÑEVAR[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, Email: {sag,cic,grs}@cs.uns.edu.ar

[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina

Abstract. *In this paper we present a preliminary framework for reasoning with possibly inconsistent Description Logic ontologies using Possibilistic Defeasible Logic Programming. We present a case study where it is showed how the proposed approach works. We contend that the proposal presented is apt for being used in the context of Semantic Web ontologies as it can be applied to the Web Ontology Language OWL, which is the current standard.*

1. Introduction

The *Semantic Web* (SW) [1] is a vision of the Web where resources have precise meaning defined in terms of ontologies. The Web Ontology Language (OWL) [2] whose semantics is based on *Description Logics* [3] is the *de facto* standard for the SW. Agents in the SW are supposed to reason over web resources by using standard reasoning systems, thus being able to compute an implicit hierarchy of concepts defined in an ontology and then checking the membership of individuals to those concepts. Over the last few years an alternative approach to reasoning with ontologies called *Description Logic Programming* (DLP) [4] has gained interest. The DLP approach relies on translating DL ontologies into the language of logic programming, so standard Prolog environments can be used to reason on them.

A possible anomaly in ontologies is *inconsistency*. An inconsistency is characterized by a logical contradiction. Inconsistencies in ontologies prevent standard reasoners from obtaining conclusions. Normally, this situation renders an ontology useless unless the knowledge engineer *debugs* it (*i.e.*, repairs the ontology for making it consistent). As the knowledge engineer could not be available, alternative approaches to automatically repairing the ontology consist of using Belief Revision [5] techniques to either extract a maximally consistent subset of the ontology or discard a minimally inconsistent subset of the ontology. Yet another approach consists of using a non-standard reasoning mechanism for accepting inconsistency and just *dealing* with it (for instance using Paraconsistent Logics [6]). In this line of work, Gómez *et al.* [7] have applied defeasible argumentation (in particular

Defeasible Logic Programming [8] to reason with possibly inconsistent ontologies.

Possibilistic Defeasible Logic Programming (P_DeLP) [9] is a logic programming language which combines features from argumentation theory and logic programming, incorporating as well the treatment of possibilistic uncertainty and fuzzy knowledge at object-language level. In this article, we show a preliminary approach to reason with possibly inconsistent DL ontologies in P-DeLP. For this we define the concept of weighted DL ontology which is an ontology whose axioms have given numerical weights indicating their degree of certainty, then the ontology can be interpreted as a P-DeLP program.

Outline: In Section 2, we present the fundamentals of Description Logics. Section 3 reviews the fundamentals of Possibilistic Defeasible Logic Programming. In Section 4, we introduce a framework for reasoning with possibly inconsistent weighted ontologies in PDeLP. Finally, Section 5 concludes.

2. Fundamentals of Description Logics

Description Logics (DL) [3] are a family of knowledge representation formalisms based on the notions of *concepts* (unary predicates, classes) and *roles* (binary relations) that allow building complex concepts and roles from atomic ones. Let *C*, *D* stand for concepts, *R* for a role and *a*,*b* for individuals. Concept descriptions are built from concept names using the constructors conjunction$(C \sqcap D)$, disjunction $(C \sqcup D)$, complement $(\neg C)$, existential restriction $(\exists R.C)$, and value restriction $(\forall R.C)$. To define the semantics of concept descriptions, concepts are interpreted as subsets of a domain of interest, and roles as binary relations over this domain. Further extensions are possible including inverse (P^-) and transitive (P^+) roles. A DL ontology consists of two finite and mutually disjoint sets: a *Tbox* which introduces the *terminology* and an *Abox* which contains facts about particular objects in the application domain. Tbox statements have the form $C \sqsubseteq D$ (*inclusions*) and $C \equiv D$ (*equalities*), where *C* and *D* are (possibly complex) concept descriptions. Objects in the Abox are referred to by a finite number of *individual names* and these names may be used in two types of assertional statements: *concept assertions* of the type *a:C* and *role assertions* of the type <*a*,*b*>:*R*.

A knowledge representation system based on DL is able to perform specific kinds of reasoning, its purpose goes beyond storing concept definitions and assertions. As a DL ontology has a semantics that makes it equivalent to a set of axioms of first-order logic, it contains implicit knowledge that can be made explicit through inferences. Inferences in DL systems are usually divided into Tbox reasoning and Abox reasoning. In this paper we are concerned only with Abox reasoning, more precisely with *instance checking* [3]. Instance checking consists of determining if an assertion is entailed from an Abox. For instance, $T \cup A \models a{:}C$ indicates that the individual a is a member of the concept C w.r.t. the Abox A and the Tbox T.

3. Possibilistic Defeasible Logic Programming

The P-DeLP [9] language **L** is defined from a set of ground fuzzy atoms (fuzzy propositional variables) $\{p, q, \ldots\}$ together with the conectives $\{\sim, \wedge, \leftarrow\}$. The symbol ~ stands for *negation*. A literal $L \in$ **L** is a ground (fuzzy) atom ~q, where q is a ground (fuzzy) propositional variable. A *rule* in **L** is a formula of the form $Q \leftarrow L_1 \wedge \ldots \wedge L_n$, where $Q, L_1, \ldots, L_n$ are literals in **L**. When n=0, the formula Q← is called a *fact*. The term *goal* will refer to any literal $Q \in$ **L**. Facts, rules and goals are the well-formed formulas in **L**.

Definition 1 (Certainty-weighted clause) A *certainty-weighted clause*, or simply weighted clause, is a pair (γ, α), where γ is a w.f.f. in **L** and $\alpha \in [0,1]$ expresses a lower bound for the certainty of γ in terms of a necessity measure.

The original P-DeLP language is based on Possibilistic Gödel Logic or PGL, which is able to model both uncertainty and fuzziness and allows for a partial matching mechanism between fuzzy propositional variables. For simplicity, Chesñevar *et al.* [9] restrict themselves to the fragment of P-DeLP built on non-fuzzy propositions, and hence based on the necessity-valued classical propositional Possibilistic logic. As a consequence, possibilistic models are defined by possibility distributions on the set of classical interpretations and the proof method for P-DeLP formulas, written |-, is defined based on the generalized modus ponens rule:

$$\frac{\begin{array}{c}(L_0 \leftarrow L_1 \wedge \ldots \wedge L_k, \gamma)\\ (L_1, \beta_1), \ldots, (L_k, \beta_k)\end{array}}{(L_0, \min(\gamma, \beta_1, \ldots, \beta_k)}$$

which is a particular instance of the possibilistic resolution rule, and which provides the *non-fuzzy* fragment of P-DeLP with a complete calculus for determining the maximum degree of possibilistic entailment for weighted literals.

In P-DeLP *certain* and *uncertain* clauses can be distinguished. A clause (γ, α) is referred as certain if α=1 and uncertain otherwise. A set of clauses Γ is deemed as *contradictory*, denoted Γ|-$\perp$, whenever Γ|- (q, α) Γ|-$(\sim q, \beta)$, with α>0 and β>0, for some atom in **L**. A P-DeLP program is a set of weighted rules and facts in **L** in which certain and uncertain information is distinguished. As an additional requirement, certain knowledge is required to be non-contradictory. Formally:

Definition 2 (Program) A *P-DeLP program P* (or just *program P*) is a pair (Π,Δ), where Π is a non-contradictory finite set of certain clauses, and Δ is a finite set of uncertain clauses.

Definition 3 (Argument. Subargument) Given a program $P=(\Pi,\Delta)$, a set $A \subseteq \Delta$ of uncertain clauses is an *argument* for a goal Q with necessity degree $\alpha>0$, denoted $<A,Q,\alpha>$, iff: (i) Π U A |- (Q, α); (ii) Π U A is non-contradictory, and (iii) there is no $A_1 \subset A$ such that Π U A_1 |- (Q,β), $\beta>0$. Let $<A,Q,\alpha>$ and $<S,R,\beta>$ be two arguments, $<S,R,\beta>$ is a *subargument* of $<A,Q,\alpha>$ iff $S \subseteq A$.
Conflict among arguments is formalized by the notions of counterargument and defeat.

Definition 4 (Counterargument) Let P be a program, and let $<A_1,Q_1,\alpha_1>$ and $<A_2,Q_2,\alpha_2>$ be two arguments in P. We say that $<A_1,Q_1,\alpha_1>$ *counterargues* $<A_2,Q_2,\alpha_2>$ iff there exists a subargument (called *disagreement subargument*) $<S,Q, \beta>$ of $<A_2,Q_2,\alpha_2>$ such that $P \cup \{(Q_1, \alpha_1), (Q, \beta)\}$ is contradictory. The literal (Q, β) is called *disagreement literal.*

Defeat among arguments involves a *preference criterion* on conflicting arguments, defined on the basis of necessity measures associated with arguments.

Definition 5 (Defeat) Let P be a P-DeLP program, and let $<A_1,Q_1,\alpha_1>$ and $<A_2,Q_2,\alpha_2>$ be two arguments in P. We will say that $<A_1,Q_1,\alpha_1>$ is a *defeater* for $<A_2,Q_2,\alpha_2>$ iff $<A_1,Q_1,\alpha_1>$ counterargues argument $<A_2,Q_2,\alpha_2>$ with disagreement subargument $<A,Q,\alpha>$, with $\alpha_1 \geq \alpha$. If $\alpha_1 > \alpha$ then $<A_1,Q_1,\alpha_1>$ is called a *proper defeater*, otherwise ($\alpha_1 = \alpha$) it is called a *blocking defeater.*

Definition 6 (Argumentation line) An *argumentation line* λ starting in an argument $<A_0,Q_0,\alpha_0>$ is a finite sequence of arguments $[<A_0,Q_0,\alpha_0>, <A_1,Q_1,\alpha_1>, ..., <A_n,Q_n,\alpha_n>, ...]$ such that every $<A_i,Q_i,\alpha_i>$ defeats $<A_{i-1},Q_{i-1},\alpha_{i-1}>$, for $0<i \leq n$, satisfying certain *dialectical constraints* (see below). Every argument $<A_i,Q_i,\alpha_i>$ in λ has *level i*. We will distinguish the sets $S_\lambda^k = \bigcup_{i=0,2,\ldots,2\lfloor k/2 \rfloor} \{\langle A_i, Q_i, \alpha_i \rangle \in \lambda\}$ and $I_\lambda^k = \bigcup_{i=1,3,\ldots,2\lfloor k/2 \rfloor+1} \{\langle A_i, Q_i, \alpha_i \rangle \in \lambda\}$ associated with even-level (resp. odd-level) arguments in λ up to the k-th level ($k \leq n$).

An argumentation line 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 w.r.t. a P-DeLP program P, namely:

1. *Non-contradiction:* given an argumentation line λ of length n the set S_λ^n associated with the proponent (resp. I_λ^n for the opponent) should be *non-contradictory* w.r.t. P.

2. *No circular argumentation:* no argument $<A_j,Q_j,\alpha_j>$ in λ is a sub-argument of an argument $<A_i,Q_i,\alpha_i>$ in λ, $i<j$.
3. *Progressive argumentation*: every blocking defeater $<A_i,Q_i,\alpha_i>$ in λ is defeated by a proper defeater $<A_{i+1},Q_{i+1},\alpha_{i+1}>$ in λ.

To determine whether a given argument is ultimately undefeated (or warranted) w.r.t. a program P, the P-DeLP framework relies on an exhaustive dialectical analysis. Such analysis is modeled in terms of a dialectical tree:

Definition 7 (Dialectical tree). Let *P* be a program, and let $<A_0,Q_0, \alpha_0>$ be an argument w.r.t. *P*. A *dialectical tree* for $<A_0,Q_0, \alpha_0>$, denoted $T_{<A0,Q0,\alpha0>}$, is a tree structure defined as follows:

1. The root node of $T_{<A0,Q0,\alpha0>}$ is $<A_0,Q_0, \alpha_0>$.
2. $<B',H',\beta'>$ is an immediate child of $<B,H,\beta>$ iff there exists an argumentation line $\lambda=[<A_0,Q_0,\alpha_0>, <A_1,Q_1,\alpha_1>, ..., <A_n,Q_n,\alpha_n>, ...]$ such that there are two elements $<A_{i+1},Q_{i+1},\alpha_{i+1}>=<B',H',\beta'>$ and $<A_i,Q_i,\alpha_i>=<B,H,\beta>$, for some i=0,...,n-1.

Nodes in a dialectical tree 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 the leaves 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.

Definition 8 (Warrant). An argument $<A_0,Q_0, \alpha_0>$ is ultimately accepted as valid (or *warranted*) with a necessity degree α_0 w.r.t. a program *P* iff the root of the tree $T_{<A0,Q0,\alpha0>}$ is marked as a U-node.

4. Reasoning with DL Ontologies as P-DeLP Programs

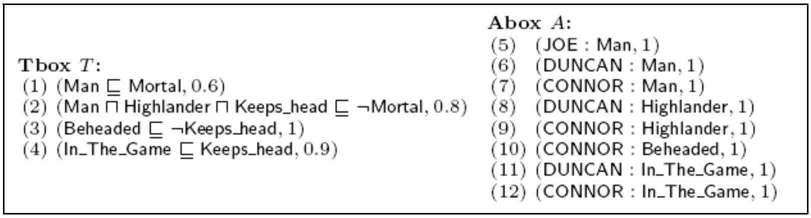
In the presence of inconsistency, traditional DL reasoners issue an error message and stop further processing of ontologies. Thus the burden of repairing the ontology is on the knowledge engineer. However, the knowledge engineer is not always available and in some cases, such as when dealing with imported ontologies, he has neither the authority nor the expertise to correct the source of inconsistency. Therefore, we are interested in coping with inconsistencies such that the task of dealing with them is automatically solved by the reasoning system. We propose performing such a task by translating DL ontologies into P-DeLP programs. By doing so we gain the capability of reasoning with inconsistent ontologies. However we also lose some expressiveness in the involved ontologies. As Def. 11 shows, certain restrictions will have to be imposed on DL ontologies in order to be expressed in the P-DeLP language.

Our proposal is based in part in the work of [4] who show that the processing of ontologies can be improved by the use of techniques from the area of logic programming. In particular they have identified a subset of DL languages that can be effectively mapped into a Horn-clause logics.

Definition 9 (Weighted axiom. Weighted assertion). Let *C*, *D* stand for concept names, *P*, *Q* for role names, and *a*, *b* for individual names. Let *A* be an axiom of the form $C \sqsubseteq D$, $C \equiv D$, $\top \sqsubseteq \forall P.D$, $\top \sqsubseteq \forall P^{-}.D$, $P \sqsubseteq Q$, $P \equiv Q$, $P \equiv Q^{-}$, or $P \sqsubseteq P^{+}$. Let ω be a real number such that $0 \leq \omega \leq 1$. A *weighted axiom* is pair (*A*, ω). Let *B* be an assertion of the form *a*:*C* or <*a*,*b*>:*P*. A *weighted assertion* is a pair (*B*, ω).

Definition 10 (Weighted ontology). Let *T* be a set of weighted axioms and *A* be a set of weighted assertions. A *weighted ontology* ∑ is a pair (*T*,*A*). The set *T* is called *weighted terminology* (or just *weighted Tbox*) and *A* is called *weighted assertional box* (or *weighted Abox* for short).

Example 1. In Fig. 1, we present a weighted ontology ∑= (*T*,*A*) based on the fictional universe of the *Highlander* movies. The meaning of weighted terminology *T* is as follows: Axiom (1) says that a man is apparently a mortal; axiom (2) expresses that men from the Highlands that keep their heads on are supposed to be immortals; axiom (3) implies that every beheaded man does not keep his head on, and (4) says that immortals that still are known to be *in the game* have their heads on. The weighted assertional box *A* expresses that Joe, Duncan and Connor are men; Duncan and Connor are Highlanders, Connor has been beheaded, and it is known that Connor and Duncan have been in the game.



Tbox T:
(1) (Man $\sqsubseteq$ Mortal, 0.6)
(2) (Man $\sqcap$ Highlander $\sqcap$ Keeps_head $\sqsubseteq \neg$Mortal, 0.8)
(3) (Beheaded $\sqsubseteq \neg$Keeps_head, 1)
(4) (In_The_Game $\sqsubseteq$ Keeps_head, 0.9)

Abox A:
(5) (JOE : Man, 1)
(6) (DUNCAN : Man, 1)
(7) (CONNOR : Man, 1)
(8) (DUNCAN : Highlander, 1)
(9) (CONNOR : Highlander, 1)
(10) (CONNOR : Beheaded, 1)
(11) (DUNCAN : In_The_Game, 1)
(12) (CONNOR : In_The_Game, 1)

Fig. 1. A weighted ontology Σ=(*T*,*A*)

As noted by Grosof *et al.* [4], for DL sentences to be mapped into Horn-logic rules, they must satisfy certain constraints. Conjunction and universal restrictions appearing in the right-hand side of inclusion axioms can be mapped to heads of rules (called L_h-classes). In contrast, conjunction, disjunction and existential restriction can be mapped to rule bodies whenever they occur in the left-hand side of inclusion axioms (called L_b-classes). As equality axioms "$C \equiv D$" are interpreted as two inclusion axioms "$C \sqsubseteq D$" and "$D \sqsubseteq C$"', they must belong to the intersection of L_h and L_b (called L_{hb} classes).

Definition 11 (L_h, L_b and L_{hb} classes (adapted from [4])). Let A be an atomic class name, C and D class expressions, and R a property. In the L_h language, $C \sqcap D$ is a class, and $\forall R.C$ is also a class. Class expressions in L_h are called *L_h-classes*. In the L_b language, $C \sqcup D$ is a class, and $\exists R.C$ is a class too. Class expressions in L_b are called *L_b-classes*. The L_{hb} language is defined as the intersection of L_h and L_b. Class expressions in L_{hb} are called *L_{hb} –classes*.

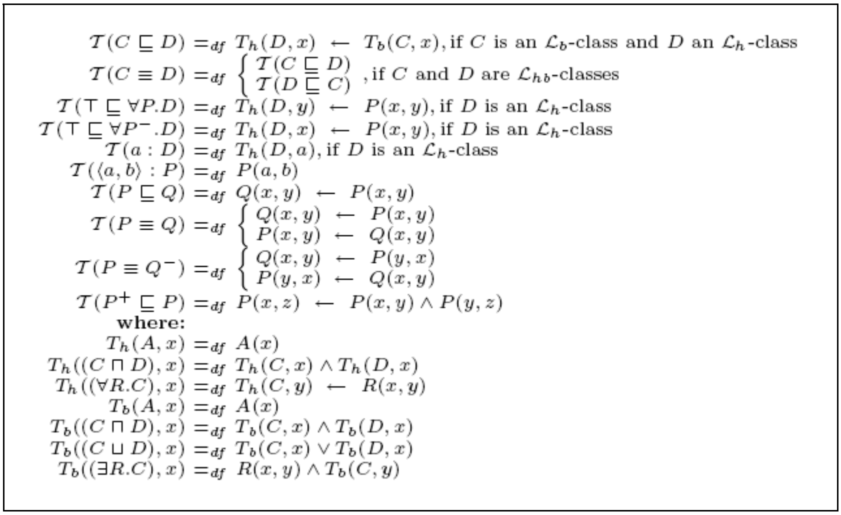
Definition 12 (T mapping from DL sentences to logic programming rules (adapted from [4])). Let A, C, D be concepts, X, Y variables, P, Q properties. The **T** mapping from the language of DL to the language of P-DeLP is defined in Fig. 2. Besides, intermediate transformations of the form "H_1 ^ H_2 ←B" will be rewritten as two rules "H_1←B"and "H_2←B". Similarly transformations of the form "H_1 ← H_2 ← B" will be rewritten as "H_1← B ^ H_2", and rules of the form "H ← B_1 v B_2" will be rewritten as two rules "H ← B_1" and "H ← B_2".

Definition 13 (Interpretation of a weighted ontology). Let $\Sigma=(T,A)$ be a weighted ontology such that $T=\{(s_1,\alpha_1), ..., (s_n,\alpha_n)\}$ and $A=\{(a_1,\beta_1), ..., (a_m,\beta_\mathrm{m})\}$, then:

$$\mathrm{TRAD}(T) = \{(\mathbf{T}(s_1), \alpha_1), ..., (\mathbf{T}(s_\mathrm{n},\alpha_\mathrm{n}))\}$$

$$\mathrm{TRAD}(A) = \{(\mathbf{T}(a_1), \beta_1), ..., (\mathbf{T}(a_m,\beta_m))\}$$

Besides, if the cardinal k of $\mathbf{T}(s_i)=\{\mathrm{f}_1, ..., \mathrm{f}_\mathrm{k}\}$ is greater than 1, then the translation of (s_i,α_i) is $(\mathrm{f}_i, \alpha_\mathrm{i}), ..., (\mathrm{f}_k, \alpha_\mathrm{i})$. Let PROP be the propositionalization operator for a first-order theory (each propositional term "*p*(*a*)" generated from a predicate "*p*(*x*)" and a constant "*a*" will be noted as "*p_a*"). The *interpretation* of Σ, noted as INTERPRETATION(Σ), is the P-DeLP program P=(PROP(**T**(T)), PROP(**T**(A))).



$$
\begin{aligned}
\mathcal{T}(C \sqsubseteq D) &=_{df} T_h(D,x) \ \leftarrow \ T_b(C,x), \text{if } C \text{ is an } \mathcal{L}_b\text{-class and } D \text{ an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(C \equiv D) &=_{df} \begin{cases} \mathcal{T}(C \sqsubseteq D) \\ \mathcal{T}(D \sqsubseteq C) \end{cases} \text{, if } C \text{ and } D \text{ are } \mathcal{L}_{hb}\text{-classes} \\
\mathcal{T}(\top \sqsubseteq \forall P.D) &=_{df} T_h(D,y) \ \leftarrow \ P(x,y), \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(\top \sqsubseteq \forall P^-.D) &=_{df} T_h(D,x) \ \leftarrow \ P(x,y), \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(a : D) &=_{df} T_h(D,a), \text{if } D \text{ is an } \mathcal{L}_h\text{-class} \\
\mathcal{T}(\langle a,b\rangle : P) &=_{df} P(a,b) \\
\mathcal{T}(P \sqsubseteq Q) &=_{df} Q(x,y) \ \leftarrow \ P(x,y) \\
\mathcal{T}(P \equiv Q) &=_{df} \begin{cases} Q(x,y) \ \leftarrow \ P(x,y) \\ P(x,y) \ \leftarrow \ Q(x,y) \end{cases} \\
\mathcal{T}(P \equiv Q^-) &=_{df} \begin{cases} Q(x,y) \ \leftarrow \ P(y,x) \\ P(y,x) \ \leftarrow \ Q(x,y) \end{cases} \\
\mathcal{T}(P^+ \sqsubseteq P) &=_{df} P(x,z) \ \leftarrow \ P(x,y) \wedge P(y,z) \\
\textbf{where:} & \\
T_h(A,x) &=_{df} A(x) \\
T_h((C \sqcap D),x) &=_{df} T_h(C,x) \wedge T_h(D,x) \\
T_h((\forall R.C),x) &=_{df} T_h(C,y) \ \leftarrow \ R(x,y) \\
T_b(A,x) &=_{df} A(x) \\
T_b((C \sqcap D),x) &=_{df} T_b(C,x) \wedge T_b(D,x) \\
T_b((C \sqcup D),x) &=_{df} T_b(C,x) \vee T_b(D,x) \\
T_b((\exists R.C),x) &=_{df} R(x,y) \wedge T_b(C,y)
\end{aligned}
$$

Fig. 2. Mapping from DL axioms to logic programming rules

Example 2. Consider again the weighted ontology Σ=(T,A) presented in Ex. 1. In Fig. 3 we present a logical program TRAD(T) U TRAD (A). And in Fig. 4, we present the P-DeLP program P=INTERPRETATION(Σ). Notice that as there are three constants (*viz.*, *joe*, *duncan* and *connor*), and three first-order rules (*viz.*, (1)-(4)), twelve rules are generated in the propositional program, *i.e.* four for every instantiation of each rule with each one of the three constants.

Definition 14 (Instance checking). Let Σ=(T,A) be a weighted ontology. Let C be a concept name, a an individual name. Let ε be real number such that $0 \leq \varepsilon \leq 1$. The individual a is a member of the concept C with strength ε iff there is a warranted argument <A,C(a),ε> w.r.t. INTERPRETATION(Σ).

Example 3. Consider the program P in Ex. 4 that corresponds to the interpretation of the ontology Σ from Ex. 1. We will show how the operation of instance checking works in P-DeLP for deciding the membership of the individuals Joe, Duncan and Connor to the concept Mortal. First, consider the case of Joe: An argument <A, *mortal_joe*, 0.6> can be obtained, where A = {(*mortal_joe* ← *man_joe*, 0.6)}. This argument has no defeaters and is thus warranted, therefore we conclude that JOE is a member of the concept Man with strength 0.6 (see Fig. 5.(a)). Second, consider the case of Duncan: As in the case of Joe, there is an argument <B_1, *mortal_duncan*, 0.6>, with B_1= {(*mortal_duncan* ← *man_duncan*, 0.6)}. But this argument is defeated by <B_2, *~mortal_duncan*, 0.8>, where

B_2 = {(*~mortal_duncan* ← *man_duncan* ^ *highlander_duncan* ^
keeps_head_duncan, 0.8),
(*keeps_head_duncan* ← *in_the_game_duncan*, 0.9)}.

As this argument B_2 is undefeated, we reach the conclusion that Duncan is a member of the concept ¬Mortal with strength 0.8 (see Fig. 5.(b)). Last, consider the case of Connor: Yet again there is an argument expressing that Connor is mortal since he is a man: <C_1, *mortal_connor*, 0.6>, with

C_1 = { (*mortal_connor* ← *man_connor*, 0.6) };

as in the case of Duncan, this argument is defeated by another that says that Connor in immortal because he is a Highlander, *i.e.* <C_2, *~mortal_connor*, 0.8> where

C_2 = {(*~mortal_connor* ← *man_connor* ^ *highlander_connor* ^
keeps_head_connor, 0.8),
(*keeps_head_connor* ← *in_the_game_connor*, 0.9)}.

However, in this case there is another (undefeated) argument $\Ca_3$ that defeats C_2, namely <C_3, *~keeps_head_connor*, 1>, where

C_3 = {(*~keeps_head_connor* ← *beheaded_connor*, 1)}.

In this way, the argument C_1 gets undefeated again, and we conclude that Connor is a member of the concept Mortal with strength 0.6 (see Fig. 5.(c)).

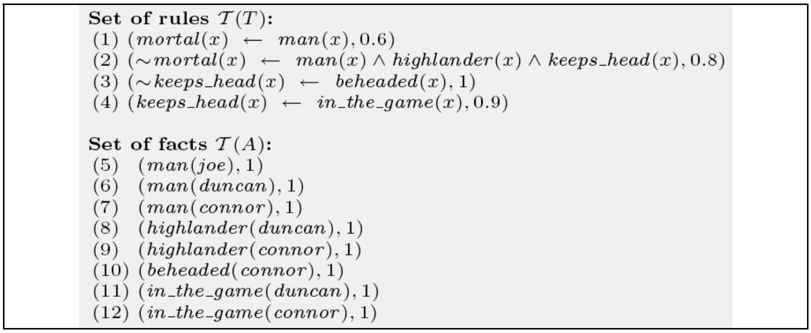
5. Conclusions and Future Work

We have presented a preliminary framework for reasoning with inconsistent ontologies by using the Possibilistic Defeasible Logic Programming machinery. The proposed approach allows for determining the degree of tentativeness for the membership of an individual to a class in the potential presence of inconsistency w.r.t. a Description Logic ontology. Axioms and assertions in an ontology are qualified with degrees of certainty which are used to determine the degree of membership of individuals to concepts. This approach continues previous work of ours [7] that translates DL ontologies into Defeasible Logic Programming and uses generalized specificity to compare arguments. One advantage of the approach presented in this paper is that it allows characterizing the preference criterion between axioms and thus arguments that are built when considering them, since the comparison criterion is no longer syntactically determined by the form of the underlying program that represents the ontology. One drawback of our approach is that the propositionalization of a first-order theory produces lots of facts that are irrelevant (notice that given a knowledge base with p k-ary predicates and n constants, there are possible pn^k instantiations, see Ex. 2), impacting the computational efficiency of the method. As part of our current research work, we are interested in applying this proposal to ontology integration and studying what the intrinsic logical properties of the approach are.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scient. American (2001).
2. McGuinnes, D.L., van Harmelen, F.: OWL Web Ontology Language Overview (2004).
3. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: The Description Logic Handbook-Theory, Implementation and Applications. Cambridge University Press (2003).
4. Grosof, B.N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logics. WWW2003, May 20-24, Budapest, Hungary (2003).
5. Ribeiro, M.M., Wassermann, R.: Base Revision for Ontology Debugging, J. Log. Comput. 19 (5) 2009, 721-743.
6. Huang, Z., van Harmelen, F., ten Teije, A.: Reasoning with Inconsistent Ontologies. In Kaelbling, L.P., Safiotti, A. eds.: Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI'05), Edinburgh, Scotland (August 2005) 454-459.
7. Gómez, S.A., Chesñevar, C.I., Simari, G.R.: Reasoning with Inconsistent Ontologies Through Argumentation. Applied Artificial Intelligence 24(1) (2010) 102-148.

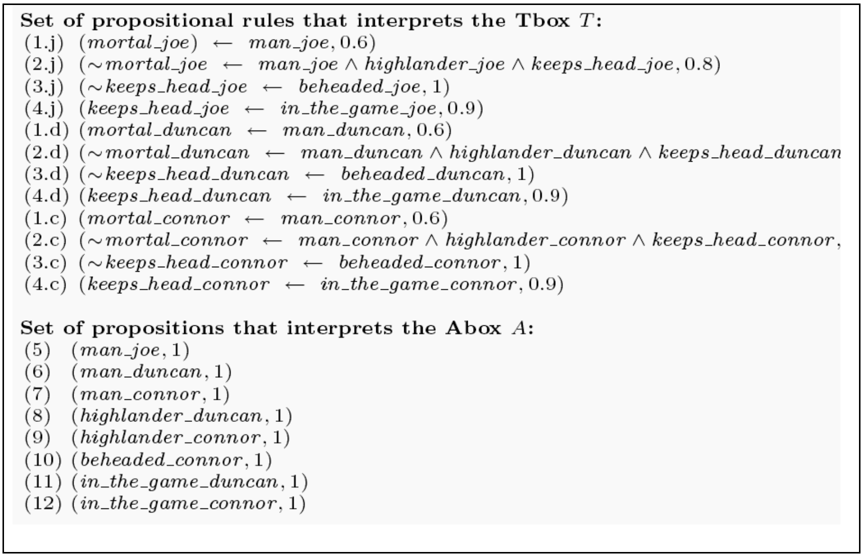
8. García, A., Simari, G.: Defeasible Logic Programming an Argumentative Approach. Theory and Practice of Logic Programmming 4(1) (2004) 95-138.
9. Alsinet, T., Chesñevar, C.I., Godo, L.: \newblock A level-based approach to computing warranted arguments in possibilistic defeasible logic programming. In Besnard, P., Doutre, S., Hunter, A., eds.: COMMA. Volume 172 of Frontiers in Artificial Intelligence and Applications., IOS Press (2008) 1-12.



Set of rules $\mathcal{T}(T)$:
(1) $(\mathit{mortal}(x) \leftarrow \mathit{man}(x), 0.6)$
(2) $(\sim\mathit{mortal}(x) \leftarrow \mathit{man}(x) \wedge \mathit{highlander}(x) \wedge \mathit{keeps\_head}(x), 0.8)$
(3) $(\sim\mathit{keeps\_head}(x) \leftarrow \mathit{beheaded}(x), 1)$
(4) $(\mathit{keeps\_head}(x) \leftarrow \mathit{in\_the\_game}(x), 0.9)$

Set of facts $\mathcal{T}(A)$:
(5) $(\mathit{man}(\mathit{joe}), 1)$
(6) $(\mathit{man}(\mathit{duncan}), 1)$
(7) $(\mathit{man}(\mathit{connor}), 1)$
(8) $(\mathit{highlander}(\mathit{duncan}), 1)$
(9) $(\mathit{highlander}(\mathit{connor}), 1)$
(10) $(\mathit{beheaded}(\mathit{connor}), 1)$
(11) $(\mathit{in\_the\_game}(\mathit{duncan}), 1)$
(12) $(\mathit{in\_the\_game}(\mathit{connor}), 1)$

Fig. 3. First order P-DeLP program that interprets ontology Σ=(*T*,*A*)



Set of propositional rules that interprets the Tbox T:
(1.j) $(\mathit{mortal\_joe}) \leftarrow \mathit{man\_joe}, 0.6)$
(2.j) $(\sim\mathit{mortal\_joe} \leftarrow \mathit{man\_joe} \wedge \mathit{highlander\_joe} \wedge \mathit{keeps\_head\_joe}, 0.8)$
(3.j) $(\sim\mathit{keeps\_head\_joe} \leftarrow \mathit{beheaded\_joe}, 1)$
(4.j) $(\mathit{keeps\_head\_joe} \leftarrow \mathit{in\_the\_game\_joe}, 0.9)$
(1.d) $(\mathit{mortal\_duncan} \leftarrow \mathit{man\_duncan}, 0.6)$
(2.d) $(\sim\mathit{mortal\_duncan} \leftarrow \mathit{man\_duncan} \wedge \mathit{highlander\_duncan} \wedge \mathit{keeps\_head\_duncan}$
(3.d) $(\sim\mathit{keeps\_head\_duncan} \leftarrow \mathit{beheaded\_duncan}, 1)$
(4.d) $(\mathit{keeps\_head\_duncan} \leftarrow \mathit{in\_the\_game\_duncan}, 0.9)$
(1.c) $(\mathit{mortal\_connor} \leftarrow \mathit{man\_connor}, 0.6)$
(2.c) $(\sim\mathit{mortal\_connor} \leftarrow \mathit{man\_connor} \wedge \mathit{highlander\_connor} \wedge \mathit{keeps\_head\_connor},$
(3.c) $(\sim\mathit{keeps\_head\_connor} \leftarrow \mathit{beheaded\_connor}, 1)$
(4.c) $(\mathit{keeps\_head\_connor} \leftarrow \mathit{in\_the\_game\_connor}, 0.9)$

Set of propositions that interprets the Abox A:
(5) $(\mathit{man\_joe}, 1)$
(6) $(\mathit{man\_duncan}, 1)$
(7) $(\mathit{man\_connor}, 1)$
(8) $(\mathit{highlander\_duncan}, 1)$
(9) $(\mathit{highlander\_connor}, 1)$
(10) $(\mathit{beheaded\_connor}, 1)$
(11) $(\mathit{in\_the\_game\_duncan}, 1)$
(12) $(\mathit{in\_the\_game\_connor}, 1)$

Fig. 4. Propositionalization of the program presented in Fig. 3

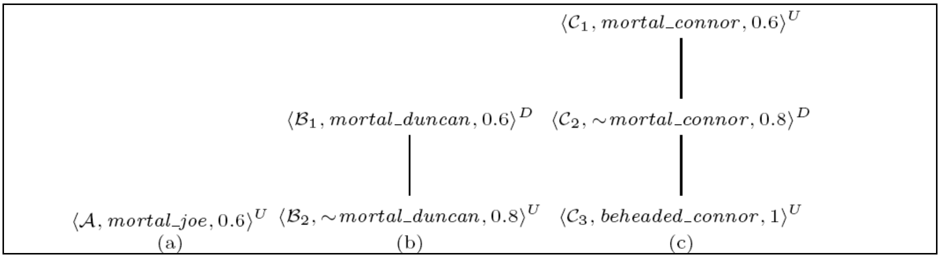

$\langle \mathcal{C}_1, mortal\_connor, 0.6\rangle^U$
$\langle \mathcal{B}_1, mortal\_duncan, 0.6\rangle^D$
$\langle \mathcal{C}_2, \sim mortal\_connor, 0.8\rangle^D$
$\langle \mathcal{A}, mortal\_joe, 0.6\rangle^U$
$\langle \mathcal{B}_2, \sim mortal\_duncan, 0.8\rangle^U$
$\langle \mathcal{C}_3, beheaded\_connor, 1\rangle^U$
(a)
(b)
(c)

Fig. 5. Dialectical trees for mortal_joe, mortal_duncan and mortal_connor

An Argumentation Framework with Backing and Undercutting

ANDREA COHEN, ALEJANDRO J. GARCÍA, GUILLERMO R. SIMARI

Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)
Artificial Intelligence Research and Development Laboratory (LIDIA)
Department of Computer Science and Engineering (DCIC)
Universidad Nacional del Sur (UNS), Bahía Blanca, Argentina
{ac, ajg, grs}@cs.uns.edu.ar

Abstract. *In this work we will combine two important notions for the argumentation community into Abstract Argumentation Frameworks (AFs). These notions correspond to Toulmin's backings and Pollock's undercutting defeaters. We will define Backing-Undercutting Argumentation Frameworks (BUAFs), an extension of AFs that includes a specialized support relation, a distinction between different attack types, and a preference relation among arguments. Thus, BUAFs will provide a more concrete approach to represent argumentative or non-monotonic scenarios where information can be attacked and supported.*

1. Introduction

The study of argumentation within the field of Artificial Intelligence has grown lately [3]. Briefly, argumentation is a form of reasoning where a claim is accepted or rejected according to the analysis of the arguments for and against it. Then, argumentation provides a reasoning mechanism where contradictory, incomplete and uncertain information may appear. In the last decade several approaches were proposed to model argumentation on an abstract basis [7], using classical logics [4], or using logic programming [8].

Argumentation models usually consider an argument as a piece of reasoning that provides a connection between some premises and a conclusion. Notwithstanding, in [13] Toulmin argued that arguments had to be analyzed using a richer format than the traditional one of formal logic. Whereas a formal logic analysis uses the dichotomy of premises and conclusion, Toulmin proposed a model for the layout of arguments that in addition to data and claim distinguishes four elements: warrant, backing, rebuttal and qualifier. However, Toulmin did not elaborate much on the nature of rebuttals, but simply stated that they provide conditions of exception for the argument. That is, without loss of generality, the notion of rebuttal can be paired to the notion of defeater for an argument, as proposed in the literature [12].

An important contribution to the field of argumentation which regards the nature of defeaters was proposed by Pollock. In [10] Pollock stated that defeasible reasons (which can be assembled to comprise arguments) have defeaters and that there are two kinds of defeaters: rebutting defeaters and undercutting defeaters. The former attack the conclusion of an inference by supporting the opposite one (*i.e.*, they are reasons for denying the conclusion), while the latter attack the connection between the premises and conclusion without attacking the conclusion directly.

In this work, we will combine the notions presented by Toulmin and Pollock into an abstract argumentation framework. We will incorporate Pollock's categorization of defeaters and the modeling of Toulmin's scheme elements, in particular, focusing in undercutting defeaters and backings. We will follow the approach of [6] in which Pollock's undercutting defeaters can be regarded as attacking Toulmin's warrants. Thus, Toulmin's backings can be regarded as aiming to defend their associated warrants against undercutting attacks, by providing support for them. In that way, we will be able to capture both attack and support for an inference, that is, for Toulmin's warrants.

We will extend Abstract Argumentation Frameworks (AFs) [7] to incorporate a specialized type of support and preference relation among arguments, as well as distinguishing between different types of attacks. In particular, the support relation will correspond to the support that Toulmin's backings provide for their associated warrants. On the other hand, we will distinguish three different types of attack within Dung's original attack relation, more specifically, rebutting attacks, undermining attacks and undercutting attacks; the former and the latter being related to rebutting and undercutting defeaters, as proposed by Pollock. The remaining type of attack we will consider corresponds to undermining defeaters, which are widely considered in the literature (see *e.g.* [11]) and originate from attacks to an argument's premise.

The rest of this work is organized as follows. Section 2 briefly reviews Dung's Abstract Argumentation Frameworks (AFs). In Section 3 we present the Backing-Undercutting Argumentation Frameworks (BUAFs), an extension of AFs that incorporates attack and support for inferences, as well as a preference relation to decide between conflicting arguments. In Section 4 we introduce the different types of defeat that can be obtained from a BUAF by applying preferences to the conflicting arguments as indicated by the attack relation. Later we define the requirements for conflict-free sets of arguments in a BUAF. Section 5 introduces some semantic notions, followed by the formal definitions of the acceptability semantics for BUAFs. Finally, in Section 6 some conclusions and related work are discussed.

2. Dung's Abstract Argumentation Frameworks

In this section we will briefly review Dung's Abstract Argumentation Frameworks, as defined in [7].

Definition 1. *An* Abstract Argumentation Framework (AF) *is a pair ⟨Args,* P⟩*, where Args is a set of arguments and* P ⊆ *Args*×*Args is an attack relation.*

Here, arguments are abstract entities that will be denoted using calligraphic uppercase letters. No reference to the underlying logic is needed since we are abstracting from argument's structure. The attack relation between two arguments A and B denotes the fact that these arguments cannot be accepted simultaneously since they contradict each other. We say that an argument A *attacks* an argument B iff (A,B) ∈ P, and it is noted as A → B. For instance, in the AF of Figure 1 A and B attack each other, B attacks X, and so on.

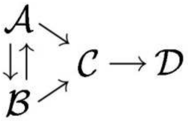


Fig. 1. A Dung's Abstract Argumentation Framework

Dung then defines the acceptability of arguments and the admissible sets of the framework.

Definition 2. *Let AF* = ⟨*Args*, P⟩ *and S* ⊆ *Args a set of arguments. Then:*

- *S is* conflict-free *iff* ʃ A, B ∈ *S s.t.* (A,B) ∈ P.
- A *is* acceptable *w.r.t. S iff* ∀B∈ *Args*: *if* (B,A) ∈ P *then* ∃ X ∈ *S s.t* .(X,B) ∈ P.
- *If S is conflict-free, then S is an* admissible set *of AF iff each argument in S is acceptable w.r.t. S.*

Intuitively, an argument A is acceptable w.r.t. *S* if for any argument B that attacks A there is an argument X in *S* that attacks B, in which case X is said to defend A. An admissible set *S* can then be interpreted as a coherent defendable position. For instance, in the AF of Figure 1, argument Δ is acceptable w.r.t. the sets {A}, {B} and {A, B}; however, only the first two of these sets are admissible.

Taking into account the notion of admissibility Dung then defines the acceptability semantics of the framework.

Definition 3. *Let AF* = ⟨*Args*, P⟩ *be an argumentation framework and S* ⊆ *Args a conflict-free set of arguments. Then:*

- *S is a* complete extension *of AF iff all arguments acceptable w.r.t. S belong to S.*
- *S is a* preferred extension *of AF iff it is a maximal (w.r.t. set-inclusion) admissible set (i.e., a maximal complete extension).*
- *S is a* stable extension *of AF iff it is a preferred extension that attacks all arguments in Args*\S.

- *S is the* grounded extension *of AF iff it is the smallest (w.r.t. set-inclusion) complete extension.*

The complete extensions of the framework in Figure 1 are ∅, {A, Δ} and {B,Δ}; the preferred and stable extensions are {A, Δ} and {B,Δ}; and the grounded extension is ∅.

3. Backing-Undercutting Argumentation Frameworks

A classical abstract argumentation framework is characterized by a set of arguments and an attack relation among them. In this section, we will introduce an extension of Dung's argumentation frameworks called Backing-Undercutting Argumentation Frameworks (BUAFs). In the extended framework we will: distinguish between different types of attack, incorporate a special kind of support relation, and include a preference relation among arguments. Thus, the BUAF will provide the means for representing both attack and support for an argument's inference, allowing to express Pollock's undercutting defeaters and Toulmin's backings.

Definition 4. (Backing-Undercutting Argumentation Framework). *A* Backing-Undercutting Argumentation Framework (BUAF) *is a tuple* ⟨A, Δ, B_k ,°⟩ *where:*

- A *is a set of arguments,*
- Δ ⊆ A×A *is a set of attacks,*
- B_k *is a backing relation, and*
- ° ⊆ A×A *is a preference relation.*

We will distinguish the three different types of attack in Δ, where the set of rebutting attacks is denoted as P_b, the set of undercutting attacks is denoted as Y_c, and the set of undermining attacks is denoted as Y_m ($\Delta = P_b \cup Y_c \cup Y_m$). In addition, when two arguments A and B are related by the preference relation (*i.e.* (A, B) ∈ °) it means that argument B is at least as preferred as argument A, denoting it as A ° B. Furthermore, following the usual convention, A π B means A ° B and B —— A.

From hereon, we may use the following notation:

- $A \dashrightarrow B$ denotes (A, B) ∈ Δ.
- $A \Longrightarrow B$ denotes (A, B) ∈ B_k.

In order to illustrate, let us consider one of Toulmin’s famous examples which discusses whether Harry is a British subject or not [13], as shown in Figure 2.

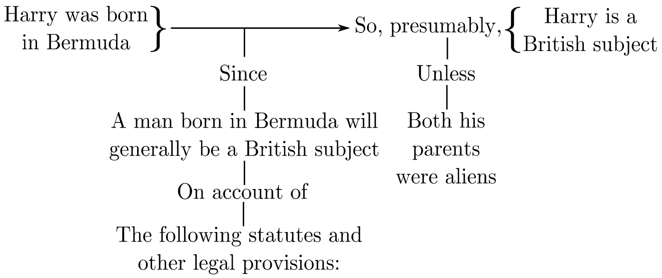

Harry was born in Bermuda
So, presumably,
Harry is a British subject
Since
Unless
A man born in Bermuda will generally be a British subject
Both his parents were aliens
On account of
The following statutes and other legal provisions:

Fig. 2. Toulmin's example about Harry

The following arguments can represent the situation depicted in Toulmin's example:

> H: *"Harry was born in Bermuda. A man born in Bermuda will generally be a British subject. So, Harry is a British subject"*
> B: *"On account of the following statutes and other legal provisions…"*
> Y: *"Both Harry's parents are aliens"*

Example 1. *A possible representation for Toulmin's example about Harry is given by the BUAF* $\Delta_1 = \langle A_1, \Delta_1, B_{k1}, ^\circ{}_1\rangle$, *where*

$$A_1 = \{H, B, Y\} \qquad B_{k1} = \{(B, H)\}$$
$$Y_{c1} = \{(Y, H)\} \qquad ^\circ{}_1 = \{(B, Y)\}$$

Here, that the statutes and other legal provisions provide support for the warrant is expressed by the pair (B, H) *in the backing relation. In addition, the fact that Harry's parents were aliens is considered as an undercut for the inference, as expressed by the pair* (Y, H) *in the attack relation.*

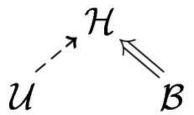


Fig. 3. The BUAF of Example 1

4. Defeat and Conflict-Freenes

Before defining any semantics-related notion, we must first consider the concept of defeat. Intuitively, given that in a BUAF there is a preference relation among arguments, an argument A would defeat an argument B iff A attacks B and A is not less preferred than B. Following this intuition, in this section we will define the notion of defeat in the context of a BUAF, where we will distinguish between two types of defeat. Then, we will define the basic restriction that an acceptable set of arguments in a BUAF must satisfy, that is, the notion of conflict-freenes for a set of arguments.

The first type of defeat we will distinguish is called *primary defeat* and is obtained directly by resolving the attacks given on the attack relation through the use of preferences. It is important to note that, in the case of undercutting attacks, the attacks will always succeed as defeats, like in [11]. On the other hand, for rebutting and undermining attacks we must compare the attacking and the attacked arguments in order to determine the existence of a defeat.

Definition 5 (Primary Defeat). *Let* $\langle \mathrm{A}, \Delta, \mathrm{B}_k, ^\circ\rangle$ *be a BUAF and* A, B ∈ A. *We will say that* A primary defeats B *iff one of the following conditions hold:*

- $(\mathrm{A}, \mathrm{B}) \in (\mathrm{P}_b \cup \mathrm{Y}_m)$ *and* $\mathrm{A} \equiv \mathrm{B}$, *or*
- $(\mathrm{A}, \mathrm{B}) \in \mathrm{Y}_c$.

Observe that in the above definition rebutting and undermining attacks are grouped together. This is because, given the abstract nature of arguments, we cannot distinguish an attack an argument's premise from an attack to its conclusion. Thus, the only way to determine the existence of a defeat in the presence of an undermining attack or a rebutting attack is to compare the attacking and attacked arguments. In contrast, for instance, if we had considered a notion of sub-argument the analysis for rebutting and undermining attacks would be different.

Example 2 *In the AF of Example 1, argument* Y *primary defeats argument* H.

As stated before, likewise [11], an undercutting attack will always result in defeat; however, in that approach the existence of arguments supporting an inference is not considered. Hence, following [6]'s approach, we will consider that backings are intended to defend their associated warrants against undercutting attacks. Therefore, it will be necessary to establish the relation between backing and undercutting arguments.

It is clear that backing and undercutting arguments are conflicting: while the latter attacks the connection between premises and conclusion of an argument, the former provides support for it. Thus, they should not be jointly accepted. Moreover, given that the conflict between backing and undercutting arguments may not always be explicit in the attack relation of a BUAF, it is necessary to ensure this acceptability restriction. To achieve this, we will define a new type of defeat called *implicit defeat*.

Definition 6 (Implicit Defeat). *Let* $\langle \mathrm{A}, \Delta, \mathrm{B}_k, ^\circ\rangle$ *be a BUAF and* A, B, X ∈ A. *We will say that* A implicitly defeats B *iff one of the following conditions hold:*

- $(\mathrm{A}, \mathrm{X}) \in \mathrm{Y}_c$ *and* $(\mathrm{B}, \mathrm{X}) \in \mathrm{B}_k$, *and* $\mathrm{A} \equiv \mathrm{B}$, *or*
- $(\mathrm{A}, \mathrm{X}) \in \mathrm{B}_k$ *and* $(\mathrm{B}, \mathrm{X}) \in \mathrm{Y}_c$, *and* $\mathrm{A} \equiv \mathrm{B}$.

Example 3 *Given the AF of Example 1, argument* Y *implicitly defeats argument* B.

Then, an argument will be defeated in a BUAF if it is primary or implicitly defeated.

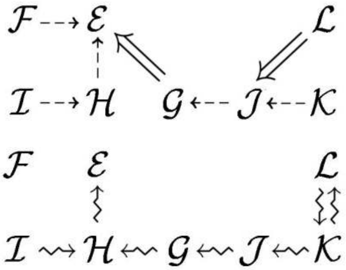
Definition 7 (Defeat). *Let* $\langle \mathrm{A}, \Delta, \mathrm{B}_k, ^\circ\rangle$ *be a BUAF and* A, B ∈ A. *Then* A defeats B, *noted as* A ∈ B, *iff* A *primary defeats or implicitly defeats* B.

From a BUAF Δ we can construct a directed graph called the *defeat graph*. The nodes in the graph are the arguments in Δ and the edges correspond the defeat relation obtained by Definition 7.

Example 4 *Consider the BUAF* $\Delta_2 = \langle \mathrm{A}_2, \Delta_2, \mathrm{B}_{k2}, ^\circ{}_2\rangle$, *where*

A_2 = {E, Φ, Γ, H, I, ϑ, K, Λ}　　Y_{m2} = {(I, H)}
P_{b2} = {(Φ, E), (ϑ, Γ)}　　B_{k2} = {(Γ, E), (Λ, ϑ)}
Y_{c2} = {(H, E), (K, ϑ)}　　$^\circ{}_2$ = {(Φ, E), (H, Γ), (Γ, ϑ), (ϑ, K)}

A graphical representation of Δ_2 *is shown below on the left and its corresponding defeat graph is shown on the right:*



The primary defeats obtained from Δ_2 *are* I ∈ H, H ∈ E, ϑ ∈ Γ *and* K ∈ ϑ*; and the implicit defeats are* Γ ∈ H, Λ ∈ K *and* K ∈ Λ.

Note that in Example 4 argument Γ is a backing for argument E, thus defending it against the undercut of H. In addition, argument I defeats argument H, becoming a defender for E. Notwithstanding, the nature of the defenses provided by Γ and I is different. The former is a backing for argument E, having the support between these two arguments explicitly determined by the backing relation; on the other hand, the latter merely defeats one of E's defeaters, in particular, the undercutting defeater H.

Next, conflict-free sets of arguments are characterized directly, by requiring the absence of defeats.

Definition 8 (Conflict-free Set). *Let* ⟨A, Δ, B_k ,°⟩ *be a BUAF. A set S* ⊆ A *is* conflict-free *iff* ⌠A, B ∈ *S s.t.* A ∈ B.

Example 5 *Given the BUAF of Example 4, some conflict-free sets of arguments are* ∅, {E} and {Φ, I, Γ, Λ}.

5. Acceptability Semantics

Since arguments in a BUAF can defeat each other, conflicting arguments should not be accepted simultaneously. Therefore, arguments in a BUAF will be subject to a status evaluation in which an argument will be accepted if it somehow "survives" the defeats it receives, or rejected otherwise. This evaluation process will be determined by the acceptability semantics.

In this section, we will define the basic semantic notions required for obtaining the set of acceptable arguments. Then, we will formally define the acceptability semantics for BUAFs. Finally, a characterization of BUAFs as Dung's AFs is presented, establishing the relation between these two frameworks.

Definition 9 (Acceptability). *Let* ⟨A, Δ, B_k ,°⟩ *be a BUAF. An argument* A ∈ A *is* acceptable *w.r.t. S* ⊆ A *iff* ∀B ∈ A *s.t.* B ∈ A, ∃X ∈ *S s.t.* X∈ B.

Intuitively, an argument A will be acceptable with respect to a set of arguments *S* iff *S* defends A against all its defeaters.

Example 6 *In the BUAF of Example 4, the argument* E *is acceptable w.r.t. the sets* {I}, {Φ, Γ}, {I, ϑ, K} *and* {Φ, I, Γ, K} *among others.*

In the literature, a usual requirement when defining the set of acceptable arguments of an AF is the conflict-freenes of the set (see *e.g.*, [7, 2]). This implies that the set of collectively acceptable arguments must be internally coherent, in the sense that no pair of arguments in the set defeats each other. Thus, it is reasonable to accept only those arguments that are acceptable. We will follow this approach and therefore, the set of accepted arguments in a BUAF will be the set of arguments that defends itself against all defeats on it, leading to a classical definition of admissibility for BUAFs.

Definition 10 (Admissibility). *Let* ⟨A, Δ, B_k ,°⟩ *be a BUAF. A set S* ⊆ A *is* admissible *iff it is conflict-free and all elements of S are acceptable w.r.t. S.*

Example 7 *From the sets of arguments listed in Example 6, only the sets* {I} *and* {Φ, I, Γ, K} *are admissible.*

Recall that acceptability semantics identify a set of extensions of an argumentation framework, namely sets of arguments which are collectively acceptable. The complete, preferred, stable and grounded extensions of a BUAF are now defined in the same way as for Dung's frameworks.

Definition 11 (Extensions). *Let* Δ = ⟨A, Δ, B_k ,°⟩ *be a BUAF and S* ⊆ A *a conflict-free set of arguments. Then:*

- *S is a* complete extension *of* Δ *iff all arguments acceptable w.r.t. S belong to S.*
- *S is a* preferred extension *of* Δ *iff it is a maximal (w.r.t. set-inclusion) admissible set of* Δ *(i.e., a maximal complete extension).*
- *S is a* stable extension *of* Δ *iff it is a preferred extension that defeats all arguments in* A\S.
- *S is the* grounded extension *of* Δ *iff it is the smallest (w.r.t. set-inclusion) complete extension.*

Given a BUAF and a semantics *s*, an argument A is *skeptically accepted* if it belongs to all *s*-extensions; A is *credulously accepted* if it belongs to some (not all) *s*-extensions; and A is *rejected* if it does not belong to any *s*-extension.

Example 8 *From the BUAF of Example 4, we can obtain the following sets of extensions:*

- *the complete extensions* {Φ, I, E}, {Φ, I, E, Γ}, {Φ, I, E, ϑ}, {Φ, I, E, Γ, K} *and* {Φ, I, E, ϑ, Λ};
- *the preferred and stable extensions* {Φ, I, E, Γ, K} *and* {Φ, I, E, ϑ, Λ}; *and*
- *the grounded extension* {Φ, I, E}.

Definitions 9, 10 and 11 correspond to those presented for Dung's argumentation frameworks. Recall that a classical argumentation framework is characterized by a set of arguments and an attack relation among them. Thus, using the defeat relation from Definition 7 and the set of arguments of a BUAF we can characterize an abstract argumentation framework which accepts exactly the same arguments as the BUAF under a given semantics.

Proposition 1. *Let* Δ = ⟨A, Δ, B_k ,°⟩ *be a BUAF. There exists an abstract argumentation framework AF* = ⟨A, ∊⟩ *such that the sets of extensions of* Δ *and AF under a given semantics are equal.*

Proof. Straightforward from definitions 2, 3, 9, 10 and 11.

,

Therefore, by Proposition 1, BUAFs will inherit all properties from abstract argumentation frameworks (refer to [7] for details). Moreover, it will be possible to determine the acceptability of arguments in a BUAF using its associated Dung's AF. We first obtain the associated AF and then, acceptability semantics are applied to this AF.

6. Conclusions and Related Work

In this work, an extension of Abstract Argumentation Frameworks called Backing-Undercutting Argumentation Frameworks (BUAFs) was proposed, inspired by the work of Pollock [10] and Toulmin [13]. This extension allows to express attack and support for an inference by distinguishing different types of attacks and incorporating a specialized support relation among arguments. In that way, the extended framework enables the representation of Toulmin's backings and Pollock's undercutting defeaters, two important notions in the argumentation community. Several approaches address these two notions separately, yet they were not widely considered together in the formalizations provided so far. For instance, in [11] an extension of AFs is presented, where arguments are partly provided of an internal structure and a categorization of defeaters is also given; however, in that work there is no consideration for support among arguments.

Likewise [1], our approach incorporates a preference relation among arguments in order to determine the success of attacks. Other works that consider preferences among arguments include [9] and [2], but the difference between those approaches and ours is that they express preferences in the object level, by incorporating attacks to attacks.

Among other approaches that address support between arguments, in addition to the attack relation, are the Bipolar Argumentation Frameworks (BAFs) [5]. A Bipolar Argumentation Framework extends Dung's framework to incorporate a support relation between arguments. The main difference between BAFs and BUAFs is that the support relation in a BAF is general, while the backing relation proposed in this work corresponds to the specific support relation between Toulmin's backings and warrants. Therefore, the implicit defeats as presented in Definition 6 could not be modeled in BAFs. On the other hand, some additional requirements for an admissible set of arguments are considered in [5], such as external coherence or consistency. Although for BUAFs we have only considered the conflict-freenes (internal consistency) of the set, those requirements are also satisfied by the notion of admissibility given in Definition 10; however, a detailed explanation is left for future work.

Finally, it was shown that BUAFs can be mapped to AFs by considering the set of arguments and the corresponding defeat relation. Thus, it is clear that the examples and applications shown for BUAFs can also be modeled with Dung's abstract frameworks. Notwithstanding, besides formalizing the backing relation and different types of attack, BUAFs will provide a more concrete approach to represent argumentative or non-monotonic scenarios where inferences can be attacked and supported.

References

1. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Ann. Math. Artif. Intell. 34(1-3), 197-215 (2002).
2. Baroni, P., Cerutti, F., Giacomin, M., Guida, G.: AFRA: Argumentation framework with recursive attacks. International Journal of Approximate Reasoning 52(1), 19-37 (2011).
3. Bench-Capon T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artificial Intelligence 171(10-15), 619-641 (2007).
4. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artificial Intelligence 128(1-2), 203-235 (2001).
5. Cayrol, C., Lagasquie-Schiex, M.C.: Bipolar abstract argumentation systems. In: Simari, G.R., Rahwan, I. (eds.) Argumentation in Artificial Intelligence, pp. 65-84. Springer US (2009).
6. Cohen, A., García, A.J., Simari, G.R.: Backing and undercutting in defeasible logic programming. In: ECSQARU. pp. 50-61 (2011).
7. Dung, P.M: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321-358 (1995).
8. García, A.J., Simari, G.R.: Defeasible logic programming: An argumentative approach. Theory and Practice of Logic Programming 4(1-2), 95-138 (2004).
9. Modgil, S.: Reasoning about preferences in argumentation frameworks. Artificial Intelligence 173(9-10), 901-934 (2009).
10. Pollock, J.L.: Defeasible reasoning. Cognitive Science 11(4), 481-518 (1987).
11. Prakken, H.: An abstract framework for argumentation with structured arguments. Argument and Computation 1, 93-124 (2009).
12. Prakken, H., Vreeswijk, G.: Logics for defeasible argumentation. In: Gabbay, D., Guenthner, F. (eds.) Handbook of Philosophical Logic, vol. 4, pp. 218-319. Kluwer Academic Pub. (2002).
13. Toulmin, S.E.: The Uses of Argument. Cambridge University Press (1958).

Face Recognition using SIFT descriptors and Binary PSO with velocity control

JUAN ANDRÉS MAULINI, LAURA LANZARINI

III-LIDI (Institute of Research in Computer Science LIDI)
Faculty of Computer Science, National University of La Plata
La Plata, Buenos Aires, Argentina
{ jmaulini, laural }@lidi.info.unlp.edu.ar

Abstract. *In this paper, a strategy for face recognition based on SIFT descriptors of the images involved is presented. In order to reduce the number of false positives and computation time, a selection of the most representative feature descriptors is carried out by applying a variation of the binary PSO method. This version improves its operation by a suitable positioning of the velocity vector. To achieve this, a new modified version of the continuous gBest PSO algorithm is used. The results obtained allow stating that the descriptors can be successfully selected through the strategy proposed solving the problems initially mentioned.*

Keywords. *Face Recognition, SIFT descriptors, Swarm Intelligence, Binary PSO, Velocity Control.*

1. Introduction

Face recognition is a biometric technique that is widely used in various areas such as security and access control, forensic medicine, and police controls. It involves determining if the image of the face of any given person matches any of the face images stored in a database. This problem is hard to solve automatically due to the changes that various factors, such as facial expression, aging and even lighting, can cause on the image.

In this paper, a method using only those SIFT descriptors that best represent the image is proposed, and good recognition results are achieved while solving the two major problems of this characterization method: false positive detection and the time required for the recognition process. The selection of SIFT descriptors is carried out by means of a variation of binary PSO (Particle Swarm Optimization), and it is applied only to database image descriptors; therefore, SIFT descriptors processing is done before the recognition stage of the process.

This paper is organized as follows: In Section 2, a brief description of previous related works using similar techniques is included. In section 3, the basic components of the PSO algorithm, both in its continuous and binary versions, are described. In Section 4 some clarifications regarding the binary PSO variation used are presented; whereas in Section 5, the method that allows obtaining SIFT descriptors from an image is described. In Section 6, implementation details are provided, and in Section 7 the results obtained are described. Finally, in Section 8 the conclusions obtained are presented.

2. Related work

There are currently various solutions to this problem that use SIFT descriptors. It has been shown [1] that using SIFT descriptors for the face recognition process is better than Eigenfaces and Fisherfaces algorithms. Training datasets were of various sizes, which allowed establishing that performance decreases as dataset size decreases. As regards the significant number of SIFT descriptors required for a reliable comparison, it was observed that, with a lower number of descriptors, performance is better than that obtained with Eigenfaces and Fisherfaces.

In order to tackle the issue of comparing very long feature vectors for all images in a database, a biased classification of the features that make SIFT descriptors, is proposed and used to reduce the length of SIFT descriptors used for face recognition [2]. Thus, the number of comparisons is reduced and the recognition process is faster. This process also filters out those descriptors that are irrelevant for face recognition, thus increasing recognition accuracy.

On the other hand, a face recognition algorithm that uses the binary PSO algorithm to explore the solution space for an optimum subset of features in order to increase recognition rate and class separation is presented in [3]. This algorithm is applied to feature vectors extracted using the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT).

3. Particle swarm optimization

3.1 Continuous particle swarm optimization

In PSO, each individual represents a possible solution to the problem and adapts following three factors: its knowledge of the environment (its fitness value), its previous experiences (its memory), and the previous experiences of the individuals in its neighborhood [4]. In this type of technique, each individual is in continuous movement within the search space and never dies.

Each particle is composed by three vectors and two fitness values:

- Vector $x_i=(x_{i1},x_{i2},\ldots,x_{in})$ stores the current position of the particle
- Vector $pBest_i=(p_{i1},p_{i2},\ldots,p_{in})$ stores the best solution found for the particle
- Velocity vector $v_i=(v_{i1},v_{i2},\ldots,v_{in})$ stores the gradient (direction) based on which the particle will move.
- The fitness value $fitness\_x_i$ stores the suitability value of the current solution.

- The fitness value $fitness\_pBest_i$ stores the suitability value of the best local solution found so far (vector $pBest_i$)

The position of a particle is updated as follows:

$$x_i(t+1)=x_i(t)+v_i(t+1) \tag{1}$$

As explained above, the velocity vector is modified taking into account its experience and environment. The expression is:

$$v_i(t+1)=w.v_i(t)+\square_1.rand_1.(p_i-x_i(t))+\square_2.rand_2.(g_i-x_i(t)) \tag{2}$$

where w represents the inertia factor [5], $\square_1$ and $\square_2$ are acceleration constants, $rand_1$ and $rand_2$ are random values belonging to the (0,1) interval, and g_i represents the position of the particle with the best *pBest* fitness in the environment of x_i (*lBest* or *localbest*) or the entire swarm (*gBest* or *globalbest*). The values of w, $\square_1$ and $\square_2$ are important to ensure the convergence of the algorithm. For detailed information regarding the selection of these values, please see [6] and [7].

3.2 Binary particle swarm optimization

PSO was originally developed for a space of continuous values and it therefore poses several problems for spaces of discrete values where the variable domain is finite. Kennedy and Eberhart [8] presented a discrete binary version of PSO for these discrete optimization problems.

In binary PSO, each particle uses binary values to represent its current position and the position of the best solution found. The velocity vector is updated as in the continuous version, but determining the probability that each bit of the position vector becomes 1. Since this is a probability, the velocity vector should be mapped in such a way that it only contains values within the [0,1] range. To this end, the sigmoid function indicated in (3) is applied to each of its values.

$$v'_{ij}(t)=sig(v_{ij}(t))=\frac{1}{1+e^{-v_{ij}(t)}} \tag{3}$$

Then, the particle position vector is updated as follows

$$x_{ij}(t+1)=\begin{cases} 1 if\ rand_{ij}<sig(v_{ij}(t+1)) \\ 0 if\ not \end{cases} \tag{4}$$

where $rand_{ij}$ is a number ramdomly generated by an uniform pdf in [0,1].

It should be mentioned that the incorporation of the sigmoid function radically changes the way in which the velocity vector is used to update the position of the particle. In continuous PSO, the velocity vector takes on higher values first to facilitate the exploration of the solution space, and then reduces them to allow the particle to stabilize. In binary PSO, the opposite procedure is applied. Each particle increases its exploratory ability as the velocity vector reduces its value; that is, when v_{ij} tends to zero, $\lim_{t\rightarrow\infty} sig(v_{ij}(t))=0.5$, thus allowing each binary digit to take a value of 1 with a probability of 0.5. This means that it could take on either value. On the contrary, when the velocity vector value increases, $\lim_{t\rightarrow\infty} sig(v_{ij}(t))=1$, and therefore all bits will change to 1, whereas when the velocity vector value decreases, taking negative values, $\lim_{t\rightarrow\infty} sig(v_{ij}(t))=0$ and all bits will change to 0. It should be noted that, by limiting the velocity vector values between -3 and 3, $sig(v_{ij})\in[0.0474,0.9526]$, whereas for values above 5, $sig(v_{ij})\simeq 1$ and for values below -5, $sig(v_{ij})\simeq 0$.

4. Binary PSO with velocity control

Based on the observations of the behavior of the velocity vector in the binary PSO algorithm defined in [8], and on the importance of correctly calculating the probabilities that allow changing each binary digit, a modified version of the original PSO algorithm to modify the velocity vector is proposed.

Under this new scheme, each particle will have two velocity vectors, $v1$ and $v2$. The first one is updated according to (5).

$$v1_i(t+1)=w.v1_i(t)+\square_1.rand_1.(2*p_i-1)+\square_2.rand_2.(2*g_i-1) \quad (5)$$

where the variables $rand_1$, $rand_2$, $\square_1$ and $\square_2$ operate in the same way as in (2). The values p_i and g_i correspond to the i^{th} binary digit of the $pBest_i$ and $gBest$ vectors, respectively.

The most significant difference between (2) and (5) is that in the latter, the shift of vector $v1$ in the directions corresponding to the best solution found by the particle and the best global solution does not depend on the current position of the particle. Then, each element of the velocity vector $v1$ is controlled by applying (6)

$$v1_{ij}(t)=\begin{cases}\delta 1_j & if\, v1_{ij}(t)>\delta 1_j\\ -\delta 1_j & if\, v1_{ij}(t)\leq -\delta 1_j\\ v1_{ij}(t) if\, not\end{cases} \tag{6}$$

where

$$\delta 1_j=\frac{limit1_{upper_j}-limit1_{lower_j}}{2} \tag{7}$$

That is, velocity vector $v1$ is calculated with (5) and controlled with (6). Its value is used to update velocity vector $v2$, as shown in (8).

$$v2(t+1)=v2(t)+v1(t+1) \tag{8}$$

Vector $v2$ is also controlled as vector $v1$ by changing $limit1_{upper_j}$ and $limit1_{lower_j}$ by $limit2_{upper_j}$ and $limit2_{lower_j}$, respectively. This will yield $\delta 2_j$, which will be used as in (6) to limit the values of $v2$. Then, the new position of the particle is calculated with (4) using the values of v2 as arguments of the sigmoid function.

The results of this method compared with [9] and [8] applied in function optimization can be consulted in [2].

5. SIFT Descriptors

In [10], Lowe defined a method to extract features from an image and use them to find matches between two different views of the same object. These features, called SIFT (Scale Invariant Feature Transform) features, are invariant to image scale and rotation, and quite invariant to affine distortion, as well as changes in point of view and lighting. They are also highly distinctive.

The process to determine SIFT features for an image consists in four steps:

- First, the location of potential points of interest within the image is determined. These points of interest correspond to the extreme points calculated from plane subsets of Difference of Gaussian (DoG) filters applied to the image at different scales.

- Then, the points of interest whose contrast is low are discarded. This is an improvement from the definition in [11].

- After this, the orientation of relevant points of interest is calculated.

- Using the previous orientations, the environment is analyzed for each point and the corresponding feature vector is determined.

As a result of this process, a set of 128-length feature vectors that can be compared with those from another image of the same object with a different scale, orientation, and/or point of view, is obtained.

This comparison can be done directly by measuring the distance and establishing a similarity threshold.

More detailed information about this method is available in [10].

6. Face Recognition

In order to perform face recognition, the method proposed uses a minimum-size database formed by the subset of most representative SIFT descriptors. Thus, the computing time required to make the necessary comparisons and detection of false positives are reduced. This selection process is performed before the recognition process; therefore, it does not affect the response time for the end user. Section 6.1. details how to make this selection.

The recognition of a new face involves the following steps:

- Calculating the SIFT vectors corresponding to the input image
- Comparing each vector in the database with the set of vectors corresponding to the new face, matches being accumulated not by image but rather by the number of the person to whom the database vector corresponds.
- The new face will correspond to the person with the highest number of accumulated matches.

It should be noted that the comparison of each database descriptor with the set of descriptors corresponding to the image to be recognized is a purely parallel task. If a parallel computation architecture were available, the database of SIFT descriptors could be partitioned so that each processor would have the information corresponding to one person, or, even better, to one image. Thus, the calculation of the number of matches found would be faster.As regards the recognition of the new face, a minimum threshold of matches can be used to identify faces that have no matches in the database.

6.1 Building the database

The method begins by obtaining all SIFT descriptors corresponding to each input image. The selection of the most representative SIFT descriptors is carried out by applying a variation of the method described in section 5, based on subpopulations of particles. In this case, the number of populations to use matches the number of images in the database.

The length of the position vector for each particle of a population is determined by the number of SIFT descriptors of the corresponding image. Therefore, the length of particles from different populations can be different.

That is, the vector of the j^{th} particle in the subpopulation i, has the following form

$$X_j^i=(x_{j1}^i,x_{j2}^i,\ldots,x_{jm_i}^i) \tag{9}$$

where m_i is the number of SIFT descriptors of image i and x_{jk}^i is 1 if the k^{th} SIFT descriptor must be included in the data base and 0 if not.

This speciation criterion allows calculating the movement of each particle using only the SIFT descriptors from one image. Thus, each population searches a different part of the solution space. The final solution is obtained by concatenation the best individuals of each population. This can be expressed as follows

$$X=(X_{best}^1,X_{best}^2,\ldots,X_{best}^M) \tag{10}$$

where M is the number of different images used to form the database and Xibest is the best individual in the ith subpopulation.

With respect to the usual parameters of PSO: In each iteration, the value of w decreases, as mentioned in [8] and elitism was used so that, if moving individuals does not allow at least maintaining the highest fitness value found thus far, the best individual of the previous iteration regains its previous position and the fitness value lost. The algorithm terminates when the maximum number of iterations was reached or when after a certain number of consecutive iterations the best fitness value has not changed.

6.2 Assessing the fitness value of each particle

In this section, the method used to measure the fitness value for each particle is described. An expression that helps reducing the number of false positives must be used. Therefore, its value increases when the selected descriptor has a match in an image of the corresponding subject, and it decreases when there are no matches.

Be X_{ij} the position vector of the j^{th} particle of sub-population i, defined in (9). Be $C1_{jk}^i$ the total number of matches between the k^{th} SIFT descriptor of image i and the rest of the images that correspond to the subject represented by image i. Be $C2_{jk}^i$ the total number of matches between the k^{th} SIFT descriptor of image i and the images that correspond to subjects other than that represented by image i. The fitness value of the j^{th} particle of sub-population i is calculated as

$$Fit_j^i=\sum_{k=1}^{m}x_{jk}^i*(\alpha_1*C1_{jk}^i-\alpha_2*C2_{jk}^i) \tag{11}$$

where α_1 and α_2 are constants with values between (0,1) and represent the significance of each term within the expression. As said above, x_{jk}^{i} is 1 if the k^{th} SIFT descriptor must be included in the data base and 0 if not.

7. Results Obtained

Measurements were carried out using two databases obtained from [12]. The first of these is the YALE faces database, containing 165 images of 15 different subjects (11 images per person). Each image has a resolution of 320x243 pixels. The second database used was the *AT&T* faces database, containing 400 images of 40 people (10 images per individual). The size of each image is 112x92 pixels. The available images were divided in two parts: Subset of input images, whose descriptors will be selected by applying the method proposed in Section 5 and subset of test images that will be compared with the selected SIFT descriptors for recognition.

The initial SIFT descriptors for each image were determined with a threshold of 0.5, as recommended in [11]. In both cases, the parameters used by PSO were the following: Initial and final inertia values: 1.2 and 0.2, respectively; maximum number of iterations = 500, α_1= 1/(number of input images), α_2= 16/(number of input images).

Thirty-five independent runs of the process described in Section 6 were carried out, varying the percentage of images used to form the base. Figure 1shows the average percentage of correct matches calculated over the test images. It can be seen that, in both cases, the selection of SIFT descriptors using PSO favors the recognition process and yields a higher success rate.

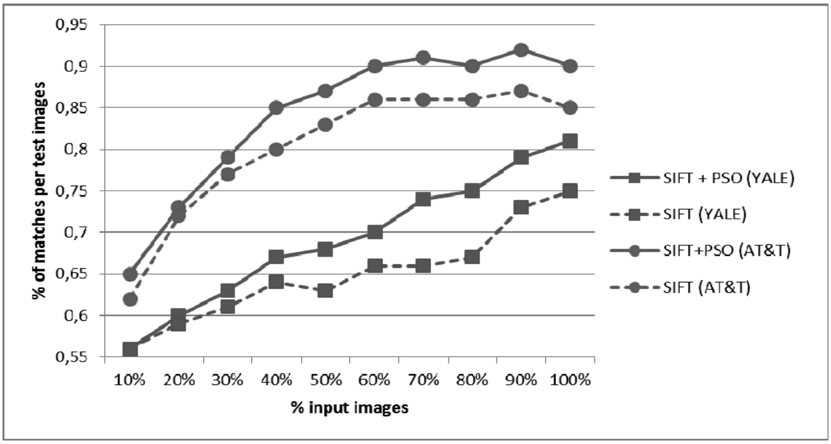


Fig. 1. Percentage of matches for test images using the method proposed (SIFT+PSO) and the original SIFT method for various percentages of images from the YALE and AT&T databases

Another aspect that should be taken into account is the accuracy of the response obtained. This is related to the similarity between each SIFT descriptor of the image to classify and the descriptors stored in the base. In order to be able to state with certainty that the result corresponds to a given image, it is important that there is a significant difference between the two best candidates found. Figure 2 shows that the average differences between the two best solutions found are greater if descriptors are selected using PSO. This allows stating that the response of the classification is more conclusive than using directly all SIFT descriptors identified by Lowe's method.

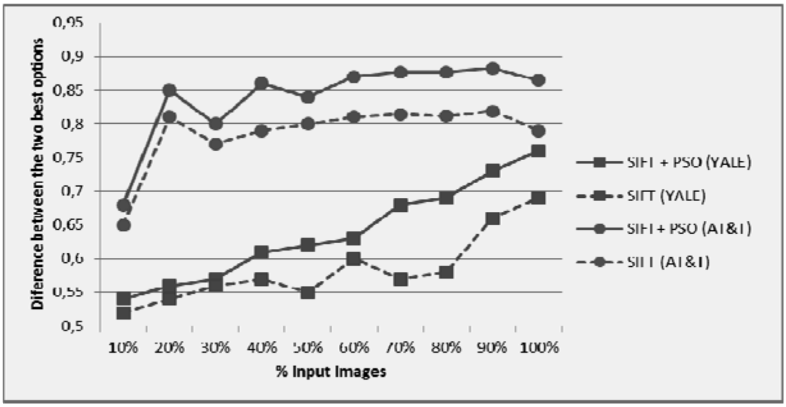


Fig. 2. Average value per image of the difference between the two highest values of correct matches, divided by the total number of matches found for the YALE and AT&T databases

Finally, Figure 3 shows the average number of SIFT descriptors used for each image in the base. It can be observed that, even though the reduction in the number of descriptors is greater for YALE than for AT&T, it is significant in both cases.

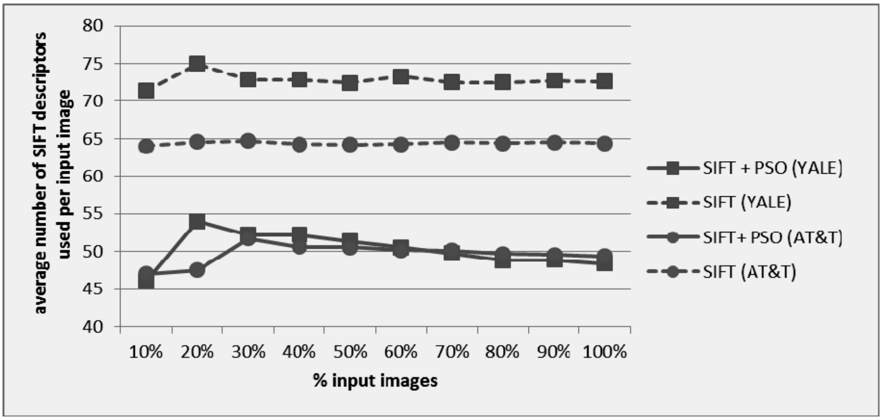


Fig. 3. Average number of SIFT descriptors used for each image in the YALE and AT&T databases

Figure 4 shows the original SIFT descriptors on the top row of images and descriptors selected by the proposed algorithm in the bottom row.

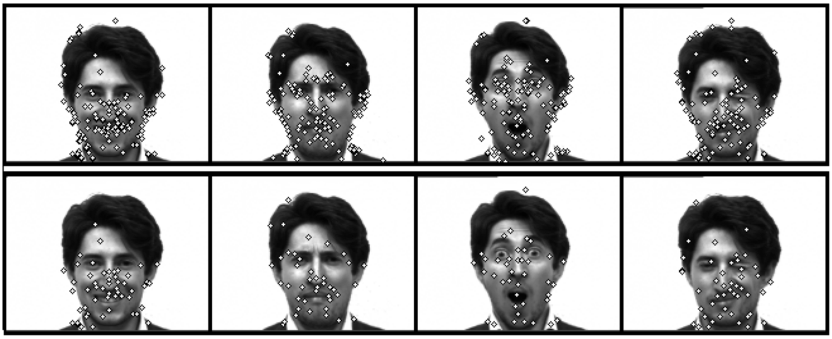


Fig. 4. SIFT descriptors of a person of the YALE database. The top row shows all descriptors found while the bottom row shows only the descriptors selected by the proposed method

8. Conclusions

A face recognition mechanism based on SIFT features that allows reducing the size of the database by using a variation of binary PSO has been described. The tests carried out with the YALE and AT&T databases have allowed reaching considerable reduction rates (50% in YALE and 25% in AT&T).

Even though the success rate for each test image using the base of descriptors selected with PSO is slightly higher than the one obtained with the process that uses all SIFT descriptors, the proportion of false positives is lower. Additionally, the smaller size of the database allows ensuring a clear reduction in the time needed for the recognition.

The parameters involved still need to be thoroughly analyzed in order to determine if a more precise adjustment would allow reducing the maximum number of iterations needed to reach an optimum selection of descriptors. The parallelization of the solution proposed also poses an interesting analysis.

References

1. Aly Mohamed. Face recognition using sift features. CNS186 Term Project, 2006.
2. Lanzarini L., López J., Maulini J., and De Giusti A. A new binary pso with velocity control. In Advances in Swarm Intelligence, Part I, volume 6728, pages 111-119. Lecture Notes in Computer Science. Springer, 2011.

3. R. Ramadan and R. Abdel Kader. Face recognition using particle swarm optimization-based selected features. International Journal of Signal Processing, Image Processing and Pattern Recognition, 2(2): 51-65, 2009.
4. J. Kennedy and R. Eberhart. Particle swarm optimization. IEEE International Conference on Neural Networks, IV:1942-1948, 1995.
5. Shi Y. and Eberhart R. Parameter selection in particle swarm optimization. 7th International Conference on Evolutionary Programming, pages 591-600, 1998.
6. Kennedy J. Clerc M. The particle swarm-explosion, stability and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation., 6(1):58-73, 2002.
7. Van den Bergh F. An analysis of particle swarm optimizers. Ph.D. dissertation. Department Computer Science. University Pretoria. South Africa, 2002.
8. Kennedy J. and Eberhart R. A discrete binary version of the particle swarm algorithm. World Multiconference on Systemics, Cybernetics and Informatics (WM-SCI), pages 4104-4109, 1997.
9. Shoorehdeli M. Khanesar M., Teshnehlab M. A novel binary particle swarm optimization. 18th Mediterranean Conference on Control and Automation., pages 1-6, 2007.
10. David G. Lowe. Distinctive image features from scale-invariant keypoints. International. journal of computer vision, 60, 2004.
11. D.G. Lowe. Object recognition from local scale-invariant features. In International Conference on Computer Vision, pages 1150-1157, 1999.
12. Face recognition homepage. www.face-rec.org/databases.

XI

Distributed and Parallel Processing Workshop

Performance Analysis of a Symmetric Cryptographic Algorithm on Multicore Architectures

ADRIÁN POUSA[1], VICTORIA SANZ[1], ARMANDO DE GIUSTI[1]

[1] Instituto de Investigación en Informática LIDI - School of Computer Science
National University of La Plata
{apousa, vsanz, degiusti}@lidi,info.unlp.edu.ar

Abstract. *In this paper, a performance analysis of the symmetric encryption algorithm AES (Advanced Encryption Standard) on various multicore architectures is presented. To this end, three implementations based on C language that use the parallel programming tools OpenMP, MPI and CUDA to be run on multicore processors, multicore clusters and GPU, respectively, were carried out. The efficiency obtained by the CUDA implementation of the algorithm as input data size increases is shown.*

Keywords: *multicore architectures, parallel programming, AES, OpenMP, MPI, CUDA, GPGPU.*

1. Introduction

The emergence of multicore architectures [1] [2] favors the use of parallel programming tools [3] [4], such as OpenMP [5] and MPI [6], to exploit the power that these architectures offer.

In recent years, GPUs (Graphic Processing Unit) [7] have gained significance due to the high performance achieved in general-purpose applications.

On the other hand, the volume of data that are transmitted through the networks has increased considerably. This information is occasionally sensitive, so it is important to encode the data to send them in a safe way through a public network such as Internet. Data encryption and decryption requires additional computation time, which can be considerable depending on data size.

AES (Advanced Encryption Standard) is a symmetrical block encryption algorithm that became a standard in 2002 [8], and is currently the most widely used algorithm to encode information. In 2003, the government of the United States announced that the algorithm was secure enough and that it could be used for the protection of national information [9]. So far, no efficient attacks are known, the only known attacks are those known as side-channel attacks[1] [10] [11] [12].

This algorithm is characterized for being simple, fast, and consuming little resources. However, the time required to encrypt and decrypt large amounts of data is significant; the possibilities offered by multicore architectures can be exploited to reduce this time.

The purpose of this paper is showing the information encryption computation speed-up with the AES algorithm taking advantage of different multicore architectures:

- Memory-sharing multicore processors [13]. In this case, the algorithm was implemented using OpenMP.
- Multicore cluster [13]. The tool used in this case is MPI.
- GPU. The algorithm was implemented using CUDA [14] [15].

In the following section, a performance analysis is carried out to show the efficiency of the implementation done for GPU.

2. Overview of the AES Algorithm

AES (Advanced Encryption Standard) is a symmetric encryption algorithm that became a standard in the year 2002, being one of the most widely used algorithms nowadays.

It is characterized for being a block-encryption algorithm. The data to be encrypted are divided in fix-sized blocks (128 bits), where each block is represented as a matrix of 4x4 bytes called *state*, as shown in Figure 1.

AE	03	1F	2A	1E	3F	01	7A	21	04	CF	7A	1C	33	11	27

128-bit block

AE	1E	21	1C
03	3F	04	33
1F	01	CF	11
2A	7A	7A	27

State

Fig. 1. AES State

[1] A side channel attack does not attack the encryption algorithm, it rather exploits implementation vulnerabilities that can reveal data as the encryption is carried out.

Each *state* goes through eleven rounds of transformation, each of them formed by a set of operations. The eleven rounds can be classified into three types: an initial round, nine standard rounds, and a final round, as detailed in Figure 2.

Since AES is a symmetric algorithm, it uses the same key to encrypt and decrypt the data; the size of this key is 128 bits as indicated by the standard. This key is called *initial key*, and it is used to generate ten more keys by means of a mathematical procedure. The ten resulting keys, together with the *initial key*, are called *subkeys* and they are used one in each of the rounds.

The initial round performs only one operation:

> *AddRoundKey*: a byte by byte XOR between the *state* and the initial key is performed.

Each of the following nine rounds, called *standard* rounds, applies 4 operations in this order:

> *SubBytes*: each state byte is replaced by another one taken from a byte-substitution table whose elements are pre-computed. The replacement byte is obtained by accessing the table taking the first 4 bits of the byte to be substituted as the row index and the last 4 bits as column index. The size of the table is 16x16 bytes.
>
> *ShiftRows*: with the exception of the first state row, which is not modified, the bytes on the remaining rows are cyclically rotated to the left: once for the second row, twice for the third, and three times for the fourth.
>
> *MixColumns*: a linear transformation is applied to each state column, and the column is substituted by the result of this operation.
>
> *AddRoundKey*: this is the same as the initial round, but using the following subkey.

The final round is formed by 3 operations:

> *SubBytes*: similar to the standard rounds.
>
> *ShiftRows*: similar to the standard rounds.
>
> *AddRoundKey*: the same as in the previous rounds, but using the last subkey.

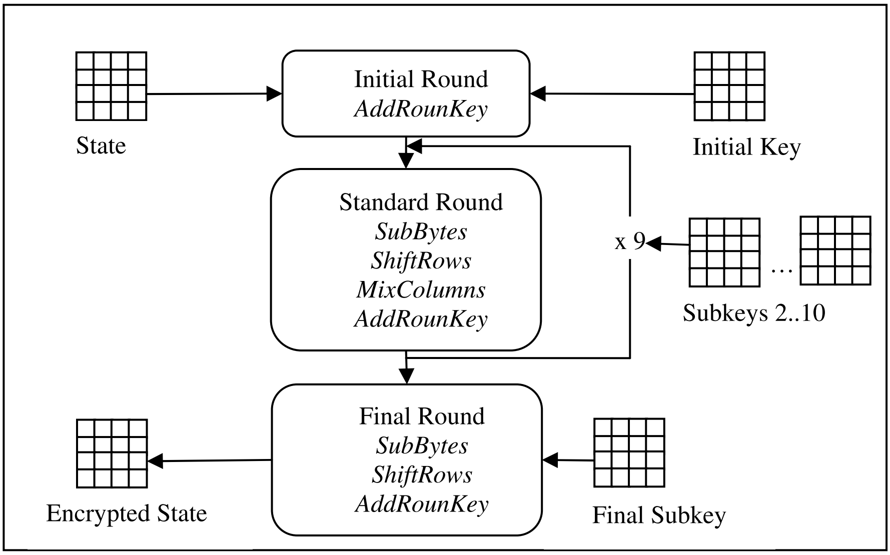


Fig. 2. Rounds of the AES algorithm applied to a state

3. Implementations of the AES Algorithm

Four implementations of the algorithm were carried out – one sequential implementation and the other three using different parallel programming tools such as OpenMP, MPI and CUDA.

3.1 Sequential Implementation

The sequential implementation of the algorithm generates the subkeys from the initial key. Then, for each 16-byte state of the data to encrypt, it applies the rounds using the subkeys that were initially generated.

3.2 Parallel Implementations

Parallel implementations consider input data as consecutive, 16-byte blocks. Additionally, there are a certain number of processes or threads, and each of them will be responsible for encrypting a set of blocks. Block distribution is proportional to the number of processes or threads, that is, if input data size is N bytes, the number of blocks will be B = N/16. If there are P processes or threads, each will have to encrypt B/P blocks.

Subkey generation is done sequentially in all the implementations because it is a very simple process and its running time is negligible. Once the subkeys are generated, all processes or threads use them for the block encryption process.

3.2.1 Implementation using OpenMP

OpenMP is an API for the C, C++ and Fortran languages that allows writing and running parallel programs using shared memory and offers the possibility of creating a set of threads that work concurrently to exploit the advantages of multicore architectures.

The proposed implementation of AES with OpenMP sequentially generates the subkeys from the initial key. Then, a set of threads is created, as many as cores are provided by the architecture, and each of the threads takes a consecutive set of 16-byte blocks to encrypt.

3.2.2 Implementation using MPI

MPI (Message Passing Interface) is an API specification for programming with distributed memory, with implementations for C, C++ and Fortran languages. It can be used in multicore machines, cluster-type architectures with several machines connected through a network, or a combination of both to exploit all the cores provided by these architectures.

The proposed implementation of AES with MPI is based on having a certain number of processes, as many as processors are available. One of these processes sequentially generates the subkeys from the initial key and then communicates them. After this, it proportionally distributes the 16-byte blocks among the processes, including itself. Each process will encrypt the blocks that were assigned to it and then return the encrypted blocks.

3.2.3 Implementation using CUDA

In recent years, GPUs (Graphic Processing Unit) have been studied due to their high performance, and it was for this reason that they began to be used for general purpose (GPGPU - General Purpose GPUs). Based on this, one of the companies that manufactured graphic boards developed a compiler and a set of development tools called CUDA (Compute Unified Device Architecture) to allow programmers use a variation of the C language to program graphic boards and exploit the computation power they offer.

GPUs are formed by a set of Streaming Multiprocessors (SMs), each with a single core, called Streaming processors (SPs). Each SM can run simultaneously a large number of threads (the limit depends on the architecture), which allows SPs to always be performing useful work, even when parts of these threads are waiting for memory access.

The computation system for a CUDA programmer is formed by the CPU, also called *host*, and one or more GPUs, called *devices.* A CUDA program is divided in phases that are run on the host and phases that are run on the device. The code that is run on the device is called *kernel*, and when the host invokes the kernel, the threads are created. These threads are grouped in a *grid*, which is divided in thread *blocks*. The threads that form a block are assigned to an SM that will run them.

GPUs have different types of memory: the global memory, with read and write access for the host and all threads; the constant memory, with read and write access for the host and read only access for the threads; a shared memory located in each SM chip, whose access is faster than for the global memory and which allows the threads in the same block to cooperate; internal records in each SM; and other texture constant memories. [16].

3.2.3.1 AES Algorithm in GPU

The AES algorithm subkey calculation is done at the host, since execution time for this task is negligible, leaving only the encryption procedure to the device.

The host copies the subkeys and the byte substitution table to the constant memory of the device, since these will be read only by the threads. It then copies the data to be encrypted to the global memory of the device.

Next, it invokes the kernel specifying both the number of blocks and the number of threads per block.

The threads belonging to a same CUDA block will work on consecutive states; each will be responsible for encrypting one state (16-byte block). Since access to global memory is very expensive, before the state encryption stage, each thread cooperates with the other threads in its block to load the information that they have to encrypt to a *shared* memory. These accesses are done in a coalescent manner. Once the encryption stage is finished, the threads cooperate in a similar way to move the data from the shared memory to the global memory.

4. Results

The sequential algorithm was run on a machine with Intel Xeon E5405 architecture [17] with 2 GB of RAM memory. The shared memory algorithm that uses OpenMP was run on a machine with 2 Intel Xeon E5405 processors with 4 cores each and 2 GB of RAM memory; whereas the algorithm in MPI was run using a cluster of 4 machines with the previously described architecture connected to a 1 Gbit Ethernet using 32 cores.

The CUDA algorithm was run on a 1-GB-RAM Nvidia Geforce GTX 560TI [18] graphics card with 384 SPs, distributed in 8 SMs, each capable of running a maximum of 768 threads. Since the blocks used ran 256 threads, the maximum number of blocks that any of these SM could run was 3, and the number of blocks depends on the size of the data to encrypt. CUDA allows creating 65535 thread blocks for a one-dimensional grid; if the number of blocks is greater than the maximum that can be run by each SM, thread blocks are assigned to the SMs as they finish running the previous blocks.

The running times presented here correspond only to the encryption time, for various input data sizes. Decryption time was not considered for being similar.

Table 1. The following table shows the average running times, in seconds, for the different implementations and for the different input data sizes

	1KB	512KB	1MB	15MB	128MB	255MB
Sequential (Intel)	0.002221	1.133039	2.266163	33.99241	290.034037	577.988805
OMP (8 cores)	0.00054	0.155369	0.29914	4.358485	37.128006	73.94553
MPI (8 cores)	0.00028	0.142951	0.286033	4.29618	36.646296	72.971179
MPI (16 cores)	0.000141	0.072217	0.143635	2.146643	18.313296	36.528222
MPI (32 cores)	0.000071	0.035668	0.071336	1.073768	9.162032	18.248819
CUDA	0.000146	0.002551	0.005033	0.067806	0.572375	1.139361

Table 2. The following table shows the speed-up of the different implementations versus the sequential implementation run on the Intel architecture

	1KB	512KB	1MB	15MB	128MB	255MB
OMP (8 cores)	4.112962	7.292568	7.575593	7.799134	7.811732	7.816413
MPI (8 cores)	7.932142	7.926065	7.922732	7.912240	7.914416	7.920782
MPI (16 cores)	15.75177	15.68936	15.77723	15.83514	15.83734	15.82307
MPI (32 cores)	31.281690	31.766261	31.767452	31.657127	31.656082	31.672669
CUDA	15.2123288	444.154841	450.260878	501.318615	506.720309	507.292074

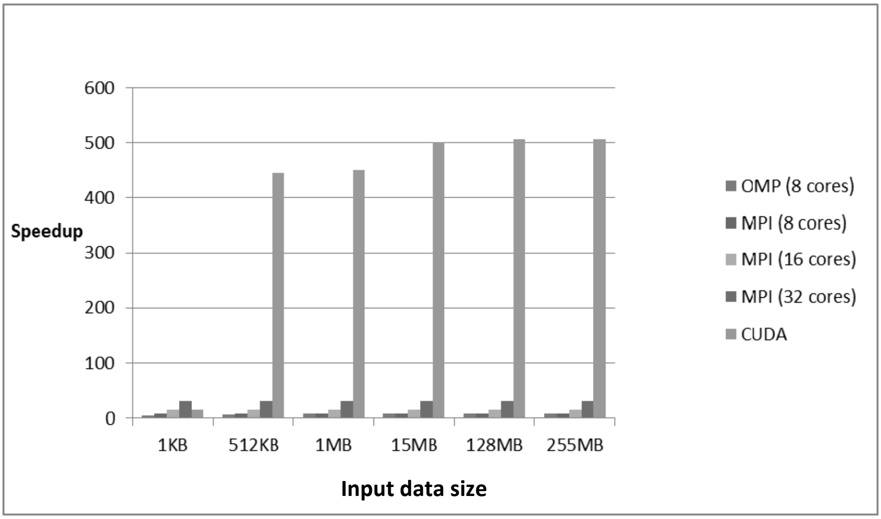


Fig. 3. Differences in speed-up of the different implementations versus the sequential implementation run on the Intel architecture

As it can be seen, running times for the algorithm that uses OpenMP and MPI (8 cores) are similar with the exception of the test with a data size of 1 KB, where MPI has a better performance. It can also be seen that running times increase linearly with the number of cores and the size of the input data for the MPI version of the algorithm. Despite this, it still does not improve the performance achieved by the implementation of the GPU that uses CUDA, which is considerably higher compared to the other implementations.

In order to run the implementations on the GPU, the data to be encrypted have to be copied to the device memory and, once the kernel execution is finished, the encrypted data must be recovered. These host-device and device-host memory copies usually add a certain overhead.

Table 3. The following table shows the average times for moving the data between host and device, encryption time, and the total resulting time

	1KB	512KB	1MB	15MB	128MB	255MB
Host-device copy	0.000007	0.000286	0.000996	0.022862	0.208334	0.404399
Device-host copy	0.000016	0.000965	0.001956	0.029013	0.247463	0.500004
Encryption time	0.000145	0.001819	0.003594	0.046689	0.392180	0.780433
Total time	0.000168	0.00307	0.006546	0.098564	0.847977	1.684836

The total running time, which includes encryption time and data transfer time, is still lower than the running time corresponding to the other algorithms, with the exception of 1KB on MPI with 16 and 32 cores, where the difference is minimal.

As already mentioned, the implementation with MPI scales in a linear manner, i.e., when the number of cores is doubled, running time is reduced by approximately half. It can be seen that the MPI implementation with an input data size of 255MB and 32 cores has a running time of the order of 18 seconds. Therefore, in order to achieve a time of 1.68 seconds with the MPI implementation, which is the time achieved by the CUDA implementation, the number of cores would have to be a bit more than 256.

5. Conclusions and Future Work

One sequential implementation and three parallel implementations of the block-based symmetric encryption algorithm AES were presented to exploit different multicore architectures.

The resulting running times of the different implementations were analyzed, the implementation efficiency of the CUDA algorithm being of note.

The encryption process of large volumes of data with the CUDA implementation of the AES algorithm proved to have very reduced times, which means that the encryption cost of data transfer can be reduced considerably.

There is currently a general-purpose cryptography library called OpenSSL [19] that implements different types of encryption algorithms, including AES, in a much more efficient manner. In the future, we intend to use this library to analyze the performance of the AES and other encryption algorithms, both symmetric and asymmetric.

Other future lines of work include systematically studying the performance of parallel algorithms (especially numeric ones [20] [21] [22]) executed on GPU-based architectures, versus multicore clusters. Also, the efficient use of energy [23] [24] [25] is of interest when scaling the problems and the number and complexity of the GPUs used.

References

1. Chapman B., The Multicore Programming Challenge, Advanced Parallel Processing Technologies; 7th International Symposium, (7th APPT'07), Lecture Notes in Computer Science (LNCS), Vol. 4847, p. 3, Springer-Verlag (New York), November 2007.
2. Suresh Siddha, Venkatesh Pallipadi, Asit Mallick. “Process Scheduling Challenges in the Era of Multicore Processors”Intel Technology Journal, Vol. 11, Issue 04, November 2007.
3. Grama A., Gupta A., Karypis G., Kumar V. “Introduction to Parallel Computing”. Second Edition. Addison Wesley, 2003.
4. Bischof C., Bucker M., Gibbon P., Joubert G., Lippert T., Mohr B., Peters F. (eds.), Parallel Computing: Architectures, Algorithms and Applications, Advances in Parallel Computing, Vol. 15, IOS Press, February 2008.
5. The OpenMP API specification for parallel programming. http://openmp.org/wp/.

6. MPI Specification http://www.mpi-forum.org/docs/mpi-2.2/mpi22-report.pdf.
7. General-Purpose Computation on Graphics Hardware http://gpgpu.org/.
8. FIPS PUB 197: the official AES Standard http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.
9. Lynn Hathaway (June 2003). "National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information"http://csrc.nist.gov/groups/ST/toolkit/documents/aes/CNSS15FS.pdf
10. D.J. Bernstein - Cache-timing attacks on AES (2005) http://cr.yp.to/antiforgery/cachetiming-20050414.pdf.
11. Dag Arne Osvik, Adi Shamir and Eran Tromer - Cache Attacks and Countermeasures: the Case of AES (2005) http://www.wisdom.weizmann.ac.il/~tromer/papers/cache.pdf.
12. A Diagonal Fault Attack on the Advanced Encryption Standard http://eprint.iacr.org/2009/581.pdf.
13. T. Rauber, G. Rünger. Parallel Programming: For Multicore and Cluster Systems. ISBN 364204817X, 9783642048173. Springer, 2010.
14. Buck I. "Gpu computing with nvidia cuda". ACM SIGGRAPH 2007 courses ACM, 2007. New York, NY, USA.
15. Cuda Home Page http://www.nvidia.com/object/cuda_home_new.html.
16. Cuda best practices guide http://developer.download.nvidia.com/compute/DevZone/docs/html/C/doc/CUDA_C_Best_Practices_Guide.pdf.
17. Intel Product Specifications http://ark.intel.com/products/33079/Intel-Xeon-Processor-E5405-(12M-Cache-2_00-GHz-1333-MHz-FSB).
18. Nvidia Geforce GTX 560TI Specifications http://www.nvidia.com/object/product-geforce-gtx-560ti-us.html.
19. The official OpenSSL www.openssl.org.
20. Basic Linear Algebra Subprograms BLAS http://www.netlib.org/blas/.
21. Linear Algebra Package LAPACK http://www.netlib.org/lapack/.
22. Automatically Tuned Linear Algebra Software ATLAS http://www.netlib.org/atlas/.
23. Computer Architecture: Challenges and Opportunities For The Next Decade - Tilak Agerwala Siddhartha Chatterjee IBM Research 2004. Published by the IEEE Computer Society
24. Green Supercomputing Comes of Age - Wu-chun Feng, Xizhou Feng & Rong Ge http://portal.acm.org/citation.cfm?id=1344283.
25. Maximizing Power Efficiency with Asymmetric Multicore Systems - Alexandra Fedorova, Juan Carlos Saez, Daniel Shelepov, and Manuel Prieto http://portal.acm.org/citation.cfm?id=1610270.

Efficiency Evaluation of the Input/Output System on Computer Clusters[1]

SANDRA MÉNDEZ, DOLORES REXACHS DEL ROSARIO,
EMILIO LUQUE FADÓN

Computer Architecture and Operating Systems Department (CAOS)
Universidad Universitat Autònoma de Barcelona, Barcelona, Spain
{sandra.mendez, dolores.rexachs, emilio.luque}@uab.es

Abstract. *The increasing in the complexity of scientific applications that use high performance computing requires more efficient Input/Output (I/O) systems. In order to efficiently use the I/O it is necessary to know its performance capacity in order to determine whether it fulfills applications I/O requirements. This paper proposes the efficiency evaluation of the I/O systems on computer clusters. This evaluation is useful to study how different I/O system will affect the application performance. This approach encompasses the characterization of the computer cluster at three different levels: devices, I/O system and application. We select different systems and we evaluate the impact on performance by considering both the application and the I/O architecture. During I/O configuration analysis we identify configurable factors that have an impact on the performance of I/O system. Furthermore, we extract information in order to determine the used percentage of I/O system by an application on a given computer cluster.*

Keywords: *Parallel I/O System, I/O Architecture, I/O Configuration, I/O Path Level, I/O inefficiency.*

1. Introduction

The increase in processing units, the advance in speed and compute power, and the increasing complexity of scientific applications that use high performance computing require more efficient I/O systems. Due to the historical "gap" between the computing performance and I/O performance, in many cases, the I/O system becomes the bottleneck of the parallel systems. The efficient use of the I/O system and the identification of I/O factors that influence the performance can help to hide this "gap". In order to efficiently use the I/O system, it is first necessary to know its performance capacity to determine if it fulfills the application's I/O requirements.

[1] Please note that the LNCS Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

There are several papers on performance evaluation of I/O system. Roth[1] presented event tracing for characterizing the I/O demands of applications on the Jaguar Cray XT of supercomputer. Fahey [2] experimented in the I/O system of the Cray XT, and the analysis was focused in the LUSTRE filesystem.

Laros [3] carried out a performance evaluation of I/O configuration. Previous papers do not directly consider the I/O characteristics of applications.

We propose the efficiency evaluation of the I/O system by analyzing each level on the I/O path. Furthermore, we take into account the application I/O requirements and the I/O architecture configuration. The proposed methodology has three phases: characterization, the analysis of I/O system, and the efficiency evaluation. In the application's characterization phase, we extract the I/O requirements of the application. In the I/O system characterization we obtained the bandwidth and IOPs (I/O operations per second) at filesystem level, interconnection network, I/O library and I/O devices. Furthermore, we identify configurable or selectable factors that have an impact on the I/O system performance.

We search these factors in the filesystem level, I/O node connection, placement and state of buffer/cache, data redundancy and service redundancy. We collect metrics of the application execution on I/O configurations In the evaluation phase, the efficiency is determined by analyzing the difference between measured values and characterized values.

The rest of this article is organized as follows: Section 2 introduces our proposed methodology. In Section 3 we review the experimental validation of this proposal. Finally, in the Section 4, we present conclusions and future work.

2. Proposed Methodology

The I/O in the computer cluster occurs on a hierarchal I/O path. We see I/O system as shown in Fig. 2(a). The application carries out the I/O operations in this hierarchical I/O path. The I/O path levels are: I/O library (high and low level), filesystem (local and global), network (I/O o shared with computing), and I/O devices. However, the placement of the filesystem and interconnection network can vary depending on the I/O system configuration. The application also can use I/O libraries of high (NetCDF, HDF5) or low level (MPI-IO). In order to evaluate the I/O system performance it is necessary to know its capacity of storage and throughput. The storage depends on amount, type and capacity of devices.

The throughput depends on IOPs (Input/Output operations per second) and the latency. Moreover, this capacity is different in each I/O system level. Furthermore, the performance depends on the connection of the I/O node, the management of I/O devices, placement of I/O node into network topology, buffer/cache state and placement, and availability of data and service. In order to determine whether an application uses I/O system capacity it is necessary to know its I/O behavior and requirements. The methodology is shown in Fig. 1. This is used to evaluate efficiency of the I/O system and identify the possible points of inefficiency. The efficiency is based on the used performance percentage by the application on each I/O path level. Also, when the cluster has different selectable or configurable parameters, the methodology is used to analyze which I/O configuration is the most appropriate for an application.

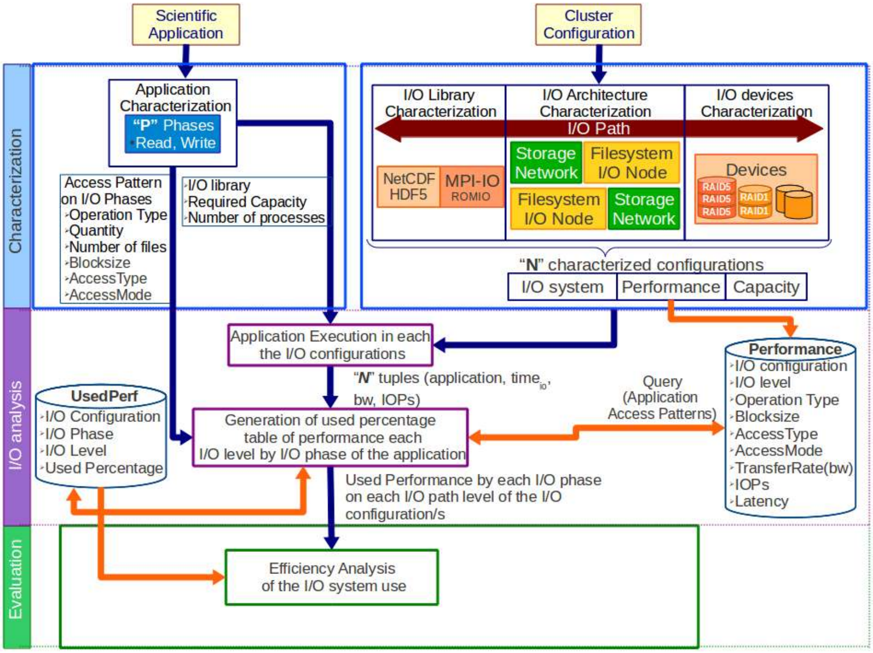


Fig. 1. Methodology for Efficiency Evaluation on I/O System

2.1 Characterization

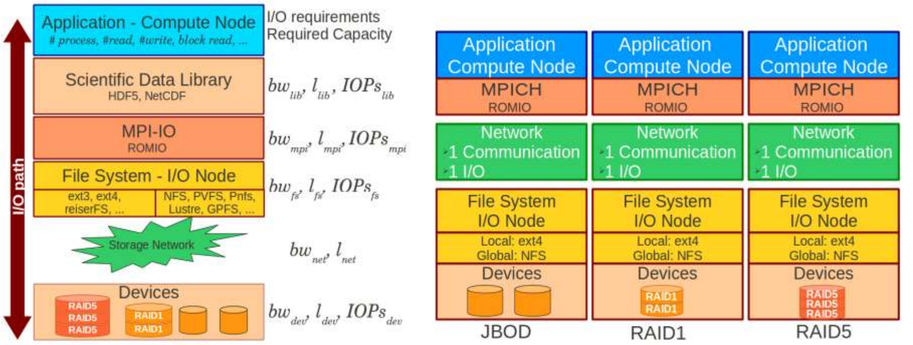
This is applied to obtain the capacity and performance of I/O system. We also obtain I/O requirements and behavior of the application. We explain the system characterization and the scientific application characterization.

A. I/O System and Devices

Parallel system is characterized at I/O library level, I/O Node (global filesystem and interconnection system) and devices (local filesystem). We characterize the bandwidth (bw), latency (l) and (IOP s) for each level, as shown in Fig. 2(a). Fig. 3(a) shows ”what” and ”how” we obtain this information for the I/O system and Devices. Furthermore, we obtain characterized configurations in each I/O path level. The data structure of I/O system performance for local and global filesystem, and I/O library following is shown:

−Operation Type (enumerate {0 (read), 1 (write)})
−Block size (double (MBytes))
−Access Type (enumerate {0 (Local), 1 (Global)})

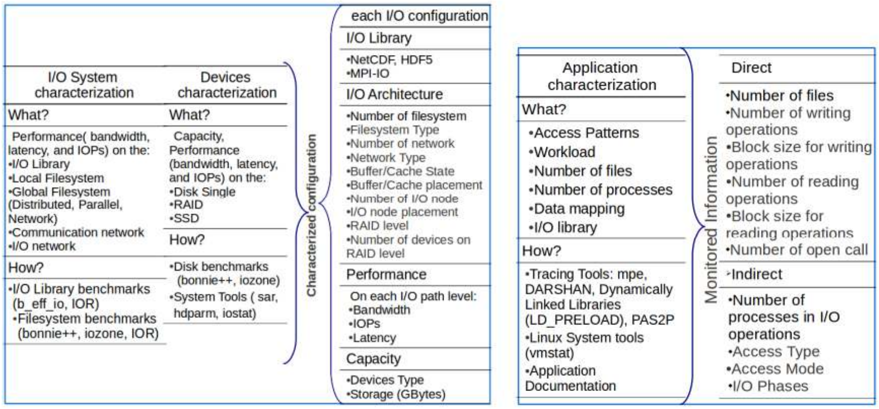
–Accesses Mode (enumerate {0 (Sequential), 1 (Strided), 2 (Random)})
–transfer Rate (double (MBytes/second))
–Latency (double (microsecond))
–IOPs (integer)



(a) I/O system Characterization (b) I/O configurations of the Aohyper cluster

Fig. 2. Characterization of I/O System

To evaluate global filesystem and local filesystem, IOzone [4] and/or bonnie++ [5] benchmarks can be used. Parallel filesystem can be evaluated with the IOR benchmark [6]. The b eff io [7] or IOR benchmarks can be used to evaluate the I/O library. To explain this phase we present the characterization for the I/O system of cluster Aohyper.

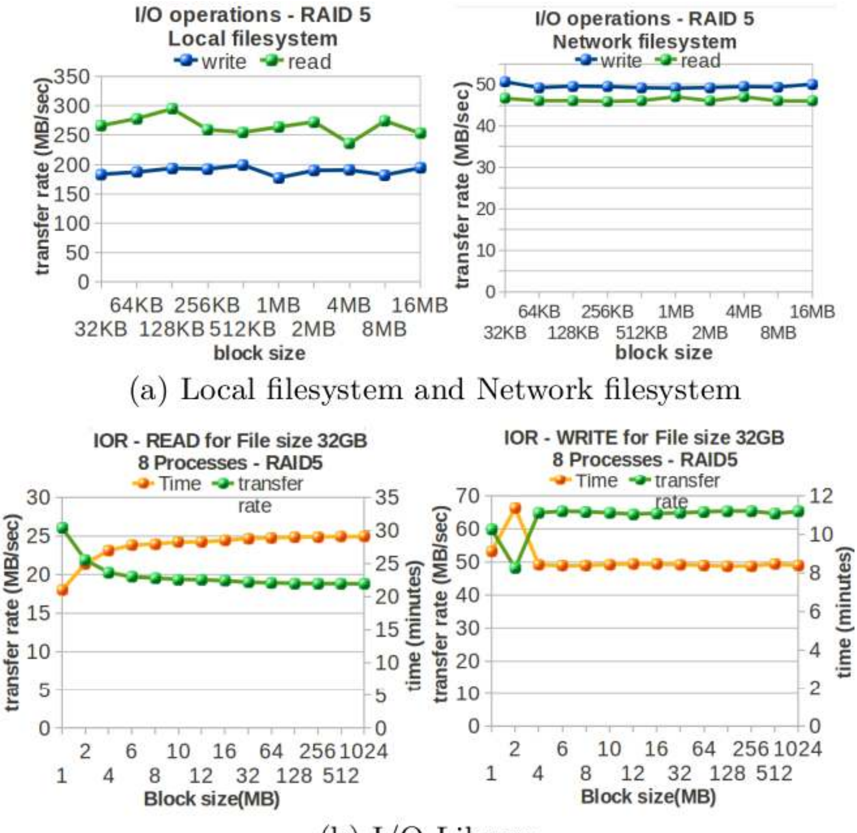

I/O System characterization
What?
Performance(bandwidth, latency, and IOPs) on the:
•I/O Library
•Local Filesystem
•Global Filesystem (Distributed, Parallel, Network)
•Communication network
•I/O network
How?
•I/O Library benchmarks (b_eff_io, IOR)
•Filesystem benchmarks (bonnie++, iozone, IOR)
Devices characterization
What?
Capacity, Performance (bandwidth, latency, and IOPs) on the:
•Disk Single
•RAID
•SSD
How?
•Disk benchmarks (bonnie++, iozone)
•System Tools (sar, hdparm, iostat)
Characterized configuration
each I/O configuration
I/O Library
•NetCDF, HDF5
•MPI-IO
I/O Architecture
•Number of filesystem
•Filesystem Type
•Number of network
•Network Type
•Buffer/Cache State
•Buffer/Cache placement
•Number of I/O node
•I/O node placement
•RAID level
•Number of devices on RAID level
Performance
On each I/O path level:
•Bandwidth
•IOPs
•Latency
Capacity
•Devices Type
•Storage (GBytes)
Application characterization
What?
•Access Patterns
•Workload
•Number of files
•Number of processes
•Data mapping
•I/O library
How?
•Tracing Tools: mpe, DARSHAN, Dynamically Linked Libraries (LD_PRELOAD), PAS2P
•Linux System tools (vmstat)
•Application Documentation
Monitored Information
Direct
•Number of files
•Number of writing operations
•Block size for writing operations
•Number of reading operations
•Block size for reading operations
•Number of open call
•Indirect
•Number of processes in I/O operations
•Access Type
•Access Mode
•I/O Phases

(a) I/O system and Devices (b) Application

Fig. 3. Characterization Phase

Cluster Aohyper has the following characteristics: 8 nodes AMD Athlon(tm) 64 X2 Dual Core Processor 3800+, 2GB RAM memory, 150GB local disk. Local filesystem is Linux ext4 and global filesystem is NFS. The NFS server has a RAID 1 (2 disks) with 230GB capacity and RAID 5 (5 disks) with stripe=256KB and 917GB capacity, both with write-cache enabled (write back); two Gigabit Ethernet networks, one for communication and the other for data. NFS server is an I/O node for shared accesses.

Also, there are eight I/O-compute nodes for local accesses and the data sharing must be done by the user. Cluster Aohyper, at device level, has three I/O configurations (Fig. 2(b)). JBOD configuration is single disks without redundancy. RAID 1 configuration has a disk with its mirror disk and RAID 5 has five active disks. The parallel system and storage devices characterization were done with IOzone. Due to space, we only show the characterization of the RAID 5 configuration. Fig. 4 shows results for network filesystem, local filesystem, and I/O library for RAID 5.



(a) Local filesystem and Network filesystem

(b) I/O Library

Fig. 4. Characterization of Configuration RAID5

The experiments were performed at block level with a file size which doubles the main memory size and block size was changed from 32KB to 16MB. The IOR benchmark was used to analyze the I/O library. It was configured for 32GB file size on RAID configurations and 12 GB on JBOD, from 1MB to 1024MB block size and transfer block size of 256KB. It was launched with 8 processes.

B. Scientific Application

We extract the type, quantity and operations size of I/O at library level. Fig. 3(b) shows ”what”, ”how”, and the monitored information of the application. This information is used in the evaluation phase to determine whether application performance is limited by the application characteristics or by the I/O system. To evaluate the application characterization at process level, an extension of PAS2P [8] tracing tool was developed. We incorporate the I/O primitives of MPI-2 standard to PAS2P. These are detected when the application is executed. To do this we used dynamic link with LD PRELOAD.

With the characterization, we propose to identify the significant phases with an access pattern and their weights. Due to the fact that scientific applications show a repetitive behavior, P phases will exist in the application.

To explain the methodology, the characterization is applied to Block Tridiagonal (BT) application of NAS Parallel Benchmark suite (NPB)[9]. The BTIO benchmark performs large collective MPI-IO writes, and reads of a nested strided datatype, and it is an important test of the performance a system which can provide for noncontiguous workloads. After every five time steps the entire solution field, consisting of five double-precision words per mesh point, must be written to one or more files. After all time steps are finished, all data belonging to a single time step must be stored in the same file, and must be sorted by vector component, x-coordinate, y-coordinate, and z-coordinate, respectively.

NAS BT-IO full subtype has 40 phases to write and 1 phase to read. Writing operation is done each 120 message sent with their respective Wait and Wait All. The reading phase consists of 40 reading operations done after all writing procedures are finished. This is done for each MPI process. Simple subtype has the same phases but each writing phase does 6,561 writing operations. The reading phase consists of 262,440 reading operations. The characterization done for the class C of NAS BT-IO in full and simple subtypes is shown in Table 1.

Table 1. NAS BT-IO Characterization - Class C - 16 and 64 processes

Parameters	Full 16p	Simple 16p	Full 64p	Simple 64p
numFiles	1	1	1	1
$numIO_{read}$	640	2,073,600 and 2,125,440	2,560	8,398,080
$numIO_{write}$	640	2,073,600 and 2,125,440	2,560	8,398,080
bk_{read}	10MB	1.56KB and 1.6KB	2.54MB	800B and 840B
bk_{write}	10MB	1.56KB and 1.6KB	2.54MB	800B and 840B
$numIO_{open}$	*32*	32	128	128
accessType	*Global*	Global	Global	Global
accessMode	*Strided*	Strided	Strided	Strided
numProcesos	*16*	16	64	64

2.2 Input/Output Analysis

The I/O configurations of cluster computer are composed of I/O library, I/O architecture and I/O devices. The configurable or selectable I/O parameters are shown in Fig. 3(a), they are labeled as “each I/O configurations”. The selection of I/O configuration depends on I/O requirements of the application and the user requirements. In order to select the configurations, we considered the I/O library, number of processes and the capacity required by the application. The RAID level will depend on what the user is willing to pay. For this article we have selected three configurations: JBOD, RAID 1 and RAID 5.

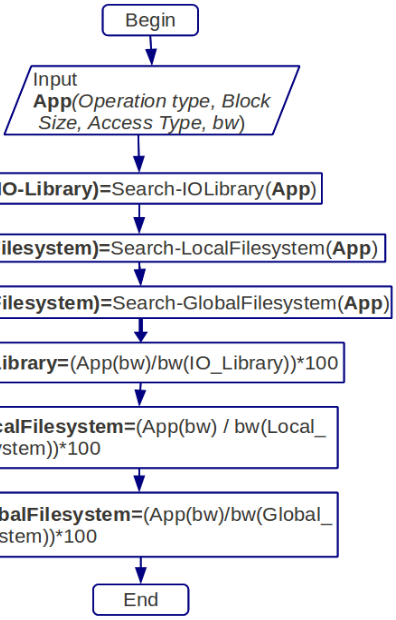

Begin
Input
App(Operation type, Block Size, Access Type, bw)
bw(IO-Library)=Search-IOLibrary(App)
bw(LocalFilesystem)=Search-LocalFilesystem(App)
bw(GlobalFilesystem)=Search-GlobalFilesystem(App)
%IO-Library=(App(bw)/bw(IO_Library))*100
%LocalFilesystem=(App(bw) / bw(Local_ Filesystem))*100
%GlobalFilesystem=(App(bw)/bw(Global_ Filesystem))*100
End

Fig. 5. Generation used percentage

We extracted I/O behavior of application in the 1st phase, now we evaluate the application in selected configurations to view its behavior. The metrics for the application are: execution time, I/O time (time to do reading and writing operations), I/O operations per second (IOPs), latency of I/O operations and throughput (number of megabytes transferred per second). A file is generated with the used percentage by the application on each of the I/O configurations, ”P” I/O phases and I/O path levels (denoted by UsedPerf in Fig. 1 on the I/O analysis phase).

The processes of generation of used percentage are presented in Fig. 5. The algorithm to search for the transfer rate on each I/O level is shown in Fig. 6; and it is applied in each searching stage of Fig. 5.

2.3 Evaluation

We evaluate the use efficiency of I/O system based in the characterized values and measured values. The efficiency evaluation of the I/O system uses a file with the characterized values for the each I/O configurations of the computer cluster (denoted by Performance in Fig. 1).
Following our example, we analyze NAS BT-IO on the Aohyper cluster. Fig. 7 shows the execution time, the I/O time and throughput for NAS BT-IO class C using 16 processes executed on the three configurations. The evaluation is for full (with collectives I/O) and simple (without collectives) subtypes. The used percentage of I/O system is shown in Table 2.

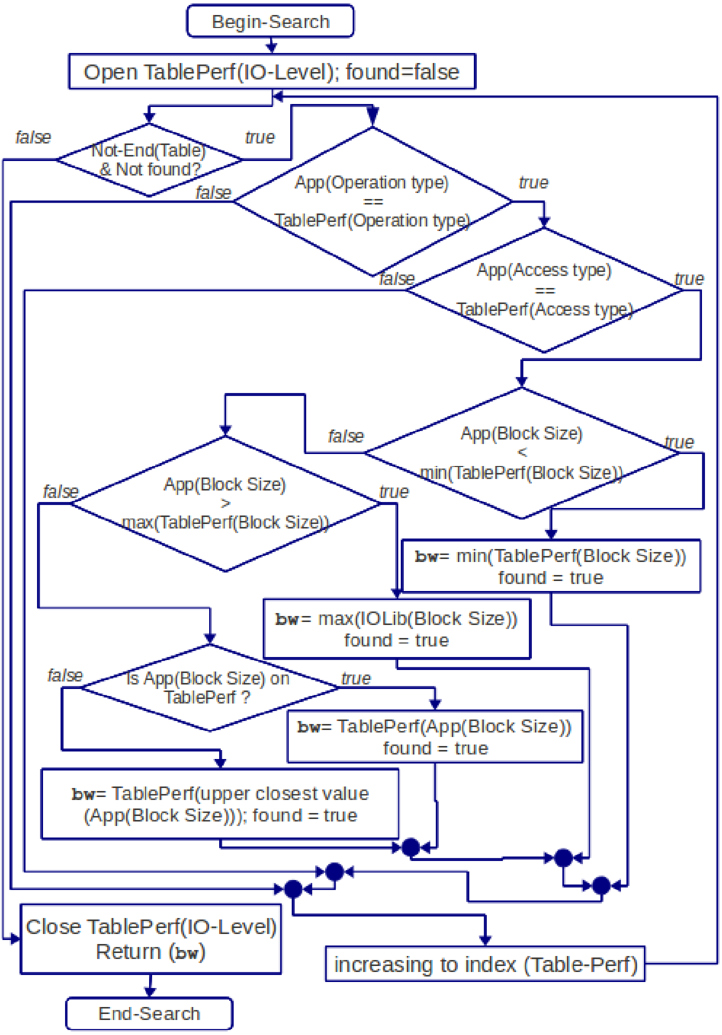

Begin-Search
Open TablePerf(IO-Level); found=false
Not-End(Table) & Not found?
false
true
false
App(Operation type) == TablePerf(Operation type)
true
false
App(Access type) == TablePerf(Access type)
true
false
App(Block Size) < min(TablePerf(Block Size))
true
false
App(Block Size) > max(TablePerf(Block Size))
true
bw= min(TablePerf(Block Size)) found = true
bw= max(IOLib(Block Size)) found = true
false
Is App(Block Size) on TablePerf ?
true
bw= TablePerf(App(Block Size)) found = true
bw= TablePerf(upper closest value (App(Block Size))); found = true
Close TablePerf(IO-Level) Return (bw)
increasing to index (Table-Perf)
End-Search

Fig. 6. Generation used percentage

Table 2. Percentage (%) of I/O system use for NAS BT-IO on I/O phases

I/O Configuration	I/O Lib. Write	NFS Write	Local FS Write	I/O Lib. Read	NFS Read	Local FS Read	Subtype
JBOD	101.47	117.70	78.00	309.74	127.93	60.00	FULL
RAID1	140.24	120.20	54.04	310.00	128.04	43.63	FULL
RAID5	88.60	115.18	29.69	303.11	125.20	22.76	FULL
JBOD	25.06	26.06	15.33	54.29	28.96	18.61	SIMPLE
RAID1	27.75	30.65	13.37	54.48	31.98	12.68	SIMPLE
RAID5	24.60	29.52	8.07	56.77	31.40	5.55	SIMPLE

The full subtype is an efficient implementation for NAS BT-IO and we observe for the class C that the capacity of I/O system is exploited. But for the simple subtype, this I/O system is used only at about 30% of performance on reading operations and less than 15% on writing operations. NAS BT-IO simple subtype carries out 4,199,040 writes and 4,199,040 reads with block sizes of 1,600 and 1,640 bytes (TABLE 1). This has a high penalization in the I/O time impacting on the execution time (Fig. 7). For this application in the full subtype the I/O is not factor bounding because the capacity of I/O system is sufficient for I/O requirements. The simple subtype does not achieve exploitation of the I/O system capacity due to its access pattern.

3. Experimentation

In order to test the methodology, an evaluation of NAS BT-IO for 16 and 64 processes in a different cluster was carried out, this cluster is called cluster A. Cluster A is composed of 32 compute nodes: 2 x Dual-Core Intel (R) Xeon (R) 3.00GHz, 12 GB of RAM, and 160 GB SATA disk Dual Gigabit Ethernet. A front-end node as NFS server: Dual-Core Intel (R) Xeon (R) 2.66GHz, 8 GB of RAM, 5 of 1.8 TB RAID and Dual Gigabit Ethernet. Cluster A has an I/O node that provides service to shared files by NFS and storage with RAID 5 level. Furthermore, there are thirty-two I/O nodes for local and independent accesses.

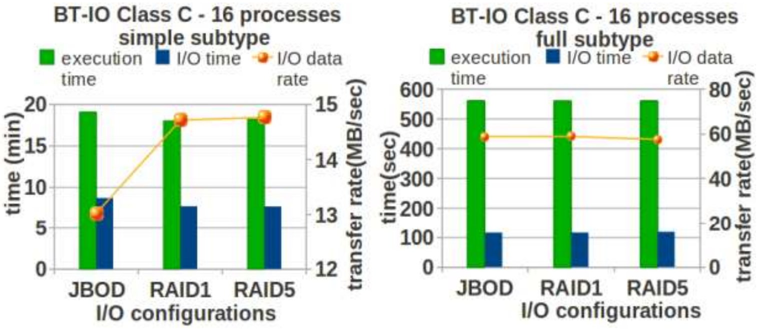


Fig. 7. Running time of the NAS BT-IO Class C 16 Processes

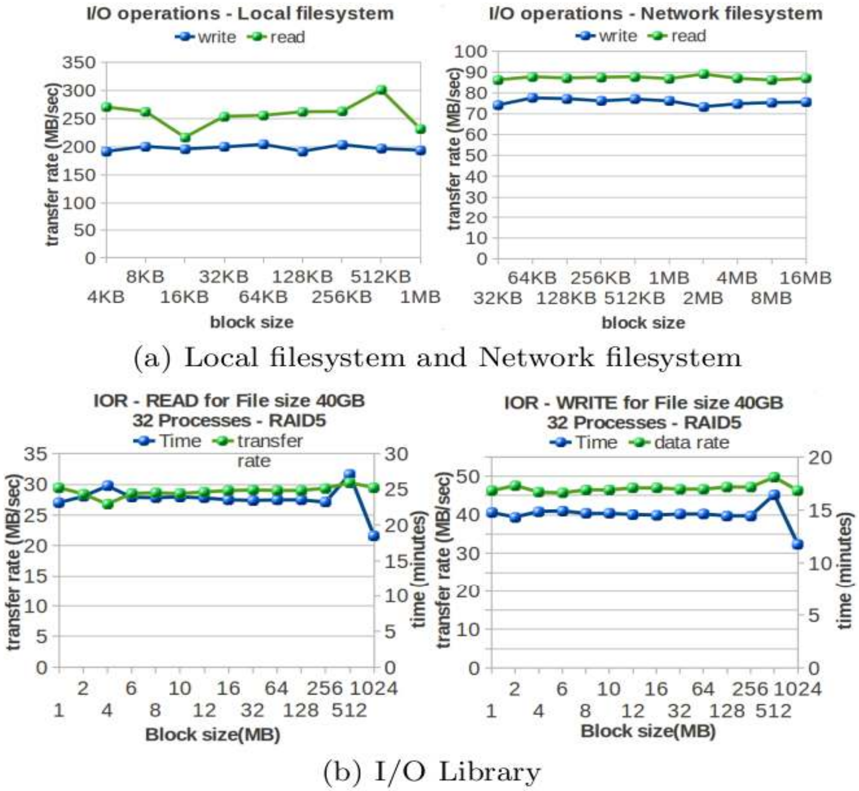
Due to the I/O characteristics of the cluster A, where there are no different I/O configurations, we used the methodology to efficiency evaluate the I/O system for NAS BT-IO. Characterization of I/O system on cluster A is presented in Fig. 8. We evaluate the local and network filesystem with IOzone. Due to this cluster is being restricted; the characterization in local file system was done by system administrators. IOR benchmark to evaluate the I/O library was done with 40 GB filesize, block size from 1 MB to 1024 MB, and 256 KB transfer block. The characterization for 16 and 64 processes is shown in Table 1.

NAS BT-IO was executed for 16 and 64 to evaluate the use of the I/O system on cluster A. Table 3 shows the used percentage on I/O library, NFS and Local filesystem. Fig. 9 shows the execution time, the I/O time and throughput for NAS BT-IO full and simple subtypes.

Table 3. Percentage (%) of I/O system use for NAS BT-IO in I/O phases

I/O Configuration	I/O Lib. Write	NFS Write	Local FS Write	I/O Lib. Read	NFS Read	Local FS Read	Subtype
16	70.74	43.39	16.27	112.21	36.16	13.56	FULL
64	80.26	49.76	18.66	128.69	41.47	15.55	FULL
16	2.45	1.58	0.57	3.86	1.28	0.45	SIMPLE
64	0.67	0.43	0.16	1.05	0.35	0.12	SIMPLE

The full subtype is an efficient implementation that achieves more than 100% of the characterized performance on the I/O library for 16 and 64 processes. However, with a greater number of processes, the I/O system influences the run time of the application. NAS BT-IO full subtype is limited in the cluster A due to the computing and/or communication. NAS BT-IO full subtype does not achieve 50% of NFS characterized values and the I/O time is increased with larger number of processes, which is due to communication among processes and the I/O operations. NAS BT-IO simple subtype is limited by I/O for this A cluster I/O configuration. The I/O time is superior to 90% of run time. For this system the I/O network and communication are bounding the application performance.



(a) Local filesystem and Network filesystem

(b) I/O Library

Fig. 8. I/O system Characterization of Cluster "A"

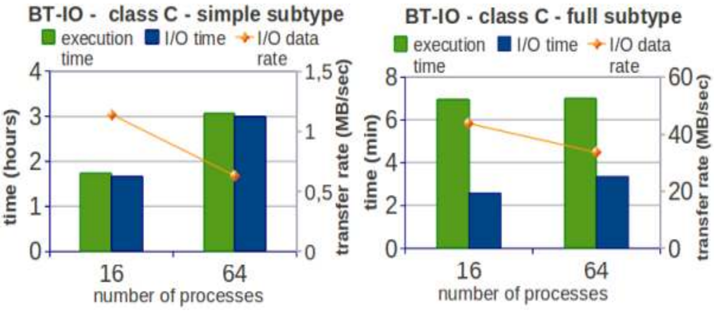


Fig. 9. Running time of the NAS BT-IO Clase C - 16 and 64 processes in Cluster "A"

4. Conclusion

A methodology for efficiency evaluation of I/O system on computer clusters was shown. Such methodology encompasses the characterization of the I/O system at three different levels: devices, I/O system and application. We analyzed and evaluated different systems and we calculated the use by the application (% of use) of the I/O system on different I/O path levels. The methodology was applied in two different clusters for NAS BT-IO benchmark. The performance of both I/O systems was evaluated using benchmarks, and we characterized the application. Also, we show the use of the I/O systems done by NAS BT-IO, which has been evaluated on each I/O path level of the I/O configurations.

As future work, we are defining an I/O model of the application to support the evaluation, design and selection of configurations. This model is based on the characteristics of the application and I/O system, and it is being developed to determine which configuration of I/O meets the performance requirements of the user, taking into account the application I/O behavior in a given system.

We will extract the functional behavior of the application, and we will define the I/O performance for the application given the functionality of application at I/O level. In order to test other configurations, we are analyzing the simulation framework SIMCAN [10] and planning to use such a tool to model I/O architectures.

References

1. P. C. Roth, “Characterizing the i/o behavior of scientific applications on the cray xt,” in PDSW ’07: Procs of the 2nd int. workshop on Petascale data storage. USA: ACM, 2007, pp. 50-55.
2. M. Fahey, J. Larkin, and J. Adams, “I/o performance on a massively parallel cray xt3/xt4,” in Parallel and Distributed Procs, 2008. IPDPS 2008. IEEE Int. Symp. on, 14-18 2008, pp. 1-12.
3. J. H. Laros et al., “Red storm io performance analysis,” in CLUSTER ’07: Procs of the 2007 IEEE Int. Conf. on Cluster Computing. USA: IEEE Computer Society, 2007, pp. 50-57.
4. W. D. Norcott, “Iozone filesystem benchmark,” Tech. Rep., 2006. [Online]. Available: http://www.iozone.org/
5. R. Coker, “Bonnie++ filesystem benchmark,” Tech. Rep., 2001. [Online]. Available: http://www.coker.com.au/bonnie++/
6. S. J. Shan, Hongzhang, “Using ior to analyze the i/o performance for hpc platforms,” LBNL Paper LBNL-62647, Tech. Rep., 2007. [Online]. Available: www.osti.gov/bridge/servlets/purl/923356-15FxGK/
7. R. Rabenseifner and A. E. Koniges, “Effective file-i/o bandwidth benchmark,” in Euro-Par ’00: Procs from the 6th Int. Euro-Par Conference on Parallel Procs. London, UK: Springer-Verlag, 2000, pp. 1273-1283.
8. A. Wong, D. Rexachs, and E. Luque, “Extraction of parallel application signatures for performance prediction,” in HPCC, 2010 12th IEEE Int. Conf. on, sept. 2010, pp. 223-230.

9. P. Wong and R. F. V. D. Wijngaart, "Nas parallel benchmarks i/o version 2.4," Computer Sciences Corporation, NASA Advanced Supercomputing (NAS) Division, Tech. Rep., 2003.
10. A. Núñez, et al., "Simcan: a simulator framework for computer architectures and storage networks," in Simutools '08: Procs of the 1st Int. Conf. on Simulation tools and techniques for communications, networks and systems & workshops. Belgium: ICST, 2008, pp. 1-8.

Implementing Sub Steps in a Parallel Cellular Automata Model

PABLO CRISTIAN TISSERA[1], A. MARCELA PRINTISTA[1],
EMILIO LUQUE FADÓN[2]

[1] Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis - Argentina
[2] Departamento de Arquitectura y Sistemas Operativos
Universidad Autónoma de Barcelona - España
{ptissera, mprinti}@unsl.edu.ar

Abstract. *Computer simulations using Cellular Automata (CA) have been applied with considerable success in different scientific areas. In this work we use CA in order to specify and implement a simulation model that allows investigating behavioral dynamics for pedestrians in an emergency evacuation. A CA model is discrete and handled by rules. However several aspects of the crown behavior should appear as a continuous phenomenon. In this paper we implement the sub steps technique in order to solve an unexpected phenomenon: the formation of holes (empty cells) both around the exit and mixed in the crowd evacuation. The holes occur when individuals of consecutive cells want move towards an exit. Additionally, the methodology allowed us to include, as a new parameter of the model, speeds associated to the pedestrians. Due to the incorporation of new features, the model complexity increases. We apply a parallel technique in order to accelerate the simulation and take advantage of modern computer architectures. We test our approaches to several environment configurations achieving important reduction of simulation time and the total evacuation time.*

Keywords: *Evacuation Simulation, Parallel Cellular Automata, Crown Dynamics.*

1. Introduction

Shopping centers, schools and dance halls are some examples of buildings commonly used in different daily activities that involve the meeting of a large number of people within a closed area. The designers of these types of building usually attempt to maximize the productivity of the available space, but also it is necessary to consider a suitable planning for assuring people safety when an unusual behavior of the crowd occurs. One of the most frequent causes of this kind of behavior is the emergency evacuation due to the threat of fire. In such situation, a closed area, with a relatively small number of fixed exits, must be evacuated for a large number of people.

In the last years, the interest in models of emergency evacuation processes and pedestrian dynamics has increased. In this context, models can be characterized according to resolution, fidelity and scale [1]. The resolution is the level of detail regarding the representation of space. Models used for evaluating the evacuation processes can broadly be categorized in microscopic (high resolution) and macroscopic approaches (low resolution). The macroscopic approaches are based on differential equations that take into account the similarities with systems previously studied like dynamics of fluids. On the other hand, the microscopic approaches allow considering how the system state evolves during the model running.

This paper considers a microscopic model where the space is represented in a discrete fashion. The special discrete model that we used is called Cellular Automata (CA) model [2] [3] [4]. For microscopic models, the connection between the space (its geometry), population and the propagation of fire and smoke is made via the rules or equations.

A high fidelity model is one with many parameters that directly takes into account all the different influences (e.g., parameters like age, training, weight, etc. in the case of pedestrians). CA allow us to generate "local" and "uniform" behaviors that resemble the dynamics observed in real processes of fire and smoke propagation. However, these local features were not suitable for representing certain aspects of people behavior that require a more specific and differentiated perspective. We developed a hybrid model where the dynamics of fire and smoke propagation are modeled by means of CA and for simulating people behavior we are using goal oriented intelligent agent.

The scale is the size of the problem with respect to time, space, etc. The scale of a pedestrian simulation depends on the application of course. In our case, a model that scales linearly (with respect to computation time and memory requirement) with the number of persons or the size of the environment is desirable. While agent-based models afford higher complexity and finer granularity they also require more computational resource. This means for large populations, modeling individuals with complex rules becomes infeasible typically individuals will instead be modeled as simple particles. We used partitioning technique as a solution to this problem, allowing large scale simulations.

The simulation allows to specify different scenarios with a large number of people and environmental features, making easier the study of the complex behaviors that arise when the people interact. In section 2 we present the Agent-based CA model for the pedestrian motion. In section 3 we developed new features that are needed to adequately represent the dynamics of a crowd. In section 4, the multicolumn partitioning technique is explained. In section 5 we describe our work with different instances of the problem at hand and report the performance analysis of each case. This section also includes a few commentaries on the operation of the EVAC and EVAC*simulation systems that support the proposed models. Finally, the section 6 presents the conclusions.

2. Agent based CA Model for Evacuation Simulation

The cellular automata (CA) are discrete dynamic systems that offer an appealing alternative to deal with this kind of situations due to their capacity to develop complex behaviors from a simple set of rules. Basically, these rules allow specifying the new state of a cell based on the state of the neighboring cells. In this way, it is possible to model complex dynamic systems from the specification of the local dynamics of its components. A further advantage of these systems is the support usually provided for displaying the results in a graphical way, allowing an easier comprehension of the dynamics of the system under study. We use the CA as basis of our simulation model for to investigate the pedestrian dynamics in emergency situations. Below, we describe the main features of the proposed model:

Cellular Space*:* it is a finite bi-dimensional array (grid) with closed boundaries.

Each cell of the cellular space represents 40 × 40 cm^2. This is the space usually occupied by a person in a crowd with maximal density [5, 6, 7]. The dimensions of the cellular space are specified in meters. So, one grid of 10 × 10 m^2 will contain 25 cells by side.

States: a cell can be in one of the states of the set Q = {W, E, P,O, S, SF, PS}, where:

W: External wall cell	*E:* Empty cell	*S:* Cell with smoke
PS: Cell with a person	*P:* Cell with a person. and smoke	
O: Internal obstacle cell	*SF:* Cell with smoke and fire	

Neighborhood*:* the neighborhood considered in the model is Moore's neighborhood that includes the eight cells surrounding the central cell. With this choice we aim to provide to each individual in the system with all possible movement directions.

Initial Configuration*:* before the simulation starts diverse information related to the outer walls, inner obstacles, individuals, combustible locations, cell with fire and arrangement of the exits must be specified.

Virtual clock*:* taking into account recent studies [8,5], an updating time of 0.3 seconds by time-step was specified for our model. This value is the estimated time required by a pedestrian for walking 0.4 m (size of a cell side).

Model Evolution Rules [9]:

1. Rules about the *building*: a cell in state W or O (outer wall or obstacle) will not change its state throughout the simulation.
2. Rules about *smoke propagation*: a cell with smoke (*S, SF or PS*) at time *t*, also will have smoke in time *t+1*. If at time *t* the central cell does not have smoke, but some of its adjacent cells have smoke, the central cell also will have smoke at time *t+1* with a probability proportional to the number of adjacent cells with smoke. For example, if one cell has four adjacent cells

3. Rules about *fire propagation*: these rules are analogous to the rules for smoke propagation.
4. Rules about the *people motion*: a simple reflex intelligent agent represents the pedestrian capability to decide about its desire to move to the cell depending on its environment. To solve the collisions that arising when two or more people simultaneously attempt to occupy the same physical location belonging to an exit way, we changed the approach commonly used in other works. Instead of being the pedestrian who decides which cell to move, the current cell is in charge of choosing between the people in the adjacent cells, which pedestrian will move to the cell. The policy distributes the decision between neighboring agents. For this, each agent builds a beliefs matrix containing the distances from its neighbor to the exit. This information is relevant to the decision that it must take:
   - If the distance from the requirer cell to the exit is greater than the current distance, then the agent does not move.
   - If the distance from the requirer cell to the exit is less than the current distance, then the agent replies its desire of moving.
   - If both distances are the same, the agent checks from its neighborhood, cells that offer better placement:

     -If the number of empty cells around the agent with a better position is 0 or 1, the agent will respond positively to the move with a 50% probability.

     -If the number of empty cells around the agent with a better position is 2, the agent will respond positively to the move with a 25% probability.

     -If the number of empty cells around the agent with a better position is greater than 3, the agent says no wish to move. With increasing number of empty positions around a cell, the desire to move decreases.

The above percentage values were obtained empirically, using those that best fit the reality. For calculating the distances from each cell to an exit, the Dijkstra´s algorithm was used. This algorithm solves the single-source shortest-paths problem on a weighted graph [10]. The cellular space is considered as a graph, where each cell represents a node and all the edges connecting adjacent cells have weight 1. Cells with state W (wall) or O (obstacle) are not considered to build the graph. If the building has more than one exit, the distance computation takes into account the exit nearest to the cell.

The cell builds a list of candidates with all the agents who want to move. Later, it must select a single pedestrian. The used process for solving this point is the following: a) if more than one individual remains as candidate to occupy the cell, the one with the minor damage grade (parameter specified in the model) will be selected; b) if the conflict persists, the pedestrian will be selected at random. The decision is concentrated in the empty cell.

3. Implementation of Crown Dynamics

Whether to use a discrete or continuous representation of space is closely connected to the implementation. A strong argument in favor of discrete models is that they are simple and can be used for large scale simulations. Additionally, for pedestrian motion there is a finite reaction time, which introduces a time scale. The time is chosen to be discrete in the model and this naturally leads to a discrete representation of space.

It is possible to discretize the progress of an individual pedestrian, however the movement of a crowd should appear as a continuous phenomenon. Consider the situation in Fig. 1, which shows a mass of people around the exit is competing to quickly leave the building. Graphically, it is possible to observe the formation of holes (empty cells) both around the exit and mixed in the crowd.

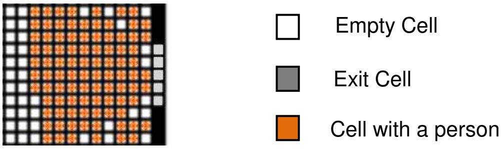


Fig. 1. Evacuation Simulation

In order to analyze this phenomenon, the Fig. 2 (top) examines in detail the progress of two people who are in adjacent cells. The time-step 0, only *B* can move a cell towards the exit. At the same time-step, *A* must remain in place because no rule can move it a cell forward. This is because A, based on their environment, cannot determine that the cell will be abandoned by the current individual. At this point a hole is formed which propagate back across the crowd.

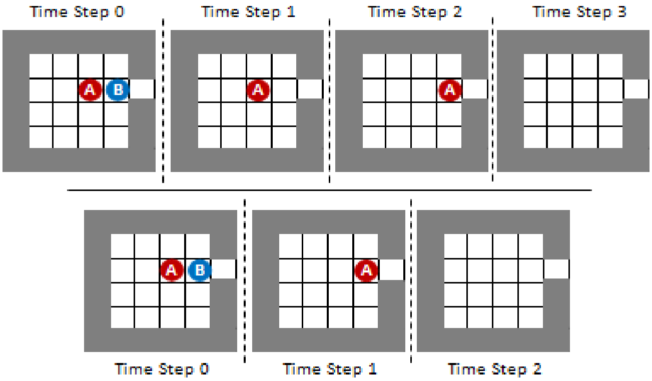
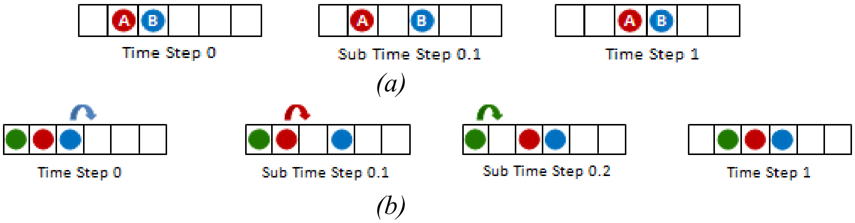


Fig. 2. Advance of two adjacent agents in a CA Model

In the next time-step, 1, *A* can move one cell forward, and the time-step 2 it will exit of building. Although both individuals were in adjacent cells (and more generally belong to a crowd), they could not evacuate in consecutive time-steps. The Fig. 2 (bottom) shows the effect of continuity should be seen in an evacuation of a crowd.

This parallel update (inherent in CA) leads to blocking of cells used during one time-step [11]. An alternative is the introduction of sub steps among two consecutive time-steps of CA, as you can see in Fig. 3(a). Clearly in the case of more than two individuals, we only need to process more than one sub step to resolve the situation. The Fig. 3(b) shows that the procedure is similar to shift all the elements of an array one position forward. In this case, with 3 pedestrian, it is necessary 2 sub steps. It is important to note that the implementation allows us to associate different speeds to individuals. Suppose the evacuation of Fig 4 in a one-dimensional environment. Consider that the individuals *A, C* and *D* have an associated speed of 1.2m/s (3 cells per s.) and individual B of 0.8m/s (2 cells per s.). Individuals *A* and *D* were able to move 3 cells (maximum speed /time-step), *B* advanced 2 cells (maximum velocity/time-step) and *C* was blocked by the individual *B*, being able to move only 2 cells. Note that in the two-dimensional cellular space, the individual *C* might then have found an alternative path and achieves at some point move to 3 cells per time-step.

In this work, the parameter sub steps/time-step has been established empirically. However the Figs. 3 and 4 give an indication of how complex it can be dynamically determine this parameter.



(a)

(b)

Fig. 3. Sub Steps among two consecutive Time-Steps

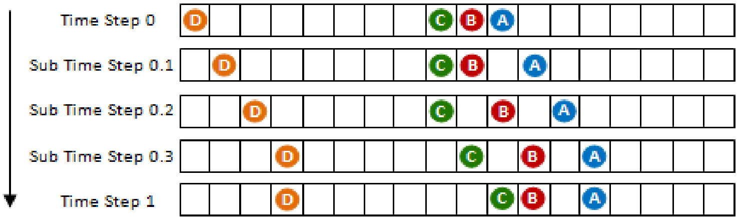


Fig. 4. Evacuation of 4 individuals with different speeds

4. Multi-column Partitioning

The CA are discrete systems that evolve over time. As a general rule, the evolution of CA is performed by repeated update of the complete set of cells, where the new state of each cell depends on the existing state of its neighborhood. CA simulation techniques used in this work, are too slow if the space to simulate is large or complex.

The aim of this work is to search for techniques to accelerate simulations exploiting the parallelism available in current multicomputers.

One of the most common methods used for resolving this problem is the Ghost Cell Pattern [12]. In this technique, the grid is geometrically divided into chunks that are processed by different processors. One challenge with this approach is that the update of points at the periphery of a chunk requires values from neighboring chunks. If a cell and its neighbors are in the same node, the update is easy. On the other hand, when nodes want to update the border cells, they must request the values of the neighboring cells on other nodes. The solution to this problem is to allocate space for a series of ghost cells around the edges of each chunk. For every iteration, each pair of neighbor exchanges a copy of their border and places the received borders in the ghost cell region (see Fig. 5). The ghost cells form a halo around each chunk that contains replicates of the borders of all immediate neighbors. These ghost cells are not updated locally, but provide stencil values when the borders of this chunk are updated. In our proposal, the cellular space of the automaton is represented by an bi-dimensional array, which contains $X \times Y$ cells. Inside of a cluster based on distributed memory system, the parallel execution using P processors (denoted p_0, p_1, $...p_{P-1}$) is performed by applying the transition function simultaneously to P chunks in a SPMD way. Due to the particular characteristics of the model, the proposed parallelization differs a bit from the traditional approach [13, 14].

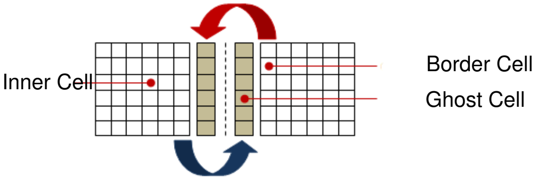


Fig. 5. Border Exchange

The Fig. 6 shows the decomposition applied in our proposal where the cellular space is divided in one-dimensional chunks (multi-column).

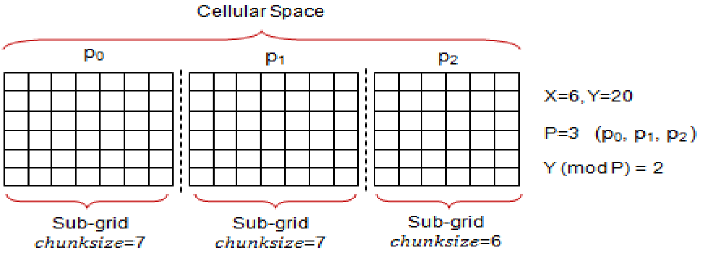


Fig. 6. Multi-column Decomposition: if *P* and *Y* are multiples then *chunksize=Y/P*; if *P* and *Y* are not multiples and *rank<Y (mod P)* then *chunksize= Y/P+1* else *chunksize=Y/P*

In each time-step, the algorithm updates each cell in the lattice. If a cell contains an individual, three things can happen in the next state: (1) the individual may leave the border cell of a chunk and move into the other chunk, (2) it can change of chunk, and (3) it can move from inside a chunk to its border. The Fig. 7 illustrates these movements. Situations 1 and 2 can lead to inconsistencies in the next state.

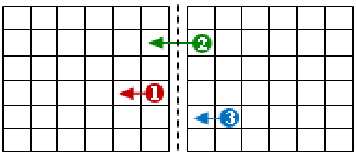


Fig. 7. Possible Movements

Suppose in the Fig. 8-left that the cell marked with *X* is being updated by the process 1. That cell offers a better position than the pedestrian currently possesses (filled circle), so after applying the transition rules it decides to move there. The resulting state would be that the individual has taken the new position, leaving the ghost cell of process 2 outdated, since in the cell still contained the pedestrian who is no longer in that position (Fig. 8-right).

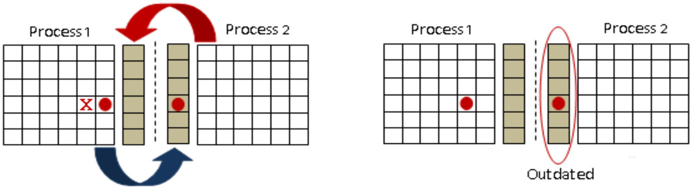
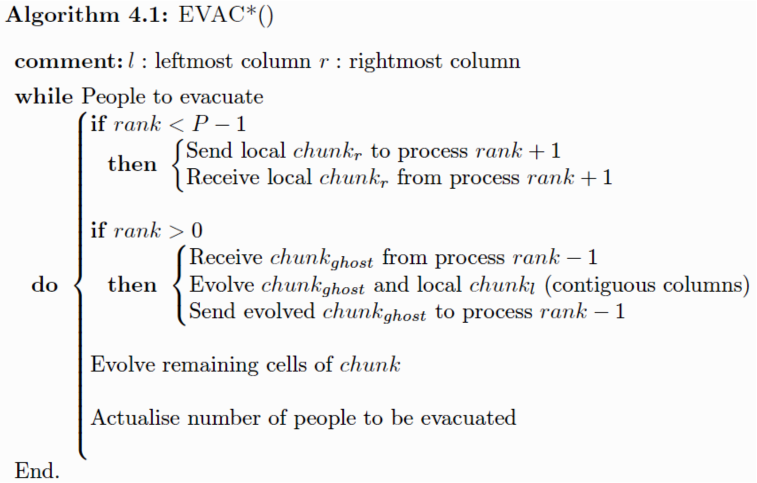


Fig. 8. Status Inconsistency

At first this does not imply any kind of error, but if the individual was placed in a position equidistant from two possible exits, the agent could take a different decision in both processes (the replicated individual moves toward two possible exits).

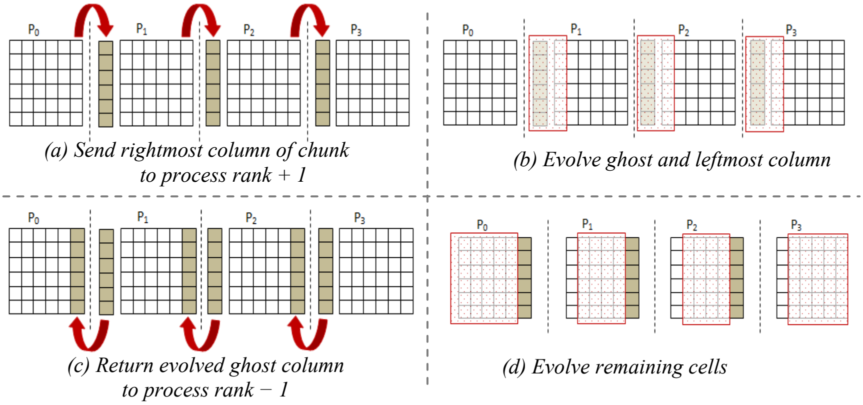


```
Algorithm 4.1: EVAC*()

comment: $l$ : leftmost column $r$ : rightmost column
while People to evacuate
  do
    if $rank < P - 1$
      then Send local $chunk_r$ to process $rank + 1$
           Receive local $chunk_r$ from process $rank + 1$

    if $rank > 0$
      then Receive $chunk_{ghost}$ from process $rank - 1$
           Evolve $chunk_{ghost}$ and local $chunk_l$ (contiguous columns)
           Send evolved $chunk_{ghost}$ to process $rank - 1$

    Evolve remaining cells of $chunk$

    Actualise number of people to be evacuated
End.
```

To keep the two processes with the updated status on the decision of an agent by making use of ghost cell pattern technique requires multiple message exchange during the same time-step. If we want to model large environments (airports, convention centers, stadiums, etc..) and simulate the evacuation of thousands of people, the communication overhead would be significant, implying a degradation of performance of the model.

The Algorithm 4.1 shows the proposed methodology that alleviates the impact of the interaction of processes against the decision to an agent.

The algorithm is illustrated by means of the Fig. 9.



(a) Send rightmost column of chunk to process rank + 1

(b) Evolve ghost and leftmost column

(c) Return evolved ghost column to process rank − 1

(d) Evolve remaining cells

Fig. 9. Multi-column Partitioning Algorithm for CA Model

5. Result of the Simulation

In this section, we present the simulation result of the explained research. The experiments were carried out with EVAC* Simulator [15], an simulation system based on parallel cellular automata. EVAC* is a system developed in C and MPI for passing message and uses the graphical interface of EVAC Simulator. EVAC is a system developed in Java that allows the design and simulation of spatial environments in an sequential way. EVAC simulator offers a friendly graphical interface which can be easily used by non expert users [9]. The experiments consider three environment configurations of the building to be evacuated (*A, B, C*), addressed those situations where several exits exists, varying the different occupation densities of the environment (number of individuals distributed evenly) and the size of the exit:

– *A, 20 × 20 m*^{2}, two exits of 1.2 m. each, 831 pedestrians.

– *B, 40 × 40 m*^{2}, three exits of 1.2 m. each, 1397 pedestrians.

– *C, 80 × 50 m*^{2}, three exits of 2.4 m. each, and 1888 pedestrian.

The experiments are designed to compare the performance of the AC model which runs sub steps between successive time-steps, called the SS experiment, with the original model called *TS* experiment. For the *SS* experiment, we set the speed to all pedestrians in 3 cells/s. The corresponding total evacuation time (*TET*) (seconds) and mean travelled distance (*MD*) (meters) per individual to the exit were obtained. The comparative results are shown in Table 1.

Case	*TET* *(Time-Step Experiment)*	*TET* *(Sub Step Experiment)*	*MD*
A	70.38	18.30	9.99
B	105.00	25.05	22.39
C	138.83	37.08	40.87

Table 1. EVAC* Total Evacuation Time and Mean Travelled Distance

The significant reduction of the evacuation times experienced for the sub step experiment is because individuals are able to move without blocking the exit at the same time-step. While our project is in development, the empirical values obtained for the evacuation time are comparable to other implementations, which have validated their results against real evacuation exercises [16] [11].

For the parallel experiments, we used a cluster equipped with 16 nodes of 64 bits with Intel Q9550 Quad Core 2.83GHz processors and RAM memory of 4GB DDR3 1333M. The nodes are interconnected by a Switch Linksys SLM2048. With both growing environment size and the occupation densities the simulation time increases. Using a partitioning strategy where the cellular space is divided in one-dimensional chunks, we can observe that as the degree of parallelism grows, the simulation performance improves (Fig. 10).

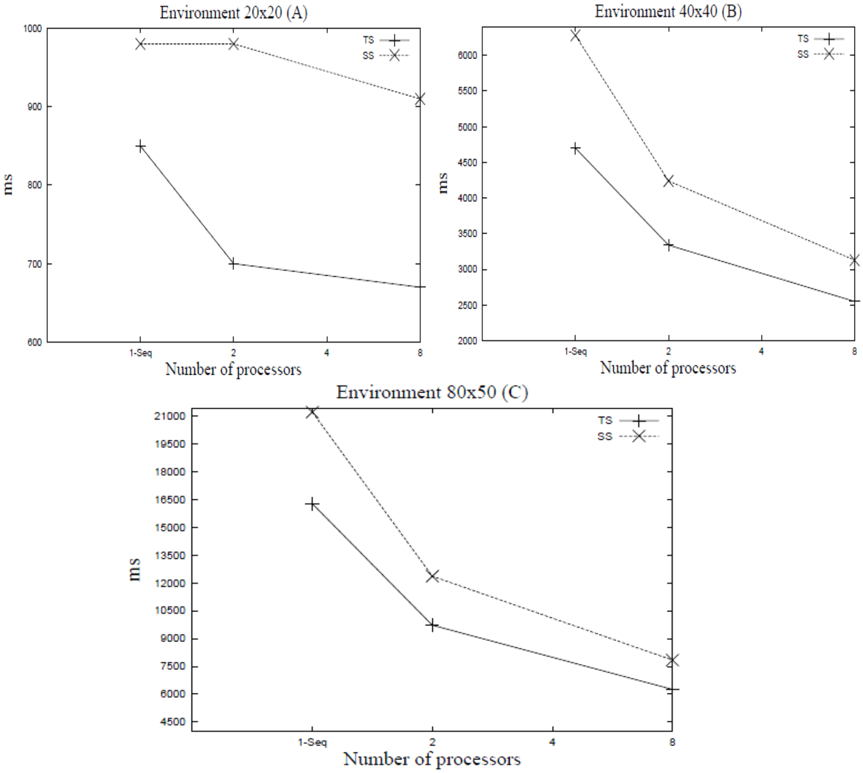


Fig. 10. EVAC*: Parallel Simulation Time

In addition, this new version of evacuation simulator requires the largest computational effort, since the tests have to be carried out with the execution of multiple *sub steps*.

6. Conclusions

In this work, we used Cellular Automata for developing and implementing a simulation model of emergency evacuations due to the fire threat.

In spite of the assumptions introduced for obtaining a simpler model, the CA resulted to be very suitable tools for modeling this class of problems achieving in many cases, results very similar to those expected to occur in a real evacuation situation. The techniques and results reported in this work are part of a research project in the long term. To date, our strategies are only empirically tested. But as future work we wish to compare our research with real evacuation exercises or previous work to validate the proposed strategies.

Simulate the movement of a crowd in a state of emergency is a complex process, not only for its mathematical modeling but also because it should be formalized in the model the behavior that arises from both the natural interaction between individuals and the reaction group (crowd) against a threat that arises in a particular building.

A primary objective of this work is to develop and select techniques of high performance computing (HPC) in order to execute and perform large-scale complex simulations. In this way, we used partitioning techniques for allowing large scale simulations. The simulation is then divided up with the set of environment and its agents being distributed equally amongst the computer nodes. However, distribution introduced its own challenges. The multicolumn partitioning method reduces the communication and synchronization time, while ensuring each agent did not perceive contradictions in the environment.

Finally, we believe that the EVAC* system is a good start point for analyzing and designing preventive safety policies and therefore, we wish to investigate further to optimize both the model and its parallel implementation.

References

1. Nagel Kai. Particle hopping models and traffic flow theory. Phys. Rev. E, 53(5):4655–4672, May 1996.
2. Von Neumann Jhon. The Theory of Self-reproducing Automata. Univ. of Illinois Press, Urbana, IL, 1966.
3. Gardner Martin. Mathematical Games: The fantastic combinations of John Conway's new solitaire game ”Life”. Scientific American, 1970.
4. Wolfram Stephen. Cellular Automata and Complexity. Addison Wesley, USA, 1994.
5. C Burstedde, K Klauck, A Schadschneider, and J Zittartz. Simulation of pedestrian dynamics using a 2-dimensional cellular automaton. Physica A: Statistical Mechanics and its Applications, 295(3-4):507525, 2001.
6. SchreckenbergMichael, Meyer-Knig Tim, and Klpfel Hubert. Simulating mustering and evacuation processes onboard passenger vessels: Model and applications. In The 2nd International Symposium on Human Factors On Board (ISHFOB, 2001.
7. Timmermans Harry. Pedestrian Behavior: Data Collection and Applications. Emerald Group Publishing Limited, 2009.
8. Klupfel Hubert, Schreckenberg Michael, and Meyer-Konig Tim. Models for crowd movement and egress simulation. In Serge P. Hoogendoorn, Stefan Luding, Piet H. L. Bovy, Michael Schreckenberg, and Dietrich E. Wolf, editors, Traffic and Granular Flow 03, pages 357–372. Springer Berlin Heidelberg, 2005.
9. Tissera Cristian. Simulador de evacuaciones basado en autómatas celulares. Informe de tesis de Licenciatura. UNSL, Julio 2006.
10. Brassard G. and Bratley P. Fundamentos de Algoritmia. Prentice Hall, 2000.
11. Klupfel Hubert Ludwig. A Cellular Automaton Model for Crowd Movement and Egress Simulation. PhD thesis, Universitat Duisburg-Essen, 2003.

12. Kjolstad Fredrik Berg and Snir Marc. Ghost cell pattern. In Proceedings of the 2010 Workshop on Parallel Programming Patterns, ParaPLoP ’10, pages 4:1-4:9, New York, NY, USA, 2010. ACM.
13. Nishinari K., Kirchner A., Namazi A., and Schadschneider A. Extended floor field CA model for evacuation dynamics. In IEICE Transactions on Information and Systems, E87-D, pages 726–732, 2004.
14. Blue Victor J. and Adler Jeffrey L. Cellular automata microsimulation for modeling bi-directional pedestrian walkways. Transportation Research Part B: Methodological, 35(3):293–312, March 2001.
15. Tissera Cristian, Printista Marcela, and Errecalde Marcelo. Multi-column partitioning for agent-based ca model. In HPC Proceeding. JAIIO. SADIO-Argentina, 2011.
16. Aik Lim Eng. Exit-selection behaviors during a classroom evacuation. International Journal of the Physical Sciences, 6(13):3218–3231, 2011. http://www.academicjournals.org/IJPS.

X

Information Technology Applied to Education Workshop

A Conversational Agent for the Improvement of Problem-Solving Skills

ELIANE VIGNERON[1,2], LIANE TAROUCO[1], ELISEO REATEGUI[1], MICHELLE LEONHARDT[1], ÁLAN GULARTE[3], ANDREA CAPRA[1]

[1]Federal University of Rio Grande do Sul, [2]Fluminense Federal Institute,
Campos-Centro/RJ campus, [3]Ritter dos Reis University Center/Porto Alegre-RS.

Abstract. *This paper outlines an approach for developing math problem-solving skills using a conversational agent. The knowledge base of the agent uses AIML markup language, built from eliciting formal and heuristic knowledge of gifted students, who won awards in the Brazilian Mathematics Olympiad for Public Schools. The paper describes the method for capturing the cognitive processes of gifted students in solving math problems and the structuring of this knowledge for the conversational agent. The paper also addresses the results achieved by students through the use of a conversational agent.*

Keywords: *Conversational Agent, Math Problems, Students.*

1. Introduction

Researchers indicate that enhancing the critical and creative thinking skills of students is fundamental for developing the ability to solve problems [1][2][3]. The improvement of these skills can be facilitated through the orientation of conversational agents, according to research findings that show that these agents can have a positive effect on the interactive experiences of students [4][5]. Gifted students, with high performance in mathematics, manifest characteristics such as flexibility, use of analogy, data organization and finding patterns and relationships during the solving of mathematical problems [6]. The possibility of capturing and storing the methods used by gifted students enables other students to benefit from these processes for the improvement of problem-solving skills. Therefore, a conversational agent was created, named Blaze, using AIML (Artificial Intelligence Markup Language). The language makes it possible to interact with the agent in natural language [7].

This paper presents the findings of this research, whose main objective was to capture the cognitive problem-solving processes of gifted students in order to store them in the knowledge base of the conversational agent and then use them with other students for improving the skills required for solving problems.

2. Conversational Agents

In the 1980s, intelligent tutoring systems with a knowledge base started to be created in order to guide students in the learning process. An intelligent tutor is a software system that uses artificial intelligence techniques to represent knowledge and interact with students for the purpose of teaching such knowledge. In the 1990s, with advances in cognitive psychology, intelligent tutoring systems evolved from being an instructional proposal aimed at structuring the environment into experimentation and discovery of knowledge [8]. However, despite these advances, the complexity of the structure of these systems greatly limited their practical application. Developing an intelligent tutoring system is a difficult task, and this is due to the complexity of the technology needed for knowledge representation, cognitive modeling, qualitative processing and causal modeling processes, in addition to the difficulty involved in domain knowledge elicitation and representation [4].

A.L.I.C.E. (Artificial Linguistic Internet Computer Entity) [9] was a very original and innovative project in the field of artificial intelligence in the 1990s [7]. It constituted an example of a conversational agent, with an open source system maintained by an active community. The system operates up until today and is composed of two parts: a conversational machine and a knowledge base constructed using AIML (Artificial Intelligence Markup Language). The language has a specific structure composed of categories, which consist of at least two elements: pattern and template, as in the example below.

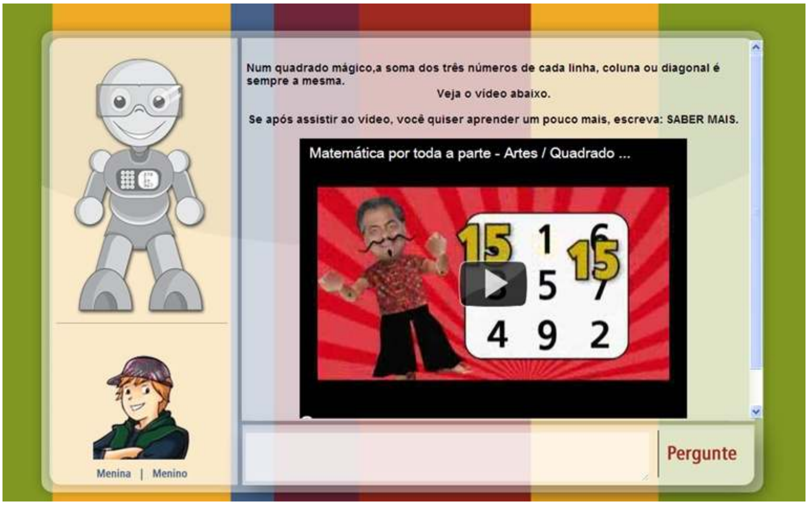
```
<category>
<pattern>possible user input</ pattern>
<template>conversational agent's response</ template>
</category>
```

The operation of agents using AIML is based on a stimulus-response model, where the stimulus (user input) is compared with patterns and when one or more pattern matches occur an associated response is determined, contained in the template that the conversational agent will show to the user [7]. All of these actions, in terms of seeing the appropriate pattern and showing the related template, are loaded by the data treatment machine.

Different conversational agents have been built using AIML language. Cybelle is an agent that interacts in Portuguese, and also in English and French [10]. It gives information about other agents, such as ALICE. Professor Elektra is an educational agent whose main goal is to serve as a complementary learning tool for students doing distance education courses [11]. CHARLIE is an agent responsible for interacting with students and an intelligent educational system, showing course content and asking about the learning material [4]. The wide range of conversational agents developed with inference engines based on the ALICE project led to this choice of a conversational agent project for the improvement of problem-solving skills.

3. Conversational Agent Blaze

The basis of conversational agent Blaze was structured in AIML language from knowledge extracted from gifted students during the course of solving math problems. These students are medalists from the Brazilian Mathematics Olympiad for Public Schools who participate in an undergraduate research project where one of the authors works. The software that enables conversational agent Blaze is hosted on a public server (Pandorabots.com). The agent is capable of responding to students who interact with it on the basis of information stored in its knowledge base. Agent Blaze does not solve problems for students, but can serve as a trained and reliable assistant during the problem-solving process. Through keywords or questions, students can converse with Blaze, which provides tips for solving new math problems. An example of a dialogue between Blaze and a student is presented in Fig. 1.



Num quadrado mágico,a soma dos três números de cada linha, coluna ou diagonal é sempre a mesma.

Veja o vídeo abaixo.

Se após assistir ao vídeo, você quiser aprender um pouco mais, escreva: SABER MAIS.

Fig. 1. Blaze's response with a video on magic squares

In this example, the agent answers a student's query on the subject of "magic squares" and displays a video containing the definition and examples of magic squares.

4. Using Agent Blaze

The research that led to the construction of agent Blaze's knowledge base and its use was developed in two phases. The first phase took place with gifted students from the Undergraduate Research Project from the Brazilian Mathematics Olympiad for Public Schools from the Campos region in Rio de

Janeiro, Brazil. Its goal was to capture the cognitive processes of these gifted students during the solving of math problems.

In the second phase, opportunities for interacting with Blaze were provided for a total of 99 students, of which 66 were high school students, 13 were Science Teaching Degree students and 20 were Mathematics Teaching Degree students.

Some of the findings obtained from the use of agent Blaze with a group of Teaching Degree students are presented in this paper. The main objective of this experiment was to determine whether agent Blaze was able to contribute to improving the math problem-solving skills of students.

4.1 Research with Gifted Students

During the first phase, in the research with gifted students, the "think aloud" method was used, a verbal method chosen to elicit the cognitive processes of the student's mathematical reasoning in regard to solving problems. In this method, it is suggested that the student freely and spontaneously verbalize, out loud, all the thoughts that occur during the execution of the task. These are recorded in audio and, occasionally, in video, after which the responses are analyzed and categorized.

For example, one of the problems given to students involved right triangle trigonometry: ***Triangle ABC has sides AB =*** $\sqrt{12}$***, BC = 4 and CA =*** $\sqrt{20}$***. Calculate the area of triangle ABC.***

The gifted student solved the problem and verbally presented the method used, through a dialogue with the teacher, as follows:

> **Student:** *I plotted the height and then separated the triangles. I used 4 as a base, one side I called b and the other 4 – b.*
> **Professor:** *4?*
> **Student:** *The sides were* $\sqrt{12}$ *,* $\sqrt{20}$ *and 4. I set it up like that. Then I used the Pythagorean Theorem with each one. First in the one with* $\sqrt{12}$ *and after in the one with* $\sqrt{20}$ *. Then one goes hooking up with the other. I found that* $20 = b^2 + h^2$ *replaces the other, I found the height to be equal to* $\sqrt{11}$ *, as required by the area, which is equal to the base four times the height* $\sqrt{11}$ *over 2. And 4 divided by 2 equals 2.*

Apart from the verbal presentation, students also represented in writing how they solved the problem, as can be seen in Fig. 2. A written representation was necessary because this problem with right triangle trigonometry requires a geometric construction of the triangle which, along with the other information given in the question, allows for a better understanding of the process used for solving it.

The different representations are complementary, since capturing the cognitive thinking of students while solving a problem only through words may not clearly reflect the process used. Often, the decisive idea that solves a problem is linked to a well-chosen word or phrase [12]. The basic reasoning mechanisms adopted by gifted students in the problem-solving process focus on retrieving concepts and, also, on combining ideas to reach the solution. The reasoning of students is based on past experience, that is, on prior knowledge which is a powerful means for people to solve problems. Apart from that, talent and high intelligence are associated with selective comparison (effectiveness in recalling similar problem situations that were previously resolved) or analogical reasoning [13]. At school, as well as in daily life, new problems arise whose solution often comes from prior learning and experiences acquired through solving similar problems.

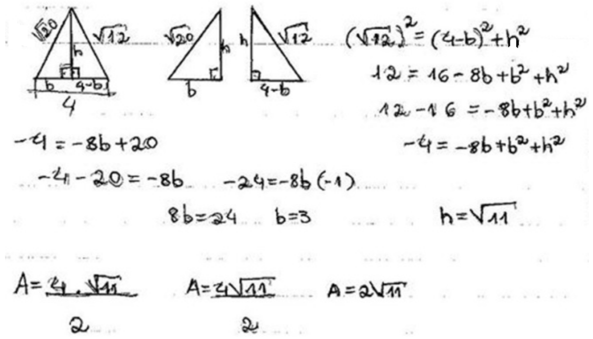


Fig. 2. Representation of the right triangle trigonometry problem

Therefore, in order for new problems to be solved by students who do not have the knowledge base and experiences of gifted students, a set of cases and respective solutions was incorporated into the knowledge base of conversational agent Blaze. Students can access this knowledge base interactively using its mechanisms of concept retrieval, comparison and combination of ideas to support their own strategy for solving problems. This approach allows more inexperienced students to improve their problem-solving skills to an "expert" level, which in this case represents the gifted students.

4.2 Interaction of Students with Blaze

In the second phase, 13 Science Teaching Degree students went to the computer lab to solve math problems with the support of agent Blaze. During this stage, students were given a handout with guidelines on Polya’s heuristic method for problem solving [12] which involves five steps: understanding the problem, representation, devising a plan for solving it, carrying out the plan and verifying the solution.

In addition, an already-solved math problem was provided showing ways students could interact with agent Blaze, that is, with tips on how to ask Blaze questions. This procedure was adopted in order to supply directions and facilitate interaction between students and Blaze, since it was the first time students were interacting with the agent.

This computer system, by which agent Blaze provides assistance to students, seeks, among other goals, to increase the performance of students as they learn individually, since beginning learners are unable to organize their problem-solving strategies for lack of experience with similar problems as their store of knowledge is smaller. So, with the help of a computer system, students can learn to solve problems more easily and improve their performance [14].

During the solving of math problems with the support of agent Blaze, students are inserted into a self-regulated learning process that has four attributes: self-motivation, planning or automatization, self-awareness concerning performance results and skill within the learning environment [15]. After solving the problems that were posed, students answered a questionnaire containing six questions. The first three checked the frequency of computer use, the fourth established the interest/engagement of the student during the study, question 5 verified the importance of conversational agent Blaze in the problem-solving process and question 6 requested students to state which of the different problems that were presented they considered themselves capable of solving without Blaze's help and why.

5. Results and Discussion

The results of the first three questions showed that all the participants were very familiar with computers and the Internet. In turn, the work done with the support of agent Blaze used the concept of cognitive scaffolding, providing orientation to help learners while solving problems. Scaffolding guides students to make predictions, experiment, reflect, write explanations, collaborate, contribute to online discussions and participate in classroom discussions [16]. Within the context of this project, therefore, the purpose of scaffolding was to supply new learners with a learning environment with limited complexity and gradually remove the limits until students became more skilled, as proposed by Young [17].

Question 4, which seeks to verify the engagement level of students in the study (Table 1), has alternatives that were prepared based on the variables deemed necessary for engagement analysis as per Blom [18]. The concept of engagement is directly related to the motivation that a participant has in truly performing a task, for which outside reward is not needed [19]. It can be noted that students were highly engaged in solving the math problems, as shown in Table 1. Concerning the evaluation of students' engagement with using a computer system, the use of a human figure can result in an increase in engagement due to factors such as student identification with the character and novelty [18]. In Table 1, 31% of students strongly agree and 54% partially agree that they were conscious of their decisions for reaching solutions to the problems. These represent the actions and behavior of learners, made with self-control. This is metacognition, which is conscious and involves reflection. Also in Table 1, 61% of students agree that they were in control of the situation and 54% agree that they were feeling good about themselves, which denotes self-evaluation, a process of metacognition, understood as an internal mental process by which learners themselves are aware of the different stages and aspects of their cognitive activity [20].

Table 1. Results presented by students in question 4 of the questionnaire

4) On a scale of 1 to 5, study the statements below and select the appropriate answer for each situation. Each item refers to your emotional state/behavior while solving the problems.						
	1	2	3	4	5	Abstain
	Strongly Disagree	Partially Disagree	Indifferent	Partially Agree	Strongly Agree	
a) While solving the problems, I was concentrated.	0 (0%)	0 (0%)	0 (0%)	8 (62%)	5 (38%)	0 (0%)
b) I was very conscious of my decisions in reaching the solution.	0 (0%)	1 (8%)	1 (8%)	7 (54%)	4 (31%)	0 (0%)
c) I was in control of the situation.	1 (8%)	4 (31%)	0 (0%)	6 (46%)	2 (15%)	0 (0%)
d) I felt good about myself.	0 (0%)	4 (31%)	1 (8%)	3 (23%)	4 (31%)	1 (8%)
e) My performance exceeded my expectations.	1 (8%)	3 (23%)	2(15%)	6 (46%)	0 (0%)	1 (8%)
f) I succeeded in solving the exercises, finding the solutions to the problems.	3 (23%)	3 (23%)	0 (0%)	7 (54%)	0 (0%)	0 (0%)
Note: For each item above, from a) to f), the frequency of responses was placed in the table by number of students and percentage, with a total of 13 students participating in the survey.						

The experiment that was conducted showed a level of student engagement equivalent to 4.15 on a scale of 1 to 5. This result demonstrated that students were very involved in the realization of the proposed activity with agent Blaze. In turn, Table 2 presents a set of questions that seek to establish whether agent Blaze provided support to students in solving problems. The results show that, among the students surveyed, 85% agree that Blaze's assistance enabled them to obtain solutions to the mathematical problems.

Table 2. Results presented by students in question 5 of the questionnaire

5) This question evaluates the importance of Blaze's assistance for solving the math problems.					
	1	2	3	4	5
	Strongly Disagree	Partially Disagree	Indiffer-ent	Partially Agree	Strongly Agree
a) The agent made suggestions that helped recall relevant information for solving the problems.	0 (0%)	0 (0%)	0 (0%)	4 (31%)	9 (69%)
b) Interaction with Blaze respects the student's own pace.	0 (0%)	0 (0%)	2 (15%)	3 (23%)	8 (62%)
c) Blaze offers students individualized help.	0 (0%)	2 (15%)	0 (0%)	2 (15%)	9 (69%)
d) Interaction with agent Blaze enabled using new ways to solve the problems.	0 (0%)	1 (8%)	1 (8%)	2 (15%)	9 (69%)

e) Interaction with the agent made for an improvement in the chain of ideas for solving the problems.	0 (0%)	1 (8%)	0 (0%)	9 (69%)	3 (23%)
f) I will be able to use the type of assistance provided by Blaze, even without being asked by the professor.	0 (0%)	0 (0%)	1 (8%)	3 (23%)	9 (69%)
g) Blaze's assistance enabled solutions to be reached for solving mathematical problems.	0 (0%)	2 (15%)	0 (0%)	7 (54%)	4 (31%)
h) I think this kind of support should also be given for problems in other fields, such as Physics, Chemistry and Biology.	0 (0%)	0 (0%)	0 (0%)	1 (8%)	12 (92%)
i) I recommend Blaze's assistance to my fellow students for solving mathematical problems.	0 (0%)	2 (15%)	0 (0%)	5 (38%)	6 (46%)
Note: For each item above, from a) to i), the frequency of responses was placed in the table by number of students and percentage, with a total of 13 students participating in the survey.					

The answers obtained in question 6 reinforced this finding in consulting with the participants in the study about: *Which of the math problems would you be able to solve without Blaze's help? Why?* Some of the comments made by students about this question are presented below:

- 31% would not be able to solve the questions since:
  - ✓*I wasn't familiar with terms that were in the questions, such as what a "magic square" or "golden ratio" is. It was necessary to research the meaning of these terms and only then begin to solve the questions;*
  - ✓*I was almost unable to solve any question, because I needed another type of help that the robot could not give me. However, without the concepts it gave me, I wouldn't even be able to start the 1st and 2nd. I also wouldn't be able to do the 4th, since it requires more research.*
- 31% would only be able to solve question 3 and explain why with reasons such as:
  - ✓*I used logical thinking;*
  - ✓*all that students needed to know were the multiples of the numbers requested in order to solve the question;*
  - ✓*the reasoning was more logical and the trial and error method could be used;*
  - ✓*because it was a question that depended more on organizing the numbers as opposed to more complex calculations.*
- 15% would only be able to solve question 2, explaining why with reasons such as:
  - ✓*I used logic to solve it;*
  - ✓*I used trial and error.*
- 15% would be able to solve questions 2 and 3 since:
  - ✓*I would be able to solve them using my knowledge of mathematical logic;*
  - ✓*because I already had previous knowledge about the magic square and the concepts of divisibility and probability.*
- 8% no response.

The importance of retrieving concepts during the problem-solving process was noteworthy in the answers given by students for item (a) of question 5 (Table 2), where 31% of students partially agree and 69% of students strongly agree that agent Blaze suggested helpful ways to recall important information. The use of different methods for presenting mathematical content made possible by agent Blaze, for example, through video, stimulated creativity as well as critical thinking in students. Students showed an understanding of the importance of learning in a diversified way in their answer to item (d) (question 5, Table 2).

The need for an individualized approach in the learning process for mathematics was emphasized in the response of option (c) (question 5, Table 2). These results demonstrated that Blaze's assistance was important during the solving of problems. The conversational agent's help provided a motivating learning environment, which supported the pursuit of strategies for solving problems, thus favoring the acquisition of problem-solving skills. It is also worth noting that other studies performed with an experimental group (which used agent Blaze) and a control group (which worked without the help of agent Blaze) revealed situations where the control group could not solve certain problems, while the experimental group was able to obtain the solutions with the help of the agent.

6. Conclusion

This paper presented a conversational agent model capable of representing the problem-solving processes of gifted students in order to provide support to other students for solving mathematical problems. The conversational agent developed was able to interact with students, showing applicable strategies for the process of solving new problems, thereby helping develop cognitive skills in a meaningful learning process. According to Ausubel [21], meaningful learning occurs when new concepts are connected with relevant prior concepts in the cognitive structure of students (subsumers). If students do not have these subsumers to serve as an anchor for meaningful learning, their interaction with agent Blaze can create opportunities for forming new subsumers which, in turn, will render them more skillful in solving problems.

In this study, the representation of agent Blaze's knowledge used AIML language. This form of representation has been improved along the way in a process where new developments can be anticipated, such as the use of generic search engines combined with the agent's work, as an extension of the body of information stored in its knowledge base. It is intended, in future studies, to implement in agent Blaze a register for storing the data of students, in addition to recording the history of the agent's dialogue with students. This feature could contribute to improving the interaction mechanism between the agent and students, as well as customize how the agent interacts with students.

References

1. SENDAG, S. and ODABASI, H. F. Effects of an online problem based learning course on content knowledge acquisition and critical thinking skills. Computers and Education, 53(1), pp. 132-141, 2009.
2. VILA, A. e CALLEJO, M. L. Matemática para aprender a pensar: o papel das crenças na resolução de problemas. Porto Alegre: Artmed, 2006, 212 p.
3. KALAYCI, N. Sosyal bilgilerde problem çözme ve uygulamalar. (Problem solving and applications in social sciences.) Ankara: Gazi Kitapevi, 2001.
4. MIKIC, F. A.; BURGUILLO, J. C.; LLAMAS, M.; RODRIGUEZ, D. A. and RODRIGUEZ, E. CHARLIE: An AIML-based Chatterbot which Works as an Interface among INES and Humans. Telematics Engineering Department, University of Vigo, Spain, 2009.
5. ANDRE, E., RIST, T. and MULLER, J. Employing AI methods to control the behavior of animated interface agents. Applied Artificial Intelligence, Vol. 13, Num. 4-5, May 1999, p. 415-448.
6. JOHNSEN, S. K. Definitions, Models, and Characteristics of Gifted Students. In: Identifying Gifted Students: A Practical Guide, 2004. Available at: <http://www.prufrock.com/client/client_pages/Definitions_and_Characteristics/Definitions_and_Characteristics_of_Gifted_Students.cfm> Acesso em 13 maio de 2010.
7. WALLACE, R. ALICE – Artificial Linguistic Internet Computer Entity – The A.L.I.C.E A.I. Foundation. 1995. Available at: <http://alicebot.blogspot.com/> Acesso em 10 jul. 2010.
8. MURRAY, T. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art, Internat. Journal of Artificial Intelligence in Education, 10, 98-129, 1999.
9. A.L.I.C.E. Artificial Intelligence Foundation. Available at: <http://www.alicebot.org/>. Accessed in April, 2011.
10. Agentland. CYBELLE. 2002. Available at: <http://www.agentland.com> Accessed in April, 2011.
11. LEONHARDT, M. D. Doroty: um Chatterbot para treinamento de profissionais atuantes no gerenciamento de redes de computadores. Dissertação de Mestrado. Porto Alegre: UFRGS, 2005.
12. POLYA, G. A arte de resolver problemas. Rio de Janeiro: Interciência, 2006.
13. MANTARAS, R. L. et al. Retrieval, reuse, revision and retention in case based reasoning. The Knowledge Engineering Review, vol. 20 (3), p. 215 – 240, 2006.
14. SHIH, K.-P.; CHEN, H.-C.; CHANG, C.-Y. and KAO, T.-C. The development and implementation of scaffolding-based self-regulated learning system for e/m-learning. Educational Technology & Society, 13(1), pp.80 – 93, 2010.
15. SCHUNK, D. H. and ZIMMERMAN, B. J. Self-regulation of learning and performance: Issues and educational applications. Hilldale: New Jersey: Lawrence Erlbaum. 1994.
16. LEE, H.-S.; LINN, M. C.; VARMA, K. and LIU, O. L. How do technology-enhanced inquiry science units impact classroom learning? Journal of Research in Science Teaching. vol.47, n. 1, pp 71 -90, 2010.
17. YOUNG, M. F. Instructional design for situated learning. Educational Technology Research & Development. 41(1), pp.43-58, 1993.
18. BLOM, J. Psychological Implications of Personalised User Interface. Doctor of Philosophy. University of York, England. 2002.

19. PAAS, F.G.W.C. and VAN MERRIËNBOER, J. J. G. An instructional design model for the training of complex cognitive skills. Tijdschrift voor Onderwijsresearch, 17, 17-27, 1993.
20. SANTOS, L. Auto-avaliação regulada: porquê, o quê e como? Universidade de Lisboa, 2001. Accessed in May, 2010. Available at: <http://area.fc.ul.pt/en/artigos%20publicados%20nacionais/F.pdf>
21. AUSUBEL, D. P.; NOVAK, J. D. e HANESIAN, H. Psicologia Educacional, Interamericana Editora, 2. ed., 1978.

ECALEAD - Quality Assessment in Distance Education. Analysis of the proposed model

GLADYS GORGA[1], CECILIA SANZ[1], CRISTINA MADOZ[1]

[1] Instituto de Investigación en Informática LIDI – School of Computer Science – National University of La Plata. 50 y 120, La Plata, Buenos Aires, Argentina
{ggorga, csanz, cmadoz}@lidi.info.unlp.edu.ar

Abstract. *A proposal of a model to assess ICT-mediated educational processes is presented. The education quality background is analyzed, particularly in relation to distance education. The model proposed is described, and then, its strengths and weaknesses are discussed, based on its application to some educational experiences. Finally, the results are analyzed and some conclusions are presented.*

Keywords: *quality, education, ICTs, assessment model.*

1. Introduction

The evolution of our society is currently supported, among other aspects, by the development of new Information and Communication Technologies, driven by advances in computer science and telematics, that affect every aspect of life.

Higher Education Institutions (IES) are not exempt from this situation, and they must take advantage of these resources to their maximum potential, so that they can generate innovative proposals that help them achieve their goals.

In this sense, IES will have to set clear quality standards and criteria to achieve their goals taking into account, mainly, the critical role they play in the current knowledge society.

Nowadays, there is a vast rather than intense proliferation and production of distance education programs, but there is no associated reflexive or deep assessment process. To do this, criteria such as filters should be conceived and established to check if these educational proposals are reliable and offer quality education [1]

To determine the set of quality criteria to be used by the IES, either for on-site or distance programs, the specific contexts in which these activities take place should be analyzed, also including, but not limited to, their components, critical issues, and the actors and their characteristics, needs and demands. All these aspects related to the goals sought will determine the roads that the educational project will follow within a quality and continued improvement context.

2. Educational Quality Background

There is no consensus regarding the concept of quality, particularly in higher education. Many authors state that the concept of quality is relative. However, the current trend is to consider that the University should analyze its quality in relation to its nature and role, its purposes and goals. Thus, quality includes the relevance of university goals and the extent to which these are achieved [2]

García Aretio states that "…the various power or thought groups can consider that quality has been achieved based on the starting premise and the intended goals." On the other hand, he mentions that a large number of the various proposals and models are rooted on the European Quality Model (TQM), since they focus their interest mainly on customer satisfaction, which is based on continued improvement, process measurement and maximum attention to processes, team work, and individual responsibilities [3].

However, there are other authors, such as Barberá et al [4], who state that "…the first mistake is copying, almost to the T, the business quality models and apply them to education; neither form nor substance can be transferred to educational environments as it is being done."

As mentioned earlier, there are different opinions on how to consider quality and its assessment.

In the case of ICT-mediated educational processes, there is also a diversity of aspects to consider when defining quality criteria and how to achieve it.

In [5], several renowned university members with experience in distance education programs analyze various strategic components that affect quality. For instance, at Monash University in Australia, digital educational materials are considered to be a key aspect, as well as the appropriate training of both teachers and students to be able to use these materials. At the University of Texas, the issue of quality is considered to be based on three main cornerstones – the quality of educational materials, presentation, and student attention; whereas at the US Open University, quality is viewed from the following perspectives: content and design of educational materials, teacher-student communication and with the institution in general, and the use of appropriate technology to support these processes.

In [6], continuous assessment processes are considered as a tool to ensure education quality. For ICT-based learning to be successful, its organization, program consistency including teacher and student training, course design, the appropriate incorporation of emerging technologies, and an effective assessment of the educational process have to be considered.

García Aretio [7] proposes a model that he describes as being inclusive and that allows developing and monitoring total quality, and that is linked to the context, goals, inputs, processes, results, and improvements. The criteria to be considered in this model are: **Functionality** (consistency between goals and educational results in relation to needs and contextual reality), **Efficacy or effectiveness** (consistency between educational goals and the results obtained), **Efficiency** (consistency between inputs, processes, means and achievements or educational results), **Availability** (consistency between the goals and objectives proposed by the institution and human, material and economic resources that are available to start the process), **Information** (consistency between the results obtained and the improvement proposals that are offered in the corresponding report), **Innovation** (consistency between the specific catalogue of improvements required – developing strengths and correcting weaknesses – to achieve goals and the decision to innovate and review goals, inputs and processes).

In the following section, the proposal designed by the authors of this paper is presented. The goal of this model is assessing the quality of a distance educational system, program or process, and it incorporates some of the elements included on the previous list. This model can also be adapted to hybrid educational processes, where on-site meetings are combined with distance work and learning strategies.

3. ECALEAD - Educational Model to Assess the Quality of Distance Processes

ECALEAD is a quality assessment model for distance educational systems or hybrid models. It is based on a layered assessment that starts with a first layer that analyzes general aspects, and ends with a third layer that applies certain measures to analyze the criteria and indicators defined in the two previous layers to assess a specific context. As layer depth increases, they become more specific in relation to aspects typical of ICT-mediated education. The model can be adapted based on the specific purpose of the assessment, context, and the specific needs in each case.

To better explain each of the layers, some of the components that are part of an educational system have been summarized (Figure 1). Thus, the educational institution is inserted in a certain socio-cultural, political and economical context that has certain demands and needs. There are indicators that the IES have into account to learn what the context requires or the direction in which they are evolving.

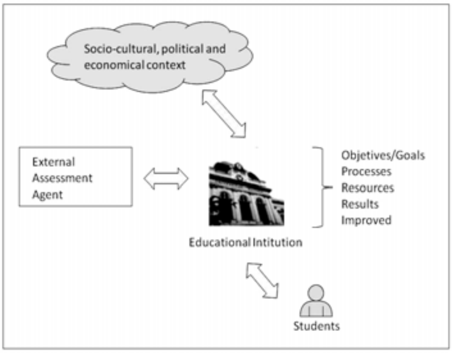


Fig. 1. System components to be taken into account for the quality assessment model proposed in this paper

Even though students are part of this context we refer to, they have been assigned a central role as component in the system, since they have specific interests in relation to the institution, they are part of it, and expect certain actions or results from the IES. They can provide information about their needs and expectations. Similarly, the IES have an effect on students (based on their goals/objectives). Students experience the institution and can give their opinion about it. External Assessment Agents are another essential component in the system; they gather information about the institution and give their specific feedback to achieve evolution in educational quality.

In order to operate, the institution defines its objectives/goals and the processes to achieve them, it has various types of resources (which are organized and distributed based on certain strategies), it pursues results that are assessed by means of internal and/or external process and, ultimately, it determines the aspects that can be improved. We understand that, if the institution achieves functionality, availability, efficacy, efficiency, information and innovation (based on Aretio's definitions), it has progressed significantly towards ensuring the quality of the system.

In the following paragraphs, each of the layers in the model proposed is described.

Layer 1: In this layer, the general criteria to be considered in the model are defined. The criteria from García Aretio's model have been considered here, but in relation to the components described in the previous paragraph (goals/objectives, processes, resources, results, improvements). These can be seen in Figure 2.

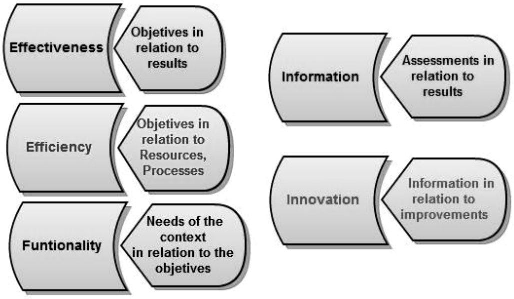

Effectiveness
Objetives in relation to results
Efficiency
Objetives in relation to Resources, Processes
Funtionality
Needs of the context in relation to the objetives
Information
Assessments in relation to results
Innovation
Information in relation to improvements

Fig. 2. Criteria to be considered in Layer 1, based on the components related to an IES, in the context of this paper

It should be noted that those involved in applying this model to assess a specific case should define whether all these criteria will be considered or not. Layer 1 involves this type of decisions by adjusting criteria based on context needs.

Layer 2: For the second layer of the model, some indicators that are directly related to the criteria in the first layer are proposed. To this end, the processes, resources and results that are considered to be of interest for a distance education system must be determined. When applying the second layer to a specific assessment, the indicators proposed by the model need to be adjusted based on the context. Not all of them will be used every time, and in some cases it might be necessary to add new ones. The model presents only a set of indicators that are considered to be relevant for the assessment of distance education systems in general. In the following paragraphs, the most significant processes that should be considered, as well and a few possible indicators, are described, also taking into account goals, resources and results.

Administration and Management: this process involves establishing activities for the promotion, enrollment, attention of administrative queries, student and teacher management (maintaining, recording and providing information about courses passed, grades, certifications obtained, etc.). Activities to hand in access credentials for virtual teaching and learning environments (if any), etc. Some possible indicators:
Related to functionality: for instance, number of students and teachers who express their satisfaction with the administrative and query services available at the institution.
Related to efficacy and efficiency: number of enrollments (analyzing student origin) received in relation to the number of human resources that handle those enrollments and available options for on-site or distance enrollment; number of students initiating the process after enrollment in relation to the number of enrolled students. These can show a lack of appropriate administrative information; number of administrative tasks carried out during the day in relation to the number of tasks finished; number/percentage of queries received through the various available media.
Related to information and innovation: verification of the existence of reports/statistics about Management and Administration processes (these can be based on the indicators defined for the other criteria); number of improvements and changes implemented versus the improvement plan determined by the institution.

Teaching and Learning: this process includes the following, but not limited to these: content planning and organization, definition of specific goals for each course, definition of strategies and appropriate teaching activities, design and development of teaching materials supported in various media for the different learning styles (related to the process of designing and producing materials mentioned below), analysis and definition of methodologies and support strategies with ICTs, definition of interaction and communication strategies (analysis of their relation to the technology supports available, use feasibility), collaborative and/or cooperative work strategies, content assessment and redefinition, assessment of media incorporated, assessment of individual learning. Some possible indicators are:
Related to functionality: learning expectations on the part of students in relation of the goals/objectives of the proposal; teaching expectations on the part of the teachers involved in relation to the proposed goals/objectives; number of graduates inserted in the job market in relation to the total number

of students graduated (this indicator could show if system goals are in agreement with market job needs).
Related to efficacy: number of students who have obtained their accreditation versus number of students enrolled; number of students, teachers and coordinators who have expressed their satisfaction with the teaching and learning process (this could be refined for the various aspects of the process, such as those related to teaching individualization strategies, interaction, interactivity, integration enhancement, critical thinking, etc.); number of students with a satisfactory performance in courses taken after the current one; number of satisfactory opinions (from certain work environments) in relation to the performance of the graduates produced by the institution, in fields related to the contents and competencies taught.
Related to efficiency: number of students assigned to each tutor; number of queries received in the teaching process in relation to the number of queries answered; number of technology resources used as part of the teaching strategy in relation to available resources; number of teaching activities proposed in relation to the duration of the course and time available.
Related to information and innovation: existence of reports prepared by those involved in the teaching and learning processes to express their opinion on that regard (these can be based on the indicators defined for the other criteria); number of improvements and changes implemented versus the improvement plan determined by the institution.

Development of Policies and Legal Regulations: definition of minimum methodological criteria, definitions on issues such as intellectual property, software licenses, institutional recommendations, criteria for the selection of teachers and resources, etc. Some possible indicators are:
Related to functionality: number of situations or events occurred that were not contemplated by the established rules and regulations.
Related to efficacy and efficiency: number of exceptions to the established regulations and policies; number of criteria and regulations that are actually implemented by the human resources working at the institution versus number of criteria and regulations established in each case.
Related to information and innovation: existence of reports/statistics that analyze the appropriateness of the existing policies and regulations, as well as the detection of new needs (these can be based on the indicators defined for the other criteria in this process); number of improvements and changes implemented versus the improvement plan determined by the institution.

Fund Management and Financial/Economic Support: channels for obtaining funds, balance distribution of available funds and resources in accordance to priorities in goals/objectives. Analysis of economic supports to face the costs of the resources involved, etc. Some possible indicators are:
Related to functionality: number of state and private organizations that finance, encourage, and grant awards, scholarships and incentives for the IES to have distance education programs.
Related to efficacy and efficiency: number of resources proposed in relation to the economic resources assigned; number of needs/resources cast aside due

to economic resources; number of goals achieved in relation to invested money; money invested by students in on-site systems versus the distance system that is being analyzed; money invested by the institution in an on-site system versus the distance system being analyzed.
Related to information and innovation: existence of reports/statistics that analyze the relation between the money invested, the goals/objectives set and the results obtained (these can be based on the indicators defined for the other criteria in this process); number of improvements and changes implemented versus the improvement plan determined by the institution.

Teacher Selection, Formation and Training: definition of teacher selection strategies and criteria, definition of tutorial roles (administrative, technological, academic, assessment, and any other the institution considers appropriate), tutorial training and formation, identification of areas with formation needs, etc. Some possible indicators are:
Related to functionality: student expectations regarding tutorial actions in relation to the formation and training provided to tutors by the institution; number of tutors incorporated to the institution with previous formation in relation to the formation requirements imposed by the institution.
Related to efficacy and efficiency: number of trained/formed tutors in relation to the number of tutors required; number of resources (technological, human, etc.) available for training/forming tutors in relation to the number of tutors to be trained, this indicator should be refined taking into account the resources required in each case.
Related to information and innovation: existence of reports/statistics that include training-related information based on the needs of the institution and the results (these can be based on the indicators defined for the other criteria in this process); number of improvements and changes implemented versus the improvement plan determined by the institution.

Design and Production of Educational Materials and Existence of Repositories: methodological definition and roles involved, material design, implementation, storage and availability strategies (including repositories). Connection with intellectual property policies. Some possible indicators are:
Related to functionality: number of topics to teach in relation to the number of topics dealt with in the educational materials; availability of flexible environments that allow storing and/or posting materials and support documents for teaching program contents; availability of diverse material formats based on the needs of students and teachers; number of materials that comply with accessibility standards; number of materials that comply with e-learning standards.
Related to efficacy and efficiency: number of resources (different experts, technology tools, etc.) assigned based on material design and production requirements; feedback from students and teachers regarding the clarity with which contents are presented; feedback from students, teachers and coordinators regarding the consistency of material contents with course or program objectives; feedback from students, teachers and coordinators regarding content relevance and how current they are; feedback from

students, teachers and coordinators regarding the appropriateness of the contents for favoring learning, critical reflection, analysis and research.
Related to information and innovation: existence of reports/statistics that analyze this process in relation to study materials, based on IES needs (these can be based on the indicators defined for the other criteria in this process); number of improvements and changes implemented versus the improvement plan determined by the institution.

Internal Assessment: definition of assessment objects, definition of the assessment plan (taking into account moments, instruments, players involved, etc.), generation of results and improvement plans. Possible indicators:
Related to functionality: extent to which the internal assessment complies with national and international standards.
Related to efficacy and efficiency: number of resources assigned in relation to the needs of the assessment processes (this aspect should be refined based on the needs detected in each case); feedback from teachers, students and management regarding the quality of the assessment instruments developed; number of aspects suggested by the human resources involved in the assessment process that were not considered by the instruments.
Related to information and innovation: existence of reports/statistics that analyze the instruments used for the assessment, their usefulness and quality (these can be based on the indicators defined for the other criteria in this process); number of improvements and changes implemented versus the improvement plan determined by the institution.

It should be noted that the set of examples of indicators provided here is not comprehensive, but it allows guiding the type of work that is to be carried out for Layer 2. As already explained, the definition of specific indicators should be set in accordance with the purpose of the assessment and its context. Readers should also check some more general examples in [8].

Layer 3: For this layer, each of the process and criteria indicators proposed are analyzed in order to carry out the assessment of the aspect under consideration (course, system, distance education project). This is the most specific layer and should be adjusted to the context. As it can be seen from the concepts developed for layer 2, there are some indicators that are quantitative, while others are qualitative. In each case, the definition of scales to determine quality will depend on this.

3.1 Application of the Model to a Study Case

The assessment model application has been presented in previous works for a specific study case. The case studied was the pre-entry distance course of the School of Computer Science of the UNLP; this course has been operational for 8 years now, and has processes that define promotion, enrollment, query channels and support for the educational process at EVEA WebUNLP. For the assessment, some of the criteria proposed for two processes were analyzed: Administration and Management, and Teaching and Learning. In particular, the Efficiency, Efficacy and Functionality criteria were considered [9] [10].

This has allowed starting a process to analyze the strengths and weaknesses of this proposal, and then tweaking it or providing a series of recommendations before applying it to the next case.

In the following section, the analysis carried out on the model proposed is presented, detailing its strengths and weaknesses together with some recommendations.

4. Analysis of the Strengths and Weaknesses of the Model Proposed

The analysis is organized as follows: first, general results are presented, and then details will be provided based on the strengths and weaknesses found.

ECALEAD presents a three-layered assessment model that allows gradually moving from the definition of processes, criteria, and indicators to the measures that should be considered for the assessment of anything from distance education systems to specific educational processes such as a distance education course. Also, it can be adjusted to analyze hybrid educational processes, such as blended learning or extended learning. This shows the flexibility of the model proposed. However, this flexibility forces those working with ECALEAD to thoroughly review which processes will be included in the assessment, and the criteria and indicators that are accessible and/or suitable based on the specific case being considered.

The assessment of a distance education system is likely to be organized by the staff from organizational and management levels at the institution, so it will be possible to consider more processes in detail, such as those related to policies and regulations, those related to economic and financial aspects, etc. It will probably be possible to consider all processes presented in this model. However, if those using ECALEAD are the teachers of a specific course, it will probably not be possible to consider the policies and regulations of the course itself, but rather those of the institution that manages or organizes it. In these cases, the information required to analyze such process may not be accessible, and therefore, only those aspects that are related to the design of the course itself will be considered. Thus, the application of the model will have to be adjusted based on the possibilities of each specific context.

The following table summarizes the strengths and weaknesses found in the model.

Table 1. Strengths and weaknesses of the model proposed

Strengths	Weaknesses
Flexibility to adapt the model based on the needs of the context and object to assess	Variety of indicators (qualitative and quantitative); this could make the definition of measurements to determine quality more difficult
Layered decisions, from higher level to lower level. This allows splitting the task, ordering it, and focusing on the specific aspects of each layer at a time.	The model does not define which measurements should be adopted in each case
The model takes into account aspects that are directly related to the educational environment, unlike other methods that have been taken from the business environment and are used for IES.	Only a set of indicators are presented for each criterion, but this set may not include some situations that are relevant for the object to assess, which must be defined by those carrying out the assessment. This may cause significant omissions during the assessment, if not correctly adjusted.
The model proposes a series of criteria and indicators that allow analyzing hybrid or fully distance educational methods, unlike other models that are applicable to on-site IES methods in general and are used for any type of method, thus disregarding some aspects that are typical of these methods.	In order to apply the entire model, a diversity of human resources is required. This includes everyone from those who make decisions to carry out the assessment to those who provide the information required to carry out the assessment in an appropriate and thorough manner. The quality of the instruments used to gather information will greatly affect the results of the assessment.
The application of the entire model allows taking into account a variety of issues that affect the quality of the educational system/process being assessed.	

The aspects mentioned in this section are key for the application of the model. They allow knowing which aspects are decided by the users, and which aspects are the elements that can affect the assessment.

5. Conclusions

In this paper, a quality assessment model for distance or hybrid educational systems/processes has been presented: ECALEAD. As a specific contribution, the strengths and weaknesses found so far in ECALEAD are detailed, so that those using this model can take this as a starting point and make appropriate and aware decisions when working with each layer. In the future, a new application of the model will be carried out to refine this analysis.

Acknowledgments

To Project FRIVIG, D/031964/10, financed by the AECID.

References

1. Fainholc, Beatriz. La calidad en la educación a distancia continúa siendo un tema muy complejo. RED Revista de Educación a Distancia, 12. http://www.um.es/ead/red/12/fainholc.pdf
2. Gonzalez Castañon, Miguel Angel. Evaluación de la Calidad en la Educación Superior a Distancia. Propuesta de la Universidad Estatal a Distancia de Costa Rica. Presentado en Congreso de Calidad y Acreditación Internacional en Educación a Distancia. AIESAD – CREAD – Virtual Educa – UTPL. 2005.
3. Sangrà M.A. Los retos de la educación a distancia. Boletín de la Red Estatal de Docencia Universitaria. Vol. 2 Nº 3.
4. Barberá, E. (coord.), Badía, A., Mominó, J. "La incógnita de la Educación a Distancia". ICE – Universidad de Barcelona – Horsori. Barcelona. 2001.
5. Davies, G., Doube, W., Lawrence-Fowler, W., Shaffer, D. Quality in Distance Education. 2001.
6. ICTs in Education. Disponible en: http://www.un-gaid.org/tabid/885/Default.aspx
7. García Aretio, L. (coord.), Corbella M., Dominguez Figaredo D. "De la Educación a Distancia a la Educación Virtual". Editorial Ariel. 1st edition. ISBN:978-84-344-2666-5. 2007.
8. Sanz C., Gorga G., Madoz C. Propuesta de un modelo de evaluación en capas. CACIC 2007. Argentina. ISBN: 978-950-656-109-3.
9. Sanz, C., Gorga, G., Madoz, C. El tema de la calidad en la educación a distancia. Propuesta de un modelo de evaluación en capas. CACIC 2010. Argentina. ISBN 978-950-9474-49-9.
10. Sanz, C. Gorga, G., Madoz, C., Propuesta para el análisis de la calidad en sistemas de Educación a Distancia. Aplicación a un caso de estudio. Tercer Congreso Virtual Iberoamericano de Calidad en Educación a Distancia, EduQ@2010.

Interactive multi-sensory environment to control stereotypy behaviours

CRISTINA MANRESA-YEE[1], RAMON MAS[1], GABRIEL MOYÀ[1],
MARIA JOSÉ ABÁSOLO[2], JAVIER GIACOMANTONE[2]

[1]Universitat de les Illes Balears. Unidad de gráficos, visión por
computador e inteligencia artificial.
Ed. Anselm Turmeda, Crta Valldemossa km 7.5, 07122 Palma, España
{cristina.manresa, ramon.mas, gabriel.moya}@uib.es

[2]Universidad Nacional de La Plata. Instituto de investigación en informática.
Facultad de Informática, 50 y 120 - 2do Piso – 1900 La Plata,
Pcia. de Buenos Aires, Argentina
{mjabasolo, jog}@lidi.info.unlp.edu.ar

Abstract. *The paper presents an interactive multi-sensory stimulation environment based on computer vision techniques for users with profound cerebral palsy to work their education curricula. We developed a set of vision-based applications with a high component of interactivity to create a controlled and safe environment to treat the users' behaviours. We analyze the user's body movement captured by a standard webcam to trigger audible, visual and/or tactile effects to produce significant stimulus in the environment.*

Keywords: *vision based interfaces, human-computer interaction, interactive multi-sensory environments*

1. Introduction

The environment is designed for users with profound cerebral palsy. Despite their limited cognitive and physical conditions, they show interest when there is a significant stimulus in the environment. Frequently, these users present self-stimulating behaviours, that is, "repetitive body movement which serves no apparent purpose in the external environment" [1], but that interferes with the daily routine. Furthermore, together with anxiety episodes, the self-stimulation can transform into self-injury. Self-injury is a destructive behaviour that implies social and personal consequences and risks the person's physical integrity like biting one's arm or banging one's head with the fist. These users also present other behavioural disorders such as screaming, banging the table and the floor or throwing objects.

Stereotypy, or self-stimulatory behaviour, can involve only one or all senses. Examples of these behaviour are: rocking, staring at lights, tapping ears, snapping fingers, making vocal sounds, rubbing the skin with one's hand or with other objects, scratching, placing body parts or objects in one's mouth, etc.

When these users receive a significant stimulus from the environment, the self-stimulating and self-injuring behaviours decrease [2, 3, 4, 5]. Users cannot access autonomously to interactive systems that could provide them with stimulus because of their capabilities. Caregivers in centres for disabled users cannot offer continuously significant stimuli to each individual. Therefore, we contribute with these interactive systems that give the control to the user in order to activate changes in the environment. Berkson and Mason [6] already related the increase of interactivity in an environment with a noticeable decrease in the self-stimulatory behaviours.

Furthermore, early stimulation is known to be a useful and necessary treatment aimed at developing as much as possible the social psychophysical potential of any person at high environmental and/or biological risk [7]. So, in the educational activities for users with this profile, we find all kind of tasks related to multisensory stimulation such as: coloured lights, tasting food, touching different textures, cold/hot sensations, etc.

This interactive environment offers caregivers and therapists alternative activities in their work for stimulation in an indirect attention mode, that is, when caregivers cannot pay total attention to one user as they have to supervise several users. Similar to other multi-sensory environments, our interactive system is a controlled and safe space with equipment designed to offer stimulation and calm. Moreover, we include interaction which can enrich the experience [8].

We present an interactive multi-sensory environment. The environment is divided in two modules. On the one hand, there is a computer with installed applications and on the other hand there is a multi-touch surface. The interaction is carried out by the movement of the user's body part. We detect and track the user's body part by means of a standard webcam and computer vision techniques. Due to the users' capabilities, they cannot access a computer; therefore, computer vision is our choice for interaction. Their movement causes a sensory stimulus change in the environment, as the computer offers auditory, visual and/or tactile stimuli adapted to each user.

The aim of the interactive multi-sensory environment is to improve the user's relationship with the environment in order to offer him or her significant stimuli and decrease the stereotypy behaviours.

The remainder of the paper is organized as follows. Section 2 describes the methodology of the design, development and evaluation process. Section 3 is the main contribution of the paper, as it explains the developed interactive multi-sensory environment. Finally, the last section concludes the paper.

2. Methodology

The methodology is a research-intervention model. The steps are:

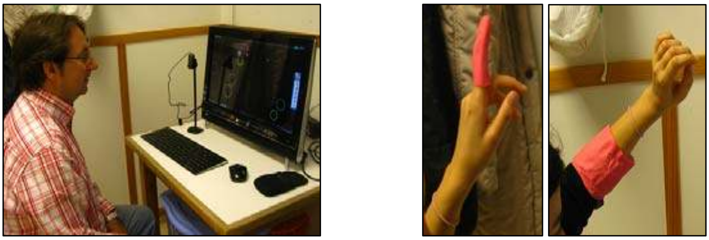
- User selection: we selected six users to analyze the system's requirements. Users were five men and a woman with ages ranging from 23-28 years. They are individuals with hearing, sight and other sensory impairments, with motor impairments and with memory, learning and cognitive impairments. However, they show interest (to a greater or lesser extent) when stimuli are produced in the environment. They are regularly engaged in self-stimulating and self-injurious behaviours. We obtained informed consent from the parents prior to their participation.
- Base line: users were recorded for two weeks in their daily routine in order to determine a base line for each of them. We registered the frequency, the duration and the type of self-stimulating and self-injuring behaviours as well as any disorderly conducts.
- Prototype design and development: considering users' capabilities and requirements, we designed and developed a prototype that will be described in the following section.
- Intervention: the system is already set up in a room and being tested. We will work with it during two months, 4 days per week, and sessions will be ten minutes long. Initially, sessions will be followed by a development member and a caregiver for helping the user, registering the user's feedback, the system's response and any other factor to take into account in the system's redesign or user's profile setting. Sessions are being recorded but just for the development group to analyze the system functioning.
- Evaluation: the users' psychologist work hypothesis is that users will decrease their self-stimulating and self-injuring behaviours. So, after working two months with the system, we will record users again for two weeks. Besides their daily work plan, the multi-sensory environment will be another activity included for indirect attention. In these two weeks, neither caregivers nor development members will help the user directly when working with the interactive multi-sensory system. Then the recordings will be analyzed and compared with the base line. We will consider the user's data when working with the interactive system and when working in other activities.

3. The interactive multi-sensory environment

In this section, we will describe the vision-based interactive multi-sensory system. As commented before, there are two modules: a set of applications and a multiuser multi-touch surface.

3.1 Interactive multi-sensory applications

In the first module, we work with a computer that counts with a webcam, loudspeakers and a radiofrequency (RF) remote plug. All applications are vision based interfaces and the caregiver can select to motivate or inhibit the body part movement depending on the user's aims.



(a) (b)

Fig. 1. (a) Hardware configuration. The user interacts with head movements, although s/he is not aware of the computer's presence (b) Different body parts to place the coloured band

As users move or stop a selected body part, the computer will offer feedback in form of auditory, visual and/or tactile effects which search to produce a change in the environment. In order to detect the movement, we can use different input systems.

On the one hand, an input is the head movement. By using the SINA interface [9,10], the user does not need to wear any sensor on the body. The system detects and tracks automatically the user's nose and this information is converted in mouse positions, which inform us on the head's movement to inhibit for example user's rocking front to back or moving the head from side-to-side. See Figure 1 (a).

On the other hand, we can detect and track any body part by means of a coloured band using the Camshift probabilistic algorithm [11]. The band is placed over the body part to motivate or to inhibit. In the case of motivation, the aim would be to increase the user's relationship with the environment. An example of inhibition would be to avoid the user snapping fingers. See Figure 1 (b).

The presence or absence of motion (depending on the therapist's aim) triggers or stops the system's output. This output is in the form of audible, visual and/or tactile effect. The outcome can be configured with the user's preferences in music, sounds and images. See Figure 2. This is very important to motivate the user. For example, we are using the parents' voices. Specifically, the computer's feedback is:

- Striking images shown on the screen
- Sounds, music or voices.
- We can switch on or off any plugged device that works in binary mode, that is, it is on or off. For example an electric mirror ball

- The system allows capturing the screen position of the play/pause button in any program; therefore, the user can start or stop any media (films or music files) by moving a body part. Viewing a presentation can be done as well.

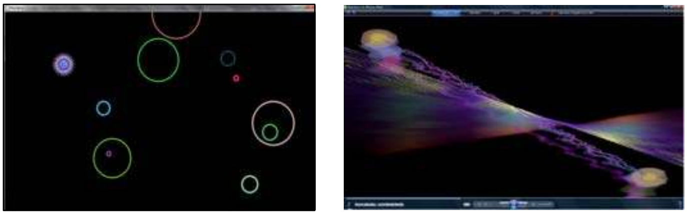


Fig. 2. Example of outputs: music is playing and circles with different radius and widths go appearing. Music played with Windows Media

3.2 Multiuser multi-touch surface

We have also designed and built an autonomous interactive multi-touch table that allows the installation of action/reaction applications that provide visual and/or auditory responses to the tactile stimuli. The table uses a DLP projector to output visual feedback on the surface and a webcam to capture the user-device interaction. We use frontal diffuse illumination (DI) from the environment, so no additional lighting is required. Stimuli can be generated according to the contact detected, so a smooth and long contact will trigger visually striking images while fast and disordered contacts, common in uncontrolled and abrupt movements, are ignored and won't produce any response.

Being multi-touch, the table allows either multiple effectors (fingers, hands, ...) or multiuser interactions. Although such devices are currently commercially available, two reasons justify the need for an adapted device. First, medium-size multi-touch devices still require a high-cost investment and second, as they are based on traditional output devices such LCD displays; they are too fragile to be used in environments where users have little control on the motion and pressure of their gestures. This could lead to dangerous situations and compromise the security of the users.

We use optical technologies because of its low cost, setup and scalability. Optical technologies require a source of light, an optical sensor and an output device. The output device is built from a projector. It displays a feedback image on the screen through a mirror. The screen is a one-centimetre-thick acrylic panel that ensures resistance to break and, thus, increases user security. A diffuser distributes light homogeneously. We have to avoid the interference of the image produced by the projector with the objects being tracked. This is accomplished using infrared light to discern the visual image displayed by the projector on the touch surface and the fingers or hands being tracked. The webcam is modified to capture only infrared light. We replace the built-in infrared filter of the webcam with a filter that removes the visible light of the electromagnetic spectrum. See Figure 3 and 4.

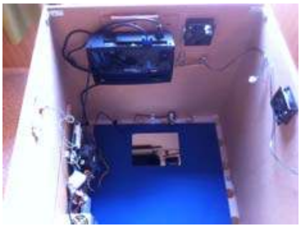


Fig. 3. Interior of the multi-touch table

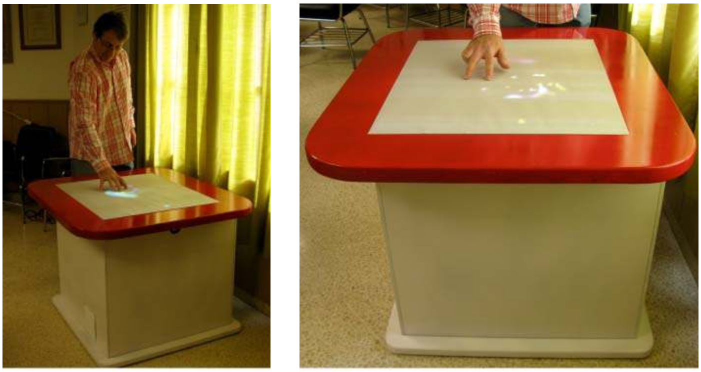


Fig. 4. Multiuser multi-touch table

4. Conclusions

In this paper we have presented an interactive multi-sensory stimulation environment based on computer vision techniques for users with profound cerebral palsy. It is composed of a multimedia environment and a multi-touch table. The environment offers caregivers and therapists alternative activities in their work for indirect attention, allowing for controlled autonomous stimulation.

The system is currently being tested and some feedback has already been received from therapists. Accessibility problems to the multi-touch table have been reported for wheelchairs and a redesign in on the way. The main question to answer is whether the users realize that their motion or steadiness cause a change in their environment. In some cases, it seems that the user connects with the environment, in others this feedback has still not appeared. We don't know if in these lasts cases the experience will teach them. It is very difficult to work with profound CP users as they cannot express themselves.

Nowadays we are carefully observing their responses: if they smile, if they stop their stereotypy behaviours, if they pay attention to the stimulus, if they make sounds, if they change their body posture or any other symptom of being engaged with the environment.

In the near future, we will be able to compare the evaluation recorded data with the base line to analyze the evolution of the users.

Acknowledgments

This research was partially supported by:

- FRIVIG. D/031964/10. Formación de Recursos Humanos e Investigación en el Área de Visión por Computador e Informática Gráfica, granted by MAE-AECID (Programa de Cooperación Interuniversitaria e Investigación Científica e/ España e Iberoamérica.
- TIN2010-16576. Modelos de Interacción basada en Visión en Interfaces Gestuales, granted by Ministerio de Ciencia e Innovación, Gobierno de España (Programa Nacional de Investigación Fundamental No Orientada).

References

1. Harris, S. L., and Wolchik, S. A. Suppression of selfstimulation: Three alternative strategies. Journal of Applied Behavior Analysis, 12, 185-198 (1979).
2. Bright, T., Bittick, K., & Fleeman, B. Reduction of selfinjurious behavior using sensory integrative techniques. American Journal of Occupational Therapy, 35, 167-172 (1981).
3. Reisman, J. Using a sensory integrative approach to treat self-injurious behavior in an adult with profound mental retardation. American J. Occupational Therapy, 47, 403-411 (1993).
4. Smith, S. A., Press, B., Koenig, K. P., and Kinnealey, M. Effects of sensory integration intervention on self-stimulating and self-injurious behaviours. American J. Occupational Therapy, 59, 418-425 (2005).
5. Singh, N., Lacioni, G.E., Winton, A. S. W., Molina, E.J., Sage, M., Brown, S., and Groeneweg, J. Effects of Snoezelen room, Activities of Daily Living skills training, and Vocational skills training on aggression and self-injury by adults with mental retardation and mental illness. Research in Developmental Disabilities 25, 3 , 285-293 (2004).
6. Berkson, G., Mason, W.. Sterotyped movements of mental defectives III. Situation effects. American Journal of mental Deficiency, 66, pp. 849-852 (1962).
7. García-Navarro, M.E., Tacoronte, M., Sarduy, I., Abdo, A., Galvizú, R., Torres, A., Leal, E. Influence of early stimulation in cerebral palsy. Rev Neurol. 16-31;31(8):716-9 (2000).

8. Fonoll Salvador, J., Lópe Álvarez, S. Recursos digitales para el aula multisensorial. En Arnaiz, P.; Hurtado, Mª.D. y Soto, F.J. (coords.) 25 años de integración escolar en España: Tecnología e Inclusión en el ámbito educativo, laboral y comunitario, pp. 1-7 (2010).
9. Varona, J., Manresa-Yee, C. Perales, F.J Hands-free Vision-based Interface for Computer Accessibility. Journal of Network and Computer Applications Volumen 31 , No 4 pp. 357-374, (2008).
10. Manresa-Yee, C., Ponsa, P., Varona, J., F.J. Perales User experience to improve the usability of a vision-based interface. Interacting with Computers Volume 22, Issue 6, pp. 594-605 (2010).
11. Bradski, G. R. Computer Vision Face Tracking For Use in a Perceptual User Interface. Intel Technology Journal, No. Q2. (1998).

IX

Graphic Computation, Imagery and Visualization Workshop

Unrestricted multivariate medians for adaptive filtering of color images

EZEQUIEL LÓPEZ-RUBIO[1], MARÍA NIEVES FLORENTÍN-NÚÑEZ[2]

[1]Department of Computer Languages and Computer Science,
University of Málaga, Spain
ezeqlr@lcc.uma.es
[2]Department of Investigations and Extensions,
National University of Itapúa, Paraguay
florentin@uni.edu.py

Abstract. *Reduction of impulse noise in color images is a fundamental task in the image processing field. A number of approaches have been proposed to solve this problem in literature, and many of them rely on some multivariate median computed on a relevant image window. However, little attention has been paid to the comparative assessment of the distinct medians that can be used for this purpose. In this paper we carry out such a study, and its conclusions lead us to design a new image denoising procedure. Quantitative and qualitative results are shown, which demonstrate the advantages of our method in terms of noise reduction, detail preservation and stability with respect to a selection of well-known proposals.*

Keywords: *Adaptive filtering, multivariate median, impulse noise, edge detection.*

1. Introduction

Filtering of color images is aimed at reducing noise while at the same time chromaticity, edges and details are preserved [1]. This can be done by either component-wise or vector methods [2]. The main difference between these families of methods is that component-wise methods can introduce new color artifacts in the resulting image, because they process each pixel color channel independently without considering the correlation between channels. On the other hand, the vector methods process the color channels of each pixel as a vector, thus avoiding inconvenient chromaticity changes in the resulting image [2], [3]. For this reason, the vector methods are more effective for noise reduction and preservation of color image chromaticity. Among classical non-linear vector based filters, we have the Vector Median Filter (VMF) [4]; the Basic Vector Directional Filter (BVDF) [5]; and the Directional Distance Filter (DDF) [6]. These filters are uniformly applied on the image; therefore they tend to modify both noisy pixels and edge pixels. Consequently, effective noise removal is achieved at the expense of blurred and distorted features. In order to better preserve the image structure, the vector median and directional weighted adaptive filters have been proposed

[7], [8]. The vector filters based on an adaptive switching scheme [3], [9], [10], [11] consider an impulse detector to determine which pixels should be filtered and which pixels should be preserved. These filters are simple and effective in preserving image details.

We propose a new filter for noise reduction in color image, based on the filter unrestricted multivariate medians, in an adaptive switching scheme. Median based filters such as the VMF are commonplace in literature; there has been little interest in studying the comparative advantages of the different multivariate medians that could be used for image denoising. Here we do such a study, and its result leads us to propose a new denoising filter for color images corrupted by impulse noise. The results of experiments and simulations have shown that the proponed filter is better than many other existing adaptive filters, in terms of capacity for noise reduction, preservation of edges, thin detail and image color.

The paper is organized as follows. Section 2 describes the different multivariate medians that can be used for color image denoising, and then carries out a study of their relative strengths. In light of the results of this study, we propose in Section 3 a new adaptive filter. Experimental results and comparisons between the proposed filter and several well-known nonlinear adaptive filters are presented in Section 4. Finally, the conclusions are set out in Section 5.

2. Median Filters

In this section we examine the use of median filters for impulse noise removal. Impulse noise is classified into two types [11], [3]:

1. Fixed-valued impulse noise, also known as salt and pepper noise, pollutes pixels with random values that can be either 0 or 255.
2. Uniform impulse noise, pollutes the pixels following a uniform random distribution over the full range [0, 255]. We have considered this filter model for the experiments.

Under impulse noise, the pixels of the *k-th* channel of a color image (*k=1,2,3*) are distorted according to the following equation:

$$y_k(x_1,x_2)=\begin{cases}\hat{y}_k(x_1,x_2), & \textit{with probability } 1-\rho\\ n_k(x_1,x_2), & \textit{with probability } \rho\end{cases} \quad \textbf{(1)}$$

where $\hat{y}_k(x_1,x_2)$ and $y_k(x_1,x_2)$ are the pixel values of channel *k* at position (x_1,x_2) of the original image and noisy image, respectively; $n_k(x_1,x_2)$ is an impulse noise value which is independently chosen in the three channels; and ρ denotes the noise ratio. In order to simplify the discussion, for the rest of this section we will assume that $n_k(x_1, x_2)$ comes from a uniform distribution (uniform impulse noise), as this is the impulse noise type which is the most difficult to remove.

Perhaps the most basic procedure for noise removal is low pass filtering. This amounts to replacing the old value by a weighted mean of the pixel color values y_j in a small window *W* around the pixel of interest. If we assume equal weights for the sake of simplicity we arrive at the sample mean:

$$smn(W) = \frac{1}{|W|} \sum_{\mathbf{y}_j \in W} \mathbf{y}_j = \arg\min_{\mathbf{z}} \sum_{\mathbf{y}_j \in W} \left\| \mathbf{z} - \mathbf{y}_j \right\|^2 \qquad \textbf{(2)}$$

where $|W|$ stands for the number of pixels of the windows *W.* However, the mean is known to be very sensitive to outliers [12]. As seen in (2), this is because the mean minimizes the sum of squared distances. Consequently, this approximation can be dominated by outliers. In this way, when the noise is made of impulses, the corrupt data dominate the computation of the weighted average, leading to poor results.

A classical approach to solve this problem is to use the component-wise median:

$$cmedian(W) = \left(\arg\min_{z_k} \sum_{\mathbf{y}_j \in W} \left\| z_k - y_{jk} \right\| \right)_{k \in \{1,2,3\}} \qquad \textbf{(3)}$$

The decision to use *cmedian* is very common when applying a denoising method designed for grayscale data to color images, since it is equivalent to filtering the three color channels separately by means of the univariate median. Despite being a robust statistic, *cmedian* does not make use of the tridimensional structure of the color space, which leads to chromatic shifting problems [3]. Moreover, it is not invariant to similarity transformations.

Robust statistics have been developed, which are fully adapted to the characteristics of multivariate data [13]. In particular, the multivariate median (also called *L_1-median*) has an efficient learning algorithm [14] and has been proven to experience little degradation when outliers are present [15], [16]. It has a key advantage over *cmedian*, namely its invariance with respect to all similarity transformations. The rationale behind the *Lmedian* statistic is to drop the square in the minimization (2) to arrive at the L1-median of the set (we use equal weights like before to simplify matters):

$$Lmedian(W) = \arg\min_{\mathbf{z}} \sum_{\mathbf{y}_j \in W} \left\| \mathbf{z} - \mathbf{y}_j \right\| \qquad \textbf{(4)}$$

so that the outliers have a much lesser impact on the minimization.

A typical simplification of *Lmedian* for color image denoising is to restrict the possible outcomes to be among the input data [10], [3], [4]. In this way we arrive to the restricted L1-median:

$$rmedian(W) = \arg\min_{\mathbf{y}_i \in W} \sum_{\mathbf{y}_j \in W} \left\| \mathbf{y}_i - \mathbf{y}_j \right\| \tag{5}$$

The four discussed strategies (*smn, cmedian, Lmedian* and *rmedian*) are estimators of the true value $\hat{\mathbf{y}}$ of the central pixel, where the input samples for the estimation come from the window *W*. Perhaps the most standard way to compare the performance of two estimators $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{y}}'$ is to compute their relative efficiency (*REF*); see [17], [18]. For a tridimensional estimated parameter $\hat{y}$, which is our case, it reads as follows [19]:

$$REF(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}') = \left(\frac{\det(\operatorname{cov}(\tilde{\mathbf{y}}'))}{\det(\operatorname{cov}(\tilde{\mathbf{y}}))} \right)^{\frac{1}{3}} \tag{6}$$

where $\operatorname{cov}(\tilde{\mathbf{y}})$ and $\operatorname{cov}(\tilde{\mathbf{y}}')$ are the covariance matrices of the estimators $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{y}}'$ respectively. The estimator $\tilde{\mathbf{y}}$ is judged to be better than $\tilde{\mathbf{y}}'$ if and only if $REF(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}') > 1$, while $\tilde{\mathbf{y}}'$ is better than if and only if $REF(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}') < 1$. Since we have four estimators to compare, and not only two, it is more convenient to compute the following *estimator efficiency*:

$$EFF(\tilde{\mathbf{y}}) = \left(\frac{1}{\det(\operatorname{cov}(\tilde{\mathbf{y}}))} \right)^{\frac{1}{3}} \tag{7}$$

Higher values of $EFF(\tilde{\mathbf{y}})$ mean that the estimator $\tilde{\mathbf{y}}$ is better, since from (6) and (7) we have:

$$REF(\tilde{\mathbf{y}}, \tilde{\mathbf{y}}') > 1 \Leftrightarrow EFF(\tilde{\mathbf{y}}) > EFF(\tilde{\mathbf{y}}') \tag{8}$$

We have tested the four considered approaches without any impulse detection method, in order to compare their intrinsic performance against uniform noise (Fig. 1); the well known Baboon image has been used for this purpose [20]. The four most used window sizes have been considered, i.e. 3 ×3, 5×5, 7×7 and 9×9. The impulse noise probability ρ has been varied from 0 to 1 in 0.01 increments.

As seen, the best performing filter is Lmedian for the most commonly considered noise levels (ρ<0.5), and all window sizes and noise types.

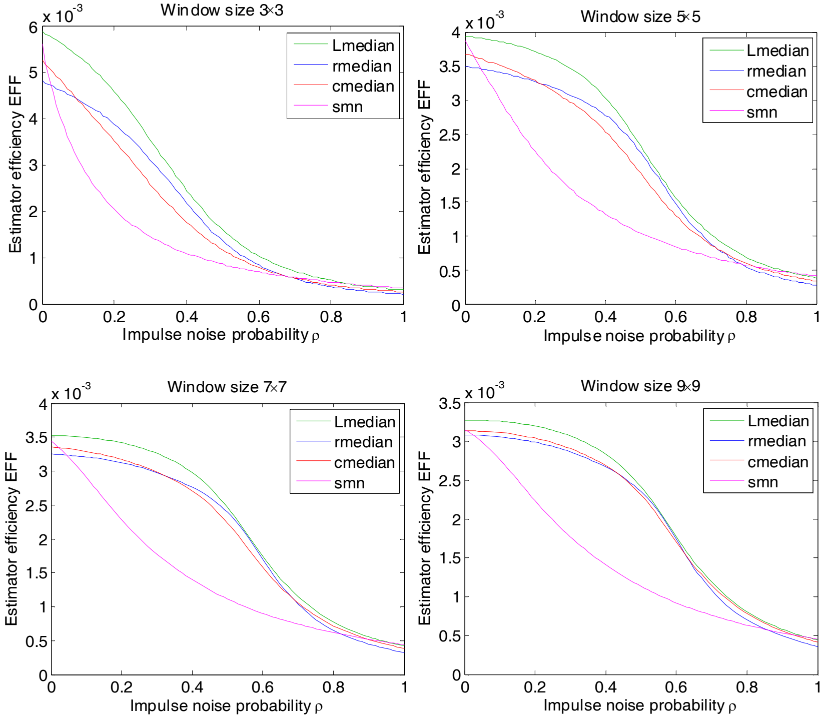


Fig. 1. Efficiency of the estimators under uniform impulsive noise for the Baboon image

The best window size is the smallest (3×3) for low noise (ρ<0.2). This is because the increased smoothing of the larger sizes does not offer any advantage at so low noise levels. However, when the noise is higher, larger sizes are better. The 5×5 size is the best performing for moderate noise levels, so it is a reasonable tradeoff, as seen before. From the preceding, it can be concluded that we can get some advantages in impulse noise removal by considering the *Lmedian* strategy, in particular with a 5×5 window size. Next we design a nonlinear image filtering scheme which is based on this observation.

3. Adaptive Filtering

Before we can apply the unrestricted multivariate median Lmedian to image denoising, we need a procedure to detect impulse corrupted pixels reliably. Let $\mathbf{x} = (x_1, x_2)$ be the position of a pixel in an image of size $NumRows \times NumCols$, and let be $\mathbf{y} : [1, NumRows] \times [1, NumCols] \to \mathbf{R}^3$, where

$$\mathbf{y}(\mathbf{x}) = (y_1(\mathbf{x}), y_2(\mathbf{x}), y_3(\mathbf{x})) \quad \mathbf{(9)}$$

is the function which gives the three-dimensional color value at position **x**. If the pixel **x** were not an impulse, it would imply that **y** is differentiable at **x**. Hence we could express each of its components as a Taylor series,

$$y_k(x_1 + \varepsilon, x_2 + \delta) = y_k(x_1, x_2) + \varepsilon \frac{\partial y_k}{\partial x_1}(x_1, x_2) + \delta \frac{\partial y_k}{\partial x_2}(x_1, x_2) + \ldots \quad \mathbf{(10)}$$

Then we get from (10) that the directional derivative in the direction (ε, δ) is zero to first order approximation:

$$\frac{y_k(x_1 + \varepsilon, x_2 + \delta) - y_k(x_1, x_2)}{\varepsilon^2 + \delta^2} \approx 0 \quad \mathbf{(11)}$$

That is, there should be a small constant $\lambda > 0$ such that

$$\frac{|y_k(x_1 + \varepsilon, x_2 + \delta) - y_k(x_1, x_2)|}{\varepsilon^2 + \delta^2} < \lambda \quad \mathbf{(12)}$$

Please note that the vector (ε, δ) points in the direction of the level curve that crosses the point (x_1, x_2). One could check whether (x_1, x_2) is not an impulse by looking for a vector (ε, δ) which fulfils (12). We restrict our search to those which correspond with easily realizable gradient estimators (edge detection filters):

$$(\varepsilon, \delta) \in \{(0,1), (1,0), (1,1), (-1,1)\} \quad \mathbf{(13)}$$

The corresponding masks for the 5×5 window size are as follows:

$$\mathbf{M}_{(0,1)} = \begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix} \mathbf{M}_{(1,1)} = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{(14)}$$

$$\mathbf{M}_{(1,0)} = (\mathbf{M}_{(0,1)})^T \ \mathbf{M}_{(-1,1)} = (\mathbf{M}_{(1,1)})^T \quad \mathbf{(15)}$$

Please note that equation (12) should hold for all three color components if the pixel were not an impulse. Consequently, we declare that pixel (x_1, x_2) is an impulse if and only if there exists a color component such that no vector (ε,δ) which satisfies (13) can be found to verify condition (12).

If the pixel at position x is not an impulse, it is not changed in the restored output $\tilde{\mathbf{y}}(\mathbf{x})$ Otherwise, we substitute it by the unrestricted multivariate median of its 5×5 window W_x, as justified in Section 2:

$$\tilde{\mathbf{y}}(\mathbf{x}) = Lmedian(W_{\mathbf{x}}) \qquad \mathbf{(16)}$$

Now that we have defined our proposal, we are ready to assess its performance, which is done in the following section.

4. Experimental Results

In this section we compare the performance of the proposal we have just presented with that of several well known impulse noise removal filters. We have considered various benchmark images of 512 × 512 pixels, 24-bit RGB. These images were obtained from the Southern California University images database [20]. We have obtained quantitative and qualitative results similar to the test images. We only present the experimental results obtained with Baboon (Fig. 2). We considered a filter window 5×5 to experiment with our filter, as previously explained. We chose a threshold value $\lambda = 5$, which proved to yield robust results across the tested benchmark images. The set of alternative multi-channel filters which has been considered for the comparative evaluations is shown in Table 1.

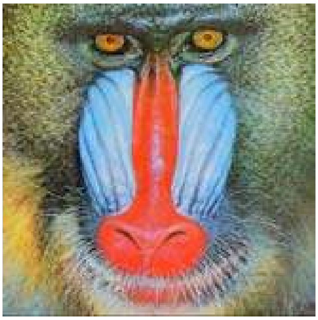


Fig. 2. Original Image (Baboon)

Table 1. Filters considered for comparison with the proposed filter. Please note that the first four filters operate using a filter window 3 × 3 and the rest of the filters operate using a filter window 5 × 5.

Notation	Filter	Parameter	Ref.
VMF	Vector Median Filter		[4]
BVDF	Basic Vector Directional Filter		[5]
DDF	Directional Distance Filter	$p = 0.25$	[6]
HVF	Hybrid Vector Filter	$Tol \geq 10$	[11]
SVMF	Switching Vector Median Filter	$Tol \geq 3$	[10]
RASVMF	Rank Adaptive Sigma Vector Median Filter	$\lambda = 5$	[9]

4.1. Quantitative results

Here we compare the methods in terms of quantitative noise reduction, faithful color reproduction, detail preservation and stability. To this end, we have selected three performance measures for the filter evaluations: Peak signal-to-noise ratio (PSNR), higher is better; mean absolute error (MAE), lower is better; and normalized color difference (NCD), lower is better. PSNR reflects noise suppression level [21]. MAE reflects the capability to preserve image details. NCD reflects the capability to preserve the image chromaticity [22]. The stability of the methods has been assessed by computing the mean value and standard deviation of these three measures over 10 simulation runs with different pseudorandom seeds for random noise generation.

In order to evaluate the stability of the filters' performance, we added impulsive noise to the test images; we processed ten times the test images with each filter, for every impulsive noise ratio considered in this paper.

Table 2 present the results obtained with the mean and standard deviation for each performance measure (PSNR, MAE and NCD), with different impulsive noise ratios.

Our method shows the best performance in chromaticity preservation (NCD) in all cases, while it also preserves details satisfactorily (MAE). This validates the theoretical results of Section 2, where we aimed to preserve the information which is associated to the three dimensional structure of the color data. On the other hand, it also attains good PSNR results, although RASVMF outperforms it. As we will see in the next subsection, RASVMF yields rather poor qualitative results in spite of having a high PSNR. This is because of its problems with color faithfulness, which we can be seen on table 2: RASVMF is the worst method with respect to NCD in several situations. This table shows that the performance of the proposed filter is more stable than that of most other filters.

Table 2. Quantitative results (standard deviations in parentheses) on Baboon image corrupted by uniform impulsive noise.

	10%			20%		
Filter	PSNR	MAE	NCD	PSNR	MAE	NCD
VMF	30.(0.01)	7.2 (0.02)	0.10 (0.00)	29 (0.01)	8.5 (0.01)	0.11 (0.00)
BVDF	29.(0.01)	9.9 (0.03)	0.12 (0.00)	29 (0.01)	11.1 (0.05)	0.13 (0.00)
DDF	31.(0.01)	6.1 (0.01)	0.09 (0.00)	30 (0.01)	6.8 (0.01)	0.09 (0.00)
SVMF	33.(0.02)	4.6 (0.02)	0.07 (0.00)	30 (0.01)	7.4 (0.02)	0.11 (0.00)
HVF	31.(0.01)	5.6 (0.02)	0.08 (0.00)	30 (0.01)	7.2 (0.02)	0.10 (0.00)
RASVMF	**38.(0.02)**	**2.0 (0.01)**	0.07 (0.00)	**35 (0.02)**	**4.7 (0.19)**	0.10 (0.02)
Proposed	34.(0.02)	3.1 (0.01)	**0.05 (0.00)**	32 (0.01)	4.9 (0.01)	**0.07 (0.00)**

4.2. Qualitative results

The proposed filter is qualitatively more adequate than the other filters, in the three above mentioned criteria. For example, in Fig. 3, it can be observed that the thin white hairs of the Baboon face, are very well preserved, and noise, successfully reduced.

It must be highlighted that bad color reproduction leads to visually deficient denoised images, even if the pixel values are numerically accurate. This can be seen in RASVMF results, where a very good PSNR does not lead to perceptually pleasant output images. The reason for this is that a relatively small number of very badly colored pixels can spoil the restored image as seen by a human.

5. Conclusions

The usage of multivariate medians for removal of impulse noise in color images has been examined. The different medians suitable for this purpose have been defined and compared. The insights provided by this comparison have leaded us to propose a new method to solve the impulse noise reduction problem in color images. Its comparative performance has been assessed with respect to several alternative proposals, both in quantitative and qualitative terms. The results of these experiments show that our method is able to reduce the impulse noise significantly and reliably, while at the same time it preserves the details and edges of the original image.

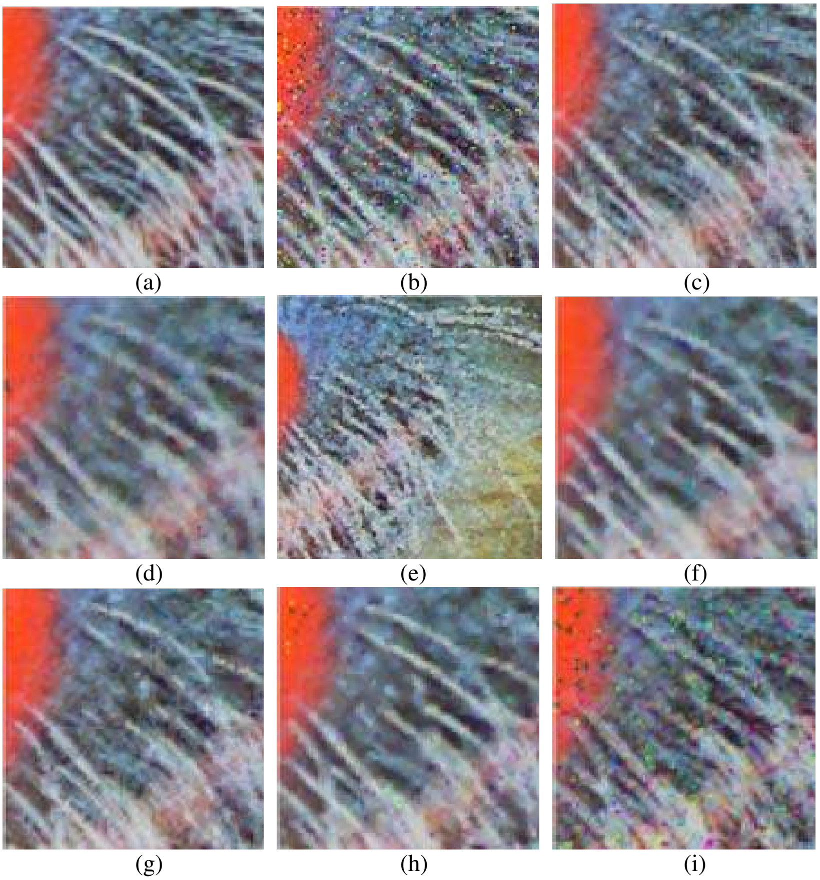


Fig. 3. Detail of the Baboon image (uniform noise). (a) Original image, (b) Image corrupted by 10% uniform impulsive noise, (c) Proposed output, (d) VMF output, (e) BVDF output, (f) DDF output, (g) SVMF output, (h) HVF output, (i) RASVMF output

References

1. Tsai, H.H., Yu, P.T. Adaptive fuzzy hybrid multichannel filters for removal of impulsive noise from color images. Signal Processing, 74 (2), 127-151 (1999).
2. Plataniotis, K.N., Venetsanopoulos, A.N. Color image processing and applications, Springer, Berlin (2000).
3. Ma, Z., Feng, D., Wu, H.R. A neighborhood evaluated adaptive vector filter for suppression of impulse noise in color images. Real-Time Imaging, 11 (5-6), 403–416 (2005).
4. Astola, J., Haavisto, P., Neuvo, Y. Vector median filters. Proceedings of the IEEE, 78 (4), 678-689 (1990).

5. Trahanias, P.E., Karakos, D.G., Venetsanopoulos, A.N. Directional processing of color images: theory and experimental results. IEEE Transactions on Image Processing, 5 (6), 868-880 (1996).
6. Karakos, D.G., Trahanias, P.E. Generalized multichannel image-filtering structures. IEEE. Transactions on. Image Processing, 6 (7), 1038-1045 (1997).
7. Viero, T., Oistamo, K., Neuvo, Y. Three-dimensional median-related filters for color image sequence filtering. IEEE Transactions on Circuits and Systems for Video Technology, 4 (2), 129-142 (1994).
8. Lukac, R., Plataniotis, K.N., Smolka, B., Venetsanopoulos, A.N. Generalized selection weighted vector filters. EURASIP Journal on Applied Signal Processing, 12 (15), 1870-1885 (2004).
9. Lukac, R., Smolka, B., Plataniotis, K.N., Venetsanopouls, A.N. Vector sigma filters for noise detection and removal in color image. Journal of Visual Communication and Image Representation, 17 (1), 1-26 (2006).
10. Jin, L., Li, D. A switching vector median filter based on the CIELAB color space for color image restoration. Signal Processing, 87, 1345–1354 (2007).
11. Dang, D., Luo, W. Color image noise removal algorithm utilizing hybrid vector filtering. AEU –International Journal of Electronic and Communications, 62 (1), 63-67 (2008).
12. Dang, X., Serfling, R., Zhou, W. Influence functions of some depth functions and application to depth-weighted L-statistics. Journal of Nonparametric Statistics, 21 (1), 49-66 (2009).
13. Gervini, D. Robust functional estimation using the median and spherical principal components. Biometrika, 95 (3), 587-600 (2008).
14. Hössjer, O., Croux, C. Generalizing univariate signed rank statistics for testing and estimating a multivariate location parameter. Non-parametric Statistics, 4, 293-308 (1995).
15. Hobza, T., Pardo, L., Vajda, I. Robust median estimator in logistic regression. Journal of Statistical Planning and Inference, 138 (12), 3822-3840 (2008).
16. López-Rubio, E. Robust location and spread measures for nonparametric probability density function estimation. International Journal of Neural Systems, 19 (5), 345-357 (2009).
17. Johnson, B.A., Abramovich, Y.I. DOA estimator performance assessment in the pre-asymptotic domain using the likelihood principle. Signal Processing, 90 (5), 1392-1401 (2010).
18. Xu, W., Hung, Y.S., Niranjan, M., Shen, M. Asymptotic mean and variance of Gini correlation for bivariate normal samples. IEEE Transactions on Signal Processing, 58 (2), 522-534 (2010).
19. Nevalainen, J., Larocque, D., Oja, H. A weighted spatial median for clustered data. Statistical Methods and Applications, 15, 355–379 (2007).
20. SIPI-USC, University of Southern California Image Database. In Internet: http://sipi.usc.edu/database/ (2010).
21. Wei, Z., Cao, Y., Newton, R.A. Digital image restoration by exposure-splitting and registration. Proceedings of the 17th International Conference on Pattern Recognition, 4, 657-660 (2004).
22. Xu, Q., Zhang, R., Sbert, M. A new approach to impulse noise removal for color image. IEEE International Conference on Multimedia and Expo, 1667-1670 (2007).

Error-Bounded Terrain Rendering Approach based on Geometry Clipmaps

LUCAS ENRIQUE GUAYCOCHEA, HORACIO ANTONIO ABBATE

Facultad de Ingeniería, Universidad de Buenos Aires
{lguaycochea,habbate}@fi.uba.ar

Abstract. *This paper introduces a terrain rendering technique based on Geometry Clipmaps. This technique includes screen-space error analysis from application's view parameters in order to provide error-bounded visualization. This is accomplished by dividing each nested patch into tiles to analyze the projected error into screen, as tiled-block terrain rendering techniques do. Finally, the implementation takes advantage of modern GPU processing power using DirectX 10 graphics library. The results obtained allow real-time navigation over large terrain extensions, consuming little CPU processing time.*

1. Introduction

Applications, in which real-time rendering of virtual 3D environments is needed, always demand more realistic experience for its users. Among this sort of applications, we will consider those where the visualization of large terrain extensions is involved. From flight simulators to GIS applications, passing through any kind of outdoor game, techniques to process elevation data and generate real-time terrain rendering are needed to fulfill the requirements presented in those applications.

Terrains are modeled using a mesh of points to represent their surface. A naïve approach is to send the whole mesh to the graphics pipeline in order to render the terrain. Despite graphics pipelines' throughput has exponentially increased over recent years thanks to modern GPUs, the brute-force approach has strong limitations in the size of the terrains supported in order to achieve real-time rendering. Consequently, several techniques have been developed to support larger terrains where the mesh is assembled using regions with different levels of detail (LOD) in order to reduce the geometry introduced into the pipeline in each frame.

Approaches using LOD are possible since the fact that the perception of details of an object (e.g. a particular region of a terrain) decreases as the distance from the viewer to the object increases. The size of the projection of these details into the final image on screen, as a consequence of the view-perspective projection transformation, may be less than the pixel resolution of the output display. The election of the appropriate level-of-detail for each region in the terrain is taken from different motivations, as can be seen along the bibliography on the theme.

This work is based on a technique called Geometry Clipmaps. It was introduced by Lossaso & Hoppe [1] and then extended to a GPU-based implementation by Asirvatham & Hoppe [2]. The authors' proposal is to represent the terrain using a set of nested regular grids of different LOD centered about the viewer. The nested grids have successive power-of-two resolutions and are translated as the viewer moves. The translation of the grids involves an incremental update of the elevation data of each nested grid.

Geometry Clipmaps decides the LOD over the terrain using only the 2D distance to the viewer. This strategy allows the technique to be independent from terrain local roughness and, therefore, to maintain the CPU work to the minimum and to guarantee constant throughputs (frame rate). Nevertheless, this approach has some limitations, as it is mentioned by its authors, to prevent the perception of changes in the surface of particular rough terrains as the viewer moves.

The technique proposed in this work is targeted to resolve the terrain rendering problem for applications where an immersive virtual reality on a well-known real-world environment must be provided to the user, such as flight simulators. In other words, the users must have an accurate real-world terrain perception without noticing any artifacts. In order to achieve this requirement, error-less or, at least, error-bounded surface terrain representation must be guaranteed by the solution.

The approach presented adds, to the state-of-the-art geometry clipmaps technique, the ability to analyze the error incurred in the use of a particular LOD in a region. The error is calculated projecting into screen-space the world-space error between the full-resolution region and the same region represented with lower detail. This projected error is calculated from the distance to the viewer and, also, from view parameters, such as the size of the window and the field-of-view of the camera. Then, the resulting screen-space error is compared with a pixel threshold defined by the application. This is done dividing each nested grid into tiles and then deciding if any of them needs to be refined to guarantee the error-bounded terrain representation. This strategy is taken from well-known tiled-blocks terrain rendering techniques [3, 4, 5, 6].

The error analysis discussed before, compared with the pure Geometry Clipmaps technique, involves more CPU process to compute the projected error and, if refinement of tiles is necessary, some CPU to GPU transfer of elevation data and more geometry to render. Even though, this obviously means a lower throughput rate from our technique, in modern hardware it performs with more than acceptable rates (see Section 3).

2. The terrain rendering technique

2.1 Terrain Representation

As it was introduced, the solution presented is based on the Geometry Clipmaps technique [1]. Then, the terrain is represented using a set of nested grids, that we call ***patches***, around the viewer's position. Each patch represents a region of the terrain with different resolution or level of detail. The most detailed level is zero (L = 0), where the spacing between the points of elevation data is at the highest resolution. Each following level covers four times more surface than the previous one, which means that the former resolution doubles the latter. So, a patch resolution (g_L) is 2^L times the finest dataset surface resolution, for levels L = 0, 1, 2….

Moreover, all patches have the same amount of samples or vertices, *n x n.* In contrast with Lossaso and Hoppe [1], we use $n = 2^k + 1$ (where *k* = 1, 2, 3…), that is needed to divide each patch in tiles. The election of this value for *n* makes us handle nine cases of relative positions between successive patches. A patch center position must lie in a vertex that belongs to next coarser-level patch. In this way, patches edges can share vertices available in both, so as not to present discontinuities in the terrain.

In order to optimize the performance, we add to patches' render size *n*, some extra elevation data that is loaded to use as a border cache. We choose to have a power-of-two border size in each direction (top, bottom, left and right). This can be also useful if the elevation data is obtained from compressed resources that apply block-compression schemes.

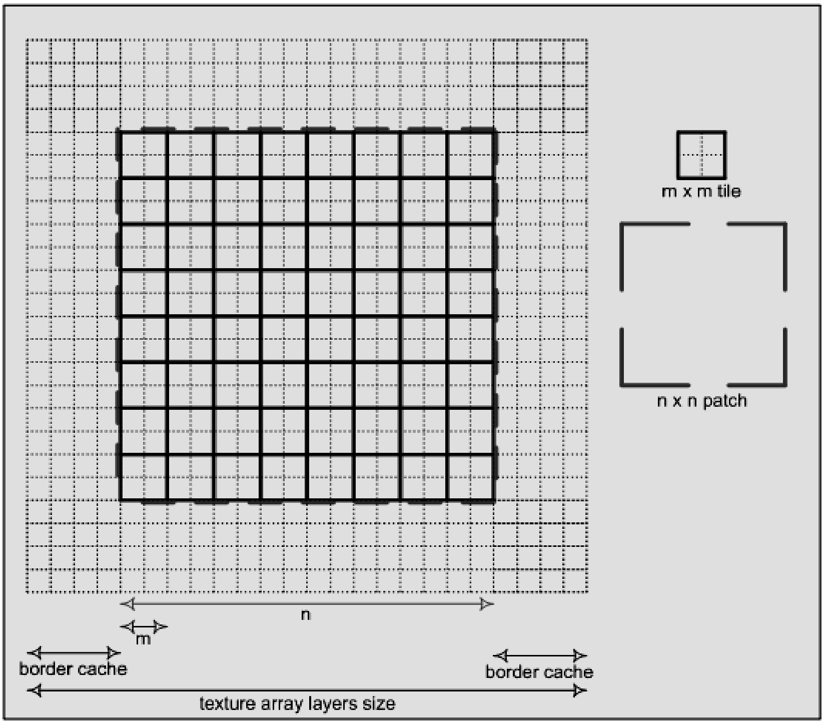


Fig. 1. Sizes of the different grids used for patches, tiles and layers of the texture array. Example: m = 3, n = 17

Elevation data will be loaded to the GPU into texture arrays (available in GPUs that support Shader Model 4.0 [7] or later). Each patch has its elevation data loaded in a different layer of the texture array. Then, using vertex buffers describing 2D "footprints" [2], the vertex' z-value is sampled in the vertex shader from the corresponding layer in the texture array.

Recalling a key-point from geometry clipmaps [1, 2], the elevation data is incrementally updated using a toroidal access with 2D wraparound addressing. With this strategy, GPU-CPU bandwidth utilization is optimized as only new regions of elevation data is updated as the viewer moves. Moreover, the use of a border cache avoids frame-to-frame updates of few, or singles, rows or columns to the texture layers.

Up to this point, the base of the surface representation approach has been described, but it lacks of view-dependent and error-bounded screen-space error properties. In order to add these properties, we decided to divide each terrain patch into tiles. These tiles allow the error analysis using view-dependent based metrics (see section 2.2).

Square tiles of size *m x m* are used, where $m = 2^j + 1$ (with j = 1, 2, 3…). The solution needs to have the patch divided in, at least, 8 x 8 tiles. This means that $k - j \geq 3$. This is necessary to manage the center position of each patch, that must lie in a corner of an own tile. On other hand, a layer in the texture array used to render each patch must have an integer count of tiles, so the border cache size must be $2i(m-1)$ (with $i = 1, 2, 3\ldots$).

Finally, Figure 1 shows relationships between the sizes of the different grids discussed in this section.

2.2 Screen-space error analysis

When the available terrain surface is approximated using a lower resolution mesh, some approximation errors will occur. The approximation error is defined as the vertical distance of a vertex present in the full-resolution mesh with respect to its interpolated position when it is removed in a particular level of detail with lower resolution. This approximation error that appears when the terrain is not represented using a full-resolution mesh is called the world-space error, as it is calculated from world-space coordinates.

World-space error can be measured in, both, relative and absolute terms. Some works, like Lindstrom et al. [8], measure it relatively between successive levels of detail. In order to satisfy accurately error-bounded property this measure must be correctly saturated [9]. Nevertheless, it is more accurate to calculate the absolute error against the full-resolution terrain, as it is done in ROAM [10].

In this solution, the world-space error is computed in a pre-process and saved into a small file. That file is loaded in the loading phase into CPU memory so as to avoid disk-access latencies. Our technique calculates absolute world-space error. The approach consists in dividing the terrain into *m x m* tiles at the different levels of detail and then saving the maximum world-space error found in each tile. Finally, as the viewer moves and the patches are incrementally updated, those pre-calculated maximized errors are queried.

The maximum world-space error for each tile, in a particular level of detail, is a necessary input to analyze if the approximation errors are perceptible to the viewer. Then, maximum screen-space error is conservatively calculated. From the distance from the viewer to the closest point in the tile's bounding volume, and view parameters, as the viewport size and the field-of-view, world-space error is projected (using a perspective projection) onto the screen space viewing plane to obtain the screen-space error measured in pixels.

Finally, the application chooses a value for a tolerable screen-space error. This is used as a threshold value to compare with. If the projection of the tile's maximum world-space error onto screen-space exceeds this threshold, then that region of terrain needs to be represented with a higher resolution mesh (Figure 2).

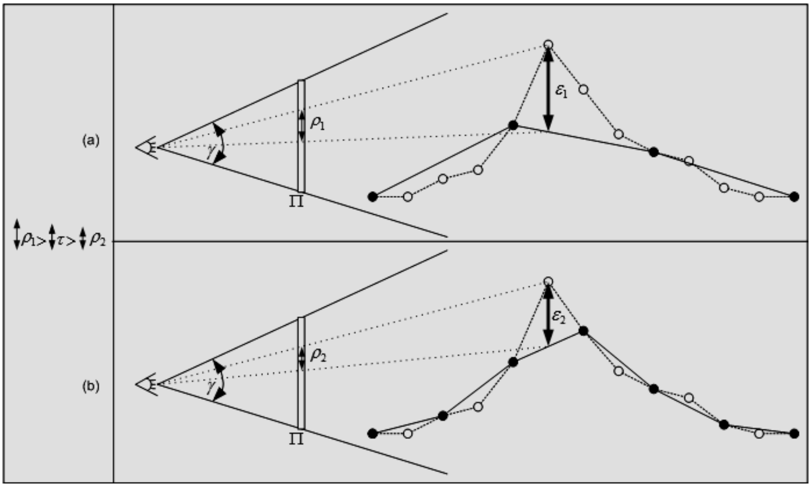


Fig. 2. World-space errors ε_1 and ε_2, originated from two different LOD representations, are projected into the projection plane Π. (a) Level 2 representation: maximum screen-space error ρ_1 exceeds threshold τ . (b) Level 1 representation: maximum screen-space error ρ_2 is smaller than threshold τ

2.3 Rendering Strategy

In this section we will describe the high-level algorithm used to generate each frame in runtime. Some useful details on implementation will be given to provide a higher performance.

First, given the viewer's position, we calculate each patch center position and its elevation data is updated if necessary. Then, the tiles that cover the render surface (*n x n*) of each patch are tested against the view frustum. At least, a coarse view-frustum culling is absolutely necessary, since it decreases nearly to a quarter the geometric rendering load to the graphics pipeline (calculated for a field-of-view of 90 degrees). The test is done using an axis-align bounding box for each tile.

Then, the screen-space error analysis is performed on those tiles which were not discarded in the view frustum culling test. As explained, the maximum world-space error present in a tile, at a particular level of detail, is projected into screen space and compared with the threshold value. If the projection exceeds the threshold, that tile needs to be represented with a higher resolution. The tile will be represented using a level of detail which maximum world-space error projection will result smaller than the threshold. That tile, that we call a ***refined tile***, will be rendered using a two, four, height, etc. times higher resolution mesh as needed (from experience it rarely needs more than two levels of refinement). Refined tiles have a size of $(r+1)*(m-1)+1\ x\ (r+1)*(m-1)+1$, where *r* is the refinement level (*r = 1, 2, 3…).*

The elevation data needed for the refined tiles is loaded into GPU memory. There is a texture array for each refinement level. Elevation data is loaded into the different layers which are managed using the least-recently-used memory management policy.

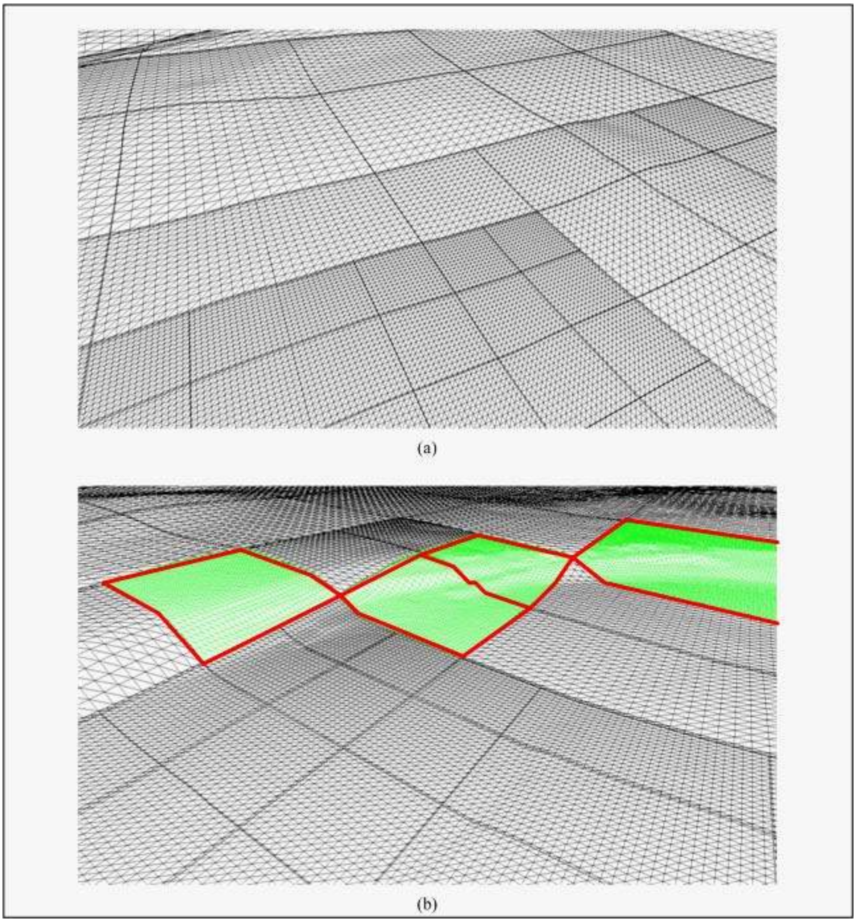


Fig. 3. Frames rendered by our solution in wire-frame mode (n = 129, m = 17). (a) Nested patches divided in tiles and different successive patches' relative position can be seen. (b) Refined tiles are drawn in green and marked in red

Rendering is done taking advantage of the instancing technique available in modern GPUs [7]. Instancing is used with a dynamic vertex buffer filled with data to instantiate each tile. It allows rendering several instances of a tile using a single *Draw* call, linking two vertex buffers to the pipeline input stage: the first containing the 2D footprint for each tile, and the second filled with the instances' data.

Finally, we render the tiles which do not need refinement first, then the refined tiles, grouped by refinement level. Figure 3 shows two frames rendered by our solution: in (a) the nested patches divided into tiles can be seen, and in (b) refined tiles are drawn in green and marked in red.

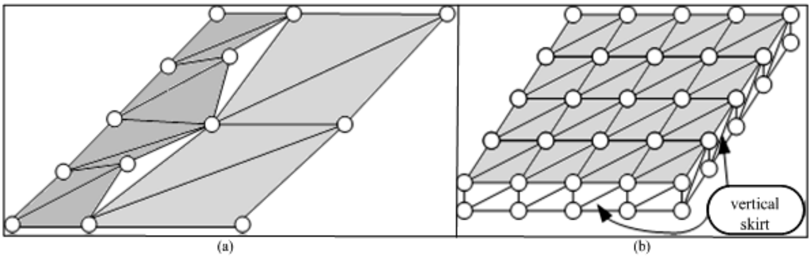


Fig. 4. (a) Gaps that may appear at the edge of tiles represented with different LOD. (b) Vertical skirts that are added around each tile

2.4 Level-of-detail approaches problems

Some artifacts may be perceptible in terrain rendering when level-of-detail approaches are used.

First, cracks in the terrain surface may appear at the edges of two regions represented using different resolutions. This problem is originated in non-continuous LOD approaches as the one described in this paper; continuous LOD approaches as [8] and [10] do not present this problem.

Tiled-blocks techniques, and also Geometry Clipmaps, have to solve this problem. Tiles at different resolution which share an edge do not have the same quantity of vertex at the edges. This can lead to gaps in the edge of two neighboring tiles (see Figure 4 (a)). Some approaches [3, 5, 6] solve it modifying the connections of the vertices in one of the tiles (usually the one with higher resolution). We will not use this approach since it requires for each tile the knowledge of the resolution of its neighbors, adding complexity and the need to manage several cases. On other hand, Ulrich [4] describes some techniques to solve the problem adding geometry. The options are to add around each tile *"flanges"*, *"ribbons"* or *"skirts"*. Geometry Clipmaps [1, 2] uses zero-area triangles to cover the perimeter of each patch. However, this requires disabling back-face culling for terrain rendering. Finally, as proposed by Ulrich [4], we decided to use vertical skirts around each tile to prevent gaps (see Figure 4 (b)).

Another artifact that may occur in level-of-detail terrain representation is known as *"popping"*. It refers to the perception of a *"pop"* in the terrain surface, a change in the terrain geometry that suddenly happens. It occurs when there is a change in the level of detail used to represent a particular terrain region while the viewer is moving. To prevent this artifact from being noticeable, some approaches manage to *slowly* change the resolution. This means that the strategy is to interpolate the vertex height between successive levels of detail when a change in the representation will occur. So, terrain geometry will slowly morph to (or from) a higher resolution representation, and for that reason, this is known as *geomorphing,* introduced by Hoppe [11] and used by many others.

As our technique has the error-bounded property, since we do not want to notice differences from the highest resolution terrain, choosing a low value for the screen-space error threshold will also guarantee that no popping will be noticeable. Pops can not exceed in screen the threshold chosen by the application.

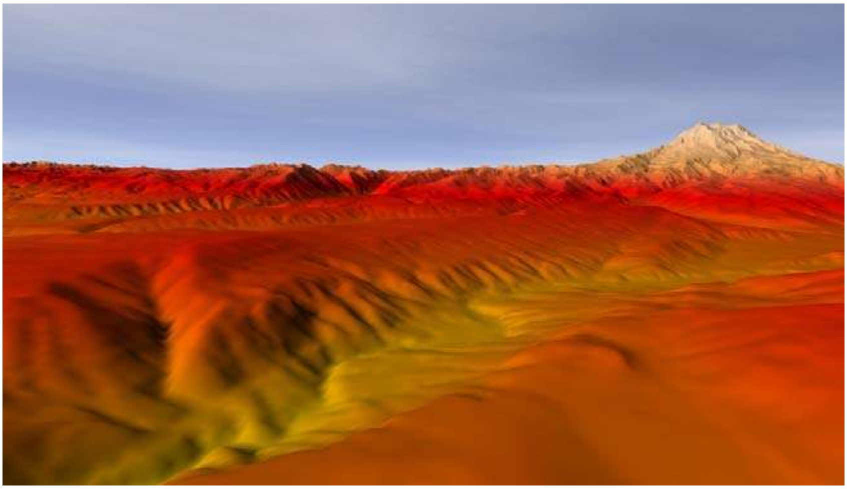


Fig. 5. Rendering result on a fly through Puget Sound 16K x 16K dataset

3. Results

Implementation was done using DirectX 10 graphics library [12] in order to access functionalities of GPU's Shader Model 4.0 [7]. Experimentation was performed using the well-known dataset of Puget Sound area. A 16,385 x 16,385 grid with 10 meters spacing covers a 163.85km x 163.85km terrain area which is suitable enough to applications, such as a flight-simulator. Height values have a 16-bit representation with 0.1m vertical resolution.

Table 1. Results from the five experiments are collected in this table

	Run 1	**Run 2**	**Run 3**	**Run 4**	**Run 5**
Flying Time	132 sec	189 sec.	180 sec.	166 sec.	140 sec.
Frames Total Count	10960	15545	13564	13118	12575
Max. Frame Time	49.5 msec.	44.1 msec.	40.7 msec.	41.5 msec.	36.0 msec.
Frame Time in 5-10 ms	49.91%	43.94%	37.14%	41.31%	49.46%
Frame Time in 10-15 ms	24.80%	14.91%	31.21%	27.19%	29.11%

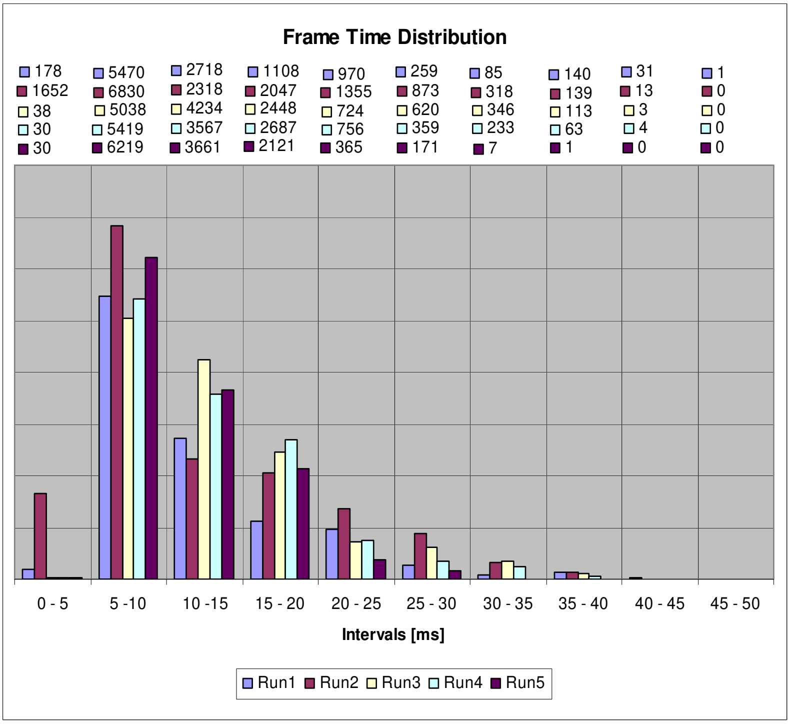


Fig. 6. Distribution of time used to generate each frame in five different flies over the terrain

Results (Figure 5) were obtained using a PC, running Windows 7 OS, with a 2.8 GHz Intel® Core™2 Quad, 4GB system memory and an Nvidia GeForce GTX 280 graphics card with 2GB of video memory.

Application was configured to run in a 1920 x 1080 full-screen window and a viewer's horizontal field-of-view of 90 degrees. Choosing a threshold value of 5 pixels (0.5% of vertical resolution), we obtain an average of 120 frames/second, which means an average of 8 milliseconds to generate each frame. On other hand, when the viewer flies near high detailed regions, for example the mountain present near the center of Puget Sound terrain, the average frame rate drops to 50 fps. Due to refinement data loading, some frame time can reach about 40 milliseconds.

We run five experiments flying over the Puget Sound terrain at 340 meters per second, from the center of the terrain to the high-detailed mountain. We measured the time to generate each frame during a flying time between two and three minutes approximately. From Figure 6, we obtained that the 70% of the frame times are between 5 and 15 milliseconds (44.35% in 5-10 ms interval, and 25.44% in 10-15ms interval). At last, from Table 1, the maximum frame time was 49.5 milliseconds.

Finally, note that average frame time allows including this technique into a graphics engine, leaving free time to other processing and rendering tasks within each frame.

4. Conclusion

This paper introduces a terrain rendering approach that is based on the state-of-the-art terrain rendering technique called Geometry Clipmaps. The contribution of this solution is to introduce an approach where the limitation presented in Geometry Clipmaps, to represent particular rough terrain without noticeable surface changes, is resolved. The strategy consists in adding view-dependent screen-space error analysis to ensure screen-space error-bounded terrain representations.

The solution presented shows a good performance in modern hardware. As it consumes little CPU processing time it can be integrated into a graphics engine to resolve the terrain rendering problem.

Finally, the approach presented targets applications where an immersive environment needs to be represented over a user well-known real-world terrain surface, such as a flight-simulator.

References

1. Lossaso, F., Hoppe, H.: Geometry Clipmaps: Terrain Rendering Using Nested Regular Grids. ACM Transactions on Graphics (SIGGRAPH) 23(3), 769-776 (2004).
2. Asirvatham, A., Hoppe, H.: Terrain Rendering Using GPU-Based Geometry Clipmaps. GPU Gems 2, Chapter 2, Addison-Wesley, March 2005.
3. de Boer, W.: Fast Terrain Rendering Using Geometrical MipMapping. E-mersion Project, October 2000.
4. Ulrich, T.: Rendering massive terrains using chunked level of detail. In: Super-size-it! Scaling up to Massive Virtual Worlds (ACM SIGGRAPH Tutorial Notes). ACM SIGGRAPH (2000).
5. Snook, G.: Simplified Terrain Using Interlocking Tiles. Game Programming Gems 2, pp. 377-383, Charles River Media, 2001.
6. Wagner, D.: Terrain Geomorphing in the Vertex Shader. Shader X2, Wordware Publishing, 2003.
7. Patidar, S., Bhattacharjee, S., Singh, J., Narayanan, P.: Exploiting the Shader Model 4.0 Architecture. *Technical Report IIIT Hyderabad*, 2006.
8. Lindstrom, P, Koller, D., Ribarsky, W., Hodges, L., Faust, N., Turner, G.: Real-Time, Continuous Level of Detail Rendering of Height Fields. *Proceedings of SIGGRAPH 96*, 109-118. August, 1996.
9. Pajarola, R., Gobbetti, E.: Survey on Semi-Regular Multiresolution Models for Interactive Terrain Rendering. The Visual Computer 23(8), 583-605, 2007.
10. Duchaineau, M., Wolinsky, M., Sigeti, D., Miller, M., Aldrich, C., Mineev-Weinstein, M.: ROAMing Terrain: Real-time Optimally Adapting Meshes. *IEEE Visualization '97*, 81-88. November, 1997.

11. Hoppe, H. Smooth view-dependent level-of-detail control and its application to terrain rendering. IEEE Visualization 1998, 35-42, October 1998.
12. Blythe, D. Direct3D 10. GPU Shading and Rendering, SIGGRAPH 06 Course, 2006.

VIII

Software Engineering Workshop

Exploring Software Engineering Techniques for Developing Robotic Systems

CLAUDIA PONS[1,2,3], ROXANA GIANDINI[2], GABRIELA ARÉVALO[1,3], DIMITRIS KARAGIANNIS[4]

[1]CONICET
[2] LIFIA, Facultad de Informática, UNLP, Buenos Aires, Argentina
[3] Universidad Abierta Interamericana, UAI, Buenos Aires, Argentina
[3] University of Vienna, Austria
{cpons, giandini}@info.unlp.edu.ar

Abstract. *The robotics community has a sufficient amount of experience on how to build complex robotics systems. However, we cannot expect significant growth with hand-crafted single-unit systems and it is mandatory to work towards applying engineering principles to cope with the complexity of robotics software systems. In most cases, we already have the knowledge about what proved to be a good solution in the software engineering field. The next step is to make this knowledge explicit and easily accessible for new systems. Applying existing technology would save time and effort that is better put into what is specific in robotics. In this paper we present an overview of ongoing activities regarding the application of modern software engineering techniques on the robotic software development process. We observe a growing tendency on the application of component based development as well as service oriented architecture and model driven software development; however those techniques are mostly applied in isolation, failing to achieve the possible benefits derived from combining the three technologies.*

Keywords: *robotic software system, model-driven software development (MDD), software engineering, Service Oriented Architecture (SOA), Component based software development (CBD).*

1. Introduction

Robotic systems (RSs) play an increasing role in everyday life. Also the need for robotic systems in industrial settings increases and becomes more demanding. While robotic systems grow to be more and more complex, the need to engineering their software development process grows as well. Traditional approaches that are used in the development process of these software systems are reaching their limits; currently used methodologies and toolsets fall short to address the needs of such complex software development process.

It is widely accepted that new approaches should be established to meet the needs of the development process of today's complex RSs. In this direction, Component-based development (CBD) (Szyperski, 2002), Service Oriented Architecture (SOA) (Bell 2008 and 2010), as well as Model Driven software Engineering (MDE) (Stahl, 2006) (Pons et al., 2010) and Domain-Specific Modeling (DSM) (Steven and Juha-Pekka, 2008) are among the key promising technologies in the RSs domain.

This paper presents a systematic review of the current use of those modern software engineering techniques for the development of robotic software systems and their actual automation level. The goal of the survey is to summarize the existing evidence concerning the application of such technologies on the robotic systems field. The paper is organized as follows. Section 2 describes the methodology we adopted to perform the review. Section 3 describes the needs for performing a review in this area; Section 4 presents the planning of the review. Section 5 reports data we extracted from each paper. Section 6 answers our research questions. Finally, we report our conclusions in Section 7.

2. Systematic Literature Reviews and Systematic Mapping Studies

A systematic literature review (SLR) (Kitchenham and Charters, 2007) (Dybå et al., 2003) is a means of identifying, evaluating and interpreting all available research relevant to a particular research question, or topic area, or phenomenon of interest. Individual studies contributing to a systematic review are called primary studies; a systematic review is a form of secondary study.

Some of the features that differentiate a systematic review from a conventional expert literature review are that: SLRs start by defining a review protocol that specifies the research question being addressed and the methods that will be used to perform the review; SLRs are based on a defined search strategy that aims to detect as much of the relevant literature as possible; SLRs document their search strategy so that readers can assess their rigor and the completeness and repeatability of the process; SLRs require explicit inclusion and exclusion criteria to assess each potential primary study and specify the information to be obtained from each primary study including quality criteria by which to evaluate each primary study.

There are other types of review that complement systematic literature reviews such as the systematic mapping studies. If, during the initial examination of a domain prior to commissioning a systematic review, it is discovered that very little evidence is likely to exist or that the topic is very broad then a systematic mapping study may be a more appropriate exercise than a systematic review. A systematic mapping study allows the evidence in a domain to be plotted at a high level of granularity. This allows for the identification of evidence clusters and evidence deserts to direct the focus of future systematic reviews and to identify areas for more primary studies to be conducted.

A SLR involves several discrete activities. Existing guidelines have slightly different suggestions about the number and order of activities However, the medical guidelines and sociological text books are broadly in agreement about the major stages in the process. Kitchenham and colleagues in (Kitchenham and Charters, 2007) summarize the stages in a systematic review into three main phases: Planning the Review (that includes the activities of identification of the need for a review, specifying the research questions, identification of research, selection of primary studies, study quality assessment, developing a review protocol, evaluating the review protocol); Conducting the Review (conformed by the following activities: data extraction and monitoring, data synthesis); Reporting the Review (conformed by the following activities: specifying dissemination mechanisms, formatting the main report, evaluating the report).

Due to the extensiveness of our topic of interest, in the present work we will perform a systematic literature review oriented to mapping studies.

3. The needs for a review in this field

Prior to undertaking a systematic review it is necessary to confirm the need for such a review. Although the complexity of robotic software is high, in most cases reuse is still restricted to the level of libraries. At the lowest level, a multitude of libraries have been created for robot systems to perform tasks like mathematical computations for kinematics, dynamics and machine vision, such as (Bruyninckx, 2001). Instead of composing systems out of building blocks with assured services, the overall software integration process for another robotic system often is still reimplementation of the glue logic to bring together the various libraries. Often, the kind of overall integration is completely driven by a certain middleware system and its capabilities. Middlewares are often used to hide complexity regarding inter-component communication, for example OpenRTM-aist (Ando et al., 2005) is a CORBA-based middleware for robot platforms that uses so-called robot technology components to model distribution of functionality. Obviously, this is not only expensive and wastes tremendous resources of highly skilled roboticists, but this also does not take advantage from a maturing process to enhance overall robustness.

We have faced this problem in our own practice. We have been programming educational robots for more than 10 years (GIRA, 2011) (CAETI, 2011) and we have observed in the last years the emergence of robotic kits oriented to non-expert users that gave rise to the development of a significant number of educational projects using robots. Those projects apply robots at different education levels, from kindergarten through higher education, especially in areas of physics and technology. In this context, one of the problems we encountered is that the hardware of the robotic kits is constantly changing; in addition its use is not uniform across different regions and even education levels. Therefore, the technical interfaces of these robots should hide these differences so that teachers are not required to change their educational material over and over again. An example of these interfaces is "Physical Etoys" (CAETI, 2011), a project in which we participated and which proposes a standard teaching platform for programming robots, regardless of whether they are based on Arduino, Lego, or other technologies.

In this context, it is widely accepted that new approaches should be established to meet the needs of the development process of today's complex RSs. Component-based development (CBD) (Szyperski, 2002), Service Oriented Architecture (SOA) (Bell 2008 and 2010), as well as Model Driven software Engineering (MDE) (Stahl, 2006), (Pons et al., 2010) and Domain-Specific Modeling (DSM) (Steven and Juha-Pekka, 2008) are among the key promising technologies in the RSs domain.

In first place, the Component-based development paradigm (Szyperski, 2002) states that application development should be achieved by linking independent parts, the components. Strict component interfaces based on predefined interaction patterns decouple the sphere of influence and thus partition the overall complexity. This results in loosely coupled components that interact via services with contracts. Components such as architectural units allow specifying very precisely, using the concept of port, both the services provided and the services required by a given component and defining a composition theory based on the notion of a connector. Component technology offer high rates of reusability and ease of use, but little flexibility with regard to the implementation platform: most existing component are linked to C/ C++ and Linux (e.g. Microsoft robotics developer studio (Microsoft, 2009), EasyLab (Barner et al., 2008), Player/Stage project (Gerkey et al., 2001)), although some achieve more independence, thanks to the use of some middleware (e.g. Smart Software Component model (Schlegel, 2007), Orocos (Bruyninckx, 2001) Orca (Brooks et al., 2005), CLARAty (Nesnas et al., 2003)).

In second place, we need a way to define interfaces and behavior at a higher level of abstraction so that they could be used in systems with different platforms. This is what prompted the idea of abstract components, which would be independent of the implementation platform but could be translated into an executable software or hardware component. Thus, the migration from code-driven designs to a model-driven development is mandatory in robotic components to overcome the current problems. A model-based description is a suitable mean to express contracts at component interfaces and to apply tools to verify the overall behavior of composed systems and to automatically derive the executable software. Instead of building tool support for each framework from scratch, one should now try to either express the needed models in standardized modeling languages like UML or any DSL, separating components from the underlying computer hardware. In the context of software engineering, the Model Driven

Development (MDD) (Stahl, 2006), (Pons et al., 2010) and Domain-Specific Modeling approach (DSM) (Steven and Juha-Pekka, 2008) have emerged as a paradigm shift from code-centric software development to model-based development. Such approaches promote the systematization and automation of the construction of software artifacts. Models are considered as first-class constructs in software development, and developers' knowledge is encapsulated by means of model transformations. The essential characteristic of MDD and DSM is that software development's primary focus and work products are models. Its major advantage is that models can be expressed at different levels of abstraction and hence they are less bound to any underlying supporting technology. This is especially relevant for software systems within the ubiquitous computing domain, which consist of dynamic, distributed applications and heterogeneous hardware platforms, such as robotic systems.

Finally, Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA will package functionality as a suite of interoperable services that can be used within multiple, separate systems from several business domains. SOA also generally provides a way for consumers of services, such as web-based applications, to be aware of available SOA-based services. SOA defines how to integrate widely disparate applications for a Web-based environment and uses multiple implementation platforms. Rather than defining an API, SOA defines the interface in terms of protocols and functionality. Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units, or services (Bell, 2008) which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format (Bell, 2010).

Summarizing, we know that these software engineering techniques offer good potential for the development of robotic systems, so we need to search for proposals in these directions and we need to detect which work is already done and which work is pending. Additionally we want to know if there is any proposal taking advantage of the combined application of CBP, SOA and MDE to robotic software system development.

4. Planning the review

A review planning specifies the methods that will be used to undertake a specific systematic review. A pre-defined planning is necessary to reduce the possibility of researcher bias.

4.1 The Research Questions

Specifying the research questions is the most important part of any systematic review. In this context, the right question is usually one that will lead either to changes in current software engineering practice or to increased confidence in the value of current practice and/or will identify discrepancies between commonly held beliefs and reality. The 5 research questions investigated in this study were:

> **RQ1** Have MDD techniques been applied to the development of robotic systems and how is the current tendency?
> **RQ2** Have CBD techniques been applied to the development of robotic systems and how is the current tendency?
> **RQ3** Have SOA techniques been applied to the development of robotic systems and how is the current tendency?
> **RQ4** Have those techniques been used in combination or in isolation?
> **RQ5** Which MDE techniques have been applied to the development of robotic systems and which is their automation level?

4.2 The Search strategy

A search strategy was used to search for primary studies. Such strategy includes search terms and resources to be searched. Resources include digital libraries, specific journals, and conference proceedings. We searched two digital libraries and one broad indexing service: IEEE Computer Society Digital Library; ACM Digital Library and SCOPUS indexing system. All searches were based on title, keywords and abstract. The searches took place in May and June 2011. We use the following Boolean query (adapted to the particular syntax of each library):

(robot*)
AND
("software development" **OR** "system development" **OR** programming)
AND
(MDD **OR** MDE **OR** "model driven" **OR** "domain specific language" **OR** "domain specific modeling" **OR** DSL **OR** "code generation" **OR** "generative programming" OR "Component based" OR CBD OR "service oriented" **OR** "service based" **OR** SOA **OR** "Web service")

Concerning the quality of the search strategy, general guidelines recommend considering the effectiveness of a question from five viewpoints (PICOC criteria):
Population: that is the application area.
Intervention: the intervention is the software methodology/tool/technology/procedure that addresses a specific issue.

Comparison: this is the software engineering methodology/tool/technology/procedure with which the intervention is being compared.
Outcomes: outcomes should relate to factors of importance to practitioners such as improved reliability, reduced production costs, and reduced time to market.
Context: this is the context in which the comparison takes place (e.g. academia or industry), the participants taking part in the study (e.g. practitioners, academics, consultants, students), and the tasks being performed (e.g. small scale, large scale).

According to this PICOC criteria our query is organized as follows,
Population: the population corresponds to the robotic domain. This is reflected in the first sub-expression of our query.
Intervention: the intervention of our survey comprises software and system development. This fact is specified in the second sub-expression of our query.
Comparison: in our case, the software engineering methodologies to be compared or analyzed are MDD, SOA and CBD. This is indicated in the third sub-expression of our query.
Outcome: we want to obtain as much outcomes as possible by collecting all the available information in the domain of study, so our query does not restrict the kind of outcomes.
Context: we apply no restriction to the context of our study.

4.3 Study Selection criteria

Study selection criteria are intended to identify those primary studies that provide direct evidence about the research question. Once the potentially relevant primary studies have been obtained, they need to be assessed for their actual relevance. Study selection criteria are used to determine which studies are included in, or excluded from, a systematic review. It is usually helpful to pilot the selection criteria on a subset of primary studies.

We undertook an initial screening of 195 papers that were found based on title, abstract and keywords. The IEEE Computer Society Digital Library contributed 55 articles (representing the 28%), while the ACM Digital Library provided 140 articles (representing the 72%). Finally, the results from searching the SCOPUS indexing system were completely included in the previous results, so SCOPUS does not contribute new articles.

In this screening we excluded studies that were obviously irrelevant, or duplicates and we eliminated 91 articles. The remaining 104 papers were then subject to a second assessment: we obtained full copies of these remaining papers and undertook a more detailed second screening using the following inclusion and exclusion criteria: -The paper should be related to software engineering rather than mathematical modeling and/or math simulation. – “service oriented” should refer to SOA but not to “robots that perform a service”. Based on those inclusion criteria we determined that 37 articles were excluded by the criteria. Finally, we analyzed the remaining 67 articles. All those sources can be retrieved from http://lifia.info.unlp.edu.ar/eclipse/robotsurvey2011.

5. Data extraction strategy

This strategy defines how the information required from each primary study will be obtained. The objective of this stage is to design data extraction forms to accurately record the information researchers obtain from the primary studies.

We elaborated a form comprising the following fields:

Field name	Type
Paper identification	Integer
Year of publication	Date
It applies SOA	Boolean
It applies CBD	Boolean
It applies MDD	Boolean

If the value of the last field is True (i.e., the paper applies MDD), then the following form is filled:

Field name	Type
Modeling Language	{UML, Profile, DSML}
Programming Language	{Any language, robotic-high-level}
Model Transformation Technique	{GPL, DSL, Black-box}
Tools	{existing tool, new tool }
Automation Level	{Full, Medium, Low}

The field named "*Modeling Language*" specifies which language is used to express the platform independent models. We found that some projects use the standard UML language, while other projects consider that UML is not expressive enough and then they define an extension through the creation of a profile. Finally, other proposals do not use UML but instead they define their own domain specific modeling language (DSML).

The field named "*Programming Language*" identifies the implementation language that is used as the target of model transformations. We observed that in most cases the PIM models are translated to different languages, which is one of the principles of MDD. However in other cases the PIM models are mapped to a specific high level language, such as Urby, or to a specific middleware, such as MSRS. The field named "*Model Transformation Technique*" indicates which is the strategy used to transform the PIM to the PSM or to the code. Some projects implement the transformation just using a general purpose programming language (GPL) such as Java, while other proposals use existing transformation DSLs, such as ATL, JET or QVT. Finally, most proposals use black-box transformations. The field named "*Tools*" denotes which kind of software tools are being used in the project. The options are as follows, using existing tools (such as EMF or MS DSL tools) or creating a new specific tool. The field named "*Automation Level*" states how much work is made automatically. The value "Full" indicates that code is fully automatically generated from models. The "Medium" value states that code is partially generated and it should be completed manually, while the "Low" value indicates that the transformation from models to code is carried out mostly by hand.

6. Data synthesis strategy: answering the questions

Data synthesis involves collecting and summarizing the results of the included primary studies. We present here the answers to our research questions and we display quantitative foundations.

Figure 1 shows the answer to the first three questions. We observe an increasing tendency in the use of all these techniques, being CBD the most applied in the robotics field.

Figure 2 answers the question number 4 showing the distribution of articles in each field. We observe little intersection among the different technologies. However there is a promising intersection between MDD and CBD showing the good potential of combining these two technologies.

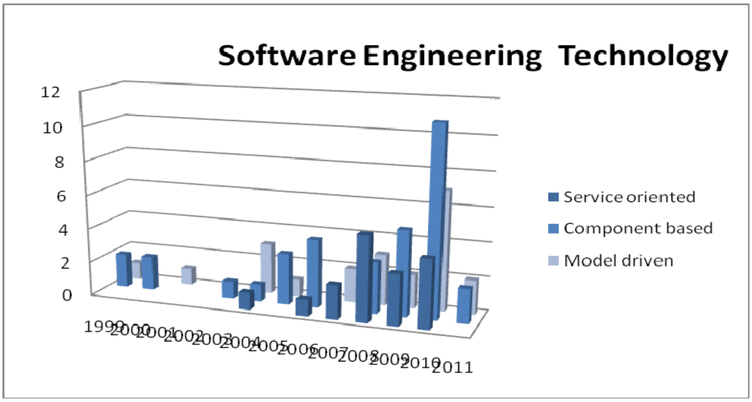


Fig. 1. Software Engineering Technology

Concerning question number 5, the figure 3 illustrates which modeling languages are being applied in the robotic projects. We observe that the definition of new domain specific languages is the most applied technique (64%), while the use of UML and its profiles come later (27% and 9% respectively). Regarding the application of MDD tools we observe in figure 4 that 65% of MDD projects take advantage of existing MDD tools such as ATL, EMF and DSL tools, while the 35% implements their own modeling and transformation tools. The reasonable tendency is that existing tools will be increasingly reused in the near future.

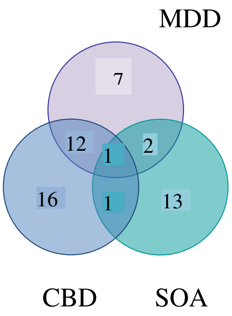


Fig. 2. Field intersection

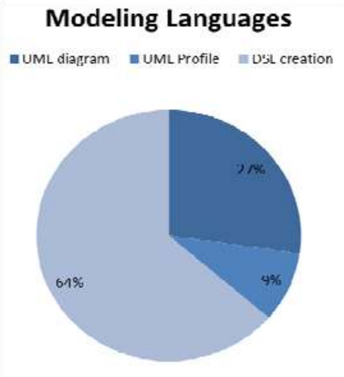


Fig. 3. Modeling languages

Finally, figure 5 shows the distribution of levels of automation in the MDD projects. Only 55% have achieve full automation in their MDD process, while 27% present an intermediate level of automation, that implies the creation of abstract models and the automatic generation of code skeleton that should be manually completed by the developers. Finally, 18% of the MDD robotic projects only reach a low level of automation consisting in the creation of abstract models but the manual derivation of code.

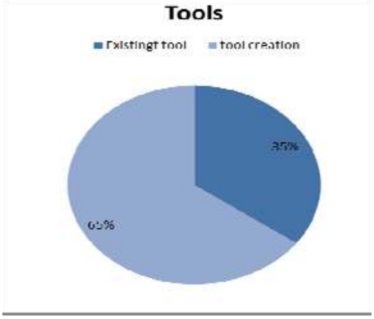


Fig. 4. Tools

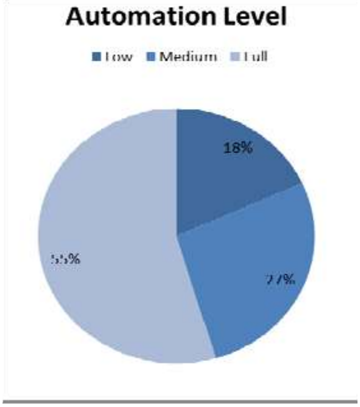


Fig. 5. Automation Level

7. Conclusion

Robots have become usual collaborators in our daily life. While robotic systems grow to be more and more complex, the need to engineering their software development process grows as well. Traditional approaches that are used in the development process of these software systems are reaching their limits; currently used methodologies and toolsets fall short to address the needs of such complex software development process. Separating robotics knowledge from short-cycled implementation technologies is essential to foster reuse and maintenance.

In this paper we have built a systematic review of the current use of modern software engineering techniques for the development of robotic software systems and their actual automation level. We observe a growing tendency on the application of Component based development as well as Service based architecture and Model driven software development, although those techniques have been mostly applied in isolation. For example, the works presented in (Basu et al, 2011), (Biggs, 2010), (Brooks et al, 2005), (Jawawi et al, 2008) and (Min Yang Jung et al, 2010) take advantage of the CBD paradigm for the development of different types of robotics systems. While the proposals described in (Amoretti et al, 2007) and (Cesetti et al, 2010) apply SOA for building autonomic robot systems. On the other hand, there are only preliminary proposal on applying model-driven development to robotics, see for example the works described in (Arney et al, 2010), (Baer et al, 2007), (Brugali and Scandurra, 2009), (Brugali and Shakhimardanov, 2010), (Hyun Seung Son et al., 2008), (Iborra et al, 2009), (Jorges et al 2007), (Jung,et al, 2005), (Sanchez et al, 2010), (Schlegel, 2009) and (Wei et al, 2009). While only one work combines the three technologies, such as the proposal introduced in (Tsai et al, 2008).

After reviewing more than 100 papers on the subject we have identified gaps in current research that open the door to further investigation. Our analysis provides a background in order to appropriately position new research activities.

References

1. Amoretti, M.; Zanichelli, F.; Conte, G.; A Service-Oriented Approach for Building Autonomic Peer-to-Peer Robot Systems Enabling Technologies: Infrastructure for Collaborative Enterprises, 2007. WETICE 2007. 16th IEEE International Workshops on, Page(s): 137-142 (2007).
2. Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T., Yoon, W.K., RT-middleware: Distributed component middleware for RT (robot technology). In: International Conference on Intelligent Robots and Systems 2005 (IROS 2005), pp. 3933-3938 (2005).

3. Arney, D.; Fischmeister, S.; Lee, I.; Takashima, Y.; Yim, M.;Model-Based Programming of Modular Robots. 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC), Page(s): 66-74 (2010).
4. Baer, P. A.; Reichle, R.; Zapf, M.; Weise,T.; Geihs, K.; A Generative Approach to the Development of Autonomous Robot Software.EASe '07. Fourth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (2007).
5. Barner, S., Geisinger, M., Buckl, C., Knoll, A.: EasyLab: Model-based development of software for mechatronic systems. In: IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications. Beijing, China (2008).
6. Basu, A.; Bensalem, B.; Bozga, M.; Combaz, J.; Jaber, M.; Nguyen, T.; Sifakis, J.; Rigorous Component-Based System Design Using the BIP Framework Software, IEEE Volume: 28 , Issue: 3 Page(s): 41-48 (2011).
7. Bell, M., "Introduction to Service-Oriented Modeling". Service-Oriented Modeling: Service Analysis, Design, and Architecture. Wiley & Sons. pp. 3. ISBN 978-0-470-14111-3 (2008).
8. Bell, M., SOA Modeling Patterns for Service-Oriented Discovery and Analysis. Wiley & Sons. pp. 390. ISBN 978-0470481974 (2010).
9. Biggs, G.; Flexible, adaptable utility components for component-based robot software. Robotics and Automation (ICRA), 2010 IEEE International Conference on, Page(s): 4615-4620 (2010).
10. Brooks, A., Kaupp, T., Makarenko, A., Oreback, A., Williams, S.: Towards component-based robotics. In: Proc. of 2005 IEEE/RSJ Int. Conf. on Intellegent Robots and Systems (IROS'05), pp. 163-168. Alberta, Canada (2005).
11. Brugali, D.; Scandurra, P.; Component-based robotic engineering (Part I) [Tutorial] Robotics & Automation Magazine, IEEE Volume: 16 , Issue: 4, Page(s): 84-96 (2009).
12. Brugali, D.; Shakhimardanov, A.; Component-Based Robotic Engineering (Part II) Robotics & Automation Magazine, IEEE Volume: 17, Issue: 1, Page(s): 100-112 (2010).
13. Bruyninckx, H., Open robot control software: The OROCOS project. In: Proceedings of 2001 IEEE International Conference on Robotics and Automation (ICRA'01), vol. 3, pp. 2523-2528 (2001).
14. CAETI (Centro de Altos Estudios en Tecnología Informática). Proyectos del área robótica. http://www.caeti.uai.edu.ar. Accedido Junio 2011.
15. Cesetti, A.; Scotti, C. P.; Di Buo, G.; Longhi, S.; A Service Oriented Architecture supporting an autonomous mobile robot for industrial applications Control & Automation (MED), 2010 18th Mediterranean Conference on, Page(s): 604-609 (2010).
16. Dybå, T., Kitchenham, B.A., Jørgensen M., Evidence-based software engineering for practitioners, IEEE Software 22 (1) 58-65 (2005).
17. Gerkey, B.P., Vaughan, R.T., Howard, A., Most valuable player: a robot device server for distributed control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1226–1231. Wailea, Hawaii, Player Stage (2001).

18. GIRA Grupo de Investigación en Robótica Autónoma del CAETI. http://tecnodacta.com.ar/gira/ (accedido en May 2011).
19. Hyun Seung Son, Woo Yeol Kim; Kim, R., Semi-automatic Software Development Based on MDD for Heterogeneous Multi-joint Robots. In Future Generation Communication and Networking Symposia, 2008. FGCNS '08.: 2008, Page(s): 93-98 (2008).
20. Iborra, A.; Caceres, D.; Ortiz, F.; Franco, J.; Palma, P.; Alvarez, B.; Design of Service Robots. Experiences Using Software Engineering. IEEE Robotics & Automation Magazine 1070-9932/09/ IEEE Page(s): 24-33. March 2009.
21. Jawawi, D.N.A.; Deris, S.; Mamat, R.; Early-Life Cycle Reuse Approach for Component-Based Software of Autonomous Mobile Robot System. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2008. SNPD '08. Ninth ACIS International Conference on, Page(s): 263-268 (2008).
22. Jorges, Sven; Kubczak, Christian; Pageau, Felix; Margaria, Tiziana; Model Driven Design of Reliable Robot Control Programs Using the jABC. Engineering of Autonomic and Autonomous Systems, 2007. EASe '07. Fourth IEEE International Workshop on, Page(s): 137-148 (2007).
23. Jung, E.; Kapoor, C.; Batory, D.; Automatic code generation for actuator interfacing from a declarative specification Intelligent Robots and Systems, 2005. (IROS 2005). IEEE/RSJ International Conference on. Page(s): 2839-2844 (2005).
24. Kitchenham, B.A., Charters, S., Guidelines for Performing Systematic Literature Reviews in Software Engineering Technical Report EBSE-2007-01, 2007.
25. Microsoft, "Microsoft robotics developer studio," 2009, http://msdn. microsoft.com /en-us/robotics/default.aspx, visited on March 11th 2009.
26. Min Yang Jung; Deguet, A.; Kazanzides, P.; A component-based architecture for flexible integration of robotic systems Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, Page(s): 6107-6112 (2010).
27. Nesnas, I., Wright, A., Bajracharya, M., Simmons, R., Estlin, T.: CLARAty and challenges of developing interoperable robotic software. In: Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 3, pp. 2428-2435 (2003).
28. Pons, C., Giandini R., Pérez, G., "Desarrollo de Software Dirigido por Modelos. Teorías, Metodologías y Herramientas", Ed: McGraw-Hill Education. ISBN: 978-950-34-0630-4 (2010).
29. Sanchez, P; Alonso, D; Rosique, F; Alvarez, B; Pastor, J; Introducing Safety Requirements Traceability Support in Model-Driven Development of Robotic Applications. Computers, IEEE Transactions on Volume: PP , Issue: 99 (2010).
30. Schlegel, C., ''Communication patterns as key towards component interoperability,'' in Software Engineering for Experimental Robotics (Series STAR, vol. 30), D. Brugali, Ed. Berlin, Heidelberg: Springer-Verlag, pp. 183-210. Smartsoftware (2007).

31. Schlegel, C., Haßler, T., Lotz, A., Steck, A., Robotic Software Systems: From Code-Driven to Model-Driven Designs. In procs. Of ICAR 2009. International Conference on Advanced Robotics. IEEE Press (2009).
32. Stahl, M Voelter. Model Driven Software Development. John Wiley (2006).
33. Steven, K., Juha-Pekka, T., Domain-Specific Modeling. John Wiley &Sons, Inc. 2008 (2008).
34. Szyperski, C., Component Software: Beyond Object-Oriented Programming. 2nd ed. Addison-Wesley Professional, Boston ISBN 0-201-74572-0 (2002).
35. Tsai, W.T., Qian Huang, Xin Sun. A Collaborative Service-Oriented SimulationFramework with Microsoft Robotic Studio® Simulation Symposium, 2008. ANSS 2008. 41st AnnualDigital Object Identifier: 10.1109/ANSS-41.2008.32, Page(s): 263-270 (2008).
36. Wei Hongxing; Duan Xinming; Li Shiyi; Tong Guofeng; Wang Tianmiao; A component based design framework for robot software architecture. Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on, Page(s): 3429-3434 (2009).

ReqGIS Classifier: A Tool for Geographic Requirements Normalization

VIVIANA E. SALDAÑO[1], AGUSTINA BUCCELLA[2], ALEJANDRA CECHICH[2]

[1] Proyecto de Investigación Área Ingeniería de Software
Unidad Académica Caleta Olivia – Universidad Nacional de la Patagonia Austral
vivianas@uaco.unpa.edu.ar
[2] Grupo de Investigación en Ingeniería de Software del Comahue (GIISCo)
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
{abuccel, acechich}@uncoma.edu.ar

Abstract. *Component Based Software Development (CBSD) is a development process based on components' reuse. One of the main difficulties for developers is selecting the most suitable component that fit in their development systems. In this paper we describe a software tool, named ReqGIS, which supports our methodology for improving components' identification in a geographic information environment. In particular, we introduce a new component named AlgSim, which completes the automation of the whole methodology. It starts analyzing user requirements specified by use cases and returns the best fitting geographic service category corresponding to those requirements.*

Keywords: *DSBC, Off-The-Shelf (OTS), GIS services, geographic component selection.*

1. Introduction

Software reuse has been incremented during last years, becoming a common practice for software products development. In particular, Component Based Software Development (CBSD) is based on components' reuse which have been developed at different times, by different people and possibly with distinct goals of use [21]. In this context, one of the main difficulties for developers is searching and selecting the most suitable components. It is known that, a wrong component selection will impact through all the software development life cycle. Therefore, searching and selecting OTS (Off-The-Shelf) components [3] are quite important.

A key mechanism which is responsible of searching and selecting components is the mediator process. In this context, a client who requires a specific component service may interrogate a mediator service for the references to those components which supply the required service. Another key issue is standardizing components' information. Service supply can be standardized so that compositions are stored in an easy access repository. The same should happen for services demand, which should also be expressed in standard terms to make search easier.

Thus, two models can be identified: *demand* and *supply*. The *supply model* concerns gathering and storing components' information in a repository in a standard way. On the other hand, the *demand model* involves identifying required services based on user requirements. The connection between these two models is the mediator service which is responsible of mapping the required services with components implementing them.

In this work, we are interested in geographic services which are necessary for implementing geographic information systems. In the last ten years, many GIS software companies have begun supplying software components to satisfy GIS software developers' needs. Therefore, a methodology and its supporting tool for facilitating the demand model and the identification of the correct components shall be very useful in this context.

The work presented in this paper is an extension of works previously presented in [17, 18, 19], in which we have proposed a methodology for improving the component identification process. In particular, this work is presented as a complement to the supply model presented in [6, 7, 8] where a publication service is defined to facilitate selection of requested components.

In this paper, we describe our supporting tool, named ReqGIS, which implements the process for searching and selecting geographic components automatically. Requirements of GIS developers are processed and classified according to a geographic services category. After classifying requirements, a mediation service is invoked to find references to components which fit in the required functionality. In this context, we have developed a geographic-services taxonomy, a use-case knowledge extraction process, and a supporting tool which classifies requirements according to service categories defined in the taxonomy.

This paper is organized as follows: next section describes a methodology for geographic services identification. Section 3 describes the supporting tool developed to classify geographic services. Then, in Section 4 we apply the whole process in a real example. Future work and conclusions are discussed afterwards.

2. Methodology for Geographic Services Identification from User Requirements

In this section we describe our methodology [18, 19] to classify services specified in textual use cases. This methodology implements the *demand model* in which a client (developer) who requires a specific component service shall ask a mediation service to find the references to those components which provide the required service category. Thus, the main goal is to identify required services from use cases in order to find the correct GIS components that provide these services. Figure 1 shows the main steps of the methodology.

As we can see, the input of the methodology are use cases. The developer provides a use case in which the main functionality required is described. In our approach, to take advantage of the natural language and to avoid ambiguities, we have analyzed use case proposals involving a restricted natural language. Thus, we have selected the proposal presented by Cockburn [4], in which templates are applied to specify the behavior within use cases. In addition, we have restricted the language in these use cases by applying a controlled natural language which structures sentences in a particular way [9]. Here, the SVDPI (Subject, Verb, Direct object, Preposition, Indirect object) pattern is applied as follows: "Sentence structure must be simple"… "Subject… verb … direct object … preposition … indirect object".

In this way, these two proposals [4, 9] are combined in order to maximize the understanding of use cases for common users and to provide, at the same time, a notation in which the automatic analysis and validation are possible.

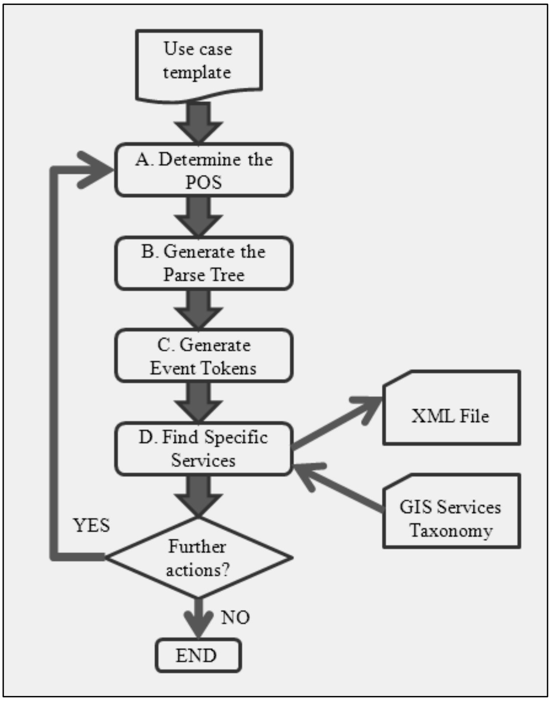


Fig. 1. Steps for extracting GIS services from use cases

In addition, in Figure 1 we can see a *GIS Services Taxonomy* component used to classify the GIS services. This taxonomy has been built by using the information provided by ISO 19119 std. This standard was developed by the Open Geospatial Consortium (OGC) and the International Standardization Organization (ISO). It proposes a geographic services classification that shall be used for all the systems compliant to this International Standard. The standard defines six categories grouping human interaction, model/information management, workflow/task management, processing, communication, and system management services. In a previous work [17], we have defined this taxonomy based on the ISO 19119 std. In addition, in order to support the matching process between user requirements and services categories, we have defined a list of keywords which describe services provided by each category [19]. In Table 1 we can see part of this taxonomy. The first column of the table is the category as defined in the standard and the second and third columns denote the keywords for service description. For instance, within the Human Interaction category, main verbs to describe services here are interact, locate, manage, etc.; and the representative objects can be catalogue, map, chain, etc.

Table 1**.** Fragment of GIS Taxonomy

Category	Service Description	
	Main Verb	Representative Object
Human Interaction	interact locate browse manage view display overlay query animate calculate edit	catalogue metadata feature coverage map spreadsheet service chain workflow view perspective texture symbol structure dataset

The other component that we can see in Figure 1 is the *XML File* component which is used to store the result of the mapped service.

According to Figure 1 the main steps of our methodology are:

A. *Determining the POS (part-of-speech)*: It analyzes each word and specifies the type (verb, noun, etc.) and the role of each of them within the sentence in which they are defined.

B. *Generating the parse tree:* Different parse trees are created according to the sentences of the main scenario of the use cases.

C. *Generating event tokens*: Event tokens are created by finding main verbs and representative objects within each sentence of the parse tree.

D. *Finding specific services:* Each event token is processed to get the corresponding geographic category according to the GIS Services Taxonomy.

The methodology applies linguistic tools to build a parse tree in which actions of predefined textual use cases are identified. Then these actions are used to discover the required GIS services. The method takes a use case specification and processes each step of the main scenario (main part of use case template) by performing the A-D steps.

The software tool implementing all the methodology, named ReqGIS, was partially implemented and described in previous works [18, 19]. Steps A-C of the methodology (Figure 1) have been implemented by using FreeLing Tool Suite [5, 15]. The tool reads a sentence (of the main scenario of the use case) and returns a parse tree with necessary information to generate the Event Token (step C). However, step D had to be made manually, that is, the developer was responsible of understanding the event token and finding the specific service in the taxonomy. Therefore, in this work, we present the *AlgSim component* which completes the implementation of the ReqGIS tool. This component implements step D by using the event token as input and returning the corresponding geographic category. The result is stored in an XML file aforementioned, which shall be used to find mappings between user requirements and the information of OTS components published on the Web. With the *AlgSim* component we fully automate the whole *demand process*. In the next section we describe the ReqGIS tool in detail, and in particular the *AlgSim* component.

3. ReqGIS: Requirements Classification Tool for GIS Services

The main goal of the ReqGIS tool is to automate the process of classifying developers' GIS requirements and speed up the demand process. In this way clients will find the most suitable component in less time.

The requirements' classification tool has been created by reusing components available on Internet. Figure 2 shows these main components that work together in order to support the steps of our methodology (Figure 1). Following, we describe each of the ReqGIS tool's components:

FreeLing Component. As we have described in the last section, FreeLing [15] is an open-source multilingual language processing library providing a wide range of language analyzers for several languages. It offers text processing and language annotation facilities to natural language processing application developers, simplifying the task of building those applications. In ReqGIS, Freeling performs steps A-C of our methodology. It receives a use case main scenario step (an english sentence in SVDPI format) and returns a parse tree of the sentence, with the corresponding syntactic analysis. This parse tree is then used to build the event token, taking the words tagged as top and direct object for event token's main verb and representative object respectively.

WordNet::Similarity Component. It is a freely available Perl software package that makes it possible to measure the semantic similarity and relatedness between a pair of concepts [16]. It provides six measures of similarity and three measures of relatedness, all of which are based on the lexical database WordNet. One of the relatedness measures calculated is Adapted Lesk Algorithm, which is the measure used by AlgSim component in order to find the most related category to the event token.

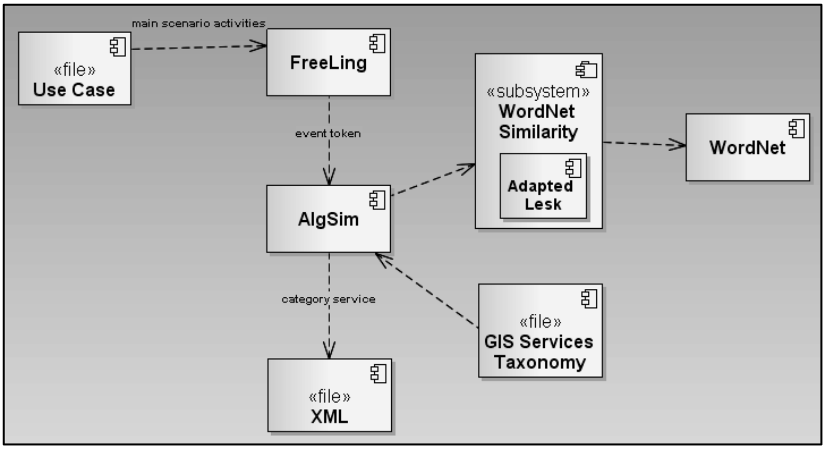


Fig. 2. ReqGIS component diagram

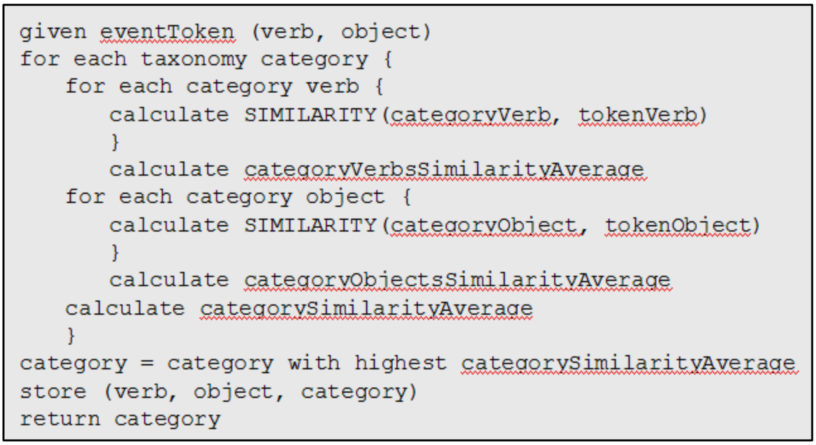
Adapted Lesk Algorithm. It is a module included in WordNet::Similarity package and it is based on Lesk algorithm which disambiguates words in short phrases. Adapted Lesk Algorithm [12] measures relatedness by evaluating words' relations in WordNet.

WordNet Component. WordNet is a large lexical database of English [2], arranged semantically. This package provides semantic information to WordNet::Similiarity component in order to compute similarity and relatedness measures.

AlgSim Component. This component, written in Perl, has been developed to achieve the main goal of classifying the required services. It implements the step D of our methodology, by taking as input the event token and processing it to obtain the required geographic category service. In order to perform this task, it performs an iterative process, shown in Figure 3, accessing information stored in the GIS Services Taxonomy and using services provided by WordNet::Similarity, AdaptedLeskAlgorithm and WordNet components. In fact, it calculates the average category relatedness for each category and selects the category with the highest relatedness. In order to calculate each category average relatedness, it computes verbs' relatedness and objects' relatedness, by evaluating pairs of verbs (category verb, event token verb) and objects (category object, event token object) within each category. After selecting the most suitable category the algorithm stores the result in an XML file.

4. Case Study

In this section we present a case study in order to show how our methodology and the classification tool work. The specification was provided by a local organization of Comodoro Rivadavia in Argentina. As it was in the Spanish language, we have translated it by considering our specification of use cases [18, 19].



```
given eventToken (verb, object)
for each taxonomy category {
    for each category verb {
        calculate SIMILARITY(categoryVerb, tokenVerb)
        }
        calculate categoryVerbsSimilarityAverage
    for each category object {
        calculate SIMILARITY(categoryObject, tokenObject)
        }
        calculate categoryObjectsSimilarityAverage
    calculate categorySimilarityAverage
    }
category = category with highest categorySimilarityAverage
store (verb, object, category)
return category
```

Fig. 3. Similarity algorithm to calculate highest relatedness

Table 2 shows a resultant use case in which a service to modify a coordinate of an electric line is presented.

Table 2. Fragment of Textual Use Case

Main Scenario	1	User selects electric line
	2	User modifies coordinate attribute
	3	System displays updated electric line

This use case is the input of the ReqGIS tool, which applies the four steps of our methodology (Figure 1) to each action defined in the main scenario of the use case specification.

Steps A-C are performed together by the *FreeLing* component. Considering the second action in the main scenario of the use case *"User modifies coordinate attribute"*, the component creates a parse tree classifying each word of the sentence. Figure 4 shows this tree. Then, ReqGIS creates the *Event Token* by finding main verbs and representative objects within each sentence of the parse tree. That is, the tool takes the root node in the parse tree tagged as **top** as the *event token main verb*, i.e. "modify", and the sentence direct object as the *event token representative object*, which is the word tagged as **dobj** in the parse tree, i.e. "attribute". So, the resultant event token is *"modify attribute"*.

Finally, in step D, the *AlgSim* component takes the event token as input, and after processing, it returns the name of the corresponding service category needed to accomplish our functional requirements. In addition the component also stores this result in an XML file.

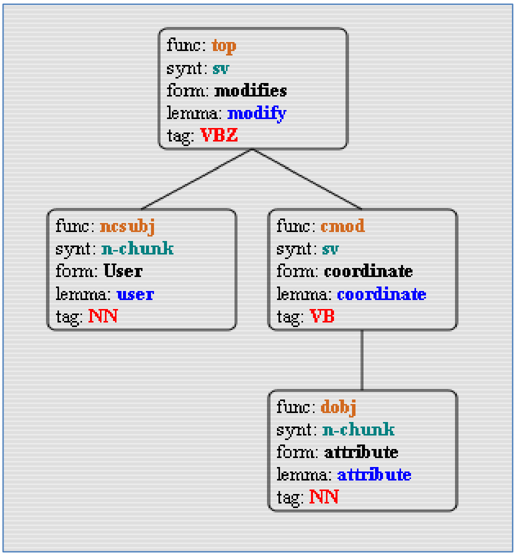
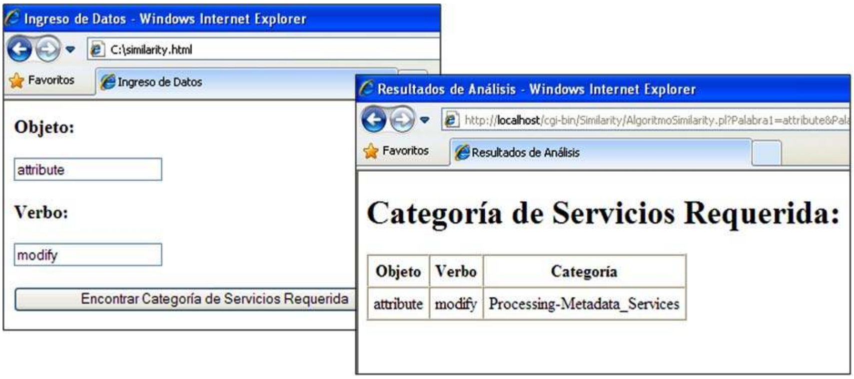


Fig. 4. Parse Tree for: “User modifies coordinate attribute”

AlgSim’s user interfaces are shown in Figure 5. In the first one, the user enters *event token’s main verb* and *event token’s representative object*. The second user interface shows the services category matching the event token, in this case, *Processing-Metadata Services* category.


Ingreso de Datos - Windows Internet Explorer
C:\similarity.html
Favoritos
Ingreso de Datos
Resultados de Análisis - Windows Internet Explorer
http://localhost/cgi-bin/Similarity/AlgoritmoSimilarity.pl?Palabra1=attribute&Pal
Favoritos
Resultados de Análisis

Objeto:

attribute

Verbo:

modify

Encontrar Categoría de Servicios Requerida

Categoría de Servicios Requerida:

Objeto	Verbo	Categoria
attribute	modify	Processing-Metadata_Services

Fig. 5. AlgSim user interfaces

In order to appreciate in more detail the process implemented by the AlgSim component, and in particular the relatedness measuring, we include a sample table (Table 3) with the scores values for each combination of Event Token Verb / Category Verb and Event Token Object / Category Object. For example to compute Category Verbs Average, the process calculates relatedness between each verb in Human Interaction category of the GIS Service Taxonomy (Table 1) and the "modify" verb (which is the Token Verb). For instance, for the first pair of verbs (interact, modify) we can see that the calculated relatedness is 158. The same process is applied to each pair of verbs and objects of the use case against the taxonomy.

These measures are taken for all categories in the GIS Services Taxonomy, calculating the average value for each category. The chosen category is the one having the highest average relatedness value. In our case study, the *AlgSim* component selects the *Processing-Metadata Services* category. The mediator service will have to map this category against the offered services to determine the components that provide them.

Table 3. Examples of relatedness measures between verbs and objects in category *Human Interaction* and event token "*modify attribute*"

Category	Category Verbs	Token Verb	Mea-sure	Category Verbs Average	Category Objects	Mea-sure	Token Object	Category Objects Average
Human Interaction	interact locate browse manage view display overlay query animate calculate edit	modify	158 30 23 12 95 70 30 33 12 122 213	72,54	Catalogue Metadata Feature Coverage Map Spreadsheet Service Chain Workflow View Perspective Texture Symbol Structure	35 8 127 8 76 13 38 31 9 64 64 60 98 179	attribute	54

5. Conclusion and Future Work

In this work we have shown our methodology for improving the demand model used by GIS developers. In particular, we have focused on the ReqGIS tool which has been implemented to make all the process automatically. The main goal is to improve the mechanisms to find required services in existing GIS components. We have analyzed several lexical analysis tools and we have developed a GIS classification tool, reusing and adapting some open source components. As future work, we will go on working on the combination with the methodology defined for publishing GIS services in order to complete the mapping between supply and demand models.

References

1. Armour, F., Miller, G.: Advanced Use Case Modeling Volume One, Software Systems. Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, (2001).
2. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A. (ed.) Computational Linguistics and Intelligent Text Processing. LNCS, vol. 2276, pp. 117-171. Springer, Heidelberg (2002).
3. Cechich A., Réquilé A., Aguirre J., Luzuriaga J.: Trends on COTS Component Identification. In: 5th IEEE International Conference on COTS-Based Software Systems, pp. 90--99. IEEE Computer Science Press, Orlando (2006).
4. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley Pub Co, (2001).
5. Freeling Home Page, http://garraf.epsevg.upc.es/freeling/
6. Gaetan, G., Cechich, A., Buccella, A.: Un Esquema de Clasificación Facetado para Publicación de Catálogos de Componentes SIG. In: XIV Congreso Argentino en Ciencias de la Computación. Chilecito, La Rioja, Argentina, (2008).
7. Gaetan, G., Cechich, A., Buccella, A.: Aplicación de Técnicas de Procesamiento de Lenguaje Natural y Web Semántica en la Publicación de Componentes para SIG. In: X Argentine Symposium on Software Engineering. Mar del Plata, Argentina (2009).
8. Gaetan G., Cechich A., Buccella A.: Extracción de Información a partir de Catálogos Web de Componentes para SIG. In: XV Congreso Argentino en Ciencias de la Computación, pp. 891-900 (2009).
9. Graham, I.: Object-Oriented Methods: Principles and Practice. Addison-Wesley (2000).
10. Kholkar, D., Krishna, G., Shrotri, U., and Venkatesh, R.: Visual Specification and Analysis of Use Cases. In: SoftVis'05: Proceedings of the 2005 ACM symposium on Software visualization, pp. 77-85. ACM, New York (2005).
11. Kulak, D., Guiney, E.: Use Cases: Requirements in Context. Addison-Wesley Longman Publishing Co. Inc., Boston (2003).
12. Lesk, M.: Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone From a Ice Cream Cone. In: 5th ACM International Conference on System Documentation, pp. 24-26. ACM, Toronto (1986).
13. OGC. Topic 12: OpenGIS Service Architecture. Open GIS Consortium (2002).
14. OMG. UML Superstructure Specification, v2.1.2. OMG Formal Document 2007-11-02 (2007).
15. Padró, L.; Collado, M.; Reese, S.; Lloberes, M.; Castellón, I. FreeLing 2.1: Five Years of Open-Source Language Processing Tools
16. Pedersen, T., Patwardhan, S., Michelizzi, J.: WordNet::Similarity- Measuring the Relatedness of Concepts. Demonstration Papers at HLT-NAACL 2004, pp. 38-41. Boston, Massachusetts (2004).

17. Saldaño, V., Buccella, A., Cechich, A.: Una Taxonomía de Servicios Geográficos para facilitar la identificación de componentes. In: XIV Congreso Argentino en Ciencias de la Computación. Chilecito, La Rioja, Argentina (2008).
18. Saldaño, V., Buccella, A., Cechich, A.: Descubrimiento de Servicios Geográficos a partir de Casos de Uso Textuales. In: XV Congreso Argentino en Ciencias de la Computación. Jujuy (2009).
19. Saldaño, V., Buccella, A., Cechich, A.: Discovering Geographic Services From Textual Use Cases, Journal of Computer Science & Technology, Vol. 10 - No. 2 - June 2010 - ISSN 1666-6038.
20. Spivey, J.: The Z Notation: A Reference Manual. Prentice Hall, 1992.
21. Szyperski, C.: Component Software-Beyond Object-Oriented Programming. Addison-Wesley, 1998.
22. Whittle, J., Jayaraman, P.: Generating Hierarchical State Machines from Use Case Charts. In: 14th IEEE International Requirements Engineering Conference (RE'06), pp. 16-25. Washington, DC, USA, IEEE Computer Society (2006).

Assessment Scheme-based Service Selection for SOC-based Applications[1]

MARTÍN GARRIGA[1,3], ANDRES FLORES[1,3], ALEJANDRA CECHICH[1],
ALEJANDRO ZUNINO[2,3]

[1] GIISCo Research Group, Facultad de Informática, Universidad Nacional del Comahue,
Neuquén, Argentina. [aflores,acechich]@uncoma.edu.ar,
[2] ISISTAN Research Institute, UNICEN,
Tandil, Argentina, azunino@isistan.unicen.edu.ar
[3] CONICET (National Scientific and Technical Research Council), Argentina.

Abstract. *Service-Oriented Computing promotes building applications by consuming reusable services. However, facing the selection of adequate services for a specific application still is a major challenge. Even with a reduced set of candidate services, the assessment effort could be overwhelming. On previous works we have presented a novel approach to assist developers on discovery, selection and integration of services, specially focusing in the selection method, which is based on a comprehensive scheme for services' interfaces compatibility. The scheme is also complemented by a framework based on black-box testing to verify compatibility on the expected behavior of a candidate service. This paper analyzes the selection method through a series of case studies, which are designed to show the scheme's potential on determining the best choice of a service among a set of candidates.*

Keywords: *Service oriented Computing, Component-based Software Eng-ineering, Web Services.*

1. Introduction

Service-Oriented Computing (SOC) promotes building distributed applications in heterogeneous environments [1]. Service-oriented applications are developed by reusing existing third-party components or services that are invoked through specialized protocols. The SOC paradigm has been widely adopted by using the Web Services technology [2], which leads to a concrete descentralization of bussiness processes and a low investment of new technologies and execution platforms. However, the efficient reuse of existing Web Services is still a major challenge. On one side, searching for candidate services on the Web implies a manual task yet, mainly exploring web catalogs usually showing poorly relevant information. On the other side, the result of a prosperous search requires skillful developers to deduce from

[1] This work is supported by projects: ANPCyT–PAE-PICT 2007-02312 and UNCo-IEUCSoft (04-E072

the set of candidates, the most appropriate service to be selected for the subsequent integration tasks. Even with a reduced set of services, the required assessment effort could be overwhelming. Not only functional and non-functional properties must be explored on candidates, but also the required adaptations for a correct integration allowing client applications to consume services while enabling loose coupling for maintainability.

In order to ease the development of SOC-based applications we presented on a previous work [3] a proposal for *discovery*, *selection* and *integration* of services, which is based on two recent approaches concerned on development and maintainability. The first approach, called EasySOC [4], provides specific semi-automated methods for both *discovery* and *integration* of services. The second approach, was initially developed as a solution for substitutability of component-based systems [5]. This approach supplies a method for *selection* of the most appropriate third-party candidate component. Since web services involve a special case of software component [6][7][8], few initial adjustments were required to apply this selection method on the context of service-oriented applications.

Particularly, this paper presents an extention of the *selection method* where a comprehensive scheme has been defined for assessing interfaces of candidate services according to requirements of internal components from a SOC-based application. The scheme allows to characterize the matchmaking process through a series of syntactic compatibility cases conveying not only the usual programming standards (e.g. names on operations and parameters), but mainly differentiating strong and potential similarity cases. The assessment process thus may produce an automatic identification of certain similarity cases to then giving the chance to improve the compatibility result by solving mismatch cases or better low equivalence results. The assessment scheme is also complemented by a framework based on black-box testing to explore the required behavior for candidate services, where the goal is to fulfill the observability testing metric [9] that identifies a component operational behavior by analyzing data transformations (input/output), which helps to understand the functional mapping performed by a component and therefore its behavior. Hence, a potential compatibility of a candidate service could be exposed – as we analyzed on a previous work [3] and was also discussed in [5][10].

Both approaches are supported by semi-automatic tools, named EasySOCPlugin and TestOOJ respectively that have been conveniently integrated, to support the whole new approach and validating the ideas proposed in this paper.

The paper is organized as follows. Section 2 presents an overview of the whole process for SOC-based application development. Section 3 gives details of the Assessment Scheme of the Selection Method. Section 4 presents a series of case studies. Conclusions and future work are presented afterwards.

2. Process for SOC-based Application Development

During development of a service-oriented application, a developer may decide to implement specific parts of a system in the form of in-house components. However, the decision could also involve the acquisition of third-party components, which in turn could be solved with the connection to web services. Figure 1 depicts our proposal intended to assist developers in the process of *discovery*, *selection* and *integration* of web services, which is briefly described as follows:

The first phase related to web services *discovery* is achieved by applying a combination of text mining and machine learning techniques. A simple input specification (in the form of a required interface I_R) is processed to form a specialized query sentence, and a search method, called WSQBE [4], returns a short list of candidate services through a mechanism for search space reduction. The second phase for services *selection* is described in more detail below according to Figure 2. The third phase related to Web services *integration* is based on an model intended to allow a client in-house component (C) being not strongly coupled to a service's interface (I_S) and also unaware of physical invocation aspects, e.g., interaction protocols, datatype formats, location, etc. Therefore, physical details for invoking services are deployed as a separate layer, through a service adapter (A_S) and a service proxy (P_S), which is placed in between to abstract client components (C) from changes on services' interfaces. Thus, client applications are able to operate with different interfaces by altering the intermediate layer, while the code implementing their in-house components remains untouched [4].

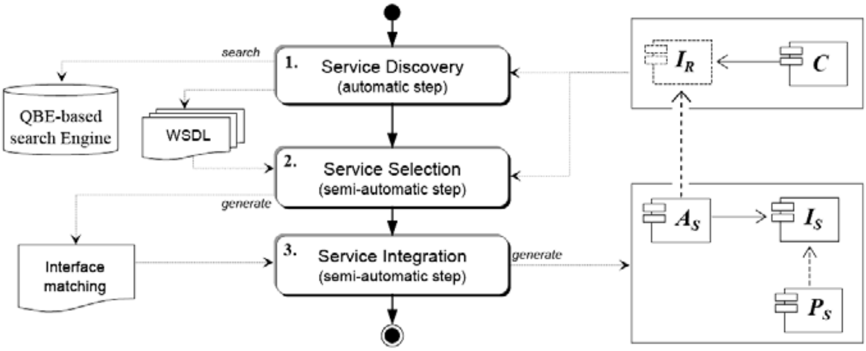
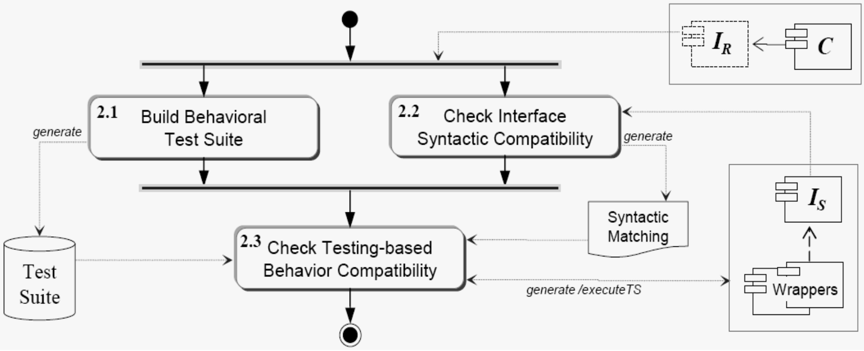


Fig. 1. Process for SOC-based Application Development

The *selection method* provides two main assessment procedures: an Interface Compatibility analysis and a Behavioral Compatibility evaluation, as shown in Figure 2. The Interface Compatibility evaluation is based on a comprehensive Assessment Scheme to recognize strong and potential matchings from a required interface (I_R) and the interface provided by candidate services (I_S). The outcome of this step is a Syntactic Matching List where each operation from I_R may have a correspondence with one or more operations from I_S [5]. Since this step is the main focus of this paper, details are given in Section 3.

The Behavioral Compatibility evaluation is intended to complement the previous assessment, where a Behavioral Test Suite (TS) is built to represent behavioral aspects from a third-party service, with required interface I_R. For this evaluation, the Syntactic Matching list produced in the previous step is processed, and a set of wrappers (adapters) is generated to allow executing the TS against the candidate service (through its provided interface I_S) to evaluate the achieved behavior compatibility [5].



I_R
C
2.1 Build Behavioral Test Suite
2.2 Check Interface Syntactic Compatibility
generate
generate
I_S
Syntactic Matching
Test Suite
2.3 Check Testing-based Behavior Compatibility
generate /executeTS
Wrappers

Fig. 2. Selection Method

Next sections provide detailed information particularly related to the Interface Compatibility step. A case study will be used to illustrate the usefulness of the Assessment Scheme into the Selection Method.

2.1 Case Study

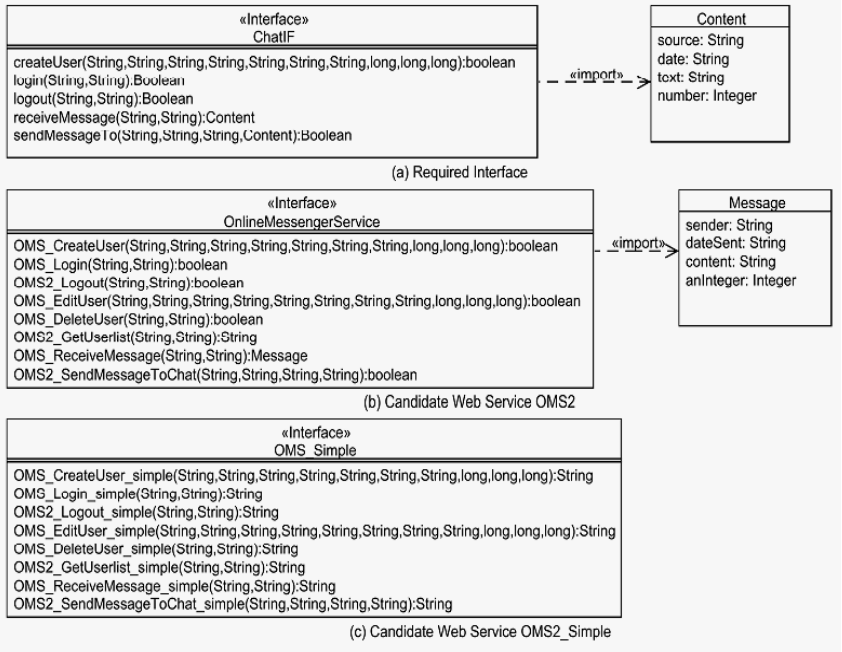
Let us suppose the development of a communication tool for exchanging instant messages with contacts from a user's contact list. We have specified the behavior of the required service in the form of operations defined into a Java interface I_R, named `ChatIF`. Figure 2(a) shows the required interface `ChatIF`, which includes a complex type named `Content`.

By running the first phase of the process, a set of web services called OMS (Online Messenger Service) has been discovered at *http://www.nims.nl/*. Particularly we are interested on two of those services: `OMS2` and `OMS2_Simple`. The former (*http://www.nims.nl/soap/oms2.wsdl*) provides an interface I_{S1} comprised of 38 operations, and the most relevant ones are shown in Figure 2(b), where another complex type named `Message` is used for enclosing the contents to be exchanged. The latter (*http://www.nims.nl/soap/oms2_simple.wsdl*), whose interface I_{S2} is shown in

Figure 2(c), uses the `String` type for the operations return, instead of any other type (built-in or complex).

3. Interface Compatibility Analysis

The Selection Method corresponds to the second phase of the whole process for SOC-based application development. Two main evaluations are applied on candidate services, from which a concrete recommendation concerning the most appropriate service is achieved. The final evaluation procedure (*step 2.3*) takes the set of candidate services to be put under test with the purpose to discover compatibility with respect to the expected behavior for the client application. Nevertheless, such final evaluation requires a previous assessment at a syntactic level on Interface Compatibility (*step 2.2*), which may provide useful preliminary information to help developers gain knowledge on several aspects.


«Interface»
ChatIF
createUser(String,String,String,String,String,String,String,long,long,long):boolean
login(String,String):Boolean
logout(String,String):Boolean
receiveMessage(String,String):Content
sendMessageTo(String,String,String,Content):Boolean
«import»
Content
source: String
date: String
text: String
number: Integer
«Interface»
OnlineMessengerService
OMS_CreateUser(String,String,String,String,String,String,String,long,long,long):boolean
OMS_Login(String,String):boolean
OMS2_Logout(String,String):boolean
OMS_EditUser(String,String,String,String,String,String,String,String,long,long,long):boolean
OMS_DeleteUser(String,String):boolean
OMS2_GetUserlist(String,String):String
OMS_ReceiveMessage(String,String):Message
OMS2_SendMessageToChat(String,String,String,String):boolean
«import»
Message
sender: String
dateSent: String
content: String
anInteger: Integer
«Interface»
OMS_Simple
OMS_CreateUser_simple(String,String,String,String,String,String,String,long,long,long):String
OMS_Login_simple(String,String):String
OMS2_Logout_simple(String,String):String
OMS_EditUser_simple(String,String,String,String,String,String,String,String,long,long,long):String
OMS_DeleteUser_simple(String,String):String
OMS2_GetUserlist_simple(String,String):String
OMS_ReceiveMessage_simple(String,String):String
OMS2_SendMessageToChat_simple(String,String,String,String):String

(a) Required Interface

(b) Candidate Web Service OMS2

(c) Candidate Web Service OMS2_Simple

Fig. 3. Instant Messenger Application-Chat

Particularly, the Interface Compatibility analysis is comprised of a practical scheme of two parts: automatic matching cases and semi-automatic potential matchings, to analyze operations from the interface I_S (of a candidate service S), respect to the required interface I_R.The outcome of this step may avoid early discarding a candidate service upon simple mismatches but also preventing from a serious incompatibility. In addition, helpful information about the adaptation effort of a candidate service may take shape for a positive integration into the consumer application.

3.1. Assessment Scheme

Table 1 presents the Assessment Scheme that is comprised of four levels to define different syntactic constraints for a pair of corresponding operations. Constraints are based on individual conditions, summarized in Table 2, according to the elements of an operation' signature (return, name, parameter, exception). Types on operations from I_S should have at least as much precision as types on I_R. However, the String type is a special case, being considered as a *wildcard* type since it is generally used in practice to allocate different kinds of data. *Parameters* (`P`) and *return type* (`R`) are the most significant signature elements of the scheme. To consider an initial strong compatibility result, a criterion of "*no inclusion*" has been defined for conditions `R3` and `P4` that are evaluated in the Automatic part of the scheme as incompatibilities (treated as conditions `R0` and `P0` respectively). Therefore, those weakest compatibility cases (`R3` and `P4`) are managed under the Semi-Automatic part of the scheme – e.g., operation `sendMessageTo` of `ChatIF` in Table 1.

Table 1. Assessment Scheme: Automatic Match and Semi-Automatic Mismatch Solving

Level	Part	Constraints
🟩 Exact Match	Auto (1 case)	Two operations must have identical signatures. (four identical conditions): [`R1`,`N1`,`P1`,`E1`]
🟨 Near Exact Match	Auto (13 cases)	Three or two identical conditions. The remaining might be second conditions: (`R2`/`N2`/`P2`/`E2`). Exceptional cases: three identical conditions with a remaining third condition (`N3`/`P3`/`E3`)
	Example: operation `logout` of `ChatIF` has *near-exact_2* match to `OMS2_Logout` of `OMS2` with a substring equivalence for the operation name ("logout"): [`R1`,`N2`,`P1`,`E1`]	
	Semi-A (1 case)	Three identical conditions with the return that may have a no equivalent complex type or lost precision: [`R3`,`N1`,`P1`,`E1`]
🟧 Soft Match	Auto (26 cases)	Similar to the previous level, but only two identical conditions. Previous exceptional cases may occur with lower equivalence conditions.
	Semi-A (13 cases)	Two identical conditions, similar to automatic scheme. Either return or parameter (not both) with a nonequivalent complex type or lost precision (`R3`/`P4`).

	Example: operation `sendMessageTo` of `ChatIF` could match operation `OMS2_SendMessageToChat`. However, the first operation includes a parame-ter of complex type (`Content`) without a match into the other operation that has only String parameters (initially evaluated as `P0`). This can be re-evaluated considering that the wildcard type String might contain a chain of all fields from the complex type – i.e. an equivalence *soft_25*: [`R1`,`N2`,`P4`,`E1`].	
■ Near Soft Match	Auto (14 cases)	There cannot be two identical conditions, i.e. all conditions can be relaxed simultaneously.
	Semi-A (40 cases)	Either two identical conditions with the condition P4 or relaxing all conditions simultaneously.

The Assessment Scheme in Table 1 is able to recognize 108 cases for Interface Compatibility (where each part is comprised of 54 cases), from the combination of individual conditions (classified into the four levels of compatibility).

For complex data types their comprising fields must be equivalent one-to-one with fields from a complex type counterpart. For example, `receiveNextMessage` of `ChatIF` and `OMS_ReceiveMessage` of `OMS2` have a complex type as a return (`Content` and `Message` respectively), which are equivalent (`R2`) because their fields are equivalent one-to-one. Thus, these operations have a *near_exact_12* match, since they also coincide on parameters and exceptions (`P1`,`E1`); with a substring equivalence on their names (`N2`) – common words "*receive*" and "*message*".

When certain mismatch cases are detected for the interface I_R, a developer may outline a likely solution with the support of context information from the application's business domain. We have identified specific cases in which a concrete compatibility can be set up providing a semi-automatic mechanism to ease this procedure. An example is given in Table 1 with the operation `sendMessageTo` of `ChatIF`.

Table 2. Syntactic Operation Matching Conditions for Interface Compatibility

urn Typ	R0: Not compatible	R1: Equal return type
	R2: Equivalent return type (subtyping, Strings or Complex types)	R3: Not equivalent complex types or lost precision
Name		N1: Equal operation name
	N2: Equivalent operation name (substring)	N3: Operation name ignored
Parameters	P0: Not Compatible	P1: Equal amount, type and order for parameters
	P2: Equal amount and type for parameters	P3: Equal amount and type at least equivalent (including subtyping, Strings or Complex types) for some parameters into the list
	P4: Not equivalent complex types or lost precision	
Excep-tions	E0: Not compatible	E1: Equal amount, type, and order for exceptions
	E2: Equal amount and type for exceptions into the list.	E3: If non-empty original's exception list, then non-empty candidate's list (no matter the type).

The second part of the Scheme is not only intended to assist on solving mismatch cases, but also to allow a developer to "force" certain correspondences even when an automatic match was identified. For a specific operation $op_R \in I_R$, there could be another correspondence that better fit for the application's context. The developer is enabled to make such prioritization, which then is considered in first order for the processing on the Selection Method's subsequent step (see Section 2).

The final outcome of the Interface Compatibility step is a matching list characterizing each correspondence according to the four levels of the Assessment Scheme, named *Interface Matching List*. For each operation $op_R \in I_R$, a list of compatible operations from I_S is shaped. For example, let be I_R with three operations op_{Ri}, $1 \leq i \leq 3$, and I_S with five operations op_{Sj}, $1 \leq j \leq 5$. The matching list might result as follows: $\{ (op_{R1}, \{op_{S1}, op_{S5}\}), (op_{R2}, \{op_{S2}, op_{S4}\}), (op_{R3}, \{op_{S3}\}) \}$.

Each compatibility case represents a specific numeric value in the Assesment Scheme. For example, the value of *exact* equivalence is 4. Therefore, a totalized value could be determined to synthetize the *degree* of Interface Compatibility between a required interface I_R and a candidate interface I_S (from a service S). Only the higher compatibility level for each operation is considered to calculate that value, named *Syntactic Distance*. The corresponding formula is shown in (1).

$$syntDist(I_R,I_S)= \frac{\sum_{i=1}^{N} Min(op_{Ri},MapComp(I_R,I_S))}{N * 4} - 1 \qquad \textbf{(1)}$$

where N is the interface's size of I_R, and *MapComp* are the values for the compatibility cases found for operation op_{Ri}.

If all operations in the *Interface Matching List* presents an *exact* equivalence, the *Syntactic Distance* between I_R and I_S is zero. This iniatially means that I_R is included into I_S, though I_S may have additional operations.

The success on the precision achieved during the Interface Compatibility step is essential to reduce the computation effort for the subsequent step of behavior evaluation (see Section 2). This is the main reason for the definition of the whole Assessment Scheme, in which different design and programming heuristics have been applied, mostly from a practical experience perspective.

4. Case Studies

This section shows the evaluation's results for the example presented in Section 2.1. Then another case study is briefly described.

4.1 Instant Messenger – Chat

Table 3 shows the automatic matching results for `ChatIF` and service `OMS2`, where a mismatch is identified for operation `sendMessageTo` of `ChatIF` (depicted with a gray cell) for which a semi-automatic solution could be set up by a *soft_25* (`R1,N2,P4,E1`) match to operation `OMS2_SendMessageToChat` of `OMS2`. The rest of the `ChatIF` interface has found a match. For example operation `createUser` has a *near-exact_2* match to operation `OMS_CreateUser` (due to the substring equivalence). Operations `login` and `logout` obtained similar result by a *near-exact_2* match to alike operations, and four *near-exact_7* matches to other operations.

Table 3. Automatic Interface Compatibility between `ChatIF` and `OMS2`

ChatIF	OMS2
boolean createUser(String, String,String,String,String, String,String,long,long,long)	[n_exact_2, boolean OMS_CreateUser (String, String, String,String,String,String,String,long,long,long), R1, N2, P1, E1]
boolean sendMessageTo (String,String, String,Content)	
Content receiveNextMessage (String, String)	[n_exact_12,Message OMS_ReceiveMessage (String, String,), R2, N2, P1, E1]
boolean logout(String, String)	[n_exact_2,boolean OMS2_Logout(String,String), R1, N2, P1, E1]
boolean login(String, String)	[n_exact_2, boolean OMS_Login(String, String), R1, N2, P1, E1]

As no automatic matching has been found for `ChatIF` and `OMS2Simple`, the mismatches have been solved in the semi-automatic step, by the notion of the String type as a *wildcard* type (see Section 3.1).

At this point, the *Interface Matching List* for both candidate services is available. Thus, the *syntactic distance* could be used to determine which of them is better to continue with the Behavioral Compatibility (*step 2.3*). Table 5 summarizes the best values found for each candidate service and each operation in `ChatIF`.

The *syntactic distance* between `ChatIF` and `OMS2` is 29/20-1 = 0,45 according to formula (1), and considering `OMS2_Simple` the *syntactic distance* is 40/20-1 = 1. Because the lower value is better, the suggested candidate service is `OMS2`.

Table 4. Interface Compatibility Summary for `ChatIF`, `OMS2` and `OMS2Simple`

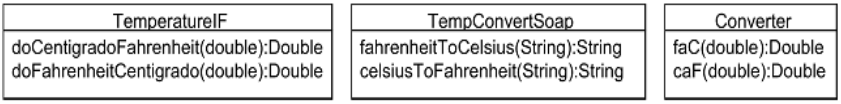
ChatIF Operations	OMS2 Best Value*	OMS2_Simple Best Value*
createUser	5	6
sendMessageTo	8	11
receiveNextMessage	6	7
logout	5	8
login	5	8
Total	29	40
Syntactic Distance	0,45	1

* Total Best Value 20 (based on `ChatIF` size)

4.2 Weather System

This case study is a system in which it is required to provide temperature information on both Celsius and Fahrenheit scales. A required interface I_R has been defined in the Java format, named `TemperatureIF`, which is shown in Figure 4(a). Candidate web services are named `TempConvert`[2] and `Converter`[3], whose interfaces I_{S1} and I_{S2} are shown in Figure 4(b) and 4(c) respectively.

The automatic Interface Matching between `TemperatureIF` and service `TempConvert`, reveals that all operations from `TemperatureIF` have found a match. Both operations from `TemperatureIF` obtained similar result by two matches to both operations of `TempConvert` service. The `String` type recognized as a *wildcard* type allows to have an equivalence on types for return and parameters (`R2`,`P3`).


TemperatureIF
doCentigradoFahrenheit(double):Double
doFahrenheitCentigrado(double):Double
TempConvertSoap
fahrenheitToCelsius(String):String
celsiusToFahrenheit(String):String
Converter
faC(double):Double
caF(double):Double

(a) Required Interface (b) Candidate Service (c) Candidate Service

Fig. 4. Weather System

After the automatic Interface Matching for `TemperatureIF` and `Converter`, the syntactic distance between `TemperatureIF` and both candidate services is calculated, as shown in Table 5, being 1 for `TempConvert` and 0,5 for `Converter`. Thus, the suggested candidate for the next step of Behavioral Compatibility is the `Converter` service.

Table 5. Interface Compatibility Summary for `TemperatureIF` and the candidates

Operations of TemperatureIF	**TempConvert Best Value***	**Converter Best Value***
doFarenheitCentigrado	8	6
doCentigradoFarenheit	8	6
Total	16	12
Syntactic Distance	1	0,5

*Total Best Value 8 (based on `TempConvert` size)

These case studies show how a developer may gain specific and valuable knowledge about an application's context by the support of the Assessment Scheme. For each likely equivalence case automatically identified, there is a clear rationale that is also reinforced by the characterization within the four levels of compatibility. In addition, different scenarios of compatibility upon low levels may be analyzed by setting up other correspondences with the

[2] *http://www.w3schools.com/webservices/tempconvert.asmx?WSDL*

[3] *http://www.elguille.info/Net/WebServices/CelsiusFahrenheit.asmx?WSDL*

semi-automatic assistance based on the second part of the scheme. In this way, a certain web service may be saved from being early discarded as a potential candidate, but also a concrete validation is given for any change on correspondences, which become very helpful for a developer to understand the required adaptation effort to achieve the service integration.

5. Conclusions and Future Work

In this paper we have presented details of a Selection Method which allows evaluating a candidate web service for its likely integration into a SOC-based application under development. This method is part of a larger process for discovery and integration of services, and provides a practical Assessment Scheme for Interface Compatibility where a synthesis of design and programming heuristics have been added, both to improve possibilities to identify potential matchings, but also to help developers to gain knowledge on the application's context for a candidate service. The syntactic distance metric provides a measurable value to mathematically support the candidate selection. Additionally, such selection might consider other aspects like *Quality of Service* parameters – e.g., performance, security, and so on.

The whole process of discovery, selection and integration has a fully support to achieve efficiency and reliability. Our current work is focused on exploring Information Retrieval techniques to better analyzing concepts from interfaces, which has been initially applied on the EasySOC approach. Another concern implies the composition of candidate services to fulfill functionality, which is particularly useful when a single candidate service cannot provide the whole required functionality. We will expand the current procedures and models mainly based on business process descriptions and service orchestration [11], [12].

References

1. Erickson, J., Siau, K.: Web service, service-oriented computing, and service-oriented architecture: Separating hype from reality. Journal of BD Management, 19(3), 42-54 (2008).
2. Bichler, M., Lin, K.: Service-oriented computing. Computer, 39(3), 99-101 (2006).
3. Flores, A., Cechich, A., Zunino, A., Polo, M.: Testing-Based Selection Method for Integrability on Service-Oriented Applications. In: 5th IEEE ICSEA'10. pp. 373-379 (2010).
4. Crasso, M., Mateos, C., Zunino, A., Campo, M.: EasySOC: Making Web Service Outsourcing Easier. Information Sciences, Elsevier (2010).
5. Flores, A., Polo, M.: Testing-based Process for Component Substitutability. Software Testing, Verification and Reliability, p. 33 (2010), [early view press]
6. Stuckenholz, A.: Component Evolution and Versioning State of the Art. ACM SIGSOFT Software Engineering Notes, 30(1), 7-20 (2005).

7. Canfora, G., Di Penta, M.: Testing Services and Service-Centric Systems: Challenges and Opportunities. IT Professional, 8(2), 10-17 (Mar/Apr 2006).
8. Kung-Kiu, L., Zheng, W.: Software Component Models. IEEE Transactions on Software Engineering, 33(10), 709-724 (2007).
9. Jaffar-Ur Rehman,M. et.al.: Testing Software Components for Integration: a Survey of Issues and Techniques. Software Test., Ver., Reliab., 17(2), 95-133 (2007).
10. Alexander, R., Blackburn, M.: Component Assessment Using Specification-Based Analysis and Testing. Tech. Rep. SPC-98095-CMC, Software Productivity Consortium, USA (1999).
11. C. Peltz, Web Services Orchestration and Choreography. IEEE Computer, 36(10), 46-52, (2003).
12. Weerawarana, S.; et al., Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WSReliable Messaging, and More. Prentice Hall PTR, (2005).

Adaptation of ISO/IEC 15504 Standard to SMBs Needs. Analysis of Required Roles to Meet Work Product Requirements

ARIEL PASINI[1], JOSÉ RAMÓN ZUBIZARRETA AIZPURU[2], PATRICIA PESADO[3]

III-LIDI- Instituto de Investigación en Informática LIDI - School of Computer Science -
National University of La Plata - La Plata - Buenos Aires - Argentina.
LSI - Departamento de Lenguajes y Sistemas Informáticos - School of Computer Science -
University of the Basque Country - San Sebastián - Basque Country - Spain
{apasini,ppesado}@lidi.info.unlp.edu.ar
joseramon.zubizarreta@ehu.es

Abstract. *An analysis is presented on the possibility of deploying the ISO/IEC 15504 standard in the various SMBs categories defined by the European Union, in particular, on the roles that would be necessary for a company and the roles that each of the SMBs categories can support. Scenarios are provided, defining a set of roles for each category, and the possibility is analyzed of meeting Work Product requirements that are part of the Engineering and Project processes, as well as the attributes of capacity levels 2 and 3.*

Keywords: *Process Models, SMBs, ISO/IEC 15504, Roles, Work Product.*

1. Introduction

Software Engineering has progressed in time with a clear tendency towards the maturity of development processes, with the purpose of considering all tasks related to software development as a process that can be controlled, measured and improved. Process-orientation in software development has achieved great popularity throughout the world through the publication of the already recognized quality standards, currently led by SEI (Software Engineering Institute) and ISO (International Standard Organization).

Software developers know in detail the factors and issues that affect their work, but without a strategy to improve, achieving the visibility of the

[1] Full-Time Associate Professor, School of Computer Science, UNLP.

[2] University Head Professor, School of Computer Science, UPV/EHU.

[3] Full-Time Head Professor, School of Computer Science, UNLP – CIC Bs. As.

efforts aimed at improving is difficult. Thus, an improvement plan that can lead the organization towards a continuous improvement is essential [1].

There are various models that propose different methods for assessing process capacity, different ways of representing the activities needed to improve, and different ways of leading the organization into maturity. However, the application of these models is oriented to large organizations. They do not consider the needs of SMBs, where their application is costly in terms of economy and effort, since a significant, long-term investment of capital, time and resources is required [2].

The objective of this work is to analyze the needs of SMBs that attempt to deploy the ISO/IEC 15504 model as improvement process; in particular, the required roles for meeting the requirements defined by work products. Even though the model does not require a specific assignment of roles to carry out the activities, any organization that is on its way to deploying an improvement process is likely to have a structure that can comply with practices. This structure is directly linked to the size of the organization – in the case of a large development company, there will probably be a range of roles to fulfill the activities requested; but in the case of micro, small and medium businesses, this range of roles is very reduced.

The analysis of role assignment to carry out these activities in different types of organizations and the feasibility of meeting the requirements established for each activity is of interest.

The scope of this work is limited in the case of Capacity Levels, to levels 2 and 3, and in the case of processes, to the areas of Engineering (ENG) and Project (PRO). This limitation is based on the fact that a software development SMBs will certainly have a structure dedicated to project development and management. In Section 2, a brief description of standard ISO/IEC 15504 is included, as well as the economic context in which SMBs work, the impact of software development SMBs, the need to deploy improvement plans, and the complexities they must face. In the next section, the practices proposed by the model are described, together with their corresponding work products as proof of their execution, the assignment of roles within the organization, and the possible adaptations that can be done depending on business structure. Finally, the conclusions of the analysis carried out are presented.

2. ISO/IEC 15504 and SMBs

2.1 ISO/IEC 15504

The ISO/IEC 15504 standard defines a framework for process assessment and improvement that can be used by organizations to plan, manage, monitor, control, and improve software acquisition, development and operation, assessment, and support [3].

This process assessment framework is based on the evaluation of one instance of a specific process on which information can be collected. Each process instance is characterized by a set of maturity level valuations. Practice valuation is an assessment of the level of compliance of the practice with the roles defined by the standard [4].

The standard defines, in an abstract way, the fundamental activities that are essential for the "good practice" of software engineering. These activities are grouped into processes, and processes are distributed in five categories**:** **Customer-Supplier** (CUS), **Engineering** (ENG), **Project** (PRO), **Support** (SUP), **Organization** (ORG). Each process in the model is described in terms of the base practices; these tasks or activities characterize the execution of a process. Activities carried out will be reflected through work products.

The evolution of process capacity is expressed in terms of maturity levels. Each level provides a significant improvement to the performance capacity of the process, compared to the previous level. The model has six levels: **0 -** *Incomplete,* **1 -** *Performed,* **2 -** *Managed,* **3 -** *Established,* **4 -** *Predictable,* **5 -** *Optimizing.*

2.2 SMBs

SMBs (Small and Medium Businesses) is a set of businesses whose dimensions are limited regarding number of employees and business volume. These restrictions are imposed by the State or the Region where these businesses operate. SMBs have an important role in the economy of every country, since they generate a high percentage of jobs.

The greatest advantage of a SMB is its ability to quickly change its production structure if market needs vary. On the flip side, due to their higher risk, it is harder for them to find financing at a suitable cost and term, and it is complicated for them to access international markets [5].

Since the analysis was carried out in collaboration between a University in Argentina and a University in Spain, a decision had to be made in relation to the definition of SMBs that would be used. It was agreed to use that of the European Community [6] because it distinguishes categories based on the number of employees and the amounts invoiced by the businesses (see Table 4), as compared with the Argentinean classification [5] that uses only the amounts invoiced.

	Micro	Small	Medium
Employees	Less than 10	Between 11 and 50	Between 51 and 250
Maximum Billing EU (EUR)	2 Million	10 Million	50 Million
Maximum Billing Argentina (ARS)	590,000	4,300,000	28,300,000

Table 1. SMBs in the EU and Argentina

SMBs have a very significant specific gravity in software development. However, this type of businesses in general does not use any explicit development methodology, and therefore they often suffer delays in their deliveries, exceed their costs, fail to comply with their commitments, etc. Consequently, the application of process improvement models becomes a complex task for this type of businesses.

The application of an improvement model in a SMB carries some problems, such as: high auditing and certification costs, costs in human resources dedicated to the improvement process, changes in work philosophy, etc. In the specific case of the organization of small or micro businesses, the following problems also appear: excessive documentation of software development and organization processes, planning, organization, and management of key processes areas oriented to large organizations, limited resources, high personnel training costs, lack of guidance on project needs and the development team [7,8].

As a consequence, the economic costs associated with these models can be a serious obstacle for their adoption

3. Analysis Carried Out

As mentioned in the previous section, one of the main obstacles for the deployment of the model in SMBs are the human resources needed to meet all the requirements established by the practices. In order to gain a deeper understanding of these requirements, the work products suggested by the standard as indicators of processes and evidence of their performance were studied in detail [9].

3.1 Work Product

Work Products (WP) are documents that include a set of characteristics that should be reflected on process input and output documentation. These documents help the assessor determine if the information recorded about the process is appropriate. One of the characteristics of these documents is that they help keep a comprehensive control of the person carrying out the practice and recording the corresponding information, but the model makes no reference to the roles that are responsible for performing the task.

The structure presented in the WP standard is informative, since organizations could have the information requested in one or several documents, with different structures, but which, as a whole, meet the characteristics requested for the WP at hand.

WPs are one of the main sources of evidence for the assessment. The full definition of WP, their set of characteristics, and their relation among processes can be found in the annexes to Part 5 of the standard [9].

For this work, of the 109 WPs of the model, those necessary for capacity levels 2 and 3 and the ENG and PRO processes were analyzed.

3.2 Roles in SMBs

Large software organizations organize their employees based on their set of responsibilities, which is called Role. There is no standard definition for role; roles are generally defined by the needs of the businesses. However, there is agreement in software-related business management literature that in large organizations, there are certain responsibilities that must be covered by a specific role. Table 2 shows a role structure for a software development business.

SMBs must cover the same responsibilities, but their composition widely differs from that of software development businesses. The proposed role structure cannot be covered by the employees an SMB has, even considering that one employee may have different, simultaneous roles within the organization if there are no conflicts between his/her responsibilities.

AB	Role
CIO	Chief Information Officer (CIO)
PL	Project Leader
ARCH	Architect
FA	Functional Analyst
TA	Technical Analyst
P	Programmer Analyst
T	Tester Analyst
QA	Quality Assurance Analyst
CA	Configuration Administrator
TEA	Testing Environment Analyst
DBA	Data Base Analyst
OP	Operator
HRM	Human Resources Manager
HDA	Help Desk Analyst

Table 2. Roles of a software development organization

Role		Level 2	Level 3
CIO	9	2	0
PL	24	23	18
ARCH	12	3	3
FA	13	4	0
TA	4	1	0
P	1	0	0
T	4	0	0
QA	14	5	5
CA	16	4	2
TEA	2	0	0
DBA	1	0	0
OP	1	0	0
HRM	4	2	1
HDA	4	1	0
Total WP	109	45	29

Table 3. WPs assigned to each role

In order to see the relationship between roles and WPs, the roles from Table 2 are assigned to the 109 WPs of the standard. The assignment was done based on the definition of the role and the characteristics of the WP. Table 3 shows the results of the distribution and allows identifying the set of WPs for which a role is responsible. For instance, the PL is responsible for 24 WPs in the entire model, 23 WPs corresponding to level 2, and 18 from level 3.

If we try to analyze the deployment of this model in SMBs, being able to meet WP requirements with a very reduced staff is very complicated. Taking the different SMB categories defined in Section 2.2 as a starting point, *micro*, *small* and *medium*, a set of possible scenarios based on these categories are presented.

A *micro* software development business has an internal structure with a maximum of 10 employees. Their composition will be at least a team of *developers*, responsible for all programming-related tasks, decisions on the codification used, support and getting the project to the production stage. It requires some *functional analyst*, responsible for the analysis of requirements, the functional analysis and the analysis of project changes feasibility. A *project leader*, who is in charge of one or more simultaneous projects and who knows how to manage each of them in particular, analyze the possible reutilization of components, manage configuration, and determine the assignment of resources based on the priorities of each project. Finally, it requires a *business manager*, who must have extensive knowledge both in the business area to ensure the subsistence of the business and also in the technical area to be aware of what the business can and cannot produce from a technical perspective. This business manager will be responsible for ensuring the quality of the product or service provided.

With a structure of up to 50 employees, a *small* business differs from a *micro* business in the incorporation of the role of *configuration management* and the detachment of developers from the responsibilities of production start-up and project leaders from the activities related to configuration management. Also, the vision of an *architect* can be added, detaching the project leader from the reusability analysis and allowing him/her to focus on the goal of leading the management of the project itself.

In a *medium* business, the size of the staff grows considerably, and therefore, it is possible to include roles specifically determined to ensure the quality of the product or service, freeing *project leaders* from the responsibility of managing and controlling metrics and *business management* from the responsibility of managing quality standards and policies. In this type of businesses, the role of the *business manager* is more oriented to the business sector rather than the technical aspects because there is a strong internal structure that supports technical feasibilities.

Table 4 shows a role structure that would be achievable for SMBs

Ab	Role	Micro	Small	Medium
M	Business Management	X	X	X
PL	Project Leader	X	X	X
FA	Functional Analyst	X	X	X
D	Developers	X	X	X
ARCH	Architect		X	X
CA	Configuration Management		X	X
PM	Process and QA Manager			X

Table 4. Role structure based on the type of SMBs

The analysis was carried out with a double-entry matrix or table, with WPs on the vertical axis and, on the upper horizontal axis, first the attributes corresponding to levels 2 and 3, and then the processes from the ENG and PRO categories. At the center of the matrix, required WPs were identified, and then each of them was assigned a role according to the type of business and the relationship between the characteristics of the WP and those of the role.

3.2.1 Analysis by Capacity Levels

Initially, the roles were assigned to the WPs required for levels 2 and 3. The assignment was carried out based on the role structure of the SMB, starting from the same principle of relationships between the responsibilities of the role and the characteristics of the WP.

Table 5 shows a quantitative analysis of the result of the assignment of roles to the WP carried out for level 2, Managed. As it can be seen, in *micro* and *small* businesses, most of the responsibilities correspond to the *project leader* and *business management*. This agrees with the model, since at this level, it is only corroborated that the practices are planned, controlled and verified as established. In the case of *medium* businesses, responsibilities are more equally distributed.

Role	Level 2			Level 3		
	Micro	Small	Med.	Micro	Small	Med.
Business Management	12	12	6	10	9	1
Project Leader	25	21	15	17	15	11
Functional Analyst	4	4	4	2	0	0
Architect	0	2	2	0	3	3
Developers	4	1	0	0	0	0
Configuration Management	0	5	5	0	2	2
Process and QA Manager	0	0	13	0	0	12
Total	**45**	**45**	**45**	**29**	**29**	**29**

Table 5. Level 2 and 3 WPs assigned to the roles of a SMB

In the case of level 3, Performed, again, in *micro* and *small* businesses the greatest responsibility lies on *business management* and the *project leader*, but in the case of *medium* businesses, it is the *Process and QA Manager* who carries most of the responsibility, which agrees with the purpose of the level of having practices based on well-defined processes.

The assignment of roles in both levels tends to find a balance as the size of the business grows and the business changes categories, mainly removing a work overload from the *project leader* (in a *micro* business, the PL is responsible for 17 WPs and in a *medium* business, for 11 WPs) and *business management*, which are, in the case of *micro* businesses, those leading the organization. Once the business reaches a *medium* size, it deals only with its own management.

3.2.2 Analysis by Process Maturity

The role assignment process was repeated for the ENG and PRO categories, with the same criteria described in the previous section, and the corresponding quantitative analysis was carried out. Table 6 shows the results for the ENG category. It was observed that the load assigned to the role of *developer* remains stable in *micro*, *small* and *medium* businesses. This category is responsible for development activities, the core of a software development business, and must be present from the minimum structure of any software development business. An overload of the *functional analyst* role can also be observed in the case of *micro* businesses, which becomes more stable in *small* and *medium* business due to the addition of the role of *architect*.

In the case of the PRO category, it can be seen that the most significant changes occur going from the *small* to the *medium* business, due to the addition of the *process and QA manager*.

Role	**ENG Category**			**PRO Category**		
	Micro	**Small**	**Med.**	**Micro**	**Small**	**Med.**
Business Management	1	1	1	22	22	6
Project Leader	6	2	2	22	21	16
Functional Analyst	12	8	7	9	5	4
Architect	0	3	3	0	4	4
Developers	4	4	4	2	1	0
Configuration Management	0	5	5	0	2	2
Process and QA Manager	0	0	1	0	0	23
Total	**23**	**23**	**23**	**55**	**55**	**55**

Table 6. WPs corresponding to categories ENG and PRO assigned to the roles of a SMB

A difference was observed between the WPs corresponding to the ENG category and those analyzed at capacity levels 2 and 3. This is because the ENG category tries to corroborate that the process was carried out, and Level 2 assumes that the process has already been done and tries to corroborate its management. In the case of the WPs from the PRO category, they matched those of levels 2 and 3 for the most part.

4. Conclusions

In this paper, a description of the ISO/IEC 15504 standard was provided, and the possibility for deployment by capacity levels or process maturity. Both methodologies use work products as one evidence of process completion. An organization that is interested in implementing an improvement process should start by getting records of the information required by these work products, and should focus on those who are responsible for them. Even though the standard does not indicate roles that are responsible for each of the documents, the organization that is interested in the improvement should have a role structure that allows meeting the characteristics of the documents. The structure of roles that are available for the practices depends on the size of the organization.

The deployment of standards such as ISO/IEC 15504 is a significant challenge, particularly as regards its associated costs, mainly in human resources required to meet the practices required by the model. In this sense, the need arises to study how the structure of SMBs responds to the requirements of the model.

Workproducts were considered, as well as the definition of roles of a standard business that can deploy the ISO/IEC 15504 model. It was found that none of the variations of SMBs, *micro*, *small* or *medium* business, would be able to support a role structure as that of the standard organization mentioned.

The problem was then thought in the reverse direction – what are the roles and responsibilities that each SMBs category can support?

In answer to that, we have a *micro* businesses which, given its characteristics of software developer, has a team of *developers*, some *functional analysts*, some *project leader* and a *business manager*. In the case of *small* businesses, in addition to the already mentioned roles, there can be some *architect* and someone responsible for *configuration management;* and in the case of *medium* businesses, a group of *process and QA management* is added. Then, roles were assigned to each of the work products, depending on the type of business.

After assigning the roles, first the necessary documents to achieve level 2, Managed, were analyzed. It can be seen that most of the responsibilities of *micro* and *small* businesses correspond to the *project leader* and *business management*. This agrees with the model, since at this level, it is only corroborated that the practices are planned, controlled and verified as established. In the case of medium businesses, responsibilities are more equally distributed.

For level 3, Produced, again in *micro* and *small* businesses the greatest responsibility is with *business management* and the *project leader*, but in the case of *medium* businesses, it is the *process and QA manager* who carries most of the responsibility. This agrees with the goal of the level – carrying out practices based on well-defined processes.

In the case of process analysis, for the ENG category, the transition from *micro* to *small* business is reflected on the roles of *functional analyst* and *project leader* and there are no significant changes when going from *small* to *medium* business. This is because the ENG category describes the specific development and includes the minimum requirements any organization should grant regardless of its size. In the case of the PRO category, as for ENG, the transition from *micro* to *small* business is observed in the role of *functional analyst*; but when going from *small* to *medium* business, there is a significant reduction of the responsibilities of the *project leader*.

Finally, it can be seen that the application of this type of standards to SMBs requires, as a first step, a structure that is capable of carrying out the necessary work to produce evidence of process completion. The load that is generated on the responsibilities of workers in a *micro* business is considerably high. If the business works in an organized way, following defined processes, standardized templates and a limited number of projects, process improvement is achievable with a significant sacrifice. In the case of *small* businesses, since there are more human resources available, other roles can be assigned. Responsibilities are a little better distributed, and the business will be able to manage more projects, although it will also require a significant sacrifice to initiate the improvement process. In the case of *medium* businesses, human resources should not be a problem, since work distribution among roles is balanced.

References

1. José Javier Dolado, Javier Tuya, Isabel Ramos Román, *Técnicas Cuantitativas Para La Gestión En La Ingeniería Del Software.* Netbiblio, 2007.
2. A. Pasini, S. Esponda, P. Pesado and R. Bertone, Aseguramiento de calidad en PYMES que desarrollan software. una experiencia desde el proyecto COMPETISOFT. 2008. pp. 957-966.
3. S. Sanchez, M. Sicilia and D. Rodríguez, Ingeniería Del Software. Un Enfoque Desde La Guía Swebok. 2011.
4. ISO. IEC 15504-2: 2003/Cor. 1: 2004 (E). Information Technology-Process Assessment-Part 2.
5. Ministerio de Industria, República Argentina, "SEPYME - Secretaria PyME - Ministerio de Industria".
6. UE, Definición de microempresas, pequeñas y medianas empresas. Recomendación 2003/361/CE de la Comisión, de 6 de mayo de 2003, sobre la definición de microempresas, pequeñas y medianas empresas [Diario Oficial L 124 de 20.5.2003].
7. H. Oktaba. Competisoft: Mejora De Procesos Software Para Pequeñas y Medianas Empresas y Proyectos. 2009.
8. INTECO, Estudio sobre la certificación de la calidad como medio para impulsar la industria de desarrollo del software en España. 2008.
9. I. ISO. IEC 15504-5: 2006 (E). Information Technology-Process Assessment-Part 5.

VIII

Database and Data Mining Workshop

Combining Methods for Searches in Nested Metric Spaces

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

[1] Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional de Rosario
Rosario, Argentina
hugogercek@gmail.com, {deco, bender}@fceia.unr.edu.ar, salvettimariano@hotmail.com
[2] Departamento de Informática. Universidad Nacional de San Luis
San Luis, Argentina
nreyes@unsl.edu.ar

Abstract. *Most search methods in metric spaces assume that the topology of the object collection is reasonably regular. However, there exist nested metric spaces, where objects in the collection can be grouped into clusters or subspaces, in such a way that different dimensions or variables explain the differences between objects inside each subspace. This paper proposes a two levels index to solve search problems in spaces with this topology. The idea is to have a first level with a list of clusters, which are identified and sorted using Sparse Spatial Selection (SSS) and Lists of Clusters techniques, and a second level having an index for each dense cluster, based on pivot selection, using SSS. It is also proposed for future work to adjust the second level indexes through dynamic pivots selection to adapt the pivots according to the searches performed in the database.*

Keywords: *metric spaces, pivots selection, similarity search*

1. Introduction

With the evolution of information technology and communications have emerged repositories of unstructured information, with types of data such as free text, images, audio and video. This scenario requires more general models, such as metric databases, and tools for efficient searches on these data types. In unstructured data repositories it is more useful a similarity search than an exact search. 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 that are similar to a given one. In order to improve objects retrieval an index can be used, because an index structure allows fast access to objects. Most of the search techniques were developed assuming that the topology of the object collection is reasonably regular, but experiments on spaces where collections of objects can be grouped into subspaces or clusters have shown that they are not so efficient. In [1] a two level structure is proposed: Sparse Spatial Selection for Nested Metric Spaces (SSSNMS), which is the first that consider this type of spaces.

This paper presents a new version of this structure for indexing and similarity searching with two levels of indexes. At the first level, clusters are identified with Sparse Spatial Selection (SSS) and are sorted into a List of Clusters (LC) [2]. At the second level, based on a measure of density, the clusters that are considered highly populated are indexed with pivots also using SSS.

The rest of the paper is organized as follows: Section 2 presents basic concepts. Section 3 discusses related work. Section 4 presents the proposed method. Finally, conclusions 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 $d(x,y)=d(y,x)$, reflexivity $d(x,x)=0$ and triangular inequality $d(x,y) \leq d(x,z)+d(z,y)$. A finite subset U of X, with $|U|=n$, 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: $L_s((x_1, ..., x_k),(y_1, ..., y_k))=(\sum |x_i - y_i|^s)^{1/s}$.

In general metric spaces it can be translated the concept of "dimensionality", even if the objects are not assumed to have coordinates [3]. One easy characterization of the intrinsic dimensionality is obtained from the histogram of distances. An *easy* instance will have a small mean distance value and large standard deviation, while a *difficult* instance will be the converse, a large mean distance value and small standard deviation.

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: $\{u \in U \,/\, d(u,q) \leq r\}$. In *k nearest neighbors* search, the k objects closest to q are retrieved, that is: $A \subseteq U$ such that $|A|=k$ and $\forall u \in A$, $v \in U\text{-}A$, $d(q,u) \leq d(q,v)$. 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* [3]. The first one splits the metric space into a set of equivalence regions, each of them represented by a *cluster center* and a *radius*. During searches, whole regions are discarded depending on the cluster center, the query points, and their radius. *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* [4], *Fixed-Queries Tree* [5], *Fixed-Height FQT* [5], *Fixed-Queries Array* [6], *Vantage Point Tree* [7], *Approximating and Eliminating Search Algorithm* [8], *Linear AESA* [9] and *SSS* [1].

3. Related Work

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. In [9] 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 [1] 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*α. 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. In [11] an improved SSS is presented, where the index suits to searches, after the index was adapted to the metric space, using a dynamic selection of pivots. The initial index is built using SSS and it is "updated" during searches.

Another improvement to SSS is the SSS-Tree [12] that uses trees and the best properties of clustering techniques. Its main feature is that cluster centers are selected using SSS, so the number of clusters in each node depends on the complexity of the subspace associated with it.

Since the indexes lose their efficiency as the intrinsic dimension of data increases, in [2] an index called List of Clusters (LC), based on the compact partition of the data set, is presented. It is shown that the LC is very resistant to the intrinsic dimensionality of the data set. In addition, due how the List of Clusters is built, a special order to its members is given: clusters in previous positions have priority over subsequent clusters, when they contain elements that are located in regions of intersection. Each cluster in the list, which is a subspace of center *c* and radius r_c, is called ball. In the LC, the first center chosen has precedence over the later in the case of overlapping balls. That is, all elements that fall under the ball of the first center are stored in that cluster even though they might be in others. Given a query (*q*, *r*) the idea is to use this feature to inspect the LC for those clusters in which the ball has query intersection, and stop the search when the query ball is completely contained within this cluster.

In [13] is presented the Sparse Spatial Selection for Nested Metric Spaces (SSSNMS) as a new approach to solve problems of indexing and searching in nested metric spaces. In this type of spaces, objects in the collection can be grouped into different clusters or subspaces. Each of these subspaces is nested within a more general one. The aim of this method is to identify subspaces and apply SSS in each of them. For this, the index constructed by SSSNMS is structured in two levels: first level selects a set of reference with SSS and it is used as centers of clusters to create a Voronoi partition. In the second level, those clusters, that are considered dense, are indexed using SSS pivots in each of them. Given a query (*q*, *r*), it is compared against all cluster centers of the first level. Those clusters $C_i=(c_i,r_c)$ for which $d(q,c_i)-r_c>$r are directly discarded from the result set as the intersection of each cluster with the result set is empty. If the not discarded cluster does not have an associated table of distances from their objects to the pivots, the query is directly compared against all objects in the cluster. If the not discarded cluster has an associated table of distances, the query is compared against pivots and this table is processed in order to eliminate as many objects as possible. Objects that cannot be discarded are directly compared against the query.

In this paper, we analyze the problem of searching in nested metric spaces, and we propose a new index structure that has as main objective to minimize the search time. For this, we use SSS and Lists of Clusters. The proposal is presented in the next section.

4. Proposed Method

Most index structures and their search methods were built to work on collections of data where the spatial distribution is fairly regular. For example, SSS belongs to the family of indexes that get good performance in regular spaces, but its performance is not the best in irregular collections. Moreover, the SSSNMS proposal yields better results in nested metric spaces.

In this paper, we analyze the problem of search in such spaces, and we propose a new index structure that has as main objective to minimize search time. For this, we propose to use SSS to identify clusters nested in the general metric space, obtaining the centers of the clusters to ensure a good coverage of general space. Each cluster remains ordered in a List of Clusters. By using this order during a search, if the query ball is totally contained within a cluster, we can omit inspecting the following clusters. This structure provides high resistance to the intrinsic dimensionality of data. Subspaces considered highly populated are indexed using pivots, based on a measure of density that is presented later, in order to get a good coverage of each subspace.

Therefore, our structure has two levels: a List of Clusters that identifies and maintains an order of each nested subspace in the general metric space, and a pivot index built using SSS for each subspace that we consider dense. This structure is dynamic and adaptive at the same time. Dynamic because it can start with an empty collection to which objects will be added. It is adaptive because it allows adapting itself to the complexity of space. This is, a priori we do not assume anything about the number of clusters needed, and their characteristics, nor on the number of pivots for each dense subspace.

4.1 Construction of the Index

The efficiency of similarity search methods depends on the set chosen as a reference, where *reference* means a pivot, for pivot-based index, or a cluster center, for clustering-based index.

The structure proposed in this paper has two levels. The first level uses SSS to identify subspaces, and to build a List of Clusters where the centers are well distributed (because of the use of SSS). Also, the order of the clusters will optimize the search (because of the use of LC). The second level uses SSS to obtain pivots in each cluster, acquiring an index where the references are well distributed.

Let (X,d) be a metric space, where $U \subseteq X$ is the database. Let M be the maximum distance between objects ($M=max\{d(x,y)/x,y \in U\}$).

The index is built as follows:

Level One: List of Clusters with SSS. In this first level, nested subspaces in the general metric space are identified and indexed. SSS is used to obtain well distributed centers of the clusters. Each cluster is maintained in a List of Cluster to obtain and preserve an order.

Given a center $c \in U$ and a radius r_c, we define the ball (c,r_c) as the subset of elements of X which are at a maximum distance r_c from center c; and where $r_c < M*\alpha$. Experimentally, in [1] it is shown that the optimal value for α must be in the range [0.35, 0.4] as the dimensionality of the collection.

We define: $I_{U,c,rc}=\{u \in U,\ 0<d(c,u)\leq r_c\}$ as the bucket of internal elements that remain inside the ball of center $c;$ and $E_{U,c,rc}=\{u \in U,\ d(c,u)>r_c\}$ as the other elements (external).

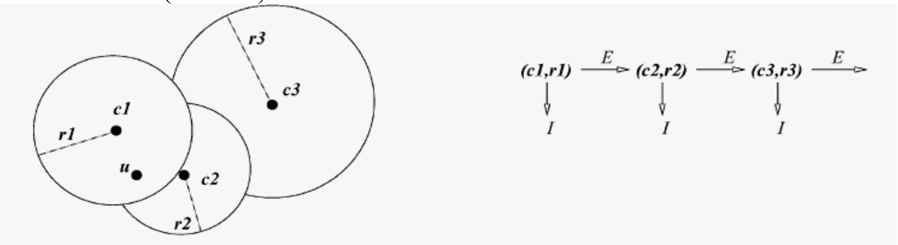


Fig. 1. Clusters representation: < (c_1, r_1, I_1), (c_2, r_2, I_2), (c_3, r_3, I_3) > , from [14]

The main idea, after selecting the first center, is to go on by selecting the SSS centers iteratively on each set *E* and get a list of triples (c_i,r_i,I_i) (center, radius, bucket), where each element represents a cluster. The data structure obtained seems to be symmetric, but it is not. The first center chosen has precedence over the later in the case of overlapping balls, as shown in Figure 1. All items that remain inside the ball of the first center (c_1 in the figure) are stored in the bucket I_1 although that might be within the buckets of subsequent centers (c_2 and c_3 in the figure). The figure shows how the data structure can be viewed as a list, where clusters in previous positions have a preference when it comes to contain elements that are located in regions of intersection on the following clusters.

This structure is dynamic. This allows us to start with an empty collection of elements. But if the initial collection is not empty, an algorithm of "bulk loading" to identify clusters can be applied. This is, to apply a variant of the SSS on the initial set *U* to obtain only a group of representative elements and the distance between them. For each representative element, distances between it and the others representative elements are averaged, and then they are ranked according to this distance from highest to lowest, and finally their appearances are removed from *U*. This sorted list is added at the beginning. This ensures that the first items examined by the algorithm will be distant, and therefore should belong to different clusters and so would get a better representation of nested subspaces in the general metric space.

The pseudo code of the algorithm of construction of this index level is as follows:

```
Build_Index(U,L,B)
R = {}
    for each $u_i \in$ U do
        if canBeCenter($u_i$,L)  //If distance between $u_i$ and each center is $\geq M*\alpha$
            setRadio($r_i$,M,$\alpha$)       //Computes radius $r_i$ which depends on M and $\alpha$.
            insertAtEndOfL(($u_i$,$r_i$,{}),L)  //Inserts triplet (($u_i$,$r_i$,{})
                                                  at the end of the List of Clusters
            updateM(M)              //Updates value of M.
        else if isInSomeBallCj($u_i$,L,($c_j$,$r_j$,$I_j$)) //If the element $u_i$
                                      belongs to any ball ($c_j$, $r_j$) of L, returns the first
                                      triplet ($c_j$,$r_j$,$I_j$) of L that satisfies this condition.
            updateI(($c_j$,$r_j$,$I_j$),$u_i$))  //Adds element $u_i$ to $I_j$.
            updateL(($c_j$,$r_j$,$I_j$),L))  //Updates the ball with center $c_j$ of L
        updateM(M)              // Updates value of M.
    else
        updateR($u_i$,R))        //Adds $u_i$ to the list L of elements to reconsider.
        UpdateM(M)
    reconsider(R,L,B)      //Reconsiders the no indexed elements.
```

This algorithm receives as parameters the set of elements to index *U* (preprocessed or not), the List of Clusters *L* empty, and an empty set *B* of elements that do not belong to any subspace, but will be indexed with SSS.

If the input is not preprocessed, and the loop *for* is considered as successive insertions for each $u_i \in U$, we would be under the assumption that it starts with an empty database that grows as elements are inserted into it. In the last line of pseudo code, the list R has two types of elements: those who should belong to some subspace of the List of Cluster L but because of the order in which the centers were chosen they do not fall into any ball (c_j,r_j); or elements that are outside from any ball of the List of Clusters L. The method *reconsider(R,L,B)* takes into account these two options: those elements that should belong to some subspace of L but that could not be observed at first are added to the respective cluster (i.e. the first from the list L if this element "falls" in more than one); and those elements that do not belong to any subspace of L are stored in a bag B of elements. Each element of B is indexed in the usual way with SSS, using each center c of L as a pivot.

r_c is the radius of the cluster of center c. Each radio is static, i.e. once chosen it cannot change, because if so, to update this value the index should be rebuild to keep the properties of the List of Clusters. According to the current values of M, it must be $r_c{<}M{*}\alpha$, so centers selection strategies with SSS does not collide with LC properties. That is, a new cluster center is not contained in an existing cluster. Therefore, the radius r_c must be equal to $M{*}\alpha{*}\rho$, where $\rho<1$.

Level Two: Choosing pivots on dense subspaces with SSS. When construction is completed the first level of the index, we have: a list of clusters L with elements (c,r,I); and a bag B of elements not contained in any subspace indexed with SSS using the centers c of L as pivots.

The *density* of each cluster is computed as the number of elements of the cluster divided by the maximum distance between them. Those clusters of L, considered dense are indexed using SSS, obtaining a reference set consisting of pivots. To compute the density of each cluster can be very costly if the maximum distance between each object is obtained by comparing all the elements of the cluster with the rest. To minimize this cost in construction time we get an approximation of the maximum distance. To do this, an object of the cluster is chosen at random and is compared against all other objects in the cluster. Its further object is compared against all other objects in the cluster to obtain it further object too. After repeating this process a few iterations, we get an approximation of the maximum distance (if it is not the current maximum distance).

We consider that the cluster C_i has high density if *density*$(Ci)>\mu+2\sigma$, where μ and σ are the mean and standard deviation of the density of all clusters. For each dense cluster, a set of objects is obtained with SSS to be used as pivots, and its table of distances for each object of the cluster to each pivot is computed and stored.

In this second step, the index stores information about the dense subspaces. An element u is chosen as pivot of the subspace with center c_i if the distance of u to each pivot of the subspace is larger than $M_i{*}\beta$, where M_i is the maximum distance between each pair of objects in the cluster of center c_i and β is a constant value near 0.4 as it is shown in [1].

4.2 Searches

Given a query *(q, r)*, *q* is compared against all the centers of clusters, following the order in the list of clusters *L*, until the end of the list or can be stopped if the ball is completely contained in one of the clusters. Each cluster that has not been discarded (i.e., clusters with which it has intersection) is a candidate cluster and should be reviewed. If it reaches the end of the list without the query ball has been completely contained in a cluster, distances to all centers of clusters have been calculated, and therefore they are used to discard some elements of the bag *B* by filtering the distances to the pivots (centers of the list of clusters). The range search algorithm is presented in the following pseudo code:

```
SearchL(L,(q,r),B, K)
if L is empty
    if  B is empty
        return K
    else
        return K $\cup$ SearchB((q,r),B)
let L = <(c, $r_c$, I):L'>
distQC  $\leftarrow$ d(q,c)
if distQC $\leq$ r                   //Query ball contains center c
    if distQC + r $\leq r_c$          //Query ball is inside the cluster
        return Pivotsearch(I,(q,r)) $\cup$ K $\cup$ {c}
    else    // Query ball contains the cluster or Query ball intersects the cluster
        K' $\leftarrow$ Pivotsearch(I,(q,r)) $\cup$ K $\cup$ {c}
        return SearchL(L',(q,r),B,K')
else                                   //Query ball does not contain the center c
    if distQC + r $\leq r_c$          //Query ball is inside the cluster
        return Pivotsearch(I,(q,r)) $\cup$ K
    else if distQC > r + $r_c$        //Query ball is outside the cluster
        return SearchL(L',(q,r),B,K)
    else                       //Query ball intersects the cluster
        K' $\leftarrow$ Pivotsearch(I,(q,r)) $\cup$ K
        return SearchL(L',(q,r),B,K')
```

The list is iterated and the relationship between each cluster and the query is established based on the distance between the query and the center of the cluster. The recursive function *SearchL* has four parameters: the list of clusters *L*, the query (*q*, *r*), the bag of elements *B* and the list *K* of candidates (which must be empty to start).

The function *Pivotsearch* gets the list of candidates for each cluster, using the pivots themselves if the cluster is dense and is indexed, and returns all elements of the cluster if it is not dense. Its parameters are: the bucket *I* of elements of the cluster, which in our case we can think as a reference to index, and the query (*q*, *r*). Given the asymmetry of the data set, the search can be pruned if the query ball is totally contained in the ball of center *c*. In this case, we do not consider the rest of the list. If the end of the list is reached without the query ball has been completely contained in a cluster and the bag of elements *B* is not empty, the method *SearchB* is responsible for discarding some elements of the bag *B* by filtering the distances to the pivots (the centers of the list of clusters).

This is an essential feature absent in other algorithms, where the search needs to go into all the partitions that are intercepted by the query ball. In this structure the consideration of relevant partitions can be stopped when the query ball is fully contained on a partition.

The function *Pivotsearch* applies the triangle inequality as follows: given an element *e* of the index, it can be discarded if $|d(p_i,e)-d(p_i,q)|>r$ for some pivot p_i of the subspace, since by the triangle inequality if this condition is true, occurs that $d(e,q)>r$.

Finally, once the list of candidates is obtained, the query is compared exhaustively against it, and the distance from centers should not be recalculated since it was previously obtained.

5. Conclusions

This paper presents a new index and similarity search method, which tries to fully exploit the advantages already known from other structures in order to obtain an efficient method for nested metric spaces. We propose a two level structure. A first level where clusters are detected and they kept sorted combining SSS and LC strategies. This allows getting a good coverage of the general metric space and a high resistance to the intrinsic dimensionality of the data set. In the second level each dense subspace is indexed with SSS getting a good coverage of the subspace. In searches, this structure is used first to exclude subspaces, and then to get the list of candidates for each subspace that has not been discarded. With the proposed algorithm, the search can be also stopped when a query ball is completely contained within a cluster from the list, which saves a significant amount of time. It is proposed as future work to use techniques of incoming pivot and outgoing pivot defined in [11], after making a certain number of searches on each subspace, in order to adapt pivots to the searches and to get better performance in future queries. The inclusion of these techniques will enable us to obtain experimental results.

References

1. Pedreira O., Brisaboa N.R.: Spatial Selection of Sparse Pivots for similarity search in metric Spaces. In: 33nd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'07), LNCS vol: 4362, pp. 434-445. Springer (2007).
2. Mamede M.: Recursive Lists of Clusters: A Dynamic Data Structure for Range Queries in Metric Spaces. CITI / Departamento de Informática Faculdade de

Ciencias e Tecnologia da UNL. Caparica, Portugal. In Proceedings of ISCIS. 2005, 843-853.

3. Chávez E., Navarro G., Baeza-Yates R., Marroquín J. L.: Searching in Metric Spaces. ACM Computing Surveys. 33(3), pp 273-321 (2001).
4. Burkhard W. A., Keller R. M.: Some approaches to best-match file searching. Communications of the ACM, 16(4): 230-236 (1973).
5. 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, pages 198-212 (1994).
6. Chavez E., Navarro G., Marroquín A.: Fixed queries array: a fast and economical data structure for proximity searching. Multimedia Tools and Applications (MTAP), 14(2):113-135 (2001).
7. Yianilos P.: Excluded middle vantage point forests for nearest neighbor search. In: 6th DIMACS Implementation Challenge: Near Neighbour searches ALENEX'99 (1999).
8. Vidal E. An algorithm for finding nearest neighbor in (approximately) constant average time. Pattern Recognition Letters 4, 145-157 (1986).
9. 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, 15: 9-17 (1994).
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, pp. 33-44. IEEE Computer Science Press (2001).
11. Salvetti M., Deco C., Reyes N., Bender C.: Adaptive and Dynamic Pivot Selection for Similarity Search. Journal of Information and Data Management, Ed. Sociedade Brasileira de Computação. Vol. 2, No. 1, February 2011, pp. 27-35.
12. Uribe Paredes R., Solar R., Brisaboa N. R., Pedreira O., Seco D.: SSSTree: Búsqueda por Similitud Basada en Clustering con Centros Espacialmente Dispersos. Encuentro Chileno de Computación. Iquique, Chile, Nov. 2007.
13. Brisaboa N. R., Luaces M. R, Pedreira O., Places Á. S., Seco D.: Indexing Dense Nested Metric Spaces for Efficient Similarity Search. In: Proceedings of the 7th International Andrei Ershov Memorial Conference on Perspectives of System Informatics (PSI 2009) - LNCS 5947, Springer, Novosibirsk (Rusia), 2010, pp. 98-109.
14. Chávez E., Navarro G.: A compact space decomposition for effective metric indexing. Pattern Recognition Letters, 26(9): 1363-1376, 2005.

Center Selection Techniques Based on Distance Histograms

ARIEL LUCERO[1], CARINA M. RUANO[1], NORMA E. HERRERA[1]

[1] Departamento de Informática, Universidad Nacional de San Luis
Ejército de los Andes 950, San Luis, Argentina
3033402@alumnos.unsl.edu.ar
{cmruano,nherrera}@unsl.edu.ar

Abstract. *The metric spaces model formalizes the similarity search concept in nontraditional databases. The goal is to build an index designed to save distance computations at query time. A large class of indexing algorithms are based on partitioning the space in zones as compact as possible. Each zone store a representative point, called center, and a few extra data that allow to discard the entire zone at query time without measuring the actual distance between the elements of the zone and the query object. How the centers are selected affect the performance of the index. In this paper, we propose a new center selection technique based on the information provided by distance histograms. This technique was evaluated using the Geometric Near-neighbor Access Tree. We experimentally show that it achieves a good performance.*

Keywords: *Databases, Metric Spaces, Metric Indexes, Center Selection.*

1. Introduction

Similarity search provides a way to find database elements that are close or similar to a given query element. Similarity search is a natural extension of the exact searching due to the fact that the databases have included the ability to store new data types such as images, sound, text, video, to name a few. As these new data types are unstructured, is not possible to organize them in records and fields, like in traditional databases. Even if a classical structuring is possible, it restricts beforehand the types of queries that can be posed later.

In [3] is shown that the similarity search problem can be expressed as follows: given a set X of objects and a distance function d, defined among them, that quantifies their similitude, the aim is to retrieve all the elements similar to a given one. This function d satisfies the properties required to be a distance function: *positivity* ($d(x, y) \geq 0$), *simmetry* ($d(x, y) = d(y, x)$) and *triangular inequality* ($d(x, y) \leq d(x, z) + d(z, y)$). The pair (X, d) is called *metric space*. A finite subset $U \subseteq X$, which will be called *database*, is the set of objects where we search.

One of the typical queries over this new database model is the *range query* which is denoted by $(q, r)_d$. Given a query $q \in X$ and a tolerance radius r, a range query consists in retrieving all the objects from the database that are within a distance r from q, that is: $(q, r)_d=\{u \in U\text{:}\ \ d(q, u) \leq r\}$.

The total query time T can be calculated as T = *#evaluations of d* x *complexity(d) + extra CPU time + I/O time*. In many applications, the evaluation of function *d* is so costly that the other terms in the formulae can be neglected. This is the complexity model we used in this work; therefore, our complexity measure will be the number of evaluations of the distance function *d*.

A range query can be trivially answered by an exhaustive examination of the database. Unfortunately, this is generally very costly in real applications. To avoid this situation, the database is preprocessed using an indexing algorithm whose aim is to build a *data structure* or *index*, designed to save distance evaluations at query time. In [3] the authors present a unifier development for all the existing solutions in this topic. Basically, two groups can be distinguished: *pivot based algorithms* and *compact partition based algorithms*.

In this paper we focus in compact partition based algorithms. These algorithms build the index based on the proximity of the elements to a predefined set of them, called *centers*. The centers selected at indexing time will not affect their effectiveness, but they dramatically impact their efficiency.

In [4] are proposed two center selection policies based on hard and soft kernel concepts [1]. One of them, called *high density zone*, consists in selecting the centers from elements which belong to the hard kernel. The authors show that this techniques significantly decrease the number of distance evaluations when compared to a random selection. However, high density zone does not actually use the hard kernels of the metric space. Instead, it only works at each selection step with the histogram of the last chosen center, therefore it works with the local hard kernel.

In this paper we propose a new center selection technique based on the information provided by distance histograms. The aim is to select centers from the hard kernel of the whole metric space instead of selecting them from a local hard kernel of a single point. We have tested the proposed selection techniques using the Geometric Near-neighbor Access Tree index (GNAT) [2].

Te remainder of the paper is organized as follows. Section 2 presents the related work, describes the GNAT index and introduces the existing center selection techniques based on distance histograms. Section 3 describes our proposal in detail. Section 4 presents the experimental evaluation and analysis of results. Finally, section 5 presents the conclusions and future work.

2. Related Work

2.1 Indexes in Metric Spaces

In [3] the authors state that the metric spaces indexing algorithms are based on, first, partitioning the space into equivalence classes and, second, a subsequent indexation of each class. Afterwards, at query time, some of these classes can be discarded using the index and an exhaustive search only takes place on the remaining classes. The difference among the existing indexing algorithms is the way they build up the equivalence relationship. Basically, two groups can be distinguished: *pivot based algorithms* and *compact partition based algorithms*.

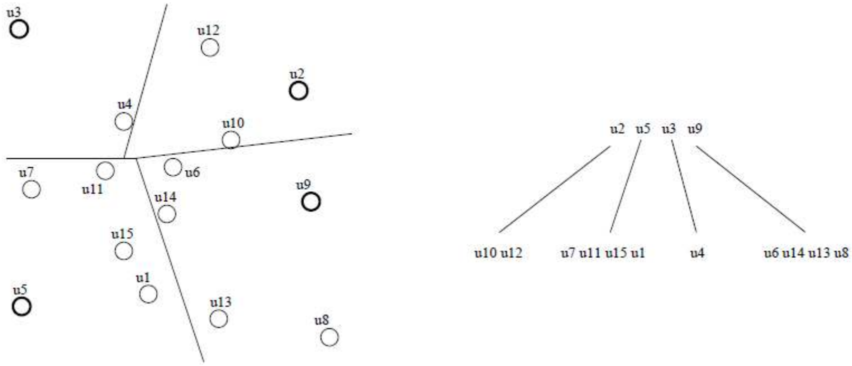


Fig. 1. On the left, the division of the space obtained when u_2, u_3, u_5, and u_9 are taken as centers. On the right, example of the first level of GNAT

In this paper we have studied metric space indexing using compact partition based algorithms. Specifically, we have focused on the index called *Geometric Near-neighbor Access Tree* (GNAT) [2]. This index builds the equivalence relationship based on the proximity of the elements to a predefined set of them, called *centers*. In this sense, two elements are equivalent if the closest center to each of them is the same center *c*.

The construction of a GNAT of arity *m* can be summarized as follows: for the first level *m* centers c_1, c_2,... , c_m, are chosen from *U*. Then, the set U_{ci} formed by all the objects of *U* closer to c_i than to any other center c_j , is related to each center c_i. If $|U_{ci}|> m$, a GNAT is recursively created as a child of node c_i, otherwise a terminal node is built using the elements of U_{ci}. Figure 1 shows an example of the first level of a GNAT with u_2, u_3, u_5 and u_9 as centers.

The GNAT stores at each node an $O(m^2)$ size table ρ_{ij}=[$\min_{x\in U_{cj}} d(c_i, x)$, $\max_{x\in U_{cj}} d(c_i, x)$], which stores minimum and maximum distances from each center c_i to each set U_{cj} . At query time, this information is used together with the triangular inequality, to limit the search. Given a range query $(q, r)_d$, the query q is compared against some center c_i and then it discards any other center c_j (and their corresponding sets U_{cj}) such that $[d(q, c_i)- r, d(q, c_i)+r] \cap \rho_{ij} =\emptyset$. This process is repeated until no one can be discarded. The search then enters recursively in each non discarded subtree. In the process, any center close enough to q is reported.

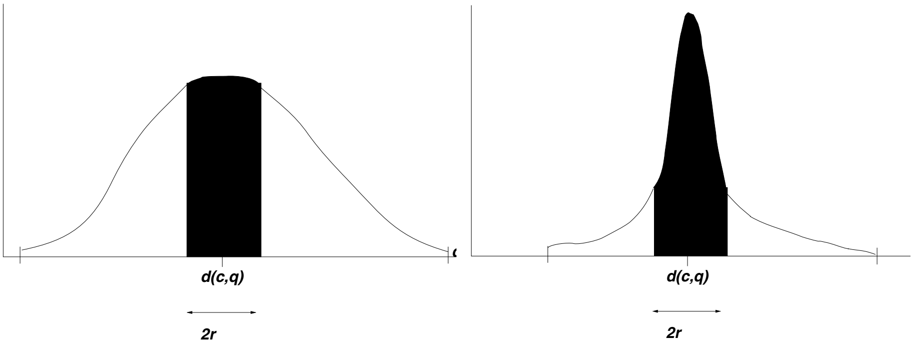


Fig. 2. A low-dimensional (left) and high-dimensional (right) histogram of distances, showing that on high dimensions the elements are concentrated around the mean

2.2 Center Selection Techniques Based on Local Histograms

One of the main issues found in the design of efficient indexing techniques is what is known as the *curse of dimensionality*. The dimensionality concept is related to the effort needed to search for an element in a given metric space.

The intrinsic dimensionality of a metric space X is defined in [3] as $\rho=\mu^2/2\sigma^2$, where μ and σ^2 are the mean and variance of the distance histogram of X. This formula indicates that, as the intrinsic dimensionality increases, so do the mean, but the variance of the histogram decreases, i.e., the distance histogram concentrates around its mean and this creates a negative effect on indexing algorithms.

Figure 2 gives an intuitive explanation of why the search problem is harder when the histogram is concentrated. The histograms in the figure illustrate a possible distance distribution with respect to a reference point c (*local histogram* of c). If we consider a random query q, the grayed areas in the figure show the points that we cannot discard. As the histogram is more and more concentrated around its mean, fewer points can be discarded using the information given by $d(c, q)$. This phenomenon is independent on the nature of the metric space and gives us a way to quantify how hard is to search on an arbitrary metric space.

It is well known that in similarity search, the most distinctive feature is the space geometry, i.e., how the dataset is distributed. As the elements are more and more concentrated, the search process becomes harder. In [1] the authors define and characterize the concept of hard and the soft kernel of a metric space. On one hand, the ***hard kernel*** is formed by those elements which lay in a densely populated zone of the metric space; this is the zone around the mean of distance histogram if it is a Gaussian shaped. On the other hand, the ***soft kernel*** is formed by the rest of the elements of the metric space.

The two center selection policies proposed in [4] are based on hard and soft kernel concepts. One of them, called *Closer Element* (*CE*), consists in selecting the centers from elements which belong to the soft kernel and the other one, called *High Density Zone (HD)*, picks the elements from the hard kernel. The authors show that the most competitive is *HD*, which achieves an important decrease in the number of distance evaluations when compared to a random selection. The *HD* technique chooses the first center c_1 randomly. Once c_i has been selected, the center c_{i+1} is chosen from the region that c_i considers as its hard kernel. In order to achieve this, the local histogram for c_i is computed and one element from the central region of the c_i local histogram is selected as the center c_{i+1}.

It is important to note that these techniques do not actually calculate the kernels of the metric space. Instead, they only works at each step with the local histogram of the last chosen center, therefore they select the next center based on the local hard kernel.

3. New Center Selection Techniques

As explained in the previous section, the HD technique chooses the center c_{i+1} based on what was identified as the hard kernel when considering c_i. If the distribution of elements in the area to which c_i belongs differs from the distribution of elements of the global space, then local histogram of c_i is significantly different from the metric space histogram. In this case, there is a low probability that c_{i+1} belongs to the hard kernel of the metric space. Our goal in this paper is to overcome this weakness.

Note that the local histogram can be very different from the global histogram. However, if several local histograms are similar, then we can predict how the dataset is distributed [3].

We have designed a new technique for center selection, called ***global high density*** **(*GHD*),** based on approaching the histogram of the metric space by the intersection of the local histograms of several centers instead of the previous center histogram. To do that, the first center (c_1) is randomly choosen. A second center (c_2) is chosen from the region that c_1 would consider as a hard kernel. Then, a new center (c_3) is chosen from the intersection between the set of the local hard kernels determined by c_1 and c_2. The process is repeated until m centers have been chosen. In general, a center c_i is chosen from the intersection between the set of the local hard kernels determined by c_1, c_2,.., c_{i-1}, that is, c_{i+1} belongs to the set θ where:

$$\theta = nd_1 \cap \ nd_2 \cap ... \cap nd_i. \quad \textbf{(1)}$$

and nd_i denotes the local hard kernel of c_i.

According to the guideline provided in [1], the hard kernel of c_i is formed by the elements surrounding the media of the histogram. That is, the element e belongs to nd_i if $e \in [\mu\text{-rc}, \mu\text{+rc}]$, where μ is the media of the histogram of c_i and *rc* is an integer number called ***cutting radius***. The most appropriate value for *rc* was experimentally obtained as will be showed further in the experimental section.

When progressing down the tree at indexing time, the number of elements in each subtree decreases and the probability that $\theta = \varnothing$ increases. In this case, the *GHD* technique cannot be used. This problem can be solved in several ways. One possibility is to choose the rest of the centers randomly. Another one is to reinitialize θ and continue the process.

In our implementation, we are used the following alternatives:

GHD-Ra: to continue the center selection randomly.
***GHD-Re*₁**: to choose the next center c_i randomly between elements in that subtree, and to reinitialize the intersections with $\theta = nd_i$.
***GHD-Re*₂**: to choose the next center c_i randomly from the last non empty θ and to reinitialize the intersections with $\theta = nd_i$.

4. Experimental Results

The experiments were performed over word dictionaries using the *edit distance* (also called *Leveshtein distance*) as distance function. This function is discrete and computes the minimum number of character insertions, deletions and replacements needed to make two strings equal. Four dictionaries were actually used, namely, Spanish, German, English, and Italian. Because of space reasons, in the following sections only the results on the Spanish dictionary will be included; the results on the other dictionaries will not be exposed but are available on demand.

In all cases, the indexes were built using 90% of the elements of each dictionary, leaving the remaining 10% as queries for range search with radii $r = 1, 2, 3, 4$. The dictionaries were indexed using GNATs with arities 32, 64, 128, and 256.

As explained before, all *GHD* variants use the *rc* value to define that each center would consider as its hard kernel. According to the guideline provided in [1], we use the values 1, 2, 3, 4, and 5 for *rc*.

The experiments have been carried out in three phases. The first phase was aimed to determine the most appropriate value of *rc* to be used for *GHD-Ra*, *GHD-Re*$_1$, and *GHD-Re*$_2$. In phase two, the experiments were run using the three proposed techniques in order to analyze which one yields better results. Finally, in phase three, the most competitive *GHD* variant was compared against *HD*.

4.1 Tunning the parameter *rc*

The rc values used by the designed policies affect the way in which the centers are selected and consequently the search performance. In this phase, our experiments try to determine the best choice for rc.

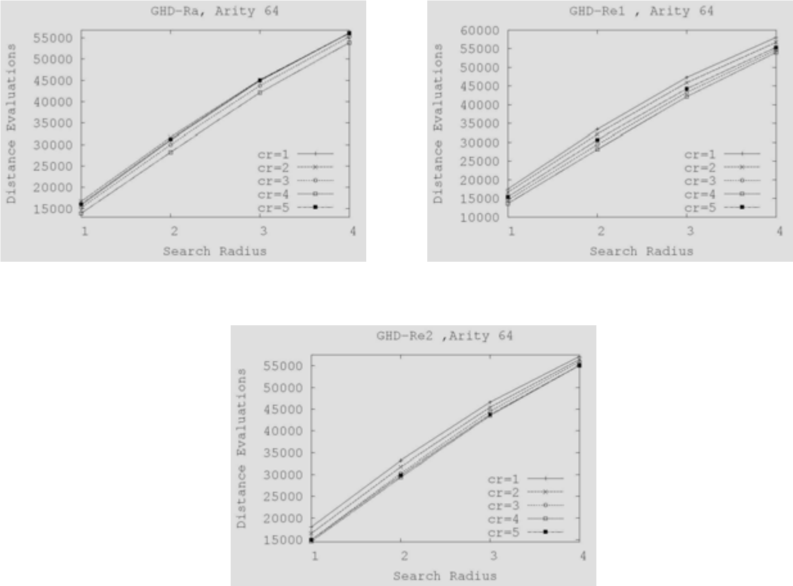


Fig. 3. Average number of distance evaluations as a function of the search radius, for different cutting radii (*cr*). On the top, *GHD-Ra* and *GHD-Re*$_1$ techniques, on the bottom *GHD-Re*$_2$ technique

Figure 3 shows the results for the different *GHD* variants using a GNAT with an arity equal to 64. The *X*-axis represents the search radii and the *Y*-axis the average number of distance evaluations done to solve a range query. As can be observed, the *GHD-Ra* and *GHD-Re*$_1$ variants using the same *rc* value to yield the best results for all search radii. For these cases, *rc=4* is the best choice. However, the previous *rc* value is not applicable to *GHD-Re*$_1$, where the best *rc* value depends on the search radius. It is important to note that the search radius is unknown at indexing time. Therefore the *rc* value cannot be selected based on the radii values. For this reason, we have decided to use *rc=4* because it yields the closest performance to the optimal one in all the cases. We have also used the same criteria to analyze the results for arities 32, 128, and 256. Table 1 shows the values that were finally selected for each case.

4.2 Evaluation of the proposed techniques

In previous subsection we have identified the best value of *rc*. Now, we analyze the performance of the *GHD* variants. In these experiments, we have used the values *rc* more appropriate for each arity (see Table 1).

Table 1. Arities and the most appropriate value *rc* for each *GHD* variants

Arity	***GHD-A***	***GHD-Re***$_1$	***GHD-Re***$_2$
32	*rc*=3	*rc*=3	*rc*=3
64	*rc*=4	*rc*=4	*rc*=4
128	*rc*=2	*rc*=2	*rc*=5
256	*rc*=1	*rc*=4	*rc*=5

Figure 4 shows the results for the different values of *r* and for the different GNAT arities. The *X*-axis represents the search radii and the *Y*-axis, the average number of distance evaluations. As can be seen, for arities greater than 64, all the center selection techniques have a common behavior in the sense that decreases the number of distance evaluations as the arity of the GNAT increases. Nevertheless, the number of distance evaluations saved differs between them.

The *GHD-Ra* obtains the best performance for arities 128 and 256 in all search radii. However, *GHD-Re*$_1$ outperforms to *GHD-Ra* for arity 32. In arities 128 and 256, *GHD-Re*$_1$ is not a good choice. In addition, we can observe that *GHD-Ra* outperforms to *GHD-Re*$_2$ in all cases, saving around 7% of distance evaluations.

As explained before, the *GHD* policies select the next center based on the set θ, which contains one approximation to the hard kernel of the metric space. The difference between the *GHD* variants is the way in which they reinitialize θ when it is empty. The experimental evaluation shows that the index performance degrades when θ is reinitialized. Therefore, we can conclude that when θ empties, it is no longer possible to obtain enough information to improve the set of centers already selected. As the the arity grows, the set θ will be empty faster, therefore *GHD-Ra* is the most suitable technique. On the other side, when the arity decreases the set θ empties slower, therefore *GHD-Re*$_1$ overcomes the other techniques.

4.3 Comparison between HD and GHD techniques

In the final set of experiments, the most competitive *GHD* variants (*GHD-Ra* and *GHD-Re*$_1$) are compared against *HD* technique. The Figure 5 shows the results for each search radius. A similar behavior is appreciated in both *HD* and *GHD* variants: for arities greater than 64 the number of distance evaluations decreases as the arity of the GNAT increases. Nevertheless, *GHD-Ra* outperforms the others techniques for arities greater than 64 on all the search radii considered. In arities 32 and 64, all techniques obtain similar results.

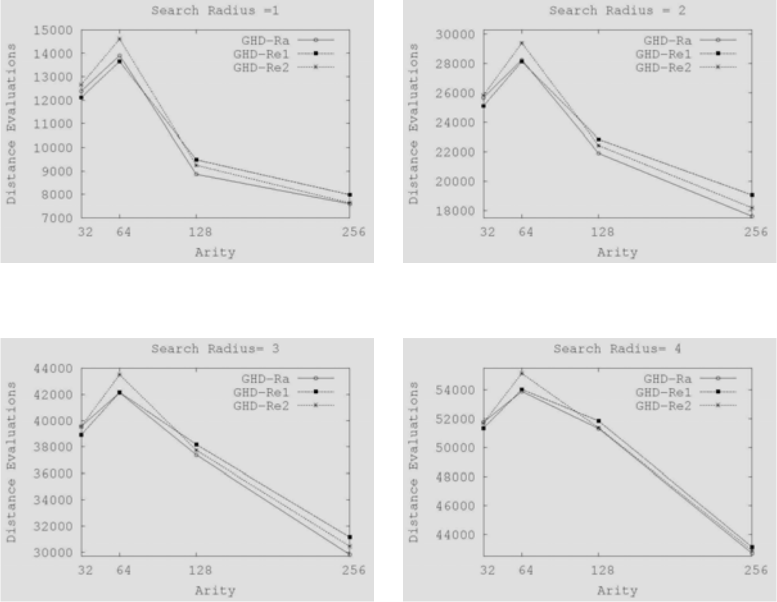


Fig. 4. Average number of distance evaluations as a function of the arity, for *GHD-Ra*, *GHD-Re*$_1$, and *GHD-Re*$_2$. On the top, search radii 1 and 2, on the bottom search radii 3 and 4

When the arity increases, *GHD-Ra* increase the number of distance evaluations saved respect to *HD*. To analyze this behavior it is important to note that *HD* at each step only uses the local histogram of the last chosen center, while the *GHD* variants use the information provided by the all centers already selected, i.e, *GHD* variants use an approximation of the global hard kernel. When the number of centers increases, *GHD-Ra* can accumulate more information in the set θ and can obtain a better approximation of the global hard kernel.

The results obtained using the metric spaces based on the rest of the dictionaries followed the same pattern as the Spanish dictionary.

4. Conclusions and Future Work

In this paper, a new center selection technique, called *global high density zone*, have been introduced. The aim was overcomes a weakness identifies in *high density zone* technique. We present three variants of *global high density zone*, denoted by *GHD-Ra*, *GHD*-Re$_1$, and *GHD-Re*$_2$.

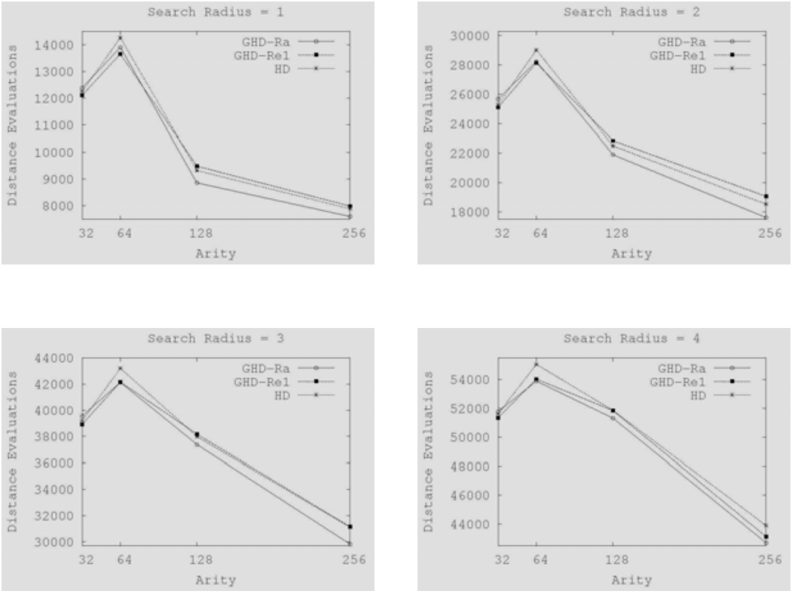


Fig. 5. Average number of distance evaluations as a function of the arity, for the most competitive *GHD* variants and *HD*. On the top, search radii 1 and 2, on the bottom search radii 3 and 4

The *GHD-Ra* and *GHD-Re*$_1$ variants were the most competitive achieving a good performance in almost all arities considered. This suggests that it is more convenient to use an approximation of the global hard kernel instead to work with the viewpoint that a single center has of the metric space.

The presented policies assume to metric spaces with bell shaped distance histograms, but some metric spaces do not satisfy this condition. As future work, we propose to study the behavior of these policies over these metric spaces.

References

1. R. Baeza-Yates, B. Bustos, E. Chávez, N. Herrera, and G. Navarro. Clustering in Metric Spaces and Its Application to Information Retrieval. Kluwer Academic Publishers, 2003. ISBN 1-4020-7682-7.
2. S. Brin. Near neighbor search in large metric spaces. In Proc. 21st Conference on Very Large Databases (VLDB'95), pages 574-584, 1995.
3. E. Chávez, G. Navarro, R. Baeza-Yates, and J.L. Marroquín. Searching in metric spaces. ACM Computing Surveys, 33(3): 273-321, September 2001.
4. N. Herrera C. Mendoza Alric. Center selection techniques for metric indexes. Journal of Computer Science & Technology, 7(1):98-104, 2007.

VI

Architecture, Nets and Operating Systems Workshop

Modifying the Behaviour of Minix System Calls through the Redirection of Messages

PABLO ANDRÉS PESSOLANI

Departamento de Sistemas de Información - Facultad Regional Santa Fe
Universidad Tecnológica Nacional - Santa Fe - Argentina
ppessolani@frsf.utn.edu.ar

Abstract. *Minix 3 is an open-source operating system designed to be highly reliable, flexible, and secure. The kernel is small and user processes, specialized servers and device drivers runs as user-mode isolated processes. Minix is a client/server operating system that uses message transfers as communication primitives between processes. Minix system calls send messages to request for services to the Process Manager Server (PM) or the File System Server (FS), and the it waits for the results. The request messages refer to destination processes with fixed endpoint numbers for each server. This article proposes changes to the Minix kernel that allow the redirection of messages to different servers other than the standard FS or PM, without changes in the source code or binary code of programs.*

Keywords: *Operating System, microkernel, IPC, message transfer.*

1. Introduction

Minix [1] is a complete, time-sharing, multitasking Operating System (OS) developed from scratch by Andrew S. Tanenbaum. It is a general-purpose OS broadly used in Computer Science degree courses.

Though it is copyrighted, the source has become widely available for universities for studying and research. Its main features are:

- *Microkernel based:* Provides process management and scheduling, basic memory management, IPC, interrupt processing and low level Input/Output (I/O) support.
- *Multilayer system*: Allows modular and clean implementation of new features.
- *Client/Server model*: All system services and device drivers are implemented as server processes with their own execution environment.
- *Message Transfer Interprocess Communications (IPC)*: Used for process synchronization and data sharing.
- *Interrupt hiding:* Interrupts are converted into message transfers.

Minix 3 is a new open-source operating system [2] designed to be highly reliable, flexible, and secure. It is loosely based somewhat on previous versions of Minix, but is fundamentally different in many key ways. Minix 1 and 2 were intended as teaching tools; Minix 3 adds the new goal of being usable as a serious system for applications requiring high reliability.

Minix 3 kernel is very small (about 5000 lines of source code) and it is the only code that runs under kernel privilege level. User processes, system servers including device drivers are isolated one from another running with lower privileges (Figure 1). These features and other aspects greatly enhance system reliability [3]. This model can be characterized as a multiserver operating system.

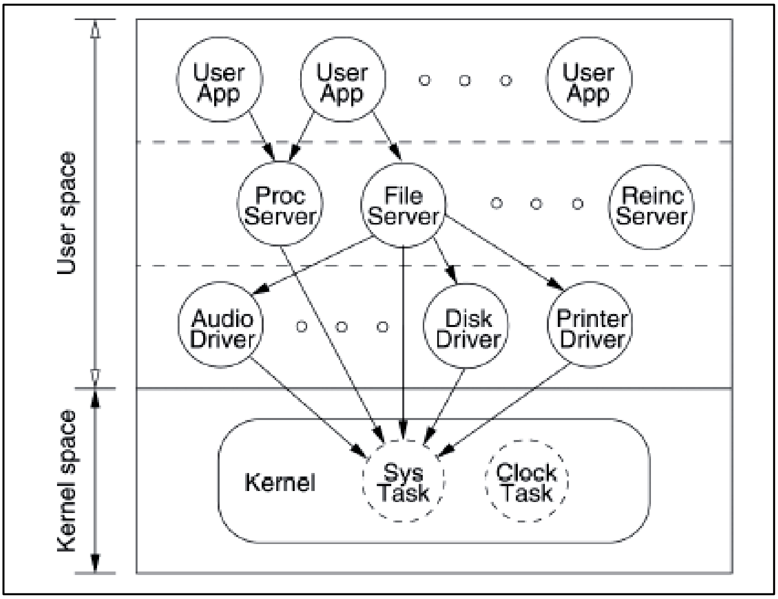


Figure 1. The Internal Structure of Minix 3 [From [4]]

A process makes system calls to request OS services, and the system calls may deliver the user requests to other functions of the OS which process the requests and return the result to the caller. Minix implements system calls using message transfers, packaging function arguments and the results in the same way as RPC does. The following source code shows how it is done:

```
PUBLIC int _syscall(who, syscallnr, msgptr)
int who; /* destination server i.e. PM or FS */
int syscallnr; /* System Call number */
register message *msgptr; /* pointer to the message */
{
  int status;
  msgptr->m_type = syscallnr; /* System Call number */
/* Send the Request and wait for the Reply */
  status = _sendrec(who, msgptr);

  if (status != 0) {msgptr->m_type = status; }
    if (msgptr->m_type < 0) {errno = -msgptr->m_type;
          return(-1);  }
        return(msgptr->m_type);
}
```

The POSIX system call *getpid()* is implemented sending a GETPID request to the PM and waiting for the reply from the server:

```
pid_t getpid()
{
  message m;
  return(_syscall(PM, GETPID, &m));
}
```

PM is a constant define as:

```
#define PM                   PM_PROC_NR
#define PM_PROC_NR  0   /* process manager */
```

The destination process is hard coded into the system call as a constant restricting that system calls can only be served by PM or FS Servers.

The development of some useful features as remote process execution, multiple filesystems support, multiple processing environments or personalities, proxy and gateway servers, security reference monitors, system call profiling, etc. would be simplified if a system call message transfer would be served up by other servers without the need of changing the program. The problem and the solution were described by the Minix`s author, prof. Andrew Tanenbaum:

"Currently FS_PROC_NR is defined as a hard constant (1). Instead, it could be a per-process entry in the process table, so when a process sent a message to 1, this would tell the kernel to look up the real number in the process table. This would mean every process could have its "own" file server. Same for PM_PROC_NR. For a specific process, the number of the "File Server" could be a user-level gateway process that had a permanent TCP connection to a remote server. The command that the user sent would then be forwarded to gateway locally and from there it would be forwarded to the remote machine and executed there. That would allow using remote file systems. On the remote machine would be another gateway process that did the work and marshalled and returned the answer…."

The Redirection of Messages can satisfy those needs. The term *redirection* basically refers to send a message to an entity but really it is deliver to other.

This article examines several approaches and pieces of modified or added source code to the Minix 3.1.2a kernel as a proof of concept of Redirection of Messages. It must be clear that it is not a definite and refined version of Minix 3.

The rest of this article is organized as follows. Section 2 is an overview of Minix 3 system calls implementation, Section 3 describes the Redirection of Messages mechanism. Section 4 refers to the proposed *relay()* IPC primitive. Finally, Section 5 presents conclusions and future works.

2. Overview Minix 3 System Calls Implementation

All processes in Minix 3 can communicate using the following IPC primitives:

- *send():* to send a message to a process.
- *receive()*: to receive a message from a specified process or from any process.

- *sendrec():* to send a request message and to receive the reply from a process.
- *notify():* a non blocking send of a special message type.

Those primitives are implemented as CPU traps that change the processor from user-mode to kernel-mode.

As it was mentioned in the previous section, Minix uses message transfers to implement system calls. Usually, the destinations of request messages are the PM server and FS server.

Minix does not have a single process table, it is scattered among servers and the kernel. The kernel process table keeps attributes, status and statistical information of each process. The FS and PM have their own process tables with fields with specific information that they need.

The kernel process table has (NR_TASKS+NR_PROCS) entries (See Figure 2) where NR_TASKS counts the following special tasks:

- *The Idle task*: It runs when no other runnable process can be scheduled.
- *The Clock task*: It accepts only messages from the timer interrupt handler. It keeps the *realtime* variable that counts timer ticks.
- *The System task*: It represents the kernel and shares its address space (it is like a kernel thread). It accepts requests for special kernel services (called *kernel calls*) from drivers and servers and carry them out.
- *A bogus Kernel task*: Really this task does not exist but its process number is used by interrupt handlers as the source process when they send *notify()* messages to device driver tasks.

NR_PROCS entries are available for servers, device drivers tasks and user processes. It can be specified in a configuration file but the operating system must be compiled completely.

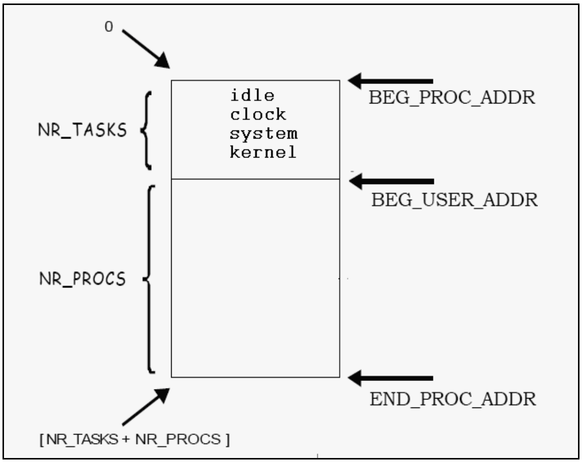


Figure 2. Minix 3 Kernel Process Table

The kernel *proc* data structure that describes a process has two fields related to message tranfers:

```
proc_nr_t    p_nr;  /* index of this entry in the table  */
int          p_endpoint; /* endpoint number */
```

The *p_nr* field is the index of the entry in the kernel process table minus NR_TASKS. Therefore, the first process in the table (*proc[0]*) has *p_nr = (-NR_TASKS)*. The reader should not confuse *p_nr* with the PID of the process. The former is a number for system internal use related to the kernel process table slot, and the latter is a number that identify Minix processes to be used as a parameter in system calls for processes management, such as adjusting the process's priority.

The *endpoint* field uniquely identifies a single processes with respect to IPC and its associate the *p_nr* field with the *generation* number of the slot. Each slot has a *generation* number that counts how many processes has occupied that slot. Each time a new process occupies the slot, the *generation* count is increased. This action prevents that a message addressed to a dead process that has previously used the slot will be delivered to the new one.

The kernel implements IPC primitives using a function with the confusing name *sys_call()* defined as follow:

```
int sys_call(call_nr, src_dst_e, m_ptr, bit_map)
```

The parameter *call_nr* really is the code of an low-level IPC primitive (SEND, RECEIVE, SENDREC, NOTIFY, explained in Section 4).

The *src_dst_e* parameter is the source/destination endpoint number.

The *m_ptr* parameter is a pointer to the request message, and the *bit_map* parameter is a bitmap of flags that change the behavior of the call.

3. Redirection of Messages

Redirection of Messages refers on resolving the process number of destinations process (*p_nr*) through a table instead of setting it as a constant. It requires adding new kernel data structures and making some modifications of the *sys_call()* function to send/receive messages to/from endpoints that are referred indirectly.

3.1. Data Structures

The following approaches were analyzed for Redirection of Messages:

- *Two new fields into the process data structure*: Adding one field for the PM process number, and other field for the FS process number. I.e. when a user processes makes a system call to the PM or FS, the kernel gets the destination's endpoints from those fields. This approach limits the Redirection of Messages only to user process and POSIX system calls.
- *A per process Servers Table*: Each process has its own servers table where the index is the *p_nr* of the server used to get the server endpoint.
- *A fixed number of system wide Servers Tables*: As possibly not all processes need Redirection of Messages, and surely some of them could use the same table, only a fixed number of servers tables are needed. Each

table represents an execution environment that can be set to each process similar to the *priv* table that Minix uses for privileges management.

A convenient table size could be (NR_PROCS+NR_TASKS) to allow the Redirection of Messages not only to user-space processes but to system processes and tasks too. The experimental version has NR_SVRTABS number of tables named *svrtab[]* with (NR_TASKS + NR_PROCS) entries each.

```
proc_nr_t   svrtab[NR_SVRTABS][NR_PROCS+NR_TASKS];
```

A new field in the kernel process descriptor was added to store the servers table that the process will use in system calls message transfers. This field is named *p_svrtab,* and represents the execution environment for the process. The servers table establishes the set of servers that will reply for system calls requests for the process environment.

3.2. Data Structures Initialization

The initialization code sets the numbers of default servers used by Minix for all tables. System programmers can change the servers numbers of a table to redirect some system calls messages to new servers leaving the other servers numbers unchanged for a standard behaviour.

The kernel *svrtab[]* is initialized in *main()* function of the kernel as it is shown in the following source code:

```
void init_svrtab(void)
{
  int i, j;
  for( j = 0; j < (NR_PROCS+NR_TASKS); j++)
        for (i = 0; i < NR_SVRTABS; i++)
               svrtab[i][j] =  (j-NR_TASKS);
}
```

All entries in *svrtab[]* are initialized with the corresponding *p_nr*, therefore the j-th entry of each table is initialize with the value (j-NR_TASKS) as it can be seen in Table 1.

Table 1. Kernel Servers Table – *svrtab[]*

Server	svrtab[0]	svrtab[1]	svrtab[2]	svrtab[3]	….
-4	-4	-4	-4	-4	….
-3	-3	-3	-3	-3	….
-2	-2	-2	-2	-2	….
….	….	….	….	….	….

The *p_svrtab* field of a process is initialized with the value 0, therefore it use *svrtab[0]* as its default table. This means that if *svrtab[0]* table has not been changed, the servers numbers will be the same as in the official Minix version, therefore the system calls will have the standard behaviour.

```
  #ifdef MSGREDIR /* rp points to the process descriptor */
  rp->p_svrtab = 0; /* Servers Table = 0  */
  #endif
```

The *p_svrtab* is a process attribute that will be inherited by its children when the process forks. The internal function of the SYSTEM task copies this field from parent to child process descriptor.

```
#ifdef MSGREDIR /* rpc points to child's  descriptor */
             /* rpp pointes to parent's descriptor */
    rpc->p_svrtab = rpp->p_svrtab;
#endif /* MSGREDIR */
```

3.3. Changes in IPC Primitives

As system calls use the *sendrec()* IPC primitive, the kernel function *sys_call()* was modified to apply Redirection of Messages as it is shown in the following source code.

```
#ifdef MSGREDIR
  if (function == SENDREC) {
    old_sd_e = src_dst_e; /* save original endpoint */
    src_dst = _ENDPOINT_P(src_dst_e);
    old_sd  = src_dst;  /* save original process number */
    new_sd =  /* get the new process number from table */
      svrtab[caller_ptr->[p_svrtab] [src_dst+NR_TASKS];
    new_sd_ptr = proc_addr(new_sd); /* get the pointer */
    /* change the original endpoint */
    src_dst_e = new_sd_ptr->p_endpoint;
  }
#else
    …. original source code of Minix ……
#endif
```

The user process will be deceived that it sends the request to the server specified in *src_dst_e* parameter, but really the request it will be sent to the server obtained from the process' *p_svrtab* servers table.

In Minix, user processes can't send any message to any other process. They can only send messages if they have the correct permissions for the destinations. Therefore, the process privileges to execute a system call are checked against the permissions to send to the standard server (***old_sd_e***) instead of the permissions of the new server to keep compatibility.

3.4. Auxiliary System Calls and Functions

Two basic auxiliary system calls were added to manage servers tables:

– *int setsti(int tabnbr, int index, int value):* The *Sets Servers Table Index* system call sets the *index*-th item of table *tabnbr* with the specified *value*.

– *int getsti(int tabnbr, int index):* The *Gets Servers Table Index* system call returns the value of the *index*-th item of table *tabnbr*.

A modified version of the *fork()* system call named *tfork()* was added to set the servers table number of the child process. The *tfork()* system call has the following C declaration:

```
pid_t tfork(int ptabnbr);
```

The parameter *ptabnbr* is the servers table number to be set for the child process.

The *tfork()* calls two auxiliary functions:

1. *sys_fork()*: This is the standard Minix function to create a new process (the child) and returns its PID.
2. *sys_setpsvrtab():* It sets the process' *p_svrtab* field to the value specified in the *ptabnbr* parameter.

The function that shows the kernel process table on system console was changed to print the ***svrtab*** field of each process. As it is shown in the following screen output, the process named **xtest** has *p_svrtab=1,* and the process **inet** has *p_svrtab=0.*

```
-nr---svrtab--endpoint--name--- -prior-quant- -user---sys----size-rts
 48        1       71126  xtest    07/07 08/08       0     1      52K -pm
 51        0       35590  inet     03/03 04/04       5     0     900K ANY
```

An auxiliary function named *svrtab_dmp()* was added to the Information Server (IS) to dump on console screen the servers tables when the **Shift-F9** keys are pressed on the console keyboard.

The following console output shows that the servers table 1 has the value 30 for the 28th entry. Those processes that have *p_svrtab=1* that make system calls to the server with process number 28, really they will make the system calls to the server with process number 30. Those processes that have *p_svrtab≠1* will make the system calls to the server with process number 28.

```
# <PRESS Shift-F9>
index  [   0]  [   1]  [   2]  [   3]  [   4]  [   5]  [   6]  [   7]  [   8]
[ 26]     26      26      26      26      26      26      26      26      26
[ 27]     27      27      27      27      27      27      27      27      27
[ 28]     28      30      28      28      28      28      28      28      28
[ 29]     29      29      29      29      29      29      29      29      29
[ 30]     30      30      30      30      30      30      30      30      30
[ 31]     31      31      31      31      31      31      31      31      31
```

4. The *relay()* IPC Primitive

A new IPC primitive named *relay()* was added to help system programmers to implement proxy services, gateways, security reference monitor, system call interception, intrusion detection system or confinement software [5].

When a user program makes a system call to a server, the request would be redirected to an alternative server (may be a proxy or gateway) using Redirection of Messages that process the requests and return the result to the caller, or it could forward the request message (perhaps previously modified) to the original destination server using *relay()* (Figure 3). A similar technique called *trampoline function* is used by other OS, but as function relay (not message relay) that *bounce* a call to other function (hence the term *trampoline*).

Minix does not have IPC primitives that allow sending a message from a source process to a destination process through a third process (the caller).

To use *relay()* the following actors must be distinguished:

- *Source*: The process (i.e. a user process) that makes a system call to a server process (i.e. PM or FS) using the *sendrec()* primitive.
- *Destination*: The process the deals with system calls (i.e. PM or FS).
- *Caller*: The process that receives the request message from the source process through Redirection of Messages and will forward it to the destination process.

The *relay()* function has the following C declaration:

```
int relay (int src_e, int dst_e)
```

where *src_e* and *dst_e* are the source and destination endpoints.

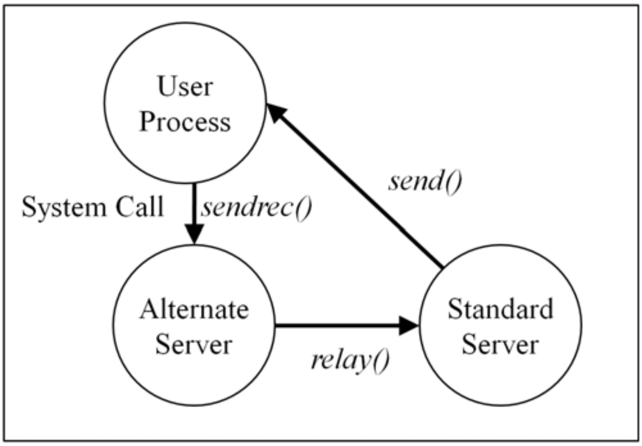


Figure 3. Sample use of *relay()*

The message itself it is not needed as an argument because it will be copied from the source message buffer to the destination message buffer.

The use of *relay()* assumes that the caller has received a request from the source process (using *sendrec()*), therefore the source process is waiting for the reply from the caller.

The kernel checks that the source process has the RECEIVING bit in the *p_rts_flags* field set to indicates that it is waiting for the reply message (line 0008), and the *p_getfrom_e* equals to the endpoint of the caller process to indicates that is waiting for the reply from it (line 0010).

```
0001 switch(function) {     /* function is the IPC code */
0002 #ifdef MSGRLY
0003 case RELAY:           /* for RELAY IPC            */
0004   src_p = _ENDPOINT_P(src_e); /* source process   */
0005   src_ptr = proc_addr(src_p); /* source endpoint */
0006   dst_p = _ENDPOINT_P(dst_e); /* dest. Process    */
0007   dst_ptr = proc_addr(dst_p); /* dest. Endpoint   */
0008   if(src_ptr->p_rts_flags != RECEIVING)
0009         return(EBADSRCDST);
0010   if(src_ptr-> p_getfrom_e !=
0011      caller_ptr->p_endpoint)
0012         return(EBADSRCDST);
0013   result = mini_relay(src_ptr, dst_ptr);
0014   break;
```

```
0015 #endif /* MSGRLY */
```

The *mini_relay()* kernel function is like *mini_send()* function that send the message to the destination process, but the caller process never blocks.

5. Conclusions and Future Works

Minix has proved to be a feasible testbench for OS development and extensions that could be easily added to it. Its modern architecture based on a microkernel and device drivers in user-mode make it a reliable operating system. The message transfer is the paradigm used by Minix to implement system calls, task calls and kernel calls.

A drawback of Minix implementation is the fact that system calls are server by FS and PM. If new system calls need to be added, some kernel source code constants must be modified and the system must be recompiled.

The proposed Redirection of Messages mechanism allows that multiple servers and drivers could execute concurrently and be interpreted as different environments for processes. A user process could use the standard filesystem server, but other process could use other servers that support EXT2/3/4, FAT16/32, VFAT, NTFS, etc., or remote filesystems through a file system proxy or gateway server.

This article describes the use of Redirection of Messages only applied to user level processes and system calls, but the same approach would be applied to servers and drivers processes. New IPC primitives, like *relay()* are needed to take advantage of those facilities.

The reliability and robustness of Minix 3 would be improved with Redirection of Messages and the *relay()* IPC primitive. The primary/backup approach [6] for servers or drivers could be implemented easily. A server or driver would receive a request from a user or server process and could replicate the request to a primary server or driver and to a backup server or driver that could be local or remote.

The proposed extensions can be used to develop a variety of security related functions such as custom auditing and logging, fine grained access control, intrusion detection or confinement.

The author is working on his PhD. thesis about a *Distributed Multiserver based Operating System as a Middleware* where servers and drivers register their services and versions on different machines, making use of Redirection of Messages and the *relay()* system call to provide new facilities in the field of modern operating systems.

References

1. Tanenbaum, Woodhull. "*Operating Systems Design and Implementation, Third Edition*". Prentice-Hall, 2006.
2. MINIX3 Home Page. http://www.minix3.org/

3. Herder, "*Towards A True Microkernel Operating System*", master degree thesis, 2005.
4. Herder, Bos, Gras, Omburg, Tanenbaum. *"Modular system programming in Minix 3".* ;Login: April 2006.
5. K. Jain, R. Sekar; "*User-Level Infrastructure for System Call Interposition: A Platform for Intrusion Detection and Confinement"*; Iowa State University.
6. Budhiraja, Marzullo, Schneider, Toueg. "*The Primary-Backup Approach*". Cornell University.

On the Road to a Global Wireless Sensor Network

EDUARDO O. SOSA[1,A], STEFAN FISCHER[2,B], FRANCISCO JAVIER DIAZ[3,C]

[1] Facultad de Ciencias Exactas Químicas y Naturales, Universidad Nacional de Misiones,
Félix de Azara 1552, N3300LQH Posadas, Argentina

[2] Institut für Telematik, Universität zu Lübeck, Ratzeburger Allee 160,
23538 Lübeck, Germany

[3] Laboratorio de Nuevas Tecnologías Informáticas, Universidad Nacional de La Plata,
Calle 50 y 120, 1900 La Plata, Argentina

[a] es@fceqyn.unam.edu.ar, [b] fischer@itm.uni-luebeck.de, [c] jdiaz@unlp.edu.ar

Abstract. *This work has considered the development, testing, configuration and implementation of a wireless sensor network infrastructure. This infrastructure on a testbed scale has been used primarily for research, oriented to an interdisciplinary approach that encompasses the hardware, software, algorithms and data. It sought to demonstrate that the heterogeneity of devices and small existing structures may coalesce into well-organized, large-scale, different existing grids that will enable a quantitative and qualitative research to a much larger scale by addressing the dynamic changes of venues, infrastructure and composition of nodes. Considering that a real implementation is usually a complicated task, we have performed a network simulation and a real testbed. Also a contribution to the establishment of a global network of wireless sensors has been highlighted, distinguishing three sub domains in the global network considered: the existing top network, the sensing devices and the servers.*

Keywords: *WSN, testbeds, infrastructure*

1. Introduction

Mark Weiser envisioned that the development of ubiquitous computing is the future scenario, and that it will interact seamlessly with human users [1], [2]. Ubiquitous computing is synonymous with the evolution of electronic devices through the miniaturization and the increased socialization of online content.

Today it is possible to use some tools to visualize, through numerical simulations, the phenomena that we cannot observe by empirical methods. This phenomenon could be understood as a result of the development of the properties and capabilities of the underlying semiconductor technology. Moore's Law establishes that the number of transistors on a circuit, and thus its storage and calculating capacity, doubles every year or two [3]. The lower costs in the market reached by the different parts of circuits, contributes to the low prices of almost all computing and communication devices. It is for this reason that IT is springing up and developing in an increasing scope of network linked physical devices. Moreover, as component technologies are becoming smaller, faster and cheaper it will continue to do so.

Miniaturization is a reality in most aspects of everyday life, as it also is the capacity of embedding computing and communications technology of spreading far and wide.

While this is achieved with the proposed improvement in calculation and storage performance, research on the application of this technology is designed to reliably determine which will be the new role of computers in science during the XXI century. IT will eventually become an invisible component of almost everything in everyone's surroundings.

Ubiquitous computing fits a large number of technologies and applications, from mobile devices, "smart" artifacts for special purposes.

A special analysis of radio frequency identification (RFID) becomes necessary. It consists of a system to storage and retrieve remote data, which is used by devices called tags, cards, transponders or RFID tags. The main purpose of this technology is to transmit the identity of an object, similarly to a code or serial number, using radio signals. The elements needed to enable the system are the RFID tags and RFID readers, elements that allow the physical object "be seen" and monitored by an existing computer network.

Scientifics and researchers talk more and more about a "ubiquitous network society" [4], [5], a society where networks and networked devices are omnipresent.

When talking about worldwide competition, efficient IT systems which can supply comprehensive and in time data, are essential, and this conveys the need of a continual stream of information from industry appliances to business applications, uniting thus many eclectic systems.

There should have some devices that could be used as an interface for the physical world to become more "friendly" with computer networks, making the conditions of objects and their surroundings perfectly and indefectibly accessible to software systems. RFID and WSN offer a wide range of networked and interconnected devices which afford significant information no matter where the user is. Home appliances, automobiles or farm machines can be in a communication range, shifting from today's Internet, the internet of data and people, to the Internet of tomorrow, the Internet of things, namely global communication network springing out of the dissemination of such devices and increasingly developing.

Often, the ultimate goal of an action is to get valid information from the field where we are interacting with. In many situations, it would be difficult or extremely expensive, to carry on a surveillance project by using a system where the interacting elements with the event are connected by physical links such as copper wires. For such situations there is a new technology based on a new paradigm, Wireless Sensor Networks (WSN). These networks should not be considered as another step in the evolution of the personal computer or the Internet, it is rather thought as the beginning of the end of personal computers.

A WSN has a number of exclusive characteristics when compared with conventional wireless networks, given that they are related to narrow bandwidth, low computation capability of the nodes, and limited lifetime. Self organization, dynamic network topology, and multi-hop routing are additional key possible features of a WSN, which make them important for many applications.

Of course it is advantageous to perform precise simulations or to develop models before deploying WSNs in the field. This is because WSNs may be deployed randomly in an ad-hoc manner with a large number of tiny nodes. Simulations help to evaluate the performance of sensor networks within certain application environments, something impossible to achieve not many years ago.

We cannot feasibly model analytically a WSN and what is more, deploying testbeds means a great effort; consequently, if we are to study WSN we must resort to simulation. To do this we need a suitable model based on solid assumptions and a suitable framework to easily accomplish it [7]. Besides, the results of the simulation which are dependent on the environment, the hardware, and certain other assumptions, often not sufficiently precise to depict the actual behavior of a WSN, jeopardize much of the results. Usually most simulators bring forth issues related to scalability and performance; this is so because of the many devices depicted on the simulated applications.

At present there is not such an active testbed in Argentina, and as result of this work, a fully operative WSN is ready to serve as scenario for any joint research-development project with national or foreign partners. Thus practical result will serve to support research and development devoted to the subject in the region.

However, in most cases, when assessing the needs at normal scale, the implementation of a real pilot testbed is usually a complicated task, often for financial reasons. Whether possible, an integrated use of possibilities should be performed, this means to investigate the network simulator and the real testbed. The physical implementation of testbeds stands for accurate and replicable testing; nevertheless, this approach experiences two serious limitations, namely a) large scale, due to costs; and b) not replicable environment. The major challenges for the use of simulators and testbeds are: 1) *Sensor node simulation*. 2) *Testbed visualization*. 3) *Interaction between simulated scenario and the practical deployment* [8].

Whenever an experience can happen in the real world it is quite engaging since it is a confident way to demonstrate that the application can perform a precise assignment wherever the technology is at hand. Yet, unpredictable environment influences make it quite difficult and nearly impossible to present the results and to set apart sources of error, that influences the problem to be considered [9]. Therefore normally testbeds are often limited to a few dozen devices [10]. In future scenarios the building of networks will consider a very high number of nodes, reaching up to several thousand or even millions [11], [12].

Although there exist frameworks for general purpose, to develop a new application in a specific field will demand large experience and skill in a wide scope of technologies. Consequently, smart devices and WSNs are yet not fully exploited. Concerning the integration of application for business, there have been so far few projects which have empowered new architectures with more flexible software. Still, the use of hybrid web applications (mashups) has mainly been used on on-line service and not on physical world analysis, which is our main intention.

The practical objective of this work intends to report the development, testing, configuration and implementation of a WSN infrastructure. This infrastructure on a testbed scale should be used primarily for research, pointing to an interdisciplinary approach that encompasses the hardware, software, algorithms and data. It seeks to demonstrate that the heterogeneity of devices and small existing testbeds structures may coalesce to form well-organized large-scale, different existing grids that will enable a quantitative and qualitative research to a much larger scale by addressing the dynamic changes of venues, infrastructure and composition of nodes.

2. Material and Methods

When we talk about WSN to monitor micro environments, we note that because of its auspicious and encouraging capability it is becoming very popular. Even though, the majority of systems used for monitoring environment that appear in the literature are those applied only in specific applications and with no use of functions that may work on the user's data processing methods. The hardware working style we have chosen is one of application-oriented, whereas the system platform to acquire information, validation, processing and visualization is orderly presented. The system capability to draw forth useful information can be guaranteed through several approaches.

Sensors connect to Internet creates endless opportunities in terms of applications and services on emerging patterns of operation. Internet users will be able to obtain real time information from our physical scenario over any item, anywhere and anytime. These investigations have been carried out in parallel, and often in isolation, making it impossible to establish a unified global framework.

The results of experiences can reveal the path reliability as well as real-time characteristics, and can also show the system viability and capability in practical use. Figure 1 shows the system architecture of the testbed installed, where the three levels involved can be observed.

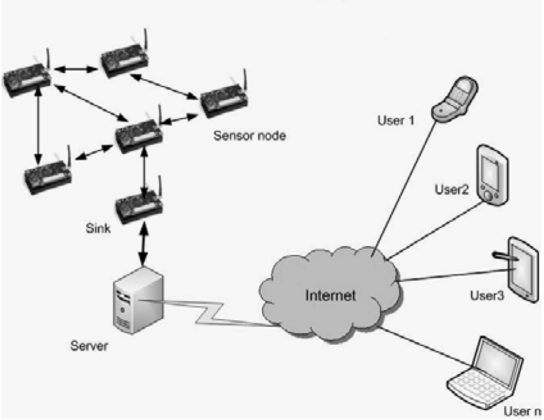


Fig. 1. System Architecture of the deployed testbed

The infrastructure level is formed by the WNS nodes and the sinks. These devices self-organize themselves into a WSN, generating continuous information data packets. This information consists of temperature, humidity and light intensity near the nodes. If we are to design this tier, the choice of the manner of sensing, of reporting and the packet routing protocol must be decisive.

The server level: captures and processes any packet transferred by the sink, in this case through an USB cable; this information afterwards is forwarded to the upper processing program where they are processed. Besides; at this level it is possible to send the packets received over the Internet, where the information can be read by the user.

The user level: Data can be accessed by remote users with web browsers in a way that they can assemble/collect and see the packets from the Server, thus being able to visualize a real-time monitoring of the system. Concerning the security of WSN, users can only survey data and they are not allowed to change parameters of the basic tier.

Although some aspects in this WSN may be generic, the specific demands of the application are significant, especially in scenarios like environmental monitoring. When doing this, a number o sensors are deployed in an area for measuring meteorological parameters, namely temperature, speed and direction of winds, moisture and pressure; and as they tend to change quite slowly, sparse sampling is conceded. Usually, nodes use to measure their contiguous space and send data packets in any of the three following ways: timer-based, event-driven, and requirement-based.

In our micro-environmental monitoring network, we did not have troubles with the energy demand of the nodes, because of the technology of the nodes applied. The only task for sensors is to measure and send the information in due packets according to approximately the internal timer frequency. The more the measuring instances are, the greater the energy used and consequently a shorter lifetime. On the other hand, lower frequencies of measurements leads to a lack of sensitivity of environment changes.

It is widely known that applying multihop techniques in WSN renders a longer lifetime saving much energy from the batteries. So far, we have not experienced such a scarcity because of the solar energy harvesting system provided by our *weather station nodes*. Furthermore as the information is relayed to sinks, scalability is enhanced.

In a typical WSN application, the observer is interested in tracking phenomena under certain restrictions of latency and accuracy. In a typical WSN, each individual sensor node performs the measurement of the values required, and disseminates this information to other members of the network, and eventually, to the observer. The different events of a given phenomena are measured as discrete samples, which will depend on the precision and accuracy of the sensors, and the location thereof. We have deployed hardware provided by iSense [13] as platform for this experimental work.

Since our WSN uses the iSense operating system and firmware, the generation of small, but complete applications, are allowed. This provides a solid foundation for rapid application development. It furnishes a C++ API to the hardware of the node, functionalities of operating system and a wide variety of network protocols.

Other personal computers were used as platforms for application development which would then be transferred to the sensing devices. In these PCs the needed packages have been installed that would allow to develop applications in C++ and compile them, so the final application could be distributed to the different nodes of the WSN. For these tasks, a Linux+PC platform under Ubuntu and Debian distributions was adopted. The necessary applications have been installed such as: make, cmake and gcc++. The iSense platform is based on a Jennic processor, which depends on the ba-elf-g++ integration of libraries and compiler to assure well done applications. The figure 2 shows the comparative size of the device to a coin.

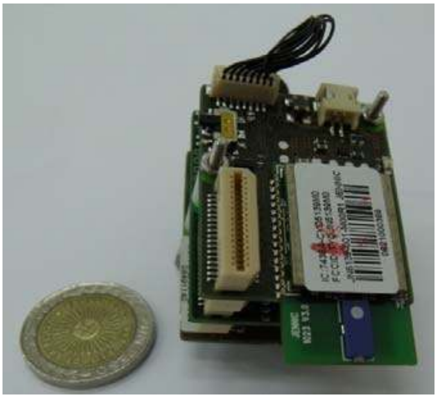


Fig. 2. iSense core and attached modules compared with a coin

During the running period we have used several modules attached to the main one, namely: gateway module, Security Sensor module (infrared PIR and 3-axis accelerometers), weather sensor module (temperature, relative humidity and barometric pressure), and environmental sensor module (thermometer and light intensity). In these experiences an energy harvesting system has been adopted, to generate energy and store it in a rechargeable battery, allowing the nodes to run all-time autonomously, using the power management module.

The program language used for developing applications for the sensor nodes is C++. The compiler used on both Linux and Microsoft Windows environment was GNU Compiler Collection. "*GCC, includes front ends for C, C Object-oriented, C++, Fortran, JAVA and ADA, as well as libraries for these languages (libstdc, libgcj, etc)*" [16].

As a simulation environment Shawn was implemented[14]. For testing, configuration and deployment of applications developed in the nodes, the iShell tool [13] was adopted

The iShell tool is the counterpart of the network firmware and it is used for testing and operation of the iSense nodes. The main benefit of using iShell is that it allows the simulation of Shawn applications without any extra effort, because applications are compiled according to different "targets".

In order to keep all the needed tools in a rational context, which simplifies the task of programming, debugging and compilation of programs; Shawn and iShell have been integrated to Eclipse [15].

Shawn has been selected as a simulator for this work because of four parameters: scalability, completeness, fidelity and bridging. To be scalable, a simulator should manage such networks conformed up to thousands of nodes in a set according to diverse configurations. The simulations are intended to be as realistic as possible by simulating the physical, data, encryption of messages, effects of wireless interference, limitations of the processor, etc.

3. Results

Several scenarios have been simulated in order to be set as a model in the sensors, having to do with communication and location, considering till 200 nodes deployed in a two square miles. Figure 3 shows the nodes with GPS (anchor nodes) as black. They "know" their location. The gray nodes are those that calculate its position by algorithms that relate to the position of the nodes *anchor*.

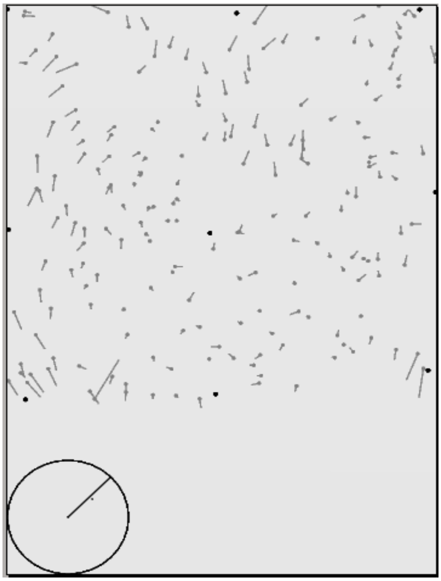
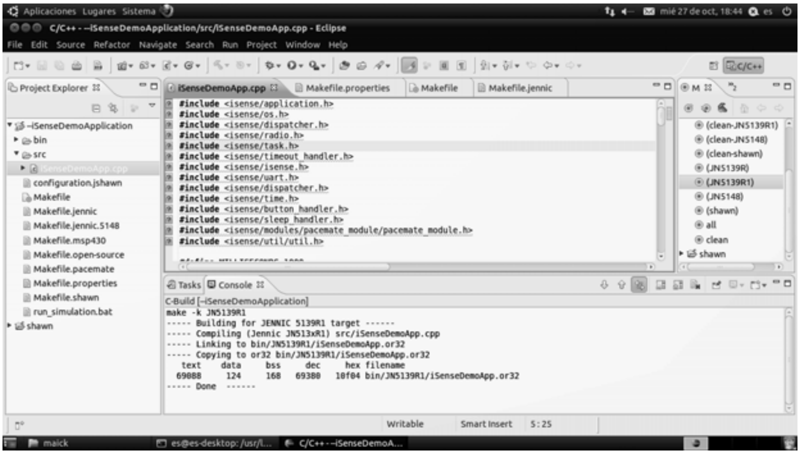


Fig. 3. Location simulation with Shawn

A great number of ready-to-run services and protocols are provided by iSense software which avoids the need of installing the applications node by node, etc.

The fact that iSense WSN software allows to build applications to be run directly in the Shawn simulator, is considerably advantageous. With this tool, applications related to solar-powered sensors have been developed as well as applications for passive infrared sensors, accelerometers, temperature, light intensity and movement detection. In addition, all codes generated are ready to be implemented in a Shawn simulator.

The iSense firmware has been imported into Eclipse, along with Shawn. In such an environment it is simple to select the coded file in C++ (.cpi) and to choose the type of processor used by the nodes. Thus, it generates the executable file compiled and linked with the corresponding libraries, either to nodes with JN5139R1 processor (like ours) or for Shawn simulator, see Figure 4.



```
#include <isense/application.h>
#include <isense/os.h>
#include <isense/dispatcher.h>
#include <isense/radio.h>
#include <isense/task.h>
#include <isense/timeout_handler.h>
#include <isense/isense.h>
#include <isense/uart.h>
#include <isense/dispatcher.h>
#include <isense/time.h>
#include <isense/button_handler.h>
#include <isense/sleep_handler.h>
#include <isense/modules/pacemate_module/pacemate_module.h>
#include <isense/util/util.h>
```

```
C-Build [~iSenseDemoApplication]
make -k JN5139R1
----- Building for JENNIC 5139R1 target ------
----- Compiling (Jennic JN513xR1) src/iSenseDemoApp.cpp
----- Linking to bin/JN5139R1/iSenseDemoApp.or32
----- Copying to or32 bin/JN5139R1/iSenseDemoApp.or32
   text    data     bss     dec     hex filename
  69088     124     168   69380   10f04 bin/JN5139R1/iSenseDemoApp.or32
----- Done  ------
```

Fig. 3. iSense firmware in the Eclipse C/C++ Development Environment

4. Discussion

Despite the fact that WSN have been largely studied in recent years, many of its issues are theoretical and they were not acquired in a practical field. Hence, when it comes for real applications on environmental surveillance, certain inconveniences are found as a rule. These inconveniences may occur affecting the network performance, when a trustworthy delivery and quality service are demanded.

In this work, a WSN testbed has been thoroughly tested in normal weather conditions, behaving as a very robust and suitable tool for these tasks. During the elapsed time of the project, we have faced many challenges. So far, we have learnt that remote management of a WSN is indeed significant.

The presence of multi-hop topologies in wireless communications scenarios is imposing its presence every day in our existence. The fundamental characteristic of these networks, unlike those we have studied many times till now, are its limitations both in terms of lifetime and the computing power of the nodes. Another important consideration is that the development of WSN applications is still complex, being a challenge for distributed applications, and integrated programming.

There exist some elements that further complicate the situation, such as resource limitations of the nodes, the unpredictable influence of the environment, and the size of the networks. Given that the project specifications are normally subject to modifications and applications that evolve over time, we find that changes are inherent to the development of the technology considered. This should be taken into account considering that changes in projects are usually long, expensive and inaccurate.

It can assuredly be foreseen that in the near future the WSN will be equipped with more powerful nodes enabling longer monitoring time, and allowing interdisciplinary work that should involve groups from data networks, as well as from distributed processing.

References

1. Weiser, Mark: The computer for the 21st century.: Scientific American (International Edition), Vol. 265, pp. 66-75, (1991).
2. Mahadev, Satyanarayanan: Pervasive computing: Vision and challenges. IEEE Personal Communications, Vol. 8. (2001).
3. D. Culler, D. Estrin, and M. Srivastava: Overview of sensor networks. IEEE Computer, pp. 37(8):41-49 (2004).
4. Telecommunication Standardization Sector: Technology Watch Briefing Report Series #4, p. 10 (2008).
5. US-National Research Council. Embedded, Everywhere: A Research Agenda for Networked Systems of Embedded Computers. Washington, D.C. : NATIONAL ACADEMY PRESS, 0-309-07568-8 (2001).
6. Guinard, D., et al.: Towards physical mashups in the Web of Things. Sixth International Conference on Networked Sensing Systems (INSS). pp. 1-4 (2009).
7. E. Egea-López, J. Vales-Alonso, A. S. Martínez-Sala, P. Pavón-Mariño, J. García-Haro: Simulation Tools for Wireless Sensor NetworksSummer Simulation Multiconference. pp. 1-9 (2005).
8. Lei Shu, Chun Wu, Yan Zhang, Jiming Chen, Lei Wang, and Manfred Hauswirth.: NetTopo: Beyond Simulator and Visualizer for Wireless Sensor Networks. Second International Conference on Future Generation Communication and Networking, vol.1, pp.17-20 (2008).
9. H. Hellbrück, M. Lipphardt, D. Pfisterer, S. Ransom and S. Fischer: Praxiserfahrungen mit MarathonNet - Ein mobiles Sensornets im Sport. PIK - Praxis der Informationsverarbeitung und Kommunikation, Vol. 4, pp 195-202 (2006)
10. Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., and Culler, D. :An analysis of a large scale habitat monitoring application. 2004. Proceedings of the 2nd international conference on Embedded networked sensor system, SenSys 2004.
11. Vijay, Kumar: Sensor: the atomic computing particle: SIGMOD Rec., 2003, Vol. 32, pp. 16-21.
12. Khan, S.U. and Hamid, M.S. [ed.]: On the optimal number of smart dust particles INMIC 2003. 7th International Multi Topic Conference. pp. 472-475 (2003).
13. Coalesenses GMbH, http://www.coalesenses.com
14. Shawn, http://shawnwiki.coalesenses.com
15. Eclipse, http://www.eclipse.org
16. C++ Library Database http://c-plusplus.org

III

Innovation in Software Systems Workshop

Cohesion among Ontologies: a Technique for Measuring Semantic Integration

MARÍA MERCEDES VITTURINI[1], PABLO RUBÉN FILLOTRANNI[1, 2]

[1] Laboratorio de Investigación y Desarrollo en Ingeniería de Software y Sistemas de Información (LISSI) – Departamento de Ciencias e Ingeniería de la Computación,Universidad Nacional del Sur. Avenidad Alem 1253 Bahía Blanca, Argentina

[2] Comisión de Investigaciones Científicas de la Provincia de Buenos Aires (CIC)

{mvitturi, prf}@cs.uns.edu.ar

Abstract. *Multiple ontologies for GIS environments are accessible via Web. Moreover, each GIS application has its own data model or application ontology. Ontology mapping could provide a common language from which several systems could exchange information and semantics. In this paper we present a novel technique that collaborates with the task of analyzing the degree of cohesion between ontologies and servers in order to anticipate the quality resulting from the integration process.*

Keywords: *Ontology, ontology integration, interoperability, GIS.*

1. Introduction

Various applications using *Geographic Information* (GI) cover a range of systems types, such as Geographic Information Systems (GIS), Spatial Data Infrastructures (SDI), and Location Based Mobile Systems (LBMS). Private companies and national and international research and education organizations are some of their major customers. Generally, GI is unwieldy, has a complex structure and usually is distributed by *theme* over different servers. If a new GIS application is needed, an extra cost to consider is the acquisition of geographic data when unavailable. Sometimes it occurs that the GI data on a topic already exists in some previous development, for instance, information about "routes and roads". The possibility of sharing information and services means lower costs and start-up times, as well as improvement of information reliability.

In the last few years, several software development problems have been faced with the need of sharing and reusing knowledge acquired for a specific domain. This is accomplished by the Semantic Web and it is linked to the notion of interoperability [7, 8, 11]. The goal is to have an unambiguous knowledge of the Web that can be interpreted by automated agents. In particular, there is a need of a Geospatial Semantic Web [4] on a framework comprising various thematic spatial ontologies. The design of some kind of solutions to GI heterogeneity problems is needed in order to share GI. In this way, data could be processed and interpreted by remote systems. Ontology semantic integration is based on several ontologies working as mediators in the communication between systems. For a successful mediation, a semantic mapping between ontologies is required. This task will be effective as long as the concepts of different ontologies are really comparable. In this paper we propose a simple technique based on relationships of concept sets that could help to anticipate the effectiveness of this communication process.

2. Heterogeneity in Geographic Information

For years, each new GIS development defined its own models of storage and visualization for spatial data, in addition to their conceptual data models. GI format diversity involves interoperability problems between GIS's.

2.1 Context

The GI concept encompasses information including spatially referenced data, i.e. linked to one or several points on the surface. GI is characterized by its inherently complex structure and volume.

A *geographic data* is an abstraction that represents a real world object, such as a route, a building, an agricultural area, etc., which has a digital representation. Each object is called geographic *feature* [10]. A geographic feature is unique and distinguishable. A *feature type* is the abstraction that represents sets of geographic features of the same class. A feature type encompasses attributes and relations that model real phenomena. Attributes of a feature type are arranged into *thematic attributes* and *spatial attributes*. The spatial component keeps reference to the Earth's surface. Thematic components maintain the description characterizing each entity. Eventually, a geographic model also includes the definition of geometrical and/or topological relationships between features. In a geographic object, metric properties include length and area -depending on the dimension of the object- and metric relations between objects such as distance and orientation. Topology refers to properties like proximity, adjacency, inclusion, and connectivity that remain invariant to morphological changes of scale or projection.

2.2 Heterogeneity Levels in Geographic Data

In any two given representations of a geographic problem, we will distinguish the following types of heterogeneities [10, 13]:

- *Syntactic Heterogeneity:* for a single phenomenon each solution provides different formats and space representation models -vector or mosaic-, and/or different coordinate representation systems.
- *Structural Heterogeneity:* refers to the "form" that each solution chooses in order to represent the same phenomenon. Many differences are expected to exist in terms of structure between models.

- *Semantic Heterogeneity:* it occurs when distinct solutions interpret different meanings for the same phenomenon.

Table 1 illustrates all these types of heterogeneities. The solution to GI heterogeneity problems encourages research in the field of Computer Sciences.

Table 1. Examples of syntactic, structural, and semantic heterogeneity

Examples of Heterogeneity using Geographic Data		
Heterogeneity	**Application A_1**	**Application A_2**
Syntactic	A_1 represents the areas according to Bahía Blanca population density under the spatial vector model.	A_2 represents the areas according to Bahía Blanca population density under the spatial raster model.
Structural	A_1 represents the areas according to Bahía Blanca population density with details about the distribution of public services.	A_2 represents the areas according to Bahía Blanca population density with details about types of constructions -buildings, private neighborhoods, etc.-.
Semantic	A_1 represents the areas according to Bahía Blanca population density. The unit of measurement used is number of inhabitants.	A_1 represents the areas according to Bahía Blanca population density The unit of measurement used is number of family groups.

3. Proposals on Geographic Information Integration

Research work to make progress towards GI integration is addressed in two different ways. On the one hand, there is research that defines standards that normalize representation models for spatial data. On the other hand, there is research on semantic difference solutions that are generally linked to the definition of ontologies that provide formal specifications. Defining integration rules is only possible if the meaning of data is known.

3.1 Standards in Geographic Information

The international standards for geographic data and services are primarily concerned with the Open Geospatial Consortium (OGC) [11] and the Technical Committee of Standardization on Geomatics and Geographic Information ISO / TC 211 [8]. OGC is an international consortium. Its participants represent business companies, government agencies, and universities. It has a consensus process to develop interface specifications applicable to open source geo processing systems. OGC solutions are referred to as *Open GIS Specifications* and provide interoperable solutions to make the GI "geo-available". OGC mission is to lead to development promoting the use of architectures that allow for the integration of geographic applications.

Meanwhile, the International Standards Organization (ISO) established the Technical Committee for Standardization in Geomatics and Geographic Information ISO / TC 211 to be responsible for defining reference standards for digital GI and for the transfer of data and services. The ISO 19100 family is a set of standards related to geographic features. These regulations deal with methods, tools, and services for managing, acquiring, processing, analyzing, accessing, presenting, and transferring digital GI among different users, systems, and locations.

OGC members also participate in ISO / TC 211 through the Joint Consultative Council ISO/TC211-OGC. Its mission is to coordinate the efforts of both organizations and to ensure a single standard reference.

3.2 Ontology Models

Ontologies unify the interpretation of concepts and terms so that such interpretation can be unique [6]. This is true among people and also when automatic agents are involved in machine communication. Personal communications can solve semantic heterogeneity caused by different conceptualizations, terminology, context or incomplete information. For example, the generalization/specialization relationship between elements is clearly understood by most of people. However, this relationship is not trivial for many search algorithms based on finding matching terms in schemas and data.

The mission of ontology is to provide the formal specification of concepts and their relationships. Figure 1 shows a simple case of ontological concept specification and its relationships. The notation used is proposed by UML. In the example there are *classes* (color) which define the common properties of the elements of the same type, and instances (white) representing a particular concept, occurrence or instance. You can see a hierarchical relationship between classes Source Of Fresh Water and River specified by *IS_A* distinguished relation. Other relations in the example are: *flow into*, which specifies that instances of River lead to instances of Ocean and *ends* which represents a Town where a River ends. The link between instances and their respective class or relation is represented by the stereotyped dependency <<*instantiate*>>. These definitions make it possible to achieve the following basic conclusions:

1. 'Río Negro' is a River.
2. 'Atlántico' is an Ocean.
3. 'Viedma' is a Town.
4. 'Río Negro' *flows into* 'Atlántico'.
5. 'Río Negro' *ends* in 'Viedma'.

And these more elaborated implicit conclusions:

1. 'Río Negro' *is a* Source of Fresh Water.
2. 'Viedma' *is near* the 'Atlántico' ocean.
3. 'Viedma' *has a* Source Of Fresh Water.

A benefit of having this explicit representation for instances and their binding model is that an automated agent could reach to these same conclusions, as if it could understand or reason.

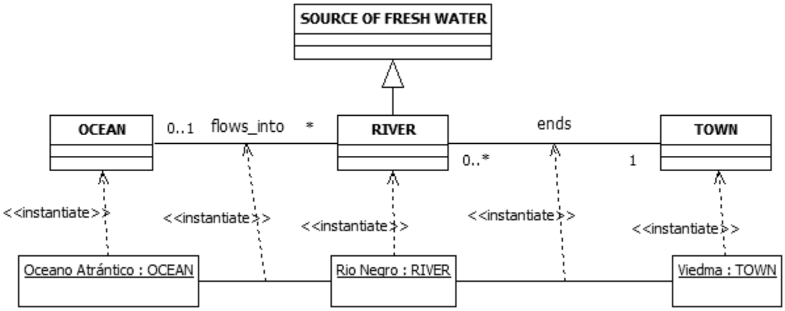


Fig. 1. Instantiation Feature-Ontology

4. Ontology-based Architecture Integration Model

As stated, to find and recover efficiently distributed heterogeneous GI is a key factor. Standards promote interoperability and classification for GI by catalogs. However, difficulties caused by semantic heterogeneities are still a challenge in integrating distributed open environment GI.

In the field of ontology, there are different ontologies built for various application domains. They vary in the level of detail they express. Ontologies can be organized according to their degree of generality as follows [1.13]:

- *Generic Ontology (Top-Level):* captures the general purpose knowledge, regardless of any particular domain, such as space, time, event, action, etc. It is expected that these ontologies will be adopted by a large community of users.
- *Domain Ontology and Task Ontology:* define the particular knowledge of a domain (for example, medicine, geography, etc.) or a specific activity (for example, trade), describing their vocabulary through the specialization of the terms introduced in the high-level ontology.
- *Application Ontology (Low-Level):* captures the knowledge needed from an individual system or application. It describes concepts that depend on both the domain and activity ontology, that are often specializations of the two previous kinds of ontology.

The options for ontology-based semantic integration systems are arranged into different styles [2, 13]. One style considers a single ontology shared by all applications. Another defines multiple ontologies along with integration functions between pairs. A more flexible option is to combine the two previous styles. The latter proposal of integration, based on *hybrid ontology,* establishes a *domain ontology* (*DO*) shared by a community of use that provides the definition of its basic terms (primitive). Hybrid ontology assumes that the common semantics of its primitives is known and understood by the community. Independently, each supplier is free to define his or her own *OGIS*$_i$ application ontology. Furthermore, the data model of each GIS solution plays the role of application ontology. Besides, the communication interface or "mapping" between *DO* ontology and *OGIS*$_i$ ontology should be established. This kind of semantic integration provides a flexible framework that respects application ontology and complies with every need, keeping the various *OGIS*$_i$ ontologies comparable [1], something crucial when making semantic searches or requiring information integration services .

5. Cohesion between Ontological Models

In compliance with the integration style based on hybrid ontology semantics, each application is free to use its own application ontology or *OGIS*$_i$ data model. An *OGIS*$_i$ could eventually be shared by more than one application, as it is the case of distributed GIS that share the data model. In the following generality level, *DO* ontology is defined. In the particular case of geographic applications, a *DO* ontology corresponds to a *theme* such as "land use". This proposal for integration assumes the existence of consensus *ODs*. Thus, each GIS is responsible for formalizing its *OGIS*$_i$ application scheme aligned to the GFM standard [9] and to define the $m(OGIS_i) \rightarrow DO$ mapping function.

Thus, the problem of solving semantic heterogeneity among different GIS solutions turns into defining the correct $m(OGIS_i) \rightarrow DO$ mapping function, with the added advantage of having concepts formalized by an ontology. However, the effectiveness of the mapping, and thus the result of integration, depends on the degree of cohesion between the world shaped by *DO* and the world shaped by *OGIS*$_i$.

This work presents a technique used to measure the degree of interrelation or cohesiveness between *DO* ontology and *OGIS*$_i$ application ontology. In particular, we want to measure the level of cohesiveness between concepts defined in *OGIS*$_i$ and concepts defined by *DO*. In order to progress with rigor, we present the following definition [12]: let C_{GIS} be a set of concepts defined by *OGIS*$_i$ ontology and C_{DO} the ontology concepts defined by *DO* ontology. It is possible to approximate the cohesiveness between the application model and the domain ontology by formalizing the following membership relations between sets:

1. $C_{GIS} \subset C_{DO} \wedge |C_{GIS}| \approx |C_{DO}|$, presents the situation of domain ontology with maximum coverage and high accuracy. This is the optimal relation between *DO* and *OGIS*$_i$ concepts. Domain ontology concepts cover the concepts required by the application. We can say that *DO* contains definitions and semantics close to the application problem. For example, suppose that C_{DO} defines the concept LOCATION while C_{GIS} provides the definition for a concept named CITY. In all instances, *city* in C_{GIS} is covered by the concepts LOCATION in C_{DO} and is hoped that its semantic definition is close.

2. $C_{GIS} \subset C_{DO} \wedge \neg(| C_{GIS} | \approx | C_{DO} |)$, represents a situation with high coverage but low precision. The relationship between concepts in domain ontology and the concepts in the GIS application can be defined as "good". In this case, *DO* also covers the concepts required by the application. However, the semantic content in *DO* is not as close to the semantic content required by the GIS, and the mapping shall be potentially less accurate. For example, C_{DO} has a definition for a concept Species while C_{GIS} considers a definition for a concept Native Species. It is expected that the description for Native Species in $OGIS_i$ will be more refined than the characterization of the concept Species provided by *DO* ontology.
3. $C_{DO} \subset C_{GIS}$. Concepts in *DO* do not cover the universe of concepts required by the $OGIS_i$ application. There are concepts defined by the GIS application for which there are no concepts to map in *DO*. The semantic relationship between $OGIS_i$ and *DO* is "not good". The greater the number of concepts excluded, the worse the cohesion ratio relation is. For example C_{GIS} contains a definition for Site whereas C_{DO} defines Location. There are *site* elements in C_{GIS} which do not map any *location* of C_{DO} and, thus, will be lost in the mapping process.
4. $\neg(C_{GIS} \subset C_{DO} \wedge C_{DO} \subset C_{GIS}) \wedge (C_{GIS} \cap C_{DO} \neq \emptyset)$. Both ontologies contain concepts that do not have a correspondence in the other universe, although they share a subset of concepts. This is the worst relationship scenario among ontologies. For example, consider a C_{GIS} set which defines concepts such as Source Of Fresh Water, while C_{DO} defines concepts such as Stream of Water. There are elements in C_{GIS} such as "*reservoir*" which do not map any concept in Stream Of Water in C_{DO}. Conversely, the semantic definition for the elements in *DO,* such as "*seas*" are not represented in the C_{GIS} of the GIS application.

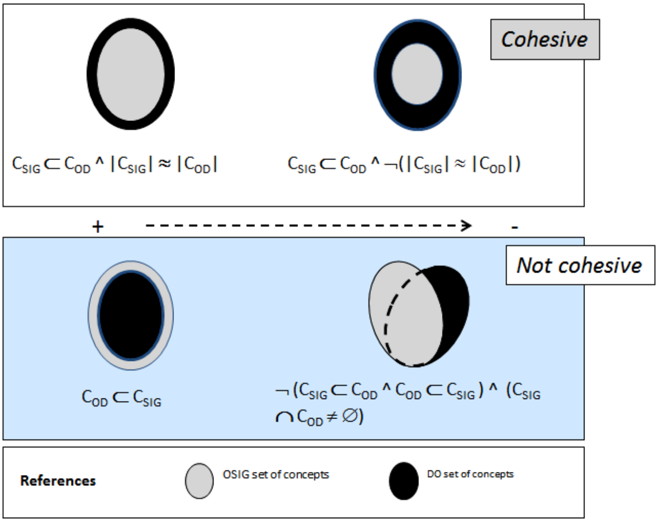

Cohesive
$C_{SIG} \subset C_{OD}$ ^ $|C_{SIG}| \approx |C_{OD}|$
$C_{SIG} \subset C_{OD}$ ^ $\neg(|C_{SIG}| \approx |C_{OD}|)$
+
-
Not cohesive
$C_{OD} \subset C_{SIG}$
$\neg(C_{SIG} \subset C_{OD}$ ^ $C_{OD} \subset C_{SIG})$ ^ $(C_{SIG} \cap C_{OD} \neq \emptyset)$
References
OSIG set of concepts
DO set of concepts

Fig. 2. Relationship between domains and GIS concepts

Figure 2 shows a graphic for the above items using traditional set representations. The kind of relationship between sets of concepts has an impact on the effectiveness of the mapping process from *OGIS*$_i$ to *OD*. We can state that the mapping function looses accuracy as we move away from the situation in the states in 1, being 4 the least desirable alternative.

As an example of the previous items, Figure 3 shows two partial solutions for a problem on "Economic Geography” with their ontological conceptual models. The analysis of the relationship between concepts present in both ontologies is shown in Table 2. As shown in this example, it is difficult to ensure an optimal relationship. One possible solution to the problem of low accuracy might be to combine several *DO*$_j$ ontology definitions. Surely this option will require further detailed analysis to resolve conflict situations between the various *DO*$_j$ ontologies concepts definition. It is expected that the definition of concepts shall raise situations of synonymy, redundancy, and/or contradictions between concepts.

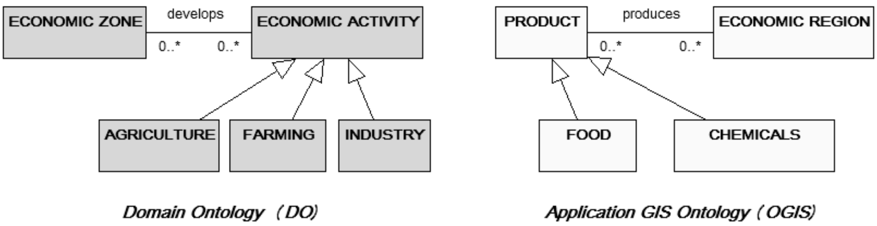


Fig. 3. DO ontology and OGIS application ontology relationship

6. Analysis of the Proposal

The distributed nature of GIS applications led to the existence of multiple overlapping domain conceptualizations. Meanwhile, continuous progress in Information and Communications Technologies (ICTs) offers the possibility of having a great amount of heterogeneous GI available. In current research in the field of Computer Sciences on the topic of ontology, researchers are looking for ways of representing and accessing knowledge in digital GI towards promoting interoperable systems.

Table 2. Analyzing the relationship between concepts

OGIS Concept	DO Concept	Map Relation	
Economic Region	Economic Zone	(.1.) Very good	
produces	*develops*	(.2.) Good	
Product	Economic Activity	(.2.) Good	
Food	Farming	(.3.) Fair	*Mapping depends upon the existence of food discriminator*
	Agriculture		
	Industry	(.4.) Worst	
Chemicals	Industry	(.2.) Good	

Research on GI semantic integration using ontologies projects in two ways. Some researchers rely on studies and suggest styles for organizing and distributing concept definition: using a common ontology and mapping to it, define intermediate ontologies and point to point mappings, among other alternatives [2, 13]. In this way, alternative activities towards integration are investigated and proposed: ontological map, ontological integration, fusion (merging), and ontological alignment. Some solutions are focused on integrating schemas or structural models, while others do the same from the integration of instances or data [14].

Other research works focus on designing and implementing tools whose function is to assist in the integration process [14, 15, 16], with a greater or lesser degree of automation, some focus on data analysis, while others compare data dictionaries. Works with comparative analysis of integration tools can be found in [3, 5]. In general, all authors agree that it is not possible to fully automate the integration process and, at least in the phase of mapping definitions, the participation of domain experts is required.

This paper proposes a novel and simple technique based on the relations among sets of concepts to conceptually anticipate the result of the integration of two distinct solution models. As described in the example developed in Figure 3 and Table 2, the method includes identifying each conceptual element in the ontology source -concepts and relationships- and classifying the mapping relationship (1 to 4) with regard to the concepts of the target ontology. As far as the target ontology covers the concepts of source ontology, it is expected that mapping to shared ontology shall be possible without loosing information. In our example, we see that between the two models there is a "good" relationship with the *DO* ontology for the concepts "Economic Region", "produces" and "Product", with a different and even impossible way of mapping the system used to classify them.

7. Conclusions

Ontology is a bridge that gains importance when looking for interoperability among heterogeneous GIS and web applications. The style of hybrid ontology-based integration provides customers with a unified abstraction layer that allows for independence from the conceptual models of each service provider. In order to make this possible, we need to define the mappings between the various implementation models and shared ontology.

Much of the effort of the research community in Semantic Integration aims at developing techniques and automated tools to ensure successful results. In this paper we propose a novel technique based on set relations in order to measure the interrelationship between different data models and to foresee the result of the integration process. As long as the relevant concepts of the application conceptual model maintain a good cohesive relationship with domain ontology concepts, it is expected that the result of integration shall be acceptable and that no information shall be lost. This proposal is under an initial research phase. In a future work, we intend to break down this first measure into specific and measurable sub-measures that shall give a result that will allow us to measure the applicability of the proposal.

References

1. A. Buccella: Integración de Sistemas de Información Geográfica. Ph Tesis. Departamento de Ciencias e Ingeniería de la Computación. UNS. Argentina (2009).
2. A Namyoun Choi, Il-Yeol Song, and Hyoil Han: A survey on ontology mapping. *SIGMOD Rec.* 35, 3, pp, 34-41 (September 2006).
3. Y. Kalfoglou and M. Schorlemmer. Ontology mapping: the state of the art. The Knowledge Engineering Review, 18, pp 1-31 (2003).
4. M. Egenhofer: Toward the semantic geospatial web. In: Proceedings of the 10th ACM international symposium on Advances in geographic information systems, pp. 1-4. ACM GIS, New York (2002).
5. B.Sean M. Falconer, Natalya Fridman Noy, Margaret-Anne D. Storey: Ontology Mapping - a User Survey. In Proceedings of OM'2007, pp 113-125 (2007).
6. F. Fonseca, J. Egenhofer et all: Using Ontologies for Integrated Geographic Information Systems. Transactions in GIS 6 (3), pp 231-257 (2002).
7. Nicola Guarino: Formal Ontology and Information Systems. In: IOS Press, pp 3-15 (1998).
8. ISO/TC 211. 1994. ISO/TC 211 Geographic Information / Geomatics. url: http://www.isotc211.org/.
9. ISO/TC211. Geographic Information - Rules for application schemas. ISO International Standard. ISO/TC211 19109. International Organization for Standardization, TC 211 (2005).
10. Longley, P. A., Goodchild, M. F., Maguire, D. J.: Geographic information systems and science. John Wily and Son (2005).
11. OGC. Open Geospatial Consortium, Inc. (OGC). 1994. url: http://www.opengeospatial.org/
12. M. Vitturini: Modelos de Representación para Información Geográfica. Mg. Tesis. Dpto. de Ciencias e Ingeniería de la Computación. UNS. Argentina (2011).
13. H. Wache, T. Vögele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Höbner: Ontology-Based Integration of Information - A Survey of Existing Approaches. In: IJCAI Workshop: Ontologies and Information Sharing, pp. 108-117 (2001).
14. Do, Hong-Hai Schema Matching and Mapping-based Data Integration. PhD thesis, Department of Computer Science, Universität Leipzig, (2006).
15. N. F. Noy, Mark A. Musen, The PROMPT suite: interactive tools for ontology merging and mapping, International Journal of Human-Computer Studies, Volume 59, Issue 6, December 2003, Pages 983-1024 (2003).
16. Rahm, E., Bernstein, Philip: A survey of approaches to automatic schema matching. In: The VLDB Journal, pp. 12-25 (2001).

An Audio File Tagging Mobile Game, mTagATune

FRANCISCO JAVIER DÍAZ, CLAUDIA ALEJANDRA QUEIRUGA, ALEJANDRO FERRARESSO, JOSÉ IGNACIO LARGHI

LINTI - Facultad de Informática - Universidad Nacional de La Plata
50 y 120 La Plata, Buenos Aires, Argentina
jdiaz@unlp.edu.ar,claudiaq@info.unlp.edu.ar,
aferraresso@cespi.unlp.edu.ar,jlarghi@cespi.unlp.edu.ar

Abstract*. mTagATune[1] is a mobile game based on TagATune[2]. mTagATune implements the concept of GWAP[3] and seizes the capabilities and wide acceptance of current smartphones[4]. GWAP promotes the creation of computer games that encourage people to do voluntary work. mTagATune implements a game that collects information on audio files to facilitate future searches on them. By means of a collaborative game, mTagATune enables an ubiquitous collection of information on audio files that can later be used in search results.*

Keywords: *TagATune, GWAP, Ericsson, Mobile Applications, JAVA.*

1. Introduction

Despite technological advances, computers still do not have the creativity or perception human beings have by nature. Due to this fact, computers today cannot subjectively classify certain sets of elements such as audio files, image files, etc.

Currently, data bases exist that contain thousands of audio files, although searching these repositories with subjective criteria is not possible due to the fact that each audio file would have to be tagged first with words that necessarily convey subjective meanings.

One solution to this problem is using the technique known as Human Computation [5]. This technique views the human brain as a processor inside a distributed system, where each can process a small part of a much larger computation.

Currently, there are millions of people around the world who use digital games as a form of entertainment. Many of these games can be accessed through the Internet. The massive expansion of mobile telephones allows users to play games where they could not play before: during trips or while queuing at the bank. The first games launched for mobile telephones were very simple due to physical limitations, but with the new generation of cellphones, the so-called smartphones, games are becoming more and more complex, with better performance, even using resources such as global positioning systems and the Internet.

One branch of Human Computation, called Games with a Purpose (GWAP) promotes the idea of creating games in which the activity people engage in forms part of a processing that produces information, which can later be used in other successive processings.

GWAP encourages people to do voluntary work, but not with the intention of obtaining income, as is the case of employment. If we think about the task of tagging music fragments, the amount of elements requiring processing is enormous which would require a tremendous workforce to complete it, yielding the task impractical due to cost.

Smartphones, with their many advantages, allow for the implementation of GWAP on mobile telephones, providing permanent access to the games and increasing the amount of players (and, as a result, the amount of hours dedicated to each game as well). This increases the data processed, which gives better use to the time players spend on each game.

2. Bases

As we have mentioned before, Human Computation posits the theory that the brain can be seen as a small processor inside a distributed system, where each brain can process a small part of a much larger computation. Human Computation is a technique in which a computation executes its function by delegating certain steps to humans. In traditional computation, humans use a computer to solve a problem: the human provides the computer with a formalized description of the problem and receives a solution they must interpret. Human Computation reverts the roles; the computer asks a human or group of humans to solve a problem and collects, interprets and integrates their solutions.

GWAP is a combination of the Human Computation technique and the billions of people around the world willing to invest time playing online. The concept of GWAP could be defined as games in which each participant processes a part of a larger computation, which is solved by combining the processings contributed by each player. In this context, "processing" makes reference to the mental exercise the player engages in to solve the part of the computation they are assigned. Some examples of GWAP can be found in work by Luis von Ahn[6]; the Google engine has an experimental version of a GWAP for the classification of images, called Google Image Labeler[7]. The goal of the Google Image Labeler game is to generate tags associated to images that can be used to improve image search results. The mechanism of the game consists of showing two users an image, and for every match, both users get points. These points motivate users to input a large amount of tags. Afterwards, when a word has been entered many times for the same image, it is assumed that it describes the image correctly and fit to be used in the search engine. Note that in this mechanism, players have no knowledge of who is their team mate during the game and have no way to communicate, thus it is impossible for users to cheat by agreeing on the words they will use.

3. Description of mTagATune

mTagATune[1], mobile TagATune[2], is a GWAP application for mobile devices, smartphones specifically, based on TagATune.

The growing trend in the use of mobile devices and the advantages they offer encourages an environment adequate for the application of GWAP in mobile phones, making it possible for people to play in more places, thus increasing the amount of data published.

3.1 How the Game Works

mTagATune is a mobile application that implements the concept of GWAP and allows for audio file tagging. The goal of mTagATune is to collect semantic information to be used in search results and further indexation.

When a user enters the game, after registering, they are assigned a partner. Because this type of application cannot ensure that the user is paying attention, once the partner is selected, both are asked to confirm that they are ready to begin the game. In case one of the users takes longer than stipulated to confirm presence, both users will be informed that the game has been canceled.

When the game begins, each participant is given an audio entry and both have to contribute words that describe it. Based on the descriptions entered by both participants, each must determine individually whether they are both listening to the same piece or not. If both participants choose the right answer, they obtain points. The goal of the game is to obtain as many points as possible.

3.2 Pair Selection

Participants pairs are selected at the beginning of each game without the users knowing who is paired with them. Following is an explanation of the mechanism implemented for this purpose.

Players are picked together on the basis of similar amounts of points. For this purpose, it is necessary to limit scoring differences, for example, if a user obtains 500 points and the limit for a game is 100 points, this user will be able to play with others that have between 400 and 600 points. The main disadvantage of this system is the delay in finding a suitable match for a certain player, which is greater the more reduced the amount of users. To solve this problem, it was decided that the limit increased with time, which allows for a greater range and increases the possibility of forming pairs. A waiting time limit was also introduced – if the limit is reached and there are no matches, the player is assigned a partner no matter their score. If there are no other players waiting, a game is created exclusively for the user. To avoid pairing new users, which might discourage them because both might enter a small amount of tags, it was determined that inexperienced users should be paired with players that have a long history of games. This way, the chances of winning games, and therefore gaining interest in the game, are greatly increased.

3.3 Description of a Game

First, the game waits for both users to download the audio file to their devices for two reasons: because both must discover whether they are listening to the same file at the same time, and because the length of the track determines the time the user is given to enter tags and decide whether it is the same fragment as their partner's. Once each user has their file, the round begins. During the round, the player must enter words that represent what they hear. As they do, the words are sent to their partner and shown in a fraction of the screen, so both players have real-time access to the words entered by each. When the track ends, no more words can be added.

Once the file has ended, players are given a few seconds to determine whether their fragment was the same as their partner's. Players will receive points each time both pick the right option.

Thus, if one or both get the wrong answer, non of them will receive points during that round. Figure 1 shows the screen players see during each round, which shows the tags both entered together with the file data and options.

It was decided that the player who got the right answer even though their partner did not would not receive points either because the goal of the game is to achieve cooperation and not competitiveness. This way, each player would have to concentrate on getting the right answer and in describing their file as well as possible to increase their chances of the other player getting the right answer as well.

A player might decide whether they are listening to the same file before the file ends, which may block out data entry and reduce the possibility of getting the same answer for both players. Although this option may make the player enter less words, notice that these cases add a new piece of information: the instant in which the choice was made.

In some cases, one player will enter tags that are completely opposite to the tags the other generates, in which case they will need no further proof to decide. Therefore, a later processing of the information could determine which words potentially express the opposite to the way in which the file is classified. At all times during the round, both players know whether their partner has decided.

Once both players have made their choice, the score they get this round is show on screen, as well as whether the answer each player gave was correct or incorrect. If one of the players chooses not provide an answer, the system will assume that they gave an incorrect answer and none of them will obtain points.

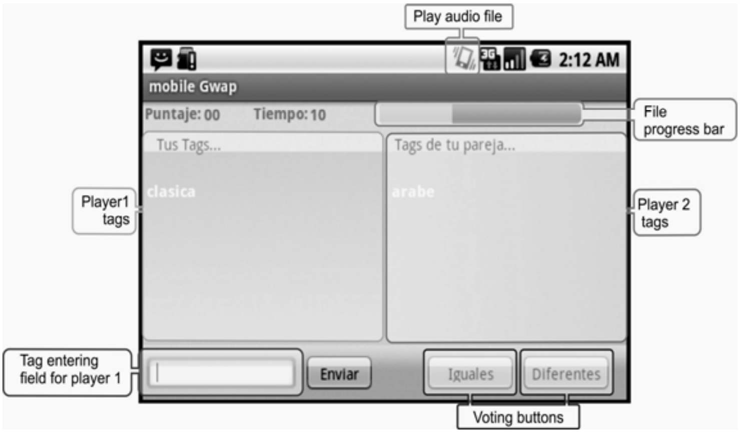


Fig. 1. Game in progress

mTagATune is a collaborative, non-competitive game in which players only receive points if both get the right answer. mTagATune gives a natural incentive for players to enter data that correctly describes the audio file. If it were a competitive game by, for example, granting points to the player who gets the right answer even if their partner does not, players would be motivated to win by harming their partner. This would cause them to enter wrong and malicious data to confuse their partner and make them pick the wrong answer, which would result in wrong data due to the implicit competitive nature of the game.

3.4 Scoring System

Players are given points in the following manner:

When both players answer correctly for the first time, each player gets 60 points. The second time, they get 70 points each and 80 if they answer correctly a third time. This way, two players that answer correctly three times during a game will obtain 210 points each. These correct answers do not necessarily have to be consecutive, that is, if a pair gets the first fragment right (and receives 60 points), fails to provide a correct answer for the second fragment and does so for the third one, both players will receive 70 points for the second correct answer in the game, earning a total of 130 points each.

The aim of this scoring system is to stimulate the attention of the user throughout the game, as a user that answers correctly in the three rounds of a game will receive more points than a user who answers correctly in three rounds from different games. In the first case, they will receive 210 points, while in the second, they will obtain 180 points. In this way, users to maintain a good performance throughout the game obtain more benefits.

3.5 Bonus Round

Another way of playing is what is known as bonus round. The bonus round is activated during a common game, when both players get the three rounds right, thus obtaining the maximum score for a game.

This round does not generate tags on the files given, but serves to create a relationship between them. When both players get the three rounds of a common game rights, they are automatically notified that they can take part in a bonus round (they can turn down the offer).

If both players agree to participate in the bonus round, the system will select three audio files that will be played for the users. When the files end, the users have 10 seconds to decide which of the three fragments is the most dissimilar, if both coincide in their choice, they get 50 points. Figure 2 shows a screen in the bonus round, which shows the controls players can use to select answer.

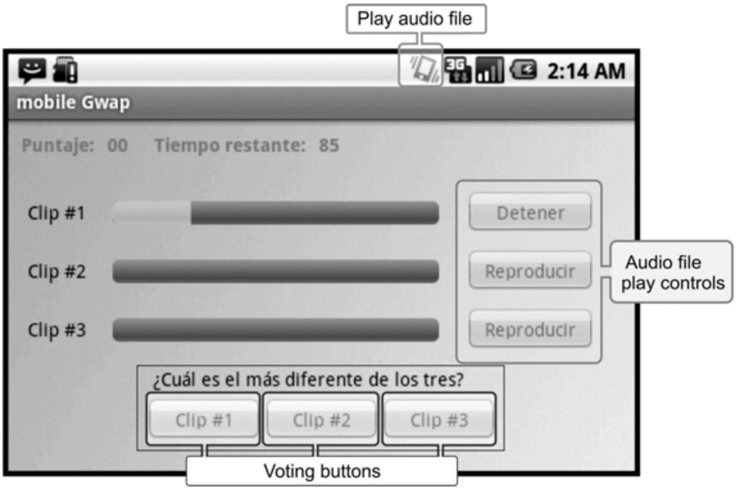


Fig. 2. Screen of a bonus round

3.6 Single-Player Mode

mTagATune has a single-player mode for when a player enters the game when there are no other players to pair them with, or the total number of players is even, making it impossible for the system to assign a partner for the player.

The single-player mode allows for a single user to start a game at any time, independently from the amount of users connected at the time. This mode is transparent to the user, as the place of the other player is occupied by a bot that reproduces a series of rounds that have already been played by real users. The bot is an algorithm that reproduces a player's behavior in a previous round.

If there is no available partner for the player upon entering the game, the player selects a saved game depending on the level of experience of the user. For rounds to be reproduced, it is a fundamental prerequisite that they resulted in a positive outcome, i.e., both players coincided in their choices for each round and that their choices were right. There has to have been a bonus round in the game as well. This is necessary for the system to be able to assume that the entered tags can be considered valid.

Once the the rounds are assigned, the user is notified that they have a partner, a bot (the user will never know that they are actually playing with a bot). During the course of each round, the bot enters the tags in the same sequence in which they were entered by the emulated user. Once the audio fragment has been reproduced in its entirety, the game evaluates the tags entered by the real user and determines whether it is the same track in both. To do this, the bot analyzes the percentage of matches between the set of tags entered by their partner and the set of tags for the audio fragment they have.

The bonus round in a single-player game is also based on a game that was stored beforehand. The game will select a saved bonus round and the bot will choose the same options the original player chose.

An important advantage is that the result of a game of this modality is just as productive as the results of ordinary games. The player will choose whether they are listening to the same fragment as their partner based on the tags entered by the bot, which is in turn based on the actions of a real user. On the other hand, the bot will take their decision based on a comparison of the tags entered by a real user and those provided earlier by another real user.

3.7 Technologies

mTagATune is a mobile application written in Java for Android phones that consists of a client and a server. The full development is based on open, free use technologies. mTagATune uses Tomcat to contain the Java servlets in charge of handling the logic and data storage. PostgreSQL was used to store data and Hibernate was used to map objects to the relational database.

For the purpose of communication between the server and its clients, mTagATune uses a Server Push mechanism called Comet [10] that allows for information to be sent asynchronously from the server to the clients. The Comet implementation was CometD, developed by the Dojo Foundation, which implements the Comet mechanism with Jetty Continuations. To handle the messages, CometD uses the Bayeux protocol, which sends the messages through named channels. These messages can be sent from the server to the client, from the client to the server or among servers, using these channels.

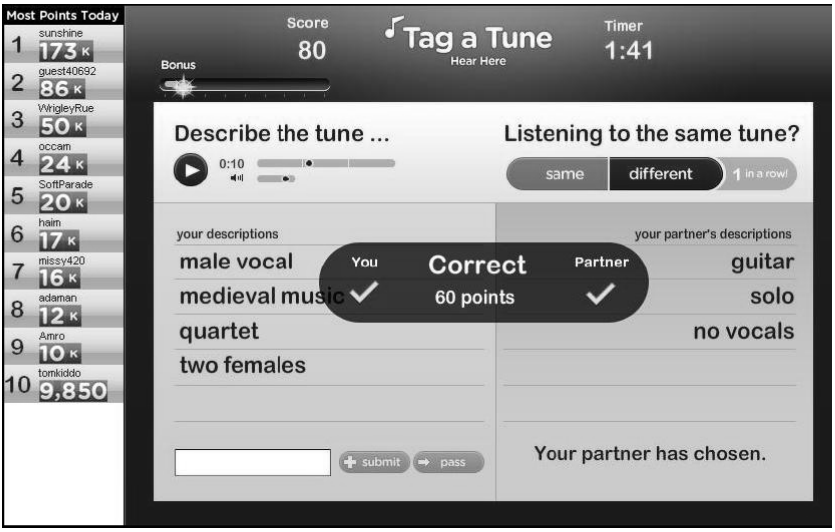
For the serialization of the Bayeux messages and the domain objects, mTagATune uses JSON. This is a light data exchange format, easy to read and write for humans, and easy to interpret and generate for machines.

The client was developed using Java for the Android operating system for many reasons, the first being its free license as well as that of the tools used for the development, and also because of the wide acceptance it has gained during the past year. The architecture used allows for the development of clients for other operating systems, such as iOS, and for users of different devices and operating systems to interact in a game.

4. Adapting TagATune for Mobile Devices

mTagATune is an adaptation of TagATune for mobile devices. mTagATune introduces some modifications to TagATune that allow it to improve the gaming experience and its performance in mobile devices. Following, we will describe the aspects of TagATune that were modified for mTagATune.

Given that the quality of Internet access in mobile devices can vary greatly, we decided that the audio files should be downloaded separately before each round starts, and each round should begin only after each file has been successfully downloaded. Although this characteristic introduces waiting time at the beginning of the rounds, it allows us to be sure that the file has been successfully downloaded and is available for full reproduction throughout the round. This also releases the connection for sending information about the round itself, so that, if the quality of the connection is low, it can be played correctly. Because mobile device screens are smaller and of lower resolution as compared to those of desktop computers, it was necessary to redesing the screens to accommodate all the elements of the user interface, eliminating unnecesary ones. Figure 3 shows a screen of a TagATune game.



Most Points Today		
1	sunshine	173 K
2	guest40692	86 K
3	WrigleyRue	50 K
4	occam	24 K
5	SoftParade	20 K
6	haim	17 K
7	missy420	16 K
8	adaman	12 K
9	Amro	10 K
10	tomkiddo	9,850

Fig. 3**.** Pantalla de una partida en TagATune

If we compare the TagATune screen to that of mTagATune, we can clearly see many changes that go beyond aesthetics. The daily high scores list was deleted together with the play and volume controls. This simplification was aimed at focusing the attention of the player on producing tags only. The warning saying that the partner has already chosen was replaced with a pop-up, and in its place we put the voting buttons. Lastly, the remaining time, score and progress bar were located in a single line to reduce the space they took up. Similar changes were introduced in the remaining screens, such as the bonus screen and the previous games screen. Another problem with mobile devices are keyboards. Firstly, the average typing speed is significantly lower than that attained with desktop computers. To mitigate this issue, we modified a feature in the original game–in TagATune, players were supposed to make up their minds during the audio reproduction, while in mTagATune, we decided to add 5 seconds of extra time before this action is allowed, so that during the game the players can focus on writing, using the extra time to analyze their partner's input and make the right choice. Secondly, many devices do not have physical keyboards, but virtual ones inside the screen itself. This is a setback for the player, which is why we have considered a redesign of the interface to integrate this kind of devices.

Because it is to be expected that players will receive incoming calls or text messages during a game, we decided to add a confirmation related to the user paying attention to the game right before each round starts. This confirmation has a timeout and, if it's reached without confirmation from the player, the game is cancelled and the partner is notified, so that they are not stalled waiting for the game to resume and can begin a new game.

The great popularity of the Android mobile operating system, its user community the proposed open development model and the agile application distribution mechanism through its online store, “Android Market”, provides an optimal media for the development and validation of the results. This was the hypothesis upon which we based our decision to choose Android [8] over iOS[9], another leader in the mobile device market.

5. Conclusions

This work shows a way to obtain information about audio files that is automatically validated by the very agreement of the players on the subject, a concept that constitutes a step towards simplicity and improvement over the manual mechanisms that are currently used for tasks such as labelling music.

With mTagATune, we demonstrate that the concept of GWAP, particularly that of TagATune, is applicable to mobile devices, so long as they have the characteristics of a smartphone, although in this particular case we have only used those with Android as their operating system.

It was also shown that this adaptation has no negative impact in quality or playability. Although changes were made in the application architecture, in some playability aspects and in the user interface, none of these changes has a negative impact on the player or represents a problem for the normal development of a game.

This adaptation for mobile devices takes the concept of GWAP to a completely accepted environment in countries with wide access to mobile technology, something that will soon come to developing countries such as our own. This will increase the amount of time users can spend on the game. The fact that these games are available for mobile devices makes it possible to play them during free time, something that was unthinkable a few years ago because they were only available for conventional computers.

We are currently planning to test the usability of mTagATune to test its performance in other mobile devices, together with an evaluation of other mobile applications such as AvatarFacedget, a widget that allows users to publish single avatars or avatar galleries in social networking sites.

References

1. Díaz J., Queiruga C., Ferraresso A., Larghi J.: “mTagATune: mobile TagATune”. In Proceedings of ICMB 2011, 10th International Conference on Mobile Business, Como, Italia, June 2011. http://www.mbusiness2011.org/
2. Law E., Von Ahn L, Dannenberg R. and Crawford M: "Tagatune: a game for music and sound annotation". In Proceedings of the 8th International Conference on Music Information Retreival, Vienna, Austria, 2007.
3. Von Ahn L.: "Games with a purpose". In IEEE Computer Magazine, June 2006. Pages 96-98.
4. Smartphones: http://es.wikipedia.org/wiki/Smartphone
5. Von Ahn L.: “Human computation”. In K-CAP '07, Proceedings of the 4th international conference on Knowledge capture, 2007.
6. Von Ahn L.:. http://www.cs.cmu.edu/~biglou/
7. Google Image Labeler, http://images.google.com/imagelabeler/
8. Android. http://developer.android.com
9. iPhone OS. http://es.wikipedia.org/wiki/IPhone_OS
10. Gravelle, Rob. "Comet Programming: Using Ajax to Simulate Server Push", http://www.webreference.com/programming/javascript/rg28/.

II

Computer Science Theoretical Aspects Workshop

DeLP marking procedure for dialectical trees is PSPACE-complete

LAURA A. CECCHI[1], GUILLERMO R. SIMARI[2]

[1]Grupo de Investigación en Lenguajes e Inteligencia Artificial
Depto. de Teoría de la Computación - Facultad de Informática
Universidad Nacional del Comahue

[2]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
{lcecchi, grsimari}@gmail.com

Abstract. *The Defeasible Logic Programming (DeLP) is a formalism that combines Logic Programming and argument-based reasoning. Its proof theory is based on a dialectical analysis where arguments for and against a literal interact. A dialectical tree is built as the result of this interaction. In this work we address the problem of studying the complexity of the marking procedure of the dialectic tree, which determines whether its root has been defeated. This point is of central importance in DeLP, in order to determine whether an argument that supports a literal L is found undefeated and therefore, L is believed by a reasoning agent. We prove that the marking procedure of a dialectical tree is* PSPACE-*complete.*

Keywords: *Argumentative Systems, Defeasible Reasoning, Logic Programming, Computational Complexity.*

1. Introduction

Defeasible Logic Programming (DeLP) is a general argumentation based tool for knowledge representation and reasoning[1][15]. Its proof theory is based on a dialectical analysis where arguments for and against a literal interact in order to determine whether this literal is believed by a reasoning agent.

Complexity theory is an important tool for comparing different formalism, and for helping to improve implementations whenever it is possible. For this reason, it is important to analyze the computational complexity and the expressive power of DeLP. The former tells us how difficult it is to answer a query, while the latter gives a precise characterization of the concepts that are definable as queries.

Historically, implementations of argumentation systems have been limited to areas with no real time response restriction (see [21, 18]).

[1] The interested reader can find an on-line interpreter for DeLP in http://lidia.cs.uns.edu.ar/DeLP.

Recently, however, several applications have been developed, and implemented using argumentation systems related, for instance, with multiagent systems and web search [2, 3, 10, 11].

Scalability and robustness of such approaches heavily depend on the computational properties of the underlying algorithms. It is hence crucial to study these properties in order to expand the application fields of argumentation systems.

Different computational complexity results [1, 4, 12, 14] have been presented on argumentation abstract framework [5,13], based on admissibility and preferability semantics. However, those results do not apply directly to DeLP, because its semantics are quite different.

Another notable study of the computational complexity of defeasible systems has been done in [16]. But, defeasible theory analyzed in this work greatly differs from DeLP in several points, such as knowledge representation (facts and strict rule, defeasible and defeaters rules) and their proof theories.

In this work we are concerned with the study of the marking procedure of the decision tree in DeLP. The marking process is of central importance in DeLP, in order to determine whether a query L will succeed, i.e., a supporting argument for L is not defeated. Therefore, the literal L is believed by a reasoning agent.

We prove that the marking procedure for a dialectical tree is PSPACE-complete considering the analogy between a dialectical tree, games and the decision problem QSAT.

The paper is structured as follows. In the following section we briefly outline the fundamentals of DeLP, and describe its theory proof. Afterwards, we analyze the computational complexity of the marking procedure of the dialectical tree. We explain the analogy between a dialectical tree and the decision problem QSAT. We introduce a transformation from QSAT to the marking process, and finally, we present the complexity results showing that the marking procedure is PSPACE-complete. In the last section, we summarize the main contributions of this work, and we present our conclusions and future research lines.

2. Defeasible Logic Programming

We will start by introducing some of the basic concepts in DeLP (see [15]). In the language of DeLP a literal L is an atom A or a negated atom ~A, where ~ represents the strong negation in the logic programming sense. The complement of a literal L, denoted as L, is defined as follows: L=~A, if L is an atom, otherwise if L is a negated atom, L=A.

Definition 1. A *strict rule* is an ordered pair, denoted “Head ← Body”, where “Head” is a ground literal, and “Body” is a finite set of ground literals. A strict rule with head L_0 and body $\{L_1,..., L_n\}$, n>0, is written as $L_0 \leftarrow L_1,..., L_n$. If body is the empty set, then we write L_0, and the rule is called a *Fact*. A *defeasible rule* is an ordered pair, denoted “Head --< Body”, where Head is a ground literal, and Body is a finite, non-empty set of ground literals. A defeasible rule with head L_0 and body $\{L_1,..., L_n\}$ n>0 is written as L_0 --< $L_1,..., L_n$. A *defeasible logic program P*, abbreviated d.l.p., is a finite set of strict rules and defeasible rules.

Intuitively, whereas Π is a set of certain and exception-free knowledge, Δ is a set of defeasible knowledge, i.e.,tentative information that could be used, whenever nothing is posed against it.

DeLP proof theory is based on developments in non monotonic argumentation systems [18, 20]. An *argument for a literal* L is a minimal subset of Δ that together with Π consistently entails L. The notion of entailment corresponds to the usual SLD derivation used in logic programming, performed by backward chaining on both strict and defeasible rules, where negated atoms are treated as a new atom in the underlying signature. Thus, an agent can explain a literal L, throughout this argument.

In order to determine whether a literal L is supported from a d.l.p. a dialectical tree for L is built. An argument for L represents the root of the dialectical tree, and every other node in the tree is a defeater argument against its parent. At each level, for a given a node we must consider all the arguments against that node. Thus every node has a descendant for every defeater.

There are certain constraints we will impose when building the dialectical tree for avoiding undesirable situations. For instance, no subargument structure can be introduced again in a dialectical tree branch, in order to avoid a circular argumentation. Furthermore, the set of arguments that support a literal L (arguments in an odd position in a dialectical tree branch) should be consistent. Analogously, the set of arguments against the supporting arguments would be consistent. Finally, if two arguments defeat each other, they will be considered blocking defeaters. If an argument A is a blocking defeater of other argument B, then A has as a child B in the dialectical tree. Whenever we have this situation, we impose that no defeater of B in the dialectical tree would be a blocking defeater.

A comparison criteria is needed for determining whether an argument defeats another. Even though there exist several preference relations considered in the literature, in this first approach we will abstract away from that issue.

We will say that a literal L is *warranted* if there is an argument for L, and in the dialectical tree each defeater of the root is itself defeated. Recursively, this leads to a marking procedure of the tree that begins by considering the fact that leaves of the dialectical tree are undefeated arguments as a consequence of having no defeaters.

The following definition specifies the marking process of the nodes in a dialectical tree. Two labels are allowed: “U” for undefeated and “D” for defeated.

Definition 2. Let *T* be a dialectical tree with the argument *A* of L as its root. The corresponding marked dialectical tree, denoted *T**, will be obtained marking every node in *T* as follows:

1. All leaves in *T* are marked as "U" in *T**.
2. Let N be an inner node of *T*. The node N will be marked as "U" iff every child of N is marked as "D". The node N will be marked as "D" iff it has at least a child marked as "U".

Finally, an agent will believe in a literal L and therefore L will be a *warranted* literal, if there exists a dialectical tree with an argument for L as its root, so that the corresponding marked dialectical tree has its root marked with "U".

There exists four possible answers for a query L: **YES** if L is warranted, **NO** if L is warranted (i.e., the complement of L is warranted), **UNDECIDED** if neither L nor L are warranted, and **UNKNOWN** if L is not in the underlying signature of the program.

Games have an analogy with a dispute and, therefore, that analogy extends to argument-based reasoning. A dialectical tree can be seen as a game between two players: the proponent and the opponent. If the game is won by the proponent (the marked dialectical tree is labelled U) then the literal is warranted. The declarative semantics *GS* for DeLP characterizes its proof theory through a trivalued game-based minimal model [6,7,8,9].

3. Complexity of the marking process of a dialectical tree

DeLP is a reasoning system where every consequence of a d.l.p. is analyzed considering every argument for and against it. Thus, a dialectical tree is built in order to determine whether a literal is warranted. The dialectical tree have an analogy with games between two players: a proponent and an opponent. Therefore, it is possible to define in a declarative way when there exists a winning strategy for the proponent [6,7,9], i.e, we can determine when the marking process of the dialectical tree finishes labeling the root with U.

QSAT is a deeply studied decision problem that has a strong relationship with two-person games. The decision problem QSAT or *quantified satisfiability* is defined as follows: given a Boolean expression φ in conjunctive normal form, with Boolean variables x_1, x_2, ... x_n, is the prenex formula in conjunctive normal form (every quantifier is at the left of the formula) $\exists x_1 \forall x_2 \exists x_3 ... Q_n x_n$ φ satisfiable? where Q_i is the quantifier ∀ if i is even and Q_i is the quantifier ∃, if i is odd.

QSAT is PSPACE-complete[17] and it is very interesting decision problem since it can be considered as a game between two players: ∃ and ∀. The two players move in an alternating manner, with ∃ moving first. A movement consists of determining the truth value of the next variable, so that ∃ fixes the value for x_i if i is odd and $\forall$ fixes the value for x_i if i is even. After n moves, where n is the number of variables, one of the two players will win the game.

Thus, to make QSAT generally applicable to games, we have the following decision problem: given a legal situation of a game and a player, does this player have a winning strategy? For instance, let's consider a two-person game where A and B are the players and A begins the game, then this question can be expressed as:

$$\exists \text{ move for A } \forall \text{ move for B } \exists \dots \text{ (A wins)}$$

Note that this expression have an analogy with the way we make the marking process of a dialectical tree: the player $\exists$ tries to make satisfiable the formula while his is not defeated and the player $\forall$ tries to make unsatisfiable the formula.

In order to prove that the marking process of the dialectical tree is PSPACE-complete we must find a reduction of a PSPACE-complete problem to it. Thus, we introduce a transformation from the decision problem QSAT to the marking process problem.

Definition 3. We define the following transformation from a quantified Boolean formula in prenex conjunctive normal form $\exists x_1 \forall x_2 \exists x_3 \dots Q_n x_n\ \varphi$ to a dialectical tree:

- The nodes of a dialectical tree are:
  n is odd: $\{\varphi, \neg\varphi, x_1, \dots, x_n, \neg x_1, \dots, \neg x_n, C_1, \dots C_m, \neg C_1, \dots \neg C_m\}$
  n is even: $\{\varphi, \neg\varphi, x_1, \dots, x_n, x_{n+1}, \neg x_1, \dots, \neg x_n, C_1, \dots C_m, \neg C_1, \dots, \neg C_m\}$
  such that φ is a Boolean formula quantifier free, x_i, $1 \leq i \leq n$ the variables in φ and $C_1, \dots, C_m$ every clause in φ, i.e., $\varphi = C_1 \wedge C_2 \wedge \dots \wedge C_m$ and x_{n+1} an auxiliary argument.
- The attack and defeat relation between nodes is the following:
  n is odd: $\{(\neg\varphi, \varphi), (x_1, \neg\varphi), (\neg x_1, \neg\varphi), (x_2, x_1), (x_2, \neg x_1), (\neg x_2, x_1), (\neg x_2, \neg x_1), (x_3, x_2), (x_3, \neg x_2), (\neg x_3, x_2), (\neg x_3, \neg x_2), \dots, (x_n, x_{n-1}), (x_n, \neg x_{n-1}), (\neg x_n, x_{n-1}), (\neg x_n, \neg x_{n-1})\}$
  n is even: $\{(\neg\varphi, \varphi), (x_1, \neg\varphi), (\neg x_1, \neg\varphi), (x_2, x_1), (x_2, \neg x_1), (\neg x_2, x_1), (\neg x_2, \neg x_1), (x_3, x_2), (x_3, \neg x_2), (\neg x_3, x_2), (\neg x_3, \neg x_2), \dots, (x_n, x_{n-1}), (x_n, \neg x_{n-1}), (\neg x_n, x_{n-1}), (\neg x_n, \neg x_{n-1}) (x_{n-1}, x_n), (x_{n-1}, \neg x_n)\}$

Finally, en both cases, every clause C_i, $1 \leq i \leq m$ attacks and defeats x_n if n is odd and every clause C_i attacks and defeats x_{n-1} if n is even and if C_i is true under the truth values of that branch, then C_i is defeated by $\neg C_i$..

Example 1. Consider the following quantifier formula

$$\varphi = \exists x_1 \forall x_2 \exists x_3\ (x_1 \vee x_2) \wedge (\neg x_1 \vee \neg x_3) \wedge (x_2 \vee \neg x_3)$$

The dialectic tree generated is shown in the Figures 1 and 2.

The root node is the Boolean formula φ and its defeater is $\neg\varphi$. From this point on, every variable defeats the variable of the upper level. Every node corresponding to the variable is equivalent to indicate whether the variable is true (node x_i) or false (node $\neg x_i$).

Note that every clause in the formula defeats the last variable. If the clause is true in such branch (according to the truth values assigned to the variables), then is defeated by the negation of such clause. The partial trees in the Figure 1 and Figure 2

show the marking process. It begins labelling the leaves with U. Note that an assignation that makes true every clause finishes assigning the label U to the movement of the existential variable. The assignations that do not make every clause true assign D to the existential variable. Figure 1 shows that there exists a value for x_1=true, such that for all value of x_2, there exists a value for x_3 (for x_2=true, x_3=false and for x_2=false, x_3=false) such that it satisfies the formula. In Figure 2 we observe that x_1=false does not satisfy the formula for all the values of x_2. In particular, it is not satisfied for x_2=false.

Remind that in the marking process of the dialectical tree, a parent node is labelled U when *all its children* are labelled D. Intuitively, we want to apply this situation when the parent node corresponds to an universal quantified variable, so we can control that *every child* satisfies the formula.

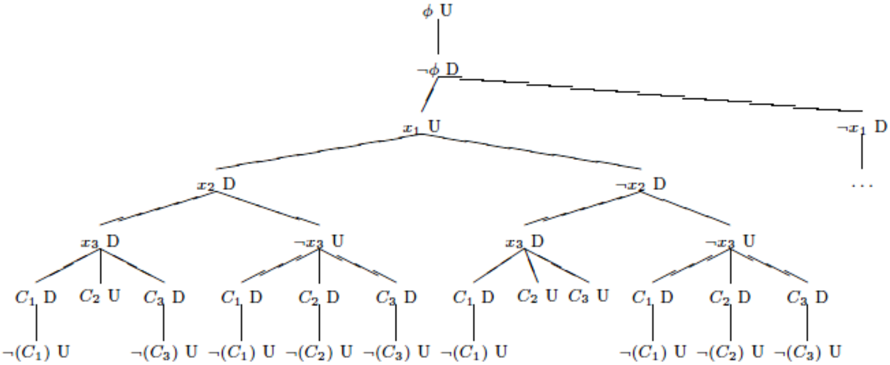


Fig.1. Partial labelled dialectical tree of the Boolean formula φ: $C_1=x_1 \vee x_2$, $C_2=\neg x_1 \vee \neg x_3$ and $C_3=x_2 \vee \neg x_3$.

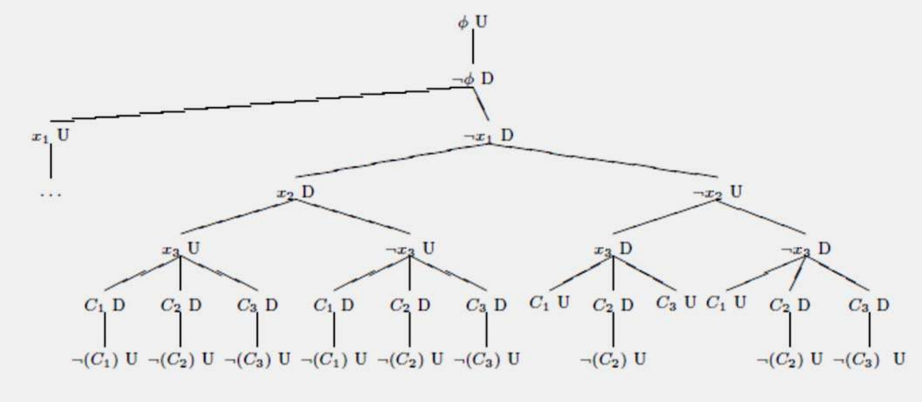


Fig. 2. Partial labelled dialectical tree of the Boolean formula
φ: $C_1=x_1$□ x_2, $C_2=\neg x_1$□$\neg x_3$ and $C_3=x_2$□ $\neg$ x_3.

On the other hand, a parent node is labelled D when at least one of its child is labelled U. This behavior is expected from a parent node that corresponds to an existential quantified variable.

Informally, we expect that the marking result of the nodes in the levels played by ∃ were all labelled D, so that the upper level, corresponding to the player ∀, were labelled U and so on.

Example 2. Consider the following quantified formula

$$\varphi = \exists x \forall y\ (x\text{□}\ y)\ \text{□}\ (\neg\ x\text{□}\ \neg\ y)$$

Figure 3 shows the dialectical tree built from φ. The formula is unsatisfiable, so the root node of the dialectical tree is labelled D. In this case, the number of variables is even, therefore we need an auxiliary node in order to achieve a dialectical tree whose first variable corresponds to the player ∀.

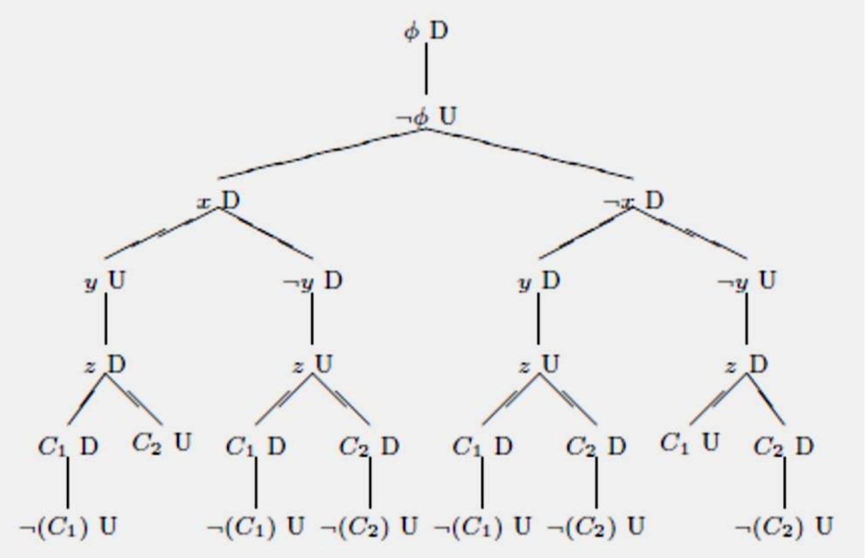


Fig. 3. Dialectical tree for the Boolean formula φ = ∃ x∀ y(x□ y) □ (¬ x□ ¬ y), where C_1=x□ y, C_2= ¬ x□ ¬ y and z is the auxiliary variable.

Theorem 1. *Given a dialectical tree T, determine whether its root is labelled U is PSPACE-complete.*

Proof: **Membership to PSPACE**: The dialectical tree is finite by definition, since a d.l.p. is finite and it cannot have cycles since it is not allowed to argument the same argument twice. A tree can be traversed in a depth-first way in polynomial space[19]. Thus, traversing a dialectical tree require polynomial space in the number of arguments in the d.l.p..

Completeness: Let φ a Boolean formula instance of the decision problem QSAT. We must prove that φ is satisfiable if and only if the dialectical tree is labelled U.

In order to prove this result, we use the transformation defined above from a Boolean formula instance of QSAT to a dialectical tree. Such transformation can be done in polynomial space.

If φ is satisfiable then in some subtree of the transformation of φ, the last two levels have the shape shown in the Figure 4. Thus, all the clauses C_1,..., C_m are true, and therefore, the node A is labelled U.

- If the formula have n variables, being n odd, then the node A corresponds with the representation of some truth value of x_n and it indicates that there is a truth value for x_n such that φ is true. The level of the parent of A is labelled D. We can prune the branch that contains C since we are analyzing ∀ x_{n-1}∃ x_n and for the value of the node B exists a value of x_n (value in A).

The formula will be satisfiable if this process repeats again with B's brother which corresponds with an alternative truth value of x_{n-1}. Thus, both B and its brother would be labelled D and its existential parent would be labelled U. The process will go on until we arrive to the first level which represents an universal quantifier and would be labelled U. Therefore , ¬ φ would be labelled D and finally , φ would be labelled U.

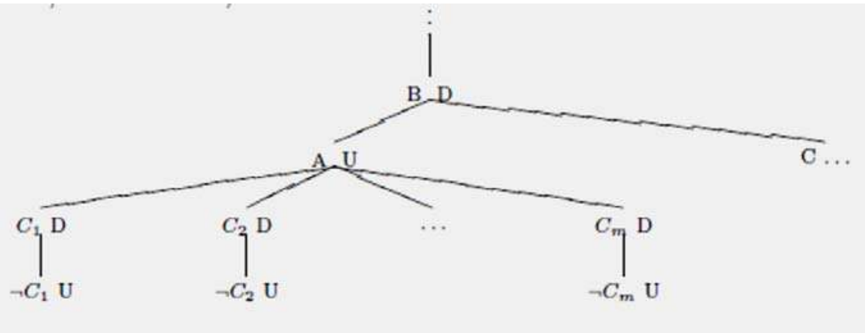


Fig. 4. Partial decision tree when all clauses are true.

- If the formula have n variables, being n even, then the node A is an auxiliary node and the node B is labelled D. This situation is shown in Figure 5. The node B is the representation of x_n with a truth value and this level represents $\forall x_n$. Since the formula is satisfiable, then process must repeat with the alternative truth value of B (node C) and therefore C will be labelled D. Thus, $\forall x_n$ is satisfied and the node E, which represents $\exists x_{n-1}$, will be labelled U. The process go on and the root will be labelled U.

If the root of the dialectical tree built through the transformation of a Boolean formula instance of QSAT is labelled U, then the second level will be labelled D (the node for ¬φ). The following level represents an existential quantifier, and therefore some node in such a level would be labelled U and its children would be labelled D, which corresponds to an universal quantifier, and so on.

- If the formula have an odd number of variables, then some node that represents $\exists x_n$ will be labelled U and its children (the clauses of the formula) will be labelled D. Thus, the nodes representing clauses cannot be leaves and therefore are true.

⅄ If the formula have an even number of variables, then the nodes that represent $\forall x_n$ will be labelled D and its auxiliary child U. Finally, the clauses of the formula will be labelled D. Thus, the nodes that represent the clauses cannot be leaves and therefore they are true.

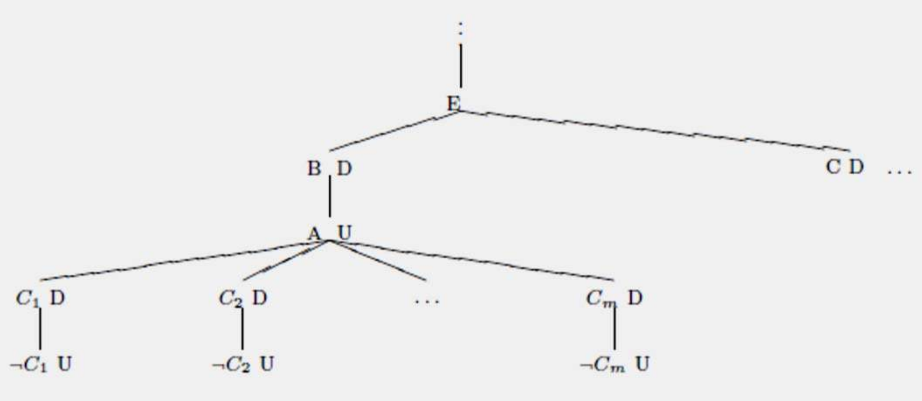


Fig. 5. Dialectical tree that represents a Boolean formula with an even number of variables n.

Therefore, the marking process of a dialectical tree is PSPACE-complete.

4. Conclusion and Future Work

We have analyzed the computational complexity of the marking process of a dialectical tree carried by DeLP. This point is of central importance in DeLP, in order to determine whether this literal is supported by the root argument of the tree and therefore, believed by a reasoning agent.

We show that there exists an analogy between a dialectical tree and a two-person game and the decision problem QSAT. The main contribution of this work is the proof that the marking process of a dialectical tree is PSPACE-complete.

In [6,7] we have introduced some relevant decision problems for DeLP and we have presented some computational complexity results. Moreover, we have study DeLP as a query language in order to use DeLP over as a database technologies.

As future work we will analyze combined complexity of the decision problems introduced in [7] using the result obtained in this work. We are studying the descriptive complexity of DeLP, in order to determine the queries expressible in DeLP.

References

1. Amgoud, L., Cayrol, C.: A reasoning model based on the production of acceptable arguments. Annals of Math and Artificial Intelligence 34, 197-215 (2002).
2. Atkinson, K., Bench-Capon, T., Mc Burney, P.: A dialogue game protocol for multiagent argument over proposals for action. Tech. Rep. ULCS-04-007, Department of Computer Science, University of Liverpool, UK (2004).
3. Bassiliades, N., Antoniou, G., Vlahavas, I.: A defeasible logic reasoner for the semantic web. In: Proc. of the Workshop on Rules and Rule Markup Languages for the Semantic Web. pp. 49-64 (2004).
4. Bench-Capon, T.J.M.: Persuasion in Practical Argument Using Value Based Argumentation Frameworks. Journal of Logic and Computation 13(3), 429-448 (2003).
5. Bondarenko, A., Dung, P., Kowalski, R., Toni, F.: An Abstract, Argumentation-Theoretic Approach to Default Reasoning. Artificial Intelligence 93(1-2), 63-101 (1997).
6. Cecchi, L.A., Fillottrani, P.R., Simari, G.R.: An Analysis of the Computational Complexity of DeLP through Game Semantics. In: XI Congreso Argentino de Ciencias de la Computación. pp. 1170-1181. Universidad Nacional de Entre Ríos, Argentina (Octubre 2005).
7. Cecchi, L.A., Fillottrani, P.R., Simari, G.R.: On the complexity of DeLP through game semantics. In: Dix, J., Hunter, A. (eds.) XI International Workshops on Nonmonotonic Reasoning. pp. 386-394. Clausthal University (2006).
8. Cecchi, L.A., Simari, G.R.: Sobre la Relación entre la Definición Declarativa y Procedural de Argumento. In: VI CACiC. pp. 465-476. Ushuaia (2000).
9. Cecchi, L.A., Simari, G.R.: Sobre la relación entre la Semántica GS y el Razonamiento Rebatible. In: X CACiC - Universidad Nacional de La Matanza. pp. 1883-1894. San Justo - Pcia. de Buenos Aires (2004).
10. Chesñevar, C., Maguitman, A.: An Argumentative Approach to Assessing Natural Language Usage based on the Web Corpus. In: Proc. of the European Conference on Artificial Intelligence (ECAI) 2004. pp. 581-585. Valencia, Spain (August 2004).
11. Chesñevar, C., Maguitman, A.: ARGUENET: An Argument-Based Recommender System for Solving Web Search Queries. In: Proc. of the 2nd IEEE Intl. IS-2004 Conference. pp. 282{287. Varna, Bulgaria (June 2004).
12. Dimopoulos, Y., Nebel, B., Toni, F.: On the Computational Complexity of Assumption-based Argumentation for Default Reasoning. Articial Intelligence 141(1), 57-78 (2002).
13. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning and logic programming and n-person games. Artificial Intelligence 77, 321-357 (1995).
14. Dunne, P.E., Wooldridge, M.: Complexity of Abstract Argumentation, pp. 85-103. Springer (2009), Argumentation in Articial Intelligence.
15. García, A.J., Simari, G.R.: Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming 4(1), 95-138 (2004).
16. Maher, M.J.: Propositional defeasible logic has linear complexity. Theory and Practice of Logic Programming 1(6), 691-711 (2001).
17. Papadimitriou, C.: Computational Complexity. Addison-Wesley Publishing Company (1994).
18. Pollock, J.: Defeasible Reasoning. Cognitive Science 11, 481-518 (1987).
19. Russell, S., Norvig, P.: Artificial Intelligence: A modern approach. Prentice Hall, New Jersey, third edn (2010).

20. Simari, G.R., Loui, R.P.: A mathematical treatment of defeasible reasoning and its implementation. Artificial Intelligence 53, 125-157 (1992).
21. Verheij, B.: Argumed - a template-based argument mediation system for lawyers. In: Hage, J., Bench-Capon, T.J., Koers, A., de Vey Mestdagh, C., Grütters, C. (eds.) Legal Knowledge Based Systems. JURIX: The Eleventh Conference. pp. 113-130 (1998).

II

Signal Processing and Real-Time Systems Workshop

Abstract Data Type for Real-Time Database Systems

CARLOS E. BUCKLE, JOSÉ M. URRIZA, DAMIÁN P. BARRY, FRANCISCO E. PÁEZ

Facultad de Ingeniería, Departamento de Informática
Universidad Nacional de La Patagonia San Juan Bosco - Puerto Madryn, Argentina
cbuckle@unpata.edu.ar, josemurriza@unp.edu.ar, demian.barry@gmail.com, franpaez@gmail.com

Abstract. *Real-Time Systems incorporate data-intensive applications within a wide range of solutions. A common problem that developers have to manage is data-oriented design with temporal constraints, since it implies considering specific rules and properties to guarantee the validity of objects at particular points of time. This work puts forward a model to facilitate this task. We present the concept of real-time data with guarantees of temporal consistency, and a set of associated classifications and definitions. Based on this, we model abstract data type that can be parameterized and encapsulates attributes and validations of temporal constraints. Thus, the application developer can be freed from these design responsibilities. The result is verified applying the model to a specific design case, in an industrial systems problem.*

Key Words: *real-time database, temporal consistency, temporal data, data-deadline, real-time software engineering, real-time transactions.*

1. Introduction

Papers related to Real Time Systems (*RTS*) modeling have considered the classic problems generally applicable to embedded systems of industrial automation, aviation, space exploration, telephone exchange and others. The evolution of developments in this area has allowed the expansion towards other areas of application that involve interaction with the user and general purpose systems, such as online stock trading systems, supervision systems, balanced scorecard for management decision-making, etc. In these systems, real-time tasks have to manage large data volume with persistence requirements. The study of these scenarios has been mainly developed by a subdiscipline called *Real-Time Database Systems* (*RTDBS*) ([1, 2]).

The *RTDBS* must handle conventional data objects as well as real-time objects. These objects are responsible of reflecting changes of the variable elements of the environment. For this purpose, their values are updated with periodic readings of the sensors that communicate with the environment. These objects have a special feature: their values *get older* until they become obsolete when they reach their *datadeadline* ([3]). The tasks that handle these data objects are called *real-time transactions* ([4]). In addition to guaranteeing the logical consistency of the involved data, should ensure the temporal consistency (validity) and meet the deadline established for their response.

In the construction of *RTDBS,* designers not only must consider the process modeling, but they also need to emphasize on data modeling with real-time constraints.

Software engineering has generated important breakthroughs in the conventional *RTS* and *DBS* disciplines, but both have worked separately. On the one hand, in *DBS,* time variable information modeling was developed by temporal databases, some works using relational approach ([5]) and some others using object-oriented approach ([6]). These *DBS* handles *time* domain, and its representational structure and different dimensions of time has to be considered in a transaction. Unfortunately, they do not contemplate the specific Real-Time context, in which data can expire during the transaction, which, in turn, has to be planned to fulfill with its deadline.

On the other hand, in *RTS* discipline, there are software engineering works, such as Douglases design patterns [7], which present solutions to common *RTS* problems and describes guidelines on development methodology. There are also recommendations such as *MARTE* ([8]) (*OMG: Modeling and Analysis of Real-Time Embedded systems*) that defines a profile for specification, design and validation stages of *RTS*. These proposals are broad and encompass the whole range of *RTS*, guiding the development process at high levels. However, they have not focused on *RTDBS*.

In the *RTDBS* subdiscipline, works are specifically geared towards data modeling with real-time constraints. In the *Real Time Semantic Objects Relationships And Constraints* (*RTSORAC*) ([9]) model, the components that a *RTDBS* model must have are defined and described. Afterwards, some of these authors present a UML package to specify real-time objects ([10]), that was taken as basis for other design profiles such as the one presented by Idoudi et al. in [11]. Although these works introduce a starting point, it is necessary to have greater levels of refinement so that they can be applied in specific designs. In addition, it is necessary to widen the scope to include other aspects, such as relative temporal consistency and real-time objects with discrete state changes.

This work presents a solution for specification of real-time elements within data model bounds. Parameterized abstract data type is defined. These data type encapsulate time properties and temporal consistency validations, so that the applications developer can be freed from these responsibilities. The data type parameter allows the implementation of any kind of object, extending its functionality with characteristics and constraints typical of real time objects. The model supports the different validation rules and classifications identified in this paper. To verify its result, it has been applied to a specific case of *RTDBS* design in the area of industrial IT.

The rest of document is organized in the following way: in section 2 we present the basic concepts of temporal consistency in *RTDBS*. In section 3 we present the resulting work and its corresponding verification. In section 4 we draw conclusions and put forward possible future work.

2. Temporal Consistency in RTDBS

RTDBS are applied in dynamic contexts in which it is necessary to detect changes in the environment and reflect them in system data, process them in database transactions, and finally transform them into outputs towards a system actuator.

These dynamics are restricted by a ruling principle: the transactions are considered satisfactory if, the results are logically and arithmetically correct and are produced before the deadline ([12]) and in addition, carried out within the *datadeadline.*

RTDBS data objects, as well as the operations, have temporal constraints ([13]). Consequently, the concept of *real-time data* arises. These are classified into r*eal-time base data* (*RTBD*)*,* which does not depend on any other data and reflect the states of an external object and *real-time derived data* (*RTDD*)*,* which are computed based on some others *RTBD* or *RTDD.*

Furthermore, the environment sensing activities can be passive or active. The *passive sensing* is started from the system, which surveys the sensor to get its value (*pull* mechanism). In the *active sensing*, on the contrary, it is the sensor that takes the initiative to transmit its value (*push* mechanism). Below we present an example that will be used throughout this document.

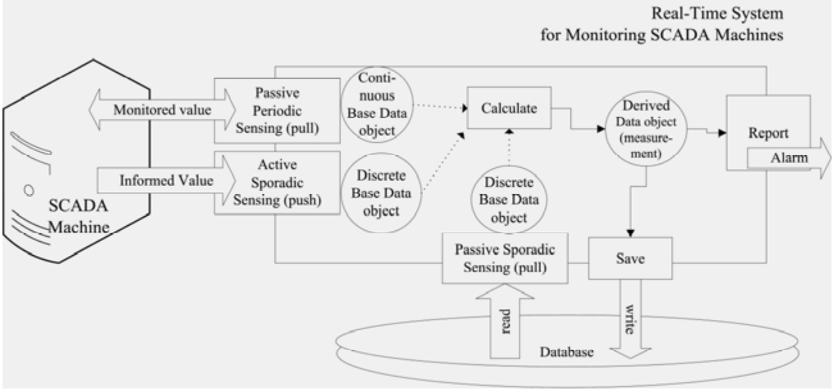

Real-Time System
for Monitoring SCADA Machines
Monitored value
Informed Value
SCADA Machine
Passive Periodic Sensing (pull)
Active Sporadic Sensing (push)
Conti-nuous Base Data object
Discrete Base Data object
Calculate
Discrete Base Data object
Derived Data object (measure-ment)
Report
Alarm
Passive Sporadic Sensing (pull)
Save
read
write
Database

Fig. 1. Example of RTDBS. Real-Time System for Monitoring SCADA Machines

This example is based on a factory that has a *SCADA* (Supervisory Control And Data Acquisition) computer network in charge of supervising industrial processes. To guarantee the correct functioning of those machines and in order to anticipate possible failures, it has implemented an automated monitoring system. The system makes hardware measurements (processor temperature, speed of coolers, etc.), operating system measures (CPU use, memory use, process execution, reboot, etc.) and measures on *SCADA* application (specific values of its database). If values obtained in measures exceeds desirable range, certain alarms are generated so that an operator can take care of the situation. Besides, some measurements need to analyze historical values to generate alarms, for example, work load variation in the last minute. The case described can schematically be seen in Figure 1.

Base Data and Absolute Temporal Consistency

The *RTBD* reflect the state of an external object and allow to detect changes in the environment. Changes can have a *continuous* or *discrete* behavior [14]. An external entity with *continuous state-change* is the one that generates continuous variations in time (for example, the processor temperature or the speed of coolers). In turn, an external entity with *discrete state-change* is the one that generates variations at discrete and spaced moments, for example, the number of processes executing in a *SCADA* machine or the average of the latest load measurements.

Based on the above, *RTBD* are classified according to the external object they reflect, into *continuous RTBD* and *discrete RTBD*. The *continuous RTBD* are updated with periodic samples. The discrete *RTDB,* in contrast, are updated sporadically, only when the external value changes.

A *RTBD* expires over time. The system must guarantee that it is working with *RTBD* that are correctly updated. The guarantee that a *RTBD* is up to date and equivalent to the environment value is called *absolute temporal consistency* or *external consistency* ([15]).

We will call VI_b to *Absolute Validity Interval* of a *RTBD b.* The *validity interval lower bound* (*VILB*) is timestamp of update of *b*, and the *validity interval upper bound* (*VIUB*) is datadeadline of *b*.

A *RTDB b* satisfies the absolute temporal consistency if the instant in which access occurs (*now*(*t*)) belongs to the VI_b interval. Therefore:

$$VILB_b(t) \le now(t) \le VIUB_b(t)$$

In a *continuous RTDB* its validity depends on its *age.* The *age* is the time from its creation or last update to the present moment. The systems establishes the *maximum age* (*MA*) ([16]) that is valid for specific *b* data. When this age is exceeded, data expires, this means: $VIUB_b(t) = VILB_b(t) + MA_b$.

This cannot be applied to a *discrete RTDB*, since in this case that value is valid as long as it does not change. Consequently, $VIUB_b(t)$ is the moment in which *b* updates its value.

Derived Data and Relative Temporal Consistency

The *RTDD* are *calculated data*. Their values are determined with operations over a *Read-Set* of other *RTD. RTDD* contain at least one *RTD* in its *Read-Set.* The *Read-Set* objects of a *RTDD* must reflect close time intervals. They must be contemporaries. This requirement is known as *relative temporal consistency.*

The *Read-Set* of a *RTDD d* fulfills relative temporal consistency if there is intersection between the *IV* of its elements. This means:

$$\bigcap\left\{VI_x(t)\middle|x \in \text{ReadSet}_d\right\} \neq \varnothing$$

This defines for a *RTDD d*, an interval of relative validity VI_d with:

$$VILB_d(t) = Max\left\{VILB_x(t)\middle|x \in \text{ReadSet}_d\right\}$$

$$VIUB_d(t) = Min\left\{VIUB_x(t)\middle|x \in \text{ReadSet}_d\right\}$$

Transactions using *RTD*

According to Stankovic ([12]), the time constraints for *RTDBS* transactions are defined by the *RTD* validity intervals involved in it and by the specific characteristics of the transactions, such as periodicity or response time previous to the *deadline*. In *RTDBS* two kinds of transactions are identified: those of *RTD* (base or derived) updating and those in which the user implements the logic of its application. We can identify:

RTBD Update Transactions (*BT*): Transactions that use *RTBD* for write-only. The system has to implement them to guarantee the absolute temporal consistency. They can be periodic or sporadic according to whether they update continuous or discrete *RTBD.*

RTDD Update Transactions (*DT*): Transactions that use some *RTD* for read and uses other *RTDD* for write. The system has to implement them to guarantee the re-calculation of derived data. In them, the relative temporal consistency over the Read-set has to be verified.

User Transactions (*UT*): They implement the logic of the user application. They are *read-only* transactions over any *RTD*, although they also use conventional data which does not have real-time restrictions.

The update transactions (*BT or DT*) can be implemented as independent transactions or as sub-transactions of a *UT* transaction, according to whether the policy is *immediate* or *on demand* update [17]. The *immediate* transaction guarantees the update of *RTD* regardless of the *UT* that use them. While *on demand* transactions are only executed when a *UT* needs access to the *RTD.*

In the case of *independent BT*, the periodic update of a *continuous RTBD b* guarantees its consistency if it is executed in a period: $P \leq MA_b/2$. Consequently, the period must not exceed a half of the maximum age ([3]). With small periods too much overhead for *BT* and *DT* chained transactions may be generated, probably due to applying an insignificant change in the external object. This introduces the concept of *Maximum Data Error* (*mde*)

([18]), which permits to discard *BT* if the variation between the data registered value and the new value is not sufficiently significant. A *BT* is discarded if:

$$|NewValue_b - OldValue_b| \leq mde.$$

RTD Versioning

If *BT*, *DT* and *UT* are independent transactions, concurrent conflicts must be considered. This is due to the fact that there can be an instantiation of *BT* transactions that may need to re-write *RTBD* before the *DT* or *UT* that use them have finished. If optimist concurrence control is used, this may result in a high rate of *DT* or *UT* transaction restart. A blocking mechanism, instead, may incorporate delays in the *BT* transactions.

Song y Liu ([19]) have proposed, for these cases, the *RTD* model of multiple versions. For each update of a *RTD* a new version is generated. Each version has its own validity interval and at a certain moment there can be more than one valid version. In this way, read transactions can be executed without performing a resource contention for write transactions. The implementation of *RTD* versioning may increase the cost of the solution but it minimizes concurrence conflicts and it has proved to improve general system performance ([19]).

3. Abstract Data Type for Real-Time Objects

The goal of this work is to achieve a model that may represent *RTD*, including the classifications mentioned in the previous section, and that may guarantee the corresponding validations of temporal consistency in an encapsulated way.

We define, then, *RTD* abstract data type that can be parameterized, and that have as minimum attributes a *value* and a validity interval (*VILB, VIUB*). Then we define specific subtypes *ContinuousRTBD*, *DiscreteRTBD* y *RTDD*. They can be used to typify transaction variables or entity attributes in a *RTDBS* in a simple way, as for example:

ContinuousRTBD<float> cpuWLoad;

With this statement, the programmer declares a *cpuWLoad* that will take a *float* type value, but which will also guarantee the properties and the rules of continuous RTBD. When creating a new object instance, parameters that allow its configuration must be indicated, for example:

cpuWLoad = new ContinuousRTBD(*MA*→*4, mde*→*0.5*)

This indicates that *cpuWLoad* will have a maximum age (*MA*) of 4 seconds and a maximum error (*mde*) of 0.5. Then, in successive *BT* transactions, the object value will be updated using the *set* operation. For example:

cpuWLoad.set(*36.52*)*;*

If this happens at the *t* instant, the system will define the validity interval of *cpuWLoad* with limits (*VILB*→*t*, *VIUB*→*t* + 4sec), and then will assign the value 36.52.

The *TD* or *TU* transactions that need to recover the *cpuWLoad* value will use *get*() and will be able to operate it transparently as *float*. For example:

cp = (*float*) *cpuWLoad.get*() ** 0.01 ;*

If this takes place in an instant *t',* the system will check the absolute temporal consistency (*cpuWLoad.VILB* ≤ *t'* ≤ *cpuWLoad.VIUB*), generating an error if it is not fulfilled.

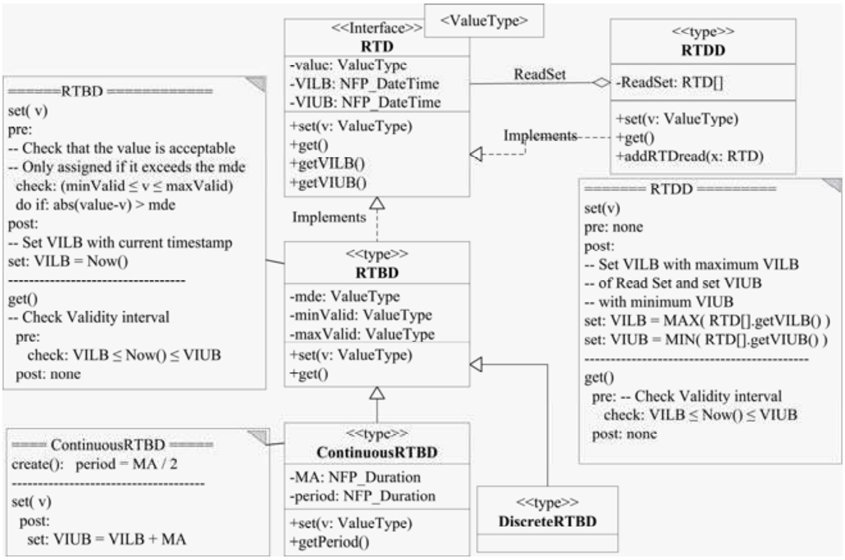


Fig. 2. UML Class Diagram for proposed model: *RTD abstract data type*

The *RTD* validity interval is updated internally every time a value is assigned, and the temporal consistency check is carried out every time that value is recovered. This is for both, *RTBD* and *RTDD*. The model proposed is presented as a UML class diagram in Figure 2.

The actions and validations performed by each subtype are described in the notes on the diagram. To declare values and time durations, non-functional data-types defined in *MARTE* (*NFP_Types*) ([8]) were used.

The *RTBD* define their absolute validity interval according to the value assignation instant, contemplating the maximum error *mde,* and considering optional definition of a valid value range to discard possible read errors (*minValid* and *maxValid*). In the special case of *ContinuousRTBD*, a *period* attribute, which is calculated according to the maximum age *MA* is used. This attribute is useful to schedule the periodic executions of *BT*.

The *RTDD* define their Read-Set as a collection of *RTD.* When creating *RTDD,* the *RTD* that make up the read-set must be added using the *addRTDRead* operation. This allows the system to validate the relative temporal consistency of *RTDD*, computing its validity interval as the intersection over its *Read-set RTD* validity intervals.

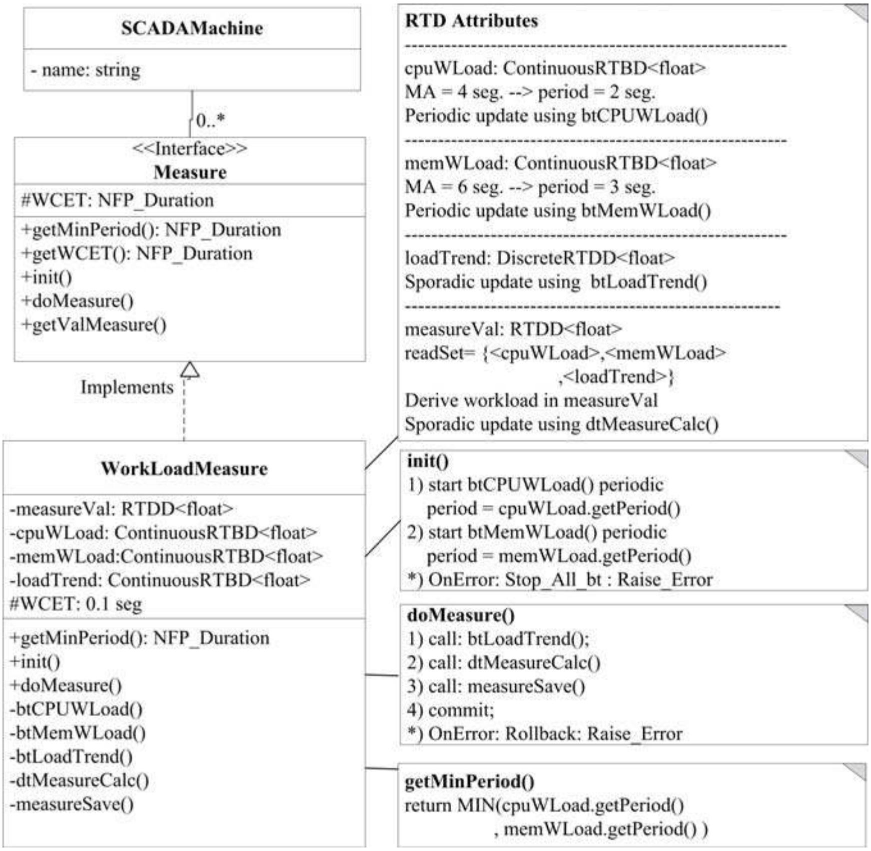

SCADAMachine
- name: string
0..*
<<Interface>>
Measure
#WCET: NFP_Duration
+getMinPeriod(): NFP_Duration
+getWCET(): NFP_Duration
+init()
+doMeasure()
+getValMeasure()
Implements
WorkLoadMeasure
-measureVal: RTDD<float>
-cpuWLoad: ContinuousRTBD<float>
-memWLoad:ContinuousRTBD<float>
-loadTrend: ContinuousRTBD<float>
#WCET: 0.1 seg
+getMinPeriod(): NFP_Duration
+init()
+doMeasure()
-btCPUWLoad()
-btMemWLoad()
-btLoadTrend()
-dtMeasureCalc()
-measureSave()
RTD Attributes
cpuWLoad: ContinuousRTBD<float>
MA = 4 seg. --> period = 2 seg.
Periodic update using btCPUWLoad()
memWLoad: ContinuousRTBD<float>
MA = 6 seg. --> period = 3 seg.
Periodic update using btMemWLoad()
loadTrend: DiscreteRTDD<float>
Sporadic update using btLoadTrend()
measureVal: RTDD<float>
readSet= {<cpuWLoad>,<memWLoad>
,<loadTrend>}
Derive workload in measureVal
Sporadic update using dtMeasureCalc()
init()
1) start btCPUWLoad() periodic
period = cpuWLoad.getPeriod()
2) start btMemWLoad() periodic
period = memWLoad.getPeriod()
*) OnError: Stop_All_bt : Raise_Error
doMeasure()
1) call: btLoadTrend();
2) call: dtMeasureCalc()
3) call: measureSave()
4) commit;
*) OnError: Rollback: Raise_Error
getMinPeriod()
return MIN(cpuWLoad.getPeriod()
, memWLoad.getPeriod())

Fig. 3. Application of *RTD* data-type. Measurements on *SCADA* machines

The verification of the model is performed by implementing the design of measurements on the example in section 2.

On a specific *SCADA* machine a set of measurements must be performed. Each one must specify its worst case execution time (*WCET*) and its minimum period of execution, as useful information for the measurements scheduler. In addition, each measurement must implement the public operations:

- *init*()*:* to perform measurement initialization tasks and start *BT* update transactions on *ContinuousRTBD.*
- *doMeasure*()*:* to calculate the measurement value and save it in the database.

In Figure 3 we show the generic measurement model and one of them in particular is implemented: the measurement of the workload on a *SCADA* machine. This measurement is calculated according to the CPU workload, the memory workload and the workload trend registered in the database in the last minute. The CPU workload and the memory workload are *ContinuousRTBD* and have a maximum *age* of 4 seconds and 6 seconds respectively. The workload trend is a *DiscreteRTBD* that is obtained reading historical records from database. The measurement value is a *RTDD* that is computed according to the previous data. The diagram notes show how the model is implemented and how the *RTD* are defined by using parameterized abstract data type.

The *init*() operation triggers two independent *BT* to obtain the *ContinuousRTBD*: *btCPUWLoad*() with a period of 2 seconds and *btMemWLoad*() with a period of 3 seconds. The *doMeasure*() operation first executes a *btLoadTrend*() *BT* to read the workload trend from database, then derives the calculated value of the present workload using *tdMeasureCalc*() and finally saves it in the database using the *masureSave*() operation.

This specific design application allows us to see how the use of *RTD* abstract data type facilitates the definition of the problem, allows to organize transactions in a modular way and frees the programmer from the temporal consistency validation.

4. Conclusions and Future Work

We have been able to encapsulate the set of rules and characteristics of real-time temporal consistency within abstract data type that can be parameterized. This can be applied directly on entity attributes or program variables in a *RTS*. It has also been possible to verify it applying it to the design of a specific application. At the verification stage, when task design was performed, it was been observed that a reusable model was possible not only for *RTD* but also for *BT*, *DT* and *UT* transactions. Future work should focus on obtaining a generic model of real-time operations that involve *RTD* in order to facilitate the design and the implementation of *RTDBS* transactions.

References

1. K. Ramamritham, "Real Time Databases", *International Journal of Distributed and Parallel Databases,* vol. 1, pp. 199-226, 1993.
2. B. Purimetla, R. Sivasankaran, K. Ramamritham, and J. Stankovic, "Real-Time Databases: Issues and Applications", in. vol. ch.20, ed: in S.Son (ed.) Advances in Real-Time Systems, Prentice Hall, 1995.
3. M. Xiong, J. A. Stankovic, K. Ramamritham, D. Towsley, and R. Sivasankaran, "Maintaining Temporal Consistency: Issues and Algorithms", in *Proceedings of International Workshop on Real-Time Database Systems*, 1996, pp. 2-7.
4. R. Abbott and H. Garcia-Molina, "Scheduling Real-Time Transactions: A Performance Evaluation", in *Proceedings of the 14th VLDB Conference*, 1988.
5. C. Combi, S. Degani, and C. S. Jensen, "Capturing Temporal Constraints in Temporal ER Models," in *Proceedings of the 27th International Conference on Conceptual Modeling*, ed Berlin, Heidelberg: Springer-Verlag, 2008, pp. 397-411.
6. Ellen Rose and A. Segev, "TOODM - A Temporal Object-Oriented Data Model with Temporal Constraints", in *Proceedings of the 10th International Conference on Entity-Relationship Approach* (*ER'91*), San Mateo, California, USA, 1991.
7. B. P. Douglass, *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*: Addison-Wesley, 2002.
8. O. M. G. (OMG). (2009, OMG Document Number: formal/2009-11-02). *A UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded systems*. Available: http://www.omg.org/spec/MARTE/1.0/
9. J. J. Prichard, L. C. DiPippo, J. Peckham, and V. F. Wolfe, "RTSORAC: A Real-Time Object-Oriented Database Model", *In The 5th International Conference on Database and Expert Systems Applications,* pp. 601-610, 1994.
10. L. C. DiPippo and L. Ma, "A UML Package for Specifying Real-Time Objects," *Computer Standards & Interfaces* vol. 22, pp. 307-321, 2000.
11. N. Idoudi, C. Duvallet, B. Sadeg, R. Bouaziz, and F. Gargouri, "Structural Model of Real-Time Databases: An Illustration", in *Object Oriented Real-Time Distributed Computing* (*ISORC*)*, 2008 11th IEEE International Symposium on*, 2008, pp. 58-65.
12. J. A. Stankovic, S. Son, and J. Hansson, "Misconceptions About Real-Time Databases," *IEEE Computer,* vol. 32, pp. 29-36, 1998.
13. K. Ramamritham, S. H. Son, and L. C. Dipippo, "Real-Time Databases and Data Services", *Real-Time Syst. Kluwer Academic Publishers,* vol. 28, pp. 179-215, 2004.
14. K. Ben, L. Kam-Yiu, B. Adelberg, R. Cheng, and T. Lee, "Maintaining temporal consistency of discrete objects in soft real-time database systems," *Computers, IEEE Transactions on,* vol. 52, pp. 373-389, 2003.
15. P. Yu, K. Wu, K. Lin, and S. H. Son, "On Real-Time Databases: Concurrency Control and Scheduling", in *Special Issue on Real-Time Systems*, Proceedings of IEEE, 1994, pp. 140-157.

16. B. Adelberg, H. García-Molina, and B. Kao, "Applying update streams in a soft real-time database system", *Proceedings of the 1995 ACM SIGMOD,* vol. 24, pp. 245-256, 1995.
17. Y. Wei, J. A. Stankovic, and S. H. Son, "Maintaining Data Freshness in Distributed Real-Time Databases", presented at the Proceedings of the 16th Euromicro Conference on Real-Time Systems, 2004.
18. M. Amirijoo, J. Hansson, and S. H. Son, "Specification and management of QoS in real-time databases supporting imprecise computations", *Computers, IEEE Transactions on,* vol. 55, pp. 304-319, 2006.
19. X. Song and J. Liu, "Maintaining temporal consistency: pessimistic vs. optimistic concurrency control", *Knowledge and Data Engineering, IEEE Transactions on,* vol. 7, pp. 786-796, 1995.

I ETHICOMP Latinoamérica

Advantages and Trade-Offs of Introducing Ethical Issues in Computing through a Dedicated Course or through Modules in Relevant Content Courses in the Curriculum

WILLIAM M. FLEISCHMAN[1], DANIEL T. JOYCE[2]

[1] Departments of Computing Sciences and Mathematical Sciences
Villanova University
Villanova, Pennsylvania 19085, U. S. A.
william.fleischman@villanova.edu
[2] Department of Computing Sciences
Villanova University
Villanova, Pennsylvania 19085, U. S. A.
daniel.joyce@villanova.edu

Abstract. *We discuss two alternatives for introducing consideration of ethical questions in the computer science curriculum. These alternatives are 1) a self-contained course on ethical issues in computing, and 2) introduction of modules devoted to ethical questions throughout the curriculum in content courses such as software engineering, databases, data mining, artificial intelligence, and systems. We discuss the advantages and the potential "hidden messages" involved in each of these approaches. By way of illustration, we list some of the pertinent points raised by two important case studies that are appropriate for inclusion in either a self-contained course or a course on software engineering.*

Keywords: *Computer ethics, Ethical questions in software engineering, Case studies.*

1. Introduction

The over-arching goal of consideration of ethical issues in computing has been given paradigmatic expression by Terry Bynum [1]:

> To integrate computing technology and human values in such a way that the technology advances and protects human values, rather than doing damage to them.

It is generally recognized that, in pursuit of this goal, curricula in computer science and computer-related fields should include explicit consideration of ethical issues raised by applications of computer and information technology in building life-critical, safety-critical, and privacy-critical systems. For undergraduate curricula in the United States, for example, the requirements of the Computing Accreditation Commission of ABET specify that to be approved as accredited any program must present documented measurable outcomes that "enable students to achieve i) an understanding of professional ethical, legal, security, and social issues and responsibilities; and ii) an ability to analyze local and global impact of computing on individuals, organizations, and society" – by the time students are eligible to graduate. [2]

The historical origin of these requirements lies in incidents such as the 1985-86 series of computer-related radiation therapy accidents related to the Therac-25 [3] and the launch in 1987 of the so-called Internet Worm [4]. Since that time, a steady stream of similar stories has provided reinforcement on a regular basis of the need to treat ethical issues in the computer science curriculum [5], [6], [7], [8], [9] and [10].

In the context of the undergraduate computer science curriculum, pursuit of the goal articulated by Terry Bynum often requires appeal to and stimulation of students' imaginations concerning situations they will face five or ten years in the future in the early stages of their professional careers. Without the ability to transcend the relatively protected idea space of their lives as university students, discussion of actual or potential ethical dilemmas may seem artificial and somewhat distant. Thus one important skill that students must develop is the exercise of their powers of imagination and empathic response to help them place themselves in the situation of individuals, often from very different backgrounds than their own, enmeshed in situations involving complex and conflicting power relationships and vulnerabilities. Lacking this ability, students are often give way to the temptation to reduce these problematic situations to simple, one-dimensional self-other oppositions. [11]

2. Some General Observations

Terry Bynum describes three modes of treatment of ethical issues in the computer science curriculum – i) a "stand-alone" course dedicated to ethical issues in computing; ii) the introduction of "case studies in every course" throughout the curriculum; and iii) a "capstone course" in software engineering integrating thorough treatment of ethical issues [1]. In this paper, we will collapse the latter two modes into a single alternative which we will refer to as the "module" approach in which courses in the computer curriculum that have significant technical and scientific content/goals also include modules devoted to ethical issues. Drawing on our own experiences, we will consider the advantages and trade-offs presented by the "stand-alone" course and the approach based on the use of modules.

Since many of the most important and useful examples for discussion of ethical issues come from the area of software engineering, it should be clear how to capitalize on the opportunities afforded adoption of modules in courses directly related to software engineering, including the capstone project-based approach. Still, the examples we cite should also suggest ways to relate some of our insights to other subject areas –for example, artificial intelligence, data mining, and robotics– within the computer science curriculum.

It is worth noting that, in our experience, a case study such as the series of Therac-25 radiation therapy accidents consistently induces astonishment and unhappy surprise among students at the extent of serious harm caused in part by faulty engineering design (involving both hardware and software) of a computer controlled system that they are able to recognize as not unlike systems they may eventually be called upon to implement. We believe strongly that consideration of case studies, carried out with appropriate seriousness and care, has an important role within either of the two approaches that we discuss.

3. Hidden Messages

Before embarking on the discussion of the advantages and trade-offs involved, it seems worthwhile to consider the "hidden messages" conveyed by each of these approaches. It may well be that every course within a given curriculum carries a set of explicit intentions or purposes that are relatively clear and straightforward. On the other hand, the status of the course within the curriculum and the manner in which it is presented often carry hidden messages that can reinforce or subvert the explicit purposes the course is presumed to serve.

What are the messages conveyed by a dedicated, required course in computer ethics? To begin with, this provides a clear signal that those who supervise the curriculum consider the subject important enough to invest precious curricular time to expose students to concepts and case studies that will be treated in depth. Next there are the potential messages associated with the identity of the faculty member who presents the course. If the course is taught by someone from the students' own department, moreover by an individual who takes a serious and informed approach to the material, the tendency will be to reinforce –perhaps very strongly– the importance students attach to the subject. The same effect can be achieved if the instructor is an external individual from the faculty of philosophy or ethics who has nonetheless taken the trouble to cultivate familiarity with the range of questions germane to this area of applied ethics and is able to present at least a few important case studies with an adequate level of understanding of the technical issues relevant to each case. A course taught jointly by a computer scientist and an ethicist would clearly present another favorable situation. On the other hand, a course presented by a disaffected instructor –either a computer scientist or an ethicist who conveyed the sense of having been given an unpleasant or unimportant assignment– would send a strong message of a contrary nature.

The modular approach has the same possibility of sending conflicting hidden messages to computing students. If each course in the curriculum incorporates a well-designed and substantial module, perhaps in the form of a case study, involving a relevant ethical problem, the message to students will be, "This is a subject that is intrinsic to virtually every aspect of the discipline. It engages the attention of all my instructors. I had better be sensitive to similar situations in my professional life." If, on the other hand, the modules are presented superficially or with embarrassment in more than a few instances, the hidden message will be one detrimental to student engagement with ethical problems in their discipline.

4. The Dedicated Course Approach

Here are some of the advantages of a dedicated course in computer ethics:

- Having the time available to lay the groundwork for a common understanding and comparison of a range of ethical theories – deontological or Kantian ethics, utilitarianism, social contract theories, and value ethics.
- Having the time to read and discuss foundational papers in computer ethics.
- The scope of a dedicated course on ethical issues clearly extends well beyond the confines of software engineering or any other single subject area in the curriculum. Thus, the fully dedicated course on computer ethics offers the possibility of treating questions that would not necessarily arise in the context of the capstone experience in software engineering or any other content course. The virtue of this breadth of coverage consists of the possibility of discovering commonalities between situations or case studies that would not be evident when considered in isolation.
- The possibility of treating in depth several case studies which, again, reveal commonalities that underscore the critical importance of various software engineering procedures.
- The possibility of treating in depth several case studies which reveal novel or unexpected aspects of the software engineering process.
- The possibility of combining, in a natural way, ethical, social and legal aspects of the implications of new computing and information technologies. Problems with new technologies rarely come neatly wrapped in a box labeled "Ethical Dilemma: Beware!".
- The possibility of cultivating a large, diverse audience for such a course. Cross-fertilization is a good thing in this context. Individuals from outside the immediate discipline often provide complementary insights to those immediately apparent to students who are absorbed in the details of software engineering practice. The obvious trade-off here is the need to dilute some of the more technical aspects of a particular case study or scenario. (Even this difficulty can be used to advantage by having technically proficient students take the responsibility of explaining the nature and implications of a technical problem to students whose backgrounds are more general, less technical).

5. The Modular Approach

Here are some of the advantages of incorporating modules devoted to ethical issues in the software engineering curriculum itself:

- Immediate relevance of a case study or question to the course content. For example, in a Systems course, the discussion of dangers and prevention of buffer overflow could be accompanied by consideration

- The possibility of motivating a deeper exploration of a technical topic –say, the use of encryption– in connection with a particular example.
- Being assured that the students are sufficiently informed about the technical/managerial/economical issues involved in the issue under consideration.
- Incorporating consideration of ethical issues as a pro-active part of specifying, designing, building, testing, delivering, and maintaining software systems, by building such consideration into the documented approaches taught and used in the specific courses which concentrate on each of these facets of software engineering.
- When consideration of an ethical question can be directly related to a technical project/problem with which the student is actively involved, there is clearly a much higher likelihood that the student will immediately perceive the importance of the topic.

6. Remarks Concerning a Couple of Case Studies

As we have indicated, we consider the treatment of case studies to be an important component of either of the two approaches to incorporating ethical issues into the computer science curriculum. In this section we note –principally in the form of bulleted lists– some of the salient points that can be addressed in connection with two compelling case studies – the series of Therac-25 radiation accidents and the chronic failure of electronic voting technologies in recent U. S. elections. In illustration of some of our earlier assertions, the two sets of bullet points overlap in a substantial number of items. This provides the opportunity to underline the importance and ethical dimensions of questions like the initial stages of design, documentation, testing, and code re-use that students might otherwise consider minor matters.

One particular common item –which we refer to as “ecology of use”– merits further comment. The term “ecology of use” was introduced implicitly by Alvarez and colleagues [12] and explicitly in a recent paper of Fleischman [13]. The concept refers particularly to the situation in which advanced forms of technology, especially life- and safety-critical systems, are placed under the control of technically underprepared personnel. This circumstance places acute emphasis on frequently overlooked elements of engineering and software engineering design, documentation, and testing during the development of such systems. The common link between ecology of use considerations as significant factors contributing to system failure in both the Therac-25 and current electronic voting technologies was discussed by Fleischman [14].

The Therac-25 Accidents [3]

- One can argue that this case study does not actually involve software engineering because when the Therac-25 was developed (the design and

- But this is clearly one of the "index cases" for incorporation of software engineering in the curriculum and for careful attention to software engineering practice.
- The case of the Therac-25 highlights the importance of good engineering design as a precondition for successful software engineering.
- The Therac-25 accidents point to problems associated with reuse of code.
- The Therac-25 accidents underscore the need for a careful and independently designed regime of testing.
- They also reveal the folly of basing safety on serial elimination of "bugs."
- Considered in a general context, the Therac-25 accidents underscore the importance of documentation in all aspects of development including the need for attention to clear and understandable documentation for non-technical personnel who may have the responsibility for operation of a life- or safety-critical system.
- Again, in the general context, the Therac-25 accidents argue for attention to a broader systems perspective than one that simply focuses on a hardware/software combination. This is sometimes referred to as the "ecology of use."
- Finally, the case study reveals the importance of robust and transparent procedures for government approval and regulation of life- and safety-critical systems.

Electronic Voting System Technology

- Here there are numerous important references including [9], [10], [15], and [16]
- Again, good engineering design must precede good software engineering. Even a system built using the best standards of software engineering can be compromised if system components can easily be "switched out" by someone with physical access to the device.
- Having students see this as a safety-critical technology. (Referring to the Software Engineering Code of Ethics which begins by emphasizing the paramount importance of working to advance the public good.)
- The importance of careful documentation both for purposes of certification and regulation, and for use by election officials and poll workers of varying levels of technological competence.
- Lapses in implementation of state-of-the-art encryption resulting in multiple paths of attack, many of them undetectable, by malicious adversaries.
- Poor or non-existent change control protocols resulting in the possibility of virtually undetectable insertion of malicious code by a malevolent member of the development team.
- Deficient or non-existent testing programs.

- “Ecology of use” considerations relating to the fact that, for poll workers, “every election day is the first (and only) day of work for many, many people.” [17]

7. Conclusions

Finally, it is important to say that the approaches described by Terry Bynum are not mutually exclusive. Perhaps the optimal solution for the treatment of ethical questions in computing would combine a dedicated course on the subject with reinforcement (or in some cases anticipation) of relevant issues in a capstone experience or through introduction of short modules in relevant content courses. From the perspective of maintaining currency in a curriculum in which there is always pressure to expand technical content, this may seem to be an infeasibly expensive proposal. One should, however, consider the real costs to society and to the reputation of the individual academic program of producing students who are oblivious to the risks associated with computer and information technology and the potential dangers arising from poor software engineering practices.

Equally, a situation in which the curriculum includes both a dedicated course on computer ethics, appropriately taught, and several instances in which well-designed and meaningful modules are incorporated in content area courses would perhaps represent the ideal form of “hidden message” concerning the centrality of ethical concerns to the discipline.

References

1. Bynum, T.: Computer Ethics in the Computer Science Curriculum, available at http://www.southernct.edu/organizations/rccs/oldsite/resources/teaching/teaching_mono/bynum/bynum_desired_outcome.html, last accessed 15 July, 2011 (2000).
2. ABET Board of Directors: ABET Criteria for Evaluating Computing Programs, available at http://www.abet.org/Linked-Documents-UPDATE/Program-Docs/abet-cac-criteria-2011-2012.pdf, last accessed 15 July, 2011 (2010).
3. Leveson, N., and Turner, C., An Investigation of the Therac-25 Accidents, IEEE Computer, volume 26, no. 7, pp. 18-41 (1993).
4. Eisenberg, T., Gries, D., Hartmanis, J., Holcomb, D., Lynn, M. S., and Santoro, T., The Cornell Commission: On Morris and the Worm, Communications of the ACM, vol. 32, no. 6, pp. 706-709 (1989).
5. Culnan, M. J. and Smith, H. J., Lotus Marketplace: Households…Managing Information Privacy Concerns, Georgetown University School of Business, Case 192-123 (1991).
6. Etzioni, A. Privacy and Safety in Electronic Communications, chapter 4 of The Common Good, Polity Press, Cambridge, MA (2004).
7. Parnas, D. L., van Schouwen, A. J., and Kwan, S. P., Evaluation of Safety Critical Software, Communications of the ACM, vol. 33, no. 6, pp. 636-648 (1990).
8. Singer, P. W., The Ethics of Killer Applications: Why Is It So Hard to Talk about Morality When It Comes to Military Technology, Journal of Military Ethics, vol. 9, no. 4, pp. 299-312 (2010).

9. Feldman, A., Halderman, J., and Felten, E., Security Analysis of the Diebold Accu-Vote-TS Voting Machine, available at http://itpolicy.princeton.edu/voting/ts-paper.pdf, last accessed 15 July, 2011 (2006).
10. Kohno, T., Stubblefield, A., Rubin, A. and Wallace, D., Analysis of an Electronic Voting System, available at http://avirubin.com/vote.pdf, last accessed 20 July 2011 (2003).
11. Fleischman, W., The Role of Imagination in a Course on Ethical Issues in Computer Science, in Proceedings of ETHICOMP 2001: Systems of the Information Society, edited by S. Rogerson, S. Szejko, and T. Ward Bynum, Gdansk, Poland, vol. 1, pp. 171-183 (2001).
12. Alvarez, R., Atkeson, L., and Hall, T., Auditing the Election Ecosystem, working paper # 85 of the Caltech/MIT Voting Technology Project, available at http://vote.caltech.edu/drupal/ files/working_paper/wp_85_pdf_4acf9bcad1.pdf, last accessed 15 July, 2011 (2009).
13. Fleischman, W., Electronic Voting Technology, the Software Engineering Code of Ethics, and Conceptions of the Public Good, in The "Backwards, Forwards, and Sideways Changes" of ICT, Proceedings of ETHICOMP 2010, the 11th International Conference, Universitat Rovira i Virgili, Tarragona, Spain, pp. 162-169. (2010).
14. Fleischman, W., Electronic Voting Systems and the Therac-25: What Have We Learned?, in The "Backwards, Forwards, and Sideways Changes" of ICT, Proceedings of ETHICOMP 2010, the 11th International Conference, Universitat Rovira i Virgili, Tarragona, Spain, pp. 170-179 (2010).
15. Theisen, E., E-Voting Failures in the 2006 Mid-Term Elections, available at http://www.votersunite.org/info/E-VotingIn2006Mid-Term.pdf, last accessed 18 July, 2011 (2006).
16. Theisen, E., Vendors are Undermining the Structure of U.S. Elections, available at http://www.votersunite.org/info/ReclaimElections.pdf, last accessed 18 July 2011 (2008).
17. Kohno, T., Testimony of Tadayoshi Kohno before the Committee on House Administration of the U.S. House of Representatives Hearing on Electronic Voting System Security, July 7, 2004, available at http://www-cse.ucsd.edu/users/tkohno, last accessed 15 July, 2011 (2004).

Some Ethical Reflections on Relations between Human Beings and Social Robots

ANNE GERDES

University of Southern Denmark, Institute of Business Communication and Information Science,
Campus Kolding, Engstien 1, 6000 Kolding, Denmark
{Gerdes@sitkom.sdu.dk)

Abstract. *The purpose of this paper is to reflect upon ethical implications of human-robot interaction. Issues are discussed within two scenarios: (1) In focusing on robots with intelligent behavior, but without consciousness, attention is paid to obstacles for forming trustful relations. Here, it is concluded that human-robot interaction will lack the kind of commitment, which stems from the fact that life is interpersonal, implying that trust is a fundamental human condition. (2) In focusing on the possibility of developing intelligent robots with a mental life of their own, issues of our responsibility as creators of robots are discussed, as well as issues dealing with the kind of relationships we might have with such robots. Here, we are faced with a Good-like responsibility and ethical obligations towards a creature, who possible will develop a mind of its own, which might turn out to be radically different from the human mind.*

Keywords: *Human-robot interaction, trust, artificial intelligence, ethics.*

1. Introduction

With the recent upcoming of more and more human-like robots, which of course still are nothing but "stupid machines", we might expect that such surprisingly human-like geminoids in a near future will be able to simulate intelligent behavior, when acting within restricted contexts.

On an epistemological level we may still argue about the status of intelligence. But, in real life people will start to form relationships with robots, whether they are intelligent or not. The fact that they look a lot like us, combined with their growing ability to behave intelligent will cause new forms of friending and bonding in connection with human-robot interaction.

The purpose of this extended abstract is to reflect upon ethical implications of such relationships. In particular issues are discussed from the perspectives below:

(1) In focusing on the possibility of developing robots with intelligent behavior, but without consciousness, attention will be paid to obstacles for establishing trustful relations in connection with human-robot interaction.

(2) In focusing on the possibility of developing intelligent robots with a mental life of their own, issues of our responsibility as creators of robots will

be discussed, as well as issues dealing with the kind of relationships we might establish with such robots[1].

2. Perspectives towards Artificial Intelligence

In what follows, I will arrange my discussion with reference to two well-known perspectives towards strong artificial intelligence; since these two positions raise similar as well as different sorts of ethical issues regarding the character of human-robot interaction.

Within a behaviorist framework, we might be concerned with the idea of artificial intelligence from a perspective of performance. Consequently, it is not considered a meaningful project to maintain a distinction between real human intelligence and artificial intelligence, if last mentioned is indistinguishable from human intelligent behavior. The behaviorist perspective focuses on appearance, in holding a definition of intelligence in which intelligence equals intelligent behavior. This idea is encapsulated in the famous Turing test [2], which has not yet been passed by any machine.

On the other hand, the perspective of reductive materialism towards intelligence assumes that consciousness is a valid concept; we do have a mental life, but mental states can be explained for in terms of the laws of physics. Hence, from a position of reductive materialism, we might argue that we can account for intelligence, emotions and consciousness within a physicalist framework - for instance by reference to neurology and bio-chemical processes. As such, we are (nothing but) nice machines ourselves; or as phrased by Marvin Minsky: “The brain is just a computer made out of meat!” [3].

The ethical implications of these positions will be discussed from a phenomenological approach. Thus, one might ask what kind of ethical issues we are faced with if robots in the future come to *look* like us (sec. 2.1) or *be* like us (sec. 2.2)?

2.1 Ethical Issues in Relation to a Behaviorist Approach

In a behaviorist framework, what can be said to characterize the kind of relations involved in human-robot interactions? Here, we are dealing with a “look-alike setting”, in which mental states are considered unnecessary. The robot’s behavior is all that counts. Similarly, if we were to deal with our human existence from a behaviorist perspective, we should only be interested in accounting for human actions with reference to complex stimuli-response patterns. In holding a pure behaviorist point of view, presupposing a symmetric relation between human and robot, ethical issues regarding human-robot interactions might be addressed within a utilitarian framework,

[1] There are of course relevant ethical related issues regarding agency and responsibility in a legal context, which I do not touch upon. These issues are discussed in an excellent paper by Ugo Pagallo [1].

in which consequences of behavior could be accounted for ethically by measuring which behavior gave rise to the greatest amount of welfare.

On the other hand, in arguing from a phenomenological position, we might maintain an asymmetric relation between the robot and ourselves, in which case the robot only looks like us, implying that even though interaction is smooth, the robot is simply a machine good at producing certain kinds of behavior, without any intentions behind it. Within this scenario, we can explore what is ethically at stake in human relationships, and discuss whether this can be carried over to human-robot interaction. Thus, it is generally acknowledged that trust is vital for the flourishing of human life, and a precondition of any cultural ordering[2]. Our fundamental human condition is rooted in the fact that life is interpersonal; we are mutually dependent on each other. Consequently, openness, in the sense of trusting, i.e., daring to risk ourselves in coming forward to meet the other, is a definitive feature of human co-existence and inherent in all communication [4]. When we place trust in others, it involves genuine risk-taking since we surrender ourselves to the other. But, in dealing with human-robot interaction, we are not faced with having to surrender ourselves to the social robot. Even though the robot act in a human-like way, and displays emotions, there is nothing at stake, and I know that this is the case about our relationship – the robot simulates and I invest without cost. This does not necessarily imply that I will be unable to respond emotionally to the robot. However, our interaction will be risk-free and without demand.

2.2 Ethical Issues from a Position of Reductive Materialism

In a physicalist framework, matters appear differently. Here, we assume that our mental states are programmable, which will enable us to develop a robot, who do not only simulate intelligence, but has a mind of its own, probably even different from the human mind. In this scenario, the above-mentioned phenomenological objections do not count, because now robots and human beings are on equal footing. If we for a moment leave aside the fact that should it ever turn out to be the case that physicalists came up with an artificial intelligence with a mind of its own, then, in general, the phenomenological position would suffer severe problems. However, for the sake of argument, I shall maintain a phenomenological perspective in the exploration of ethical issues.

Well aware that the robot might develop a mind radically different from ours, we would still have to address design issues in the first place, such as: should we set out to create a robot capable of feeling pain? Normally, we consider it morally wrong to cause somebody pain. Yet, we might argue that lack of ability to feel pain would reduce quality of life considerable for the robot, and maybe even make the robot unable to act emphatic towards others. Nevertheless, as human beings we use different kinds of enhancers to improve our life, so why not set out to design a robot in a state of permanent

[2] See for instance Løgstrup [4], Rawls [5: 433], Fukuyama [6: 126], also within the field of economics, it is commonly known that trust is in general regarded as a "critical commodity".

happiness? One objection could be that lack of challenge in life would probably make the robot unable to fulfill its potentials. But, we would not be able to take that for granted, since we might not recognize the kind of psychological developmental path the robot would follow. As such, the robot might evolve into a being entirely different from us, and demand ethical rights, which would be incomprehensible to us. Within this context of argument, we are faced with a God-like responsibility and ethical obligations towards a creature, who possible will turn out to be beyond our imagination.

3. Concluding Remarks

This paper has dealt with ethical implications related to human-robot interaction within two scenarios of artificial intelligence, of which the first is already on its way, whereas the second scenario is probably not realizable within the nearest future.

Thus, we are approaching a time in which human-like robots (capable of intelligent behavior within restricted contexts) will be able to provide us with reliable companionship. But, here we are dealing with risk-free relations without demands. Human-robot interaction will lack the kind of basic commitment, which stems from the fact that life is interpersonal. We live in a state of surrender to each other, implying that trust is a fundamental human condition, which we cannot escape. Placing trust in others thus involves genuine risk-taking, in the form of surrendering-ourselves-to-others. This is the fundamental nerve of all interpersonal interaction, which a human-robot relationship will not have.

In the second scenario, focus is on the possibility of developing robots with a mental life of their own. Here, we find ourselves faced with a God-like responsibility in deciding what kind of design we should implement. Furthermore, we might be unable to understand the robot, since it might turn out to develop a mind radically different from the human mind, and maybe even demand ethical rights of its own.

References

1. Pagallo, U.: The Human Master with a Modern Slave? Some Remarks on Robotics, Ethics, and the Law. In: Proceedings of the 11th International ETHICOMP Conference, pp.397-410. University of Rovira i Virgili, Tarragona (2010).
2. Turing, A.: Computing Machinery and Intelligence. Mind, 59, 433-460 (1950).
3. Minsky, M.: The Society of Mind. Simon and Schuster, New York (1988).
4. Løgstrup, K.E.: The Ethical Demand. University of Notre Dame Press, Notre Dame (1997).
5. Rawls, J.: A Theory of Justice - Revised Edition. Oxford University Press, Oxford (1999).
6. Fukuyama, F.: Our Posthuman Future-Consequences of the Biotechnology Revolution. Picador, New York (2003).

Esta edición de 150 ejemplares
se terminó de imprimir en Estudiocentro,
Bolívar, Buenos Aires, Argentina,
en el mes de octubre de 2012.

RedUNCI

Its objectives are:
"Coordinate academic activities related to the improvement of the teachers' training as well as the curricular update and the use of shared resources to assist the development of both the Computer Sciences careers and the Technology careers in Argentina" and "To establish a cooperative framework for the development of Postgraduate activities in Computer Sciences and Technology, in order to optimize the assignation and use of the resources".

RedUNCI:

This Network was formally created through an Agreement signed in November 1996 by five National Universities (UNSL, UBA, UNLP, UNS y UNCPBA), during the second edition of CACIC.
Actually 45 argentine Universities are active members of this network.

Regular Activities of the RedUNCI

- Arrangement of an Annual Congress on Computer Science (CACIC) since 1995.
- Arrangement of an Annual Workshop for Researchers on Computer Science (WICC) since 1999.
- Meetings for university professors of Computer Science, for Postgraduate Dissertators and for specialists in certain areas, to promote the debate of common interest topics.
- Publication of *the Journal on Computer Science &Technology* by agreement with ISTEC (Iberoamerican Science and Technology Education Consortium).
- Annual Congress on Technology in Education and Education in Technologies (TE&ET) since 2006.
- Publication of the *Iberoamerican Journal of Technology in Education and Education in Technology*, since 2007.

ISBN Nº 978-950-34-0885-8