RedUNCI

RED DE UNIVERSIDADES CON CARRERAS EN INFORMÁTICA

Computer Science & Technology Series

XXI Argentine Congress of Computer Science Selected Papers

Guillermo Eugenio Feierherd | Patricia Mabel Pesado | Claudia Cecilia Russo
(Eds.)

edulp
Editorial de la Universidad de La Plata

Computer Science & Technology Series

XXI Argentine Congress of Computer Science
Selected Papers

RedUNCI

RED DE UNIVERSIDADES CON CARRERAS DE INFORMÁTICA

Computer Science & Technology Series

XXI Argentine Congress of Computer Science
Selected Papers

Guillermo Eugenio Feierherd
Patricia Mabel Pesado
Claudia Cecilia Russo
(eds)

Editorial
de la Universidad
de La Plata

Computer Science & Technology Series : XXI Argentine Congress of Computer Science Selected Papers / Patricia Mabel Pesado ... [et al.] ; compilado por Guillermo Feierherd ; Patricia Mabel Pesado ; Claudia Cecilia Russo. - 1a ed . - La Plata : EDULP, 2016.
390 p. ; 24 x 15 cm.

ISBN 978-987-4127-00-6

1. Informática. 2. Actas de Congresos. I. Pesado, Patricia Mabel II. Feierherd, Guillermo, comp. III. Pesado, Patricia Mabel, comp. IV. Russo, Claudia Cecilia, comp.
CDD 004

Computer Science & Technology Series
XXI Argentine Congress of Computer Science
Selected Papers

Diagramación: Erica Anabela Medina

Edulp
Editorial de la Universidad de La Plata

Editorial de la Universidad Nacional de La Plata (Edulp)
47 N.° 380 / La Plata B1900AJP / Buenos Aires, Argentina
+54 221 427 3992 / 427 4898
edulp.editorial@gmail.com
www.editorial.unlp.edu.ar

Edulp integra la Red de Editoriales Universitarias (REUN)

Primera edición, 2016
ISBN 978-987-4127-00-6
Queda hecho el depósito que marca la Ley 11.723

Impreso en Argentina

Topics

XVI Intelligent Agents and Systems Workshop
Chairs Sergio A. Gómez (UNS), Marcelo Arroyo (UNRC), Guillermo Leguizamón (UNSL)

XVI Distributed and Parallel Processing Workshop
Chairs Marcela Printista (UNSL), Javier Balladini (UNCOMA), Laura De Giusti (UNLP)

XIV Information Technology Applied to Education Workshop
Chairs Cristina Madoz (UNLP), Sonia Rueda (UNS), Alejandra Malberti (UNSJ), Claudia Russo (UNNOBA)

XIII Graphic Computation, Images and Visualization Workshop
Chairs Silvia Castro (UNS), Roberto Guerrero (UNSL), Javier Giacomantone (UNLP), Kresimir Matkovic (Austria)

XII Software Engineering Workshop
Chairs Patricia Pesado (UNLP), Elsa Estevez (UNU), Alejandra Cechich (UNCOMA), Horario Kuna (UNaM)

XII Database and Data Mining Workshop
Chairs Hugo Alfonso (UNLPam), Laura Lanzarini (UNLP), Nora Reyes (UNSL), Claudia Deco (UNR)

X Architecture, Nets and Operating Systems Workshop
Chairs Jorge Ardenghi (UNS), Hugo Padovani (UMorón), Carlos Buckle (UNPSJB), Hugo Ramón (UNNOBA)

VII Innovation in Software Systems Workshop
Chairs Marcelo Estayno (UNLZ), Pablo Fillottrani (UNS), Dante Zanarini (UNR), Jorge Finocchietto (UCAECE)

VI Signal Processing and Real-Time Systems Workshop
Chairs Oscar Bría (INVAP), Osvaldo Spositto (UNLM), Horacio Villagarcía Wanza (UNLP), Daniel Pandolfi (UNPSJB)

IV Computer Security Workshop
Chairs Javier Díaz (UNLP), Antonio Castro Lechtaler (UBA), Javier Echaiz (UNS)

IV Innovation in Computer Science Education Workshop
Chairs Cecilia Sanz (UNLP), Beatriz Depetris (UNTDF), Marcelo De Vincenzi (UAI), Uriel Cukierman (UP)

Scientific Committee

Abásolo, María José (Argentina)
Acosta, Nelson (Argentina)
Alfonso, Hugo (Argentina)
Ardenghi, Jorge (Argentina)
Arroyo, Marcelo (Argentina)
Baldasarri, Sandra (España)
Balladini, Javier (Argentina)
Bertone, Rodolfo (Argentina)
Bria, Oscar (Argentina)
Brisaboa, Nieves (España)
Buckle, Carlos (Argentina)
Cañas, Alberto (EE.UU.)
Casali, Ana (Argentina)
Castro Lechtaler, Antonio (Argentina)
Castro, Silvia (Argentina)
Cechich, Alejandra (Argentina)
Chavez, Edgar (México)
Coello Coello, Carlos (México)
Constantini, Roberto (Argentina)
Cuevas, Alfredo Simón (Cuba)
Cukierman, Uriel (Argentina)
De Giusti, Armando (Argentina)
De Giusti, Laura (Argentina)
De Vincenzi, Marcelo (Argentina)
Deco, Claudia (Argentina)
Depetris, Beatriz (Argentina)
Díaz, Javier (Argentina)
Dix, Juerguen (Alemania)
Doallo, Ramón (España)
Docampo, Domingo (España)
Echaiz, Javier (Argentina)
Esquivel, Susana (Argentina)
Estayno, Marcelo (Argentina)
Estevez, Elsa (Argentina)
Falappa, Marcelo (Argentina)
Feierherd, Guillermo (Argentina)
Fillottrani, Pablo (Argentina)
Finocchietto, Jorge (Argentina)
Fleischman, William (Argentina)
García Garino, Carlos (Argentina)
García Villalba, Javier (España)
Género, Marcela (España)
Giacomantone, Javier (Argentina)
Gómez, Sergio (Argentina)
Gröller, Eduard (Austria)
Guerrero, Roberto (Argentina)
Hecht, Pedro (Argentina)
Janowski, Tomasz (Naciones Unidas)
Kantor, Raul (Argentina)
Kuna, Horacio (Argentina)
Lanzarini, Laura (Argentina)
Leguizamón, Guillermo (Argentina)
Liporace, Julio (Argentina)
Lopez Gil, Fernando (España)
Loui, Ronald Prescott (EEUU)
Luque, Emilio (España)
Madoz, Cristina (Argentina)
Malberti, Alejandra (Argentina)
Malbran, María (Argentina)
Manresa Yee, Cristina (España)
Marín, Mauricio (Chile)
Mas Sansó, Ramón (España)
Mon, Alicia (Argentina)
Motz, Regina (Uruguay)
Naiouf, Marcelo (Argentina)
Navarro Martín, Antonio (España)
Olivas Varela, José Ángel (España)
Padovani, Hugo (Argentina)
Pandolfi, Daniel (Argentina)
Pardo, Álvaro (Uruguay)
Pesado, Patricia (Argentina)
Piattini, Mario (España)
Piccoli, María Fabiana (Argentina)
Printista, Marcela (Argentina)
Puppo, Enrico (Italia)
Ramió Aguirre, Jorge (España)
Ramón, Hugo (Argentina)
Rexachs, Dolores (España)
Reyes, Nora (Argentina)
Riesco, Daniel (Argentina)
Roig Vila, Rosabel (España)
Rossi, Gustavo (Argentina)
Rosso, Paolo (España)
Rueda, Sonia (Argentina)
Russo, Claudia (Argentina)
Sanz, Cecilia (Argentina)
Simari, Guillermo (Argentina)
Spositto, Osvaldo (Argentina)
Steinmetz, Ralf (Alemania)
Suppi, Remo (España)
Tarouco, Liane (Brasil)
Tirado, Francisco (España)
Velho, Luiz (Brasil)
Vendrell, Eduardo (España)
Vénere, Marcelo (Argentina)
Villagarcía Wanza, Horacio (Argentina)
Zanarini, Dante (Argentina)

Organizing Commitee

Universidad Nacional del Noroeste de la Provincia de Buenos Aires
Buenos Aires – Argentina
Escuela de Tecnología (UNNOBA)

President
Russo, Claudia

Coordinator
Sarobe, Mónica

Members
Ahmad, Tamara
Anolles, Natalia
Di Cicco, Carlos
Lencina, Paula
Mangold, Leonardo
Pérez, Daniela
Picco, María Linda Trinidad
Rodriguez, Sabina
Serrano, Eliana
Spada, Oscar

Preface

CACIC Congress

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

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

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

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

CACIC 2015 in Junín

CACIC'15 was the 21thCongress in the CACIC series. It was organized by the School of Technology at the UNNOBA (North-West of Buenos Aires National University – www.unnoba.edu.ar) in Junín, Buenos Aires.

The Congress included 13 Workshops with 131 accepted papers, 4 Conferences, 2 invited tutorials, different meetings related with Computer Science Education (Professors, PhD students, Curricula) and an International School with 6 courses. (http://cacic2015.unnoba.edu.ar/

escuela-de-informatica/cursos/).

CACIC 2015 was organized following the traditional Congress format, with 13 Workshops covering a diversity of dimensions of Computer Science Research. Each topic was supervised by a committee of 3-5 chairs of different Universities.

The call for papers attracted a total of 202 submissions. An average of 2.5 review reports werecollected for each paper, for a grand total of 495 review reports that involved about 191 different reviewers.

A total of 131 full papers, involving 404 authors and 75 Universities, were accepted and 24 of them were selected for this book.

Acknowledgments

CACIC 2015 was made possible due to the support of many individuals and organizations. The School of Technology at the UNNOBA,RedUNCI, the Secretary of University Policies, the National ministry of Science and Technology, CIC and CONICET were the main institutional sponsors.

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

Ing. Armando De Giusti
Congresos, Publicaciones y Difusión RedUNCI

Table of Contents

XVI

XVI Intelligent Agents and Systems Workshop

Representing Traffic Congestions on Moving Objects Trajectories

MARIANO KOHAN AND JUAN M. ALE

Facultad de Ingeniería, Universidad de Buenos Aires
marianokohan@gmail.com, ale@acm.org

Abstract. *The discovery of moving objects trajectory patterns representing a high traffic density have been covered on different works using diverse approaches. These models are useful for the areas of transportation planning, traffic monitoring and advertising on public roads. Besides of the important utility, these type of patterns usually do not specify a difference between a high traffic and a traffic congestion. In this work, we propose a model for the discovery of high traffic flow patterns and traffic congestions, represented in the same pattern. Also, as a complement, we present a model that discovers alternative paths to the severe traffic on these patterns. These proposed patterns could help to improve traffic allowing the identification of problems and possible alternatives.*

Keywords: *Moving objects, trajectories, road network, traffic flow, traffic congestion*

1. Introduction

In the last years, there has been a high presence of works related to the data mining of the trajectory data generated by moving objects ([16]). From these works, there has been a lot of attention to the discovery of different type of traffic flow patterns. These patterns can be discovered from trajectories moving inside a road network like [7], [6] and [9] or with a free movement ([5], [11], [4]). Related to the first case, the concept of traffic congestions as a limitation of the road network is considered on more recent papers ([14], [1]). These works are useful on different areas like transport planning, traffic monitoring, carpooling, store locations and advertising on public roads.
In this work, we propose a model for the discovery of high traffic flow patterns in relation to traffic congestions. This relation is displayed according to the shared traffic, like presented in the hot routes model ([7]). Also we present, as a complement, a model that relates these patterns with alternative paths, according to a low traffic level and useful location inside the road network. These models present an increased utility allowing to be applied on additional cases, like the identification of paths inside the road network that require changes (for transport planning), and the redirection of the drivers

that contribute to the different congestion areas into alternative paths (for traffic monitoring).
The rest of the paper is organized as follows. Next section comments the works related to this paper. Section 3 describes selected concepts considered from these works. The proposed models are introduced on Section 4. And Sections 5 and 6 gives the definitions and algorithms for each model. Finally, Section 7 concludes this paper with information about the next steps.

2. Related Work

Works about the discovery of traffic flow patterns are related to this paper. The model of hot routes ([7]) is used as the main inspiration for this work because of its balance between the aggregate information about the moving objects and theirs specific behavior (represented in the common traffic in a sequence of edges). Li and others on [7] comments about some alternative methods to discover traffic flow patterns. First, just the aggregate behavior of individuals can be considered connecting only edges in the graph with high traffic. [6] uses this method and complements it with the discovery of the temporal evolution of the patterns. Also, in [9] the model is oriented to the traffic analysis through edges clustering. Another method is to discover moving clusters formed by moving objects, where [5] and [8] are some examples. The third method is about the clustering of trajectories. In this group we can consider some patterns like hot motion paths ([11]), the discovery of the Most Popular Route (MPR) between two locations ([4]) and the distributed parallel clustering method MCR-ACA ([15]). Besides of representing these patterns a high traffic inside the road network, they usually do not consider the cases of traffic jams, where the traffic density is close to the network capacity.

Another group of papers are related to the analysis of traffic congestions. To discover this type of patterns, it is possible to consider the road network characteristics or the moving objects data. In the first case, we have works about representative patterns for the network segments ([2]), usage patterns of road networks ([14], [13]), and the visualization of traffic jams using a GIS map service ([12]). For the second case we can consider diverse patterns like slowly flocks ([10]), the transitions within regions ([17]) and Non-Recurrent Congestion events ([1]). Also, there is the work [3] that considers both types of data in the discovered patterns. The main difference with the current work is that this second group of papers only discover patterns related to traffic congestions, but without relation to patterns with lower levels of traffic.

3. Background Models

The concepts presented in some of the related works are used as a starting point for the models proposed in this paper. In [7], the hot routes patterns are presented. These are traffic flow patterns inside a road network. The road network is represented by a graph *G(V,E),* where *E* is the set of edges representing a unit of road segment, and *V* the set of vertices representing a street intersection. Also, *T* is the set of trajectories, with each element composed of an ID (*tid*) and a sequence of edges traveled through: *(tid, <e_1, ..., e_k>)*, where $e_i \in E$. The hot routes are built up with a sequence of edges, near each other but not necessarily adjacent, that "...share a high amount of traffic between them." ([7]). The distance between the edges is based on the number of edges inside the road network graph, according to the metric *ForwardNumHops*, which represents the minimum number of edges between the end vertex of two edges. Using this metric, the *Eps-neighborhood* ($N_{Eps}(r)$) of an edge *r* defines the set of closed edges. The shared traffic considers the same trajectory identification made by each moving object. This model is used as a base for the development of patterns considering traffic congestions.

The work [10] is about the detection of potential traffic jams with slowly flock patterns. In this case, the velocity of each moving object is considered on the discovery of the flock patterns. This idea of using the velocity to identify a traffic jam is applied on the first proposed model.

The discovery of representative patterns for the network segments is proposed on [2]. On this work, the network segments are characterized according to presented network features like length, direction, capacity and density. This last concept of *segment density D*(s)* allows to identify alternative segments for an edge according to a bounding rectangle (*BR*) covering the segment, and the direction for the edges inside this *BR*. For the second model proposed, this concept is used to discover alternative paths to the traffic congestions.

[6] presents a more general pattern, called dense routes. These patterns are discovered using only the number of objects on each edge of the road network, and adjacent edges are linked if the difference on the number of moving objects is bellow a maximum threshold. A similar idea for the algorithm described on the second model is considered for the discovery of the alternative paths.

4. Models Description

We consider the model of hot routes ([7]) to be most appropriate to discover patterns with heavy traffic in a city road network, because it represents a balance between an aggregate analysis and the behaviors of the individuals. But besides of represent a high density of moving objects in a road network,

it does not consider some characteristics of this road network causing the appearance of traffic jams:

- capacity: it is associated with the edges in the road network and represents the maximum number of vehicles that are allowed to circulate into a road segment.
- velocity and time: related to the feature of capacity are the concepts of velocity and time. When the density of objects in a road segment is close to its capacity, the velocity of the moving objects starts to be decreased and the travel time is extended.
- when the velocity and time starts to be affected in an edge, the drivers from the vehicles might choose to continue its path on another alternative segments. This concept of alternative segments is designed as *Density (D*)* elsewhere.

In this paper, we propose two new models for the discovery of trajectory patterns considering these features:

1. the concept of velocity is considered in order to discover hot routes with jam sections: sections with a density close to its capacity. We call these patterns *jam routes*. This model is described on the next section.
2. the existence of paths that could be used as alternative to the traffic in a *jam route*, because of its location and low density values. These patterns are called *cold routes*. Section 6 presents this model.

5. Discovery of Jam Routes

The *jam route* pattern could be defined as a hot route with one or more subpaths identified as traffic jam. So, it is a path in a road network with heavy traffic (shared by the same objects inside a sliding window) and with one or more sectors having a traffic level close to its capacity.
In order to identify these subpaths the velocity is used. So, each trajectory is composed by its ID (*tid*) and a sequence of pairs representing each edge traveled with its respective mean velocity: $(tid, <(e_1, v_1), (e_2, v_2), \ldots, (e_k, v_k)>)$, where $e_i \in E$ and v_i is the *mean velocity* on e_i.
We consider the use of the velocity to identify traffic congestions is better than compare the density with the road capacity for two reasons:

- the data about the velocity for each moving object on each edge of the trajectory is easier to obtain than the capacity of each edge of the road network
- considering the road segments are part of a network, there also additional factors that could lead to congestions ([14])

The first concept to present is *speed*. It complements the *traffic* definition from [7] to consider the velocity in each edge.

Definition 1 (speed). The *speed(r)* for a given edge *r* is the mean of velocities v_i of the edge *r*.

In order to identify the edges affected by the conditions of a traffic jam the concept *directly traffic jam-reachable* is used.

Definition 2 (directly traffic jam-reachable). An edge *s* is *directly traffic jam-reachable* from an edge *r* with respect to parameters *Eps*, *MinTraffic* and *JamSpeed* if

1. $s \in N_{Eps}(r)$
2. $|traffic(r) \cap traffic(s)| \geq MinTraffic$
3. $speed(s) \leq JamSpeed$ or $speed(r) \leq JamSpeed$

This definition extends the concept of *directly traffic density-reachable* ([7]), to identify traffic jams but maintaining the condition of shared traffic between the edges.

Definition 3 (route traffic jam-reachable). An edge *s* is *route traffic jam-reachable* from an edge *r* with respect to parameters *Eps*, *MinTraffic* and *JamSpeed* if

1. *there is a chain of edges* $r_1, r_2, \ldots, r_n$ *with* $r_1=r$ *and* $r_n=s$, *where* r_i *is directly traffic jam-reachable from* r_{i-1} *or* r_i *is just directly traffic density-reachable from* r_{i-1}
2. *for every Eps consecutive edges in the chain,* $|traffic(r_i) \cap traffic(r_{i+1}) \cap \ldots \cap traffic(r_{i+Eps})| \geq MinTraffic$

This definition augments the concept of *route traffic density-reachable* ([7]), allowing to propose a path that relates sections with heavy traffic and sections with traffic jams.

This concept is the base for the discovery of the *jam routes*.

5.1 Algorithm

The algorithm to discover the *jam routes* presents a structure of breadth-first search on the road network graph.
It starts out the discovery from the *hot routes starts* ([7]), verifying if the *speed* on each of these edges is below the *JamSpeed* threshold. In this case, the edge is marked as a *jam*. Next, these *hot routes starts* are extended recursively to form the *jam routes*. The extension is from the last edge, finding the edges inside the N_{Eps} that satisfy the definitions of *directly traffic jam-reachable* or *directly traffic density-reachable.* Then, on each of these possible split edge from the route, the definition of *route traffic jam-reachable* is evaluated (specifically the second condition). If this definition is validated, a new *jam route* is created with the new edge. And, if the added edge is *directly traffic jam-reachable*, it is marked as a *jam*.

The algorithm is called *JamFlowScan* and its pseudo-code is presented as follows:

```
Input: Road network G, object trajectory data T,
Eps, MinTraffic, JamSpeed
Output: Jam routes R

1: Initialize R to {}
2: Let H be the set of hot route starts in G
according to T
3: for every hot route start h in H do
4:   r = new Jam Route initialized to <h> /*mark
edge as "jam" if speed(h) <= JamSpeed */
5:   Add Extend_Jam_Routes(r) to R
6: end for
7: Return R


Procedure Extend_Jam_Routes(jam route r)
1:  Let p be the last edge in r
2:  Let Q be the set of directly traffic jam-
reachable neighbors of p U the set of directly
traffic density-reachable neighbors of p
3:  if Q is non-empty then
4:    Initialize JR to {}
5:    for every split in Q do
6:      if route traffic jam-reachable condition
is satisfied then
7:        Let r' be a copy of r
8:        Append splits edges to r'
9:        if directly traffic jam-reachable
condition is satisfied then
10:         mark split edge as "jam"
11:       end if
12:       Add Extend_Jam_Routes(r') to JR
13:     end if
14:   end for
15:   return JR
16: else
17:   Return {r}
18: end if
```

To verify the definitions used in the algorithm, the *traffic* set and *speed* for every edge is required. So, the object trajectory data *T* can be converted into table structure that relates each edge with the *tid* of the trajectories that belongs to, and the mean velocity on all the trajectories. The building of this table has linear complexity with respect to the trajectories data.

The jam routes are discovered applying the definitions from the model and identifying the traffic jams on the respective cases:

- initially after the identification of the *hot routes starts* (step 4)
- on the extension of the *jam route* for each split, following the identification of an edge as *route traffic jam-reachable* (steps 9-11 from `Extend_Jam_Routes`).

So, if a traffic congestion is found during the route building, it will be properly identified on the results. Also, the order used to extend the routes adds efficiency to the search but does not omit edges. Therefore, the set of *jam routes* discovered is complete and correct.

6. Discovering Cold Routes

The *cold route* pattern is a path in a road network with low traffic (so it does not affect the network capacity) and with a location inside the road network that allows to be chosen as an alternative path to the traffic present in a *jam route*.
To allow the identification of the alternative routes the concept *BR-neighborhood* $N_{BR}(s)$ is used. It is the same concept *segment density* $D^*(s)$ from [2] (but using a name following the conventions applied to this work): considers the vicinity area of a segment (with a bounding rectangle) and the direction. So, each edge $e \in E$ from the road network graph $G(V,E)$ will be associated with a label representing its direction.

The first concept to present is *cold traffic*. It allows to identify edges with low traffic and that could be considered alternatives to edges with traffic jam (d*irectly traffic jam-reachable*).

Definition 4 (cold traffic). An edge s is considered *cold traffic* with respect to parameters *BR* and *MaxTraffic* if:

1. $|traffic(s)| \leq MaxTraffic$
2. $s \in N_{BR}(s)$ *of directly traffic jam-reachable edge*

Additionally the concept of *directly cold traffic reachable* is presented.

Definition 5 (directly cold traffic reachable). An edge s is considered *directly cold traffic reachable* from an edge r with respect to parameter *MaxTraffic* if:

1. *s is adjacent to r: start(s) = end(r) or end(s) = start(r)*
2. $|traffic(r)| \leq MaxTraffic$
3. $|traffic(s)| \leq MaxTraffic$

Both concepts are related on the definition of *route cold traffic reachable.*

Definition 6 (route cold traffic reachable). An edge *s* is considered *route cold traffic reachable* from an edge *r* with respect to parameters *BR* and *MaxTraffic* if there is a chain of edges $r_1, r_2, \ldots, r_n$ *with* $r_1=r$ *and* $r_n=s$, where:

1. *each* r_i *is directly cold traffic reachable from* r_{i-1}
2. *there exists almost one edge* r_i *that is cold traffic*

The concept of *route cold traffic reachable* allows the discovery of the *cold route* patterns.

6.1 Algorithm

Considering that *cold routes* are formed by edges with low traffic, it is better to discover them using a simple aggregate method.
The proposed algorithm starts the discovery process from the *jam routes* discovered by *JamFlowScan,* finding the *cold traffic* edges according to the N_{BR} of the *directly traffic jam-reachable* edges. Next, these edges are extended to both sides, evaluating the definition *directly cold traffic reachable* into the adjacent edges. With the two conditions of *route cold traffic reachable* satisfied, the new edge is added to the route, considering possible splits (representing different alternative paths).

The algorithm is called *ColdScan* and it pseudo-code is presented as follows:

```
Input: Road network G, object trajectory data T,
MaxTraffic, BR, JamRoutes (from JamFlowScan)
Output: Cold routes CR

1:  Initialize CR to {}
2:  Let CS the set of cold traffic edges in G
according to T and discovered Jam Routes
3:  for every cold traffic edge cs in CS do
4:    cr = new Cold Route initialized to <cs>
5:    Add Extend_Cold_Route_Forward(cs) to CRf
6:    for every route (extended forward from cs)
crf in CRf do
7:      Add Extend_Cold_Route_Backward(crf) to CR
8:    end for
9:  end for
10: return CR

Procedure Extend_Cold_Route_Forward (cold route
cr)
1:  Let p be the last edge in cr
2:  Let S be the set of directly cold traffic
reachable edges from p with end(p) = start(s)
3:  if S is non-empty then
4:    Initialize CR to {}
```

```
5:    for every edge s in S
6:      Let cr' be a copy of cr
7:      Append edge s to the end of cr'
8:      Add Extend_Cold_Route_Forward(cr') to CR
9:    end for
10:   return CR
11: else
12:   return {cr}
13: end if

Procedure Extend_Cold_Route_Backward (cold route
cr)
1:  Let p be the first edge in cr
2:  Let S be the set of directly cold traffic
reachable edges from p with start(p) = end(s)
3:  if S is non-empty then
4:    Initialize CR to {}
5:    for every edge s in S
6:      Let cr' be a copy of cr
7:      Append edge s to the beginning of cr'
8:      Add Extend_Cold_Route_Backward (cr') to CR
9:    end for
10:   return CR
11: else
12:   return {cr}
13: end if
```

The algorithm requires, to verify the definitions used, the *traffic* set for every edge. So, in this case a similar table structure built from the trajectories can be utilized, with a linear complexity with respect to the trajectory data.
ColdScan discovery process applies the definition of *route cold traffic reachable*, considering all the *jam routes* from *JamFlowScan*. Also, these routes are extended to both possible sides. So, the discovered set of *cold routes* is complete.

7. Conclusion and Future Work

In this paper we presented two models for the discovery of traffic flow patterns. The hot routes model for the discovery of high traffic routes is considered as a starting point for the development of patterns representing traffic jams and its alternative paths.
First, in the *jam routes* model the velocity of the moving objects is added in order to identify traffic jam sectors inside the patterns. The relation between these congestions and the high traffic density is according to the shared traffic in common. Next, starting with a vicinity concept the *cold routes* are presented as a path that could be used as an alternative to the traffic in the

jam routes. These patterns are identified according to a low level of traffic and comparing its location in the road network graph with respect to the congestions in the *jam routes*. The algorithms for the discovery of the proposed models are presented in order to clarify further details.
This is a work in progress. The next step is the implementation of the presented models, in order to compare the discovered patterns with the obtained using some of the related models. This will allow to confirm the utility of these models.

References

1. Berk Anbaroglu, Benjamin Heydecker, and Tao Cheng. Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. Transportation Research Part C: Emerging Technologies, 48(0):47 – 65, 2014.
2. Farnoush Banaei-Kashani, Cyrus Shahabi, and Bei Pan. Discovering patterns in traffic sensor data. In: Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on GeoStreaming, IWGS '11, pp. 10–16, New York, USA, 2011. ACM.
3. Kyoung Soo Bok, He Li, Jong Tae Lim, and Jae Soo Yoo. Discovering congested routes using vehicle trajectories in road networks. Advances in Multimedia, 2014.
4. Zaiben Chen, Heng Tao Shen, and Xiaofang Zhou. Discovering popular routes from trajectories. In: Data Engineering (ICDE), 2011 IEEE 27th International Conference on, pp 900–911, 2011.
5. Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On discovering moving clusters in spatiotemporal data. In: Claudia Bauzer Medeiros, Max J. Egenhofer, and Elisa Bertino, editors, Advances in Spatial and Temporal Databases, volume 3633 of Lecture Notes in Computer Science, pp. 364–381. Springer Berlin Heidelberg, 2005.
6. Ahmed Kharrat, Karine Zeitouni, Iulian Sandu-Popa, and Sami Faiz. Characterizing traffic density and its evolution through moving object trajectories. Fifth International IEEE Conference on Signal-Image Technologies and Internet-Based Systems, pp. 257–263, 2009.
7. Xiaolei Li, Jiawei Han, Jae-Gil Lee, and Hector Gonzalez. Traffic density-based discovery of hot routes in road networks. In: SSTD'07 - Proceedings of the 10th international conference on Advances in spatial and temporal databases, pp. 441–459, Berlin, Heidelberg, 2007. Springer-Verlag.
8. Wenting Liu, Zhijian Wang, and Jun Feng. Continuous clustering of moving objects in spatial networks. In: Ignac Lovrek, Robert J. Howlett, and Lakhmi C. Jain, editors, Knowledge-Based Intelligent Information and Engineering Systems, volume 5178 of Lecture Notes in Computer Science, pp. 543–550. Springer Berlin Heidelberg, 2008.
9. Irene Ntoutsi, Nikos Mitsou, and Gerasimos Marketos. Traffic mining in a road-network: How does the traffic flow? International Journal of Business Intelligence and Data Mining, 3:82–98(17), 2008.
10. Rebecca Ong, Fabio Pinelli, Roberto Trasarti, Mirco Nanni, Chiara Renso, Salvatore Rinzivillo, and Fosca Giannotti. Traffic jams detection using flock mining. In: Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and

Michalis Vazirgiannis, editors, Machine Learning and Knowledge Discovery in Databases, volume 6913 of Lecture Notes in Computer Science, pp. 650–653. Springer Berlin Heidelberg, 2011.
11. Dimitris Sacharidis, Kostas Patroumpas, Manolis Terrovitis, Verena Kantere, Michalis Potamias, Kyriakos Mouratidis, and Timos Sellis. On-line discovery of hot motion paths. In: Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology, EDBT '08, pp. 392–403, New York, NY, USA, 2008. ACM.
12. Anna Izabel J. Tostes, Fátima de L. P. Duarte-Figueiredo, Renato Assunção, Juliana Salles, and Antonio A. F. Loureiro. From data to knowledge: City-wide traffic flows analysis and prediction using Bing maps. In: Proceedings of the 2Nd ACM SIGKDD International Workshop on Urban Computing, UrbComp '13, pp. 12:1–12:8, New York, NY, USA, 2013. ACM.
13. Junjie Wang, Dong Wei, Kun M. He, Hang Gong, and Pu Wang. Encapsulating urban traffic rhythms into road networks. Scientific Reports, 4, 2014.
14. Pu Wang, Timothy Hunter, Alexandre M. Bayen, Katja Schechtner, and Marta C. Gonzalez. Understanding road usage patterns in urban areas. Scientific Reports, 2, 2012.
15. Jie Yang, Xiaoping Li, Dandan Wang, and Jia Wang. A group mining method for big data on distributed vehicle trajectories in wan. International Journal of Distributed Sensor Networks, 2014.
16. Yu Zheng. Trajectory data mining: An overview. ACM Transaction on Intelligent Systems and Technology, 2015.
17. Yu Zheng, Yanchi Liu, Jing Yuan, and Xing Xie. Urban computing with taxicabs. In: Ubicomp 2011. ACM, 2011.

Immune Algorithm for Solving the Smooth Economic Dispatch Problem

VICTORIA S. ARAGÓN AND SUSANA C. ESQUIVEL

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Universidad Nacional de San Luis
Ejército de los Andes 950 - (5700) San Luis, ARGENTINA
CONICET

***Abstract**. In this paper, an algorithm inspired on the T-Cell model of the immune system is presented, it is used to solve Economic Dispatch Problems with smooth objective function. The proposed approach is called IA_EDP_S, which stands for* Immune Algorithm for Economic Dispatch Problem for smooth objective function*, and it uses as differentiation process a redistribution power operator. The proposed approach is validated using five problems taken from the specialized literature. Our results are compared with respect to those obtained by several other approaches.*

***Keywords**: Artificial immune systems, economic dispatch problem, metaheuristics.*

1. Introduction

The objective of Economic Dispatch Problem (EDP) is to minimize the total generation cost of a power system while satisfying several constraints associated to the system, such as load demands, ramp rate limits, maximum and minimum limits, and prohibited operating zones. The objective function type (smooth or non smooth) and the constraints which are considered in the problem will determine how hard is to solve the problem.

Over the last years, several methods have been proposed to solve the EDP. They can be divided in three main groups: classical, based on artificial intelligence (AI) and hybrid methods. Classical methods have been proposed to solve EDP, but they suffer from some limitations (for instance, the objective functions and the constraints must be differentiable). On the other hand, modern heuristic algorithms have proved to be able to deal with nonlinear optimization problems, e.g., EDPs. Surveys about these techniques can be found in [14] and [2].

In this paper, we propose an algorithm to solve EDPs which is inspired on the T cells from the immune system. Once the algorithm has found a feasible solution, it applies a redistribution power operator in order to improve the original solution with the aim of keeping such a solution feasible at a low computational cost.

The remainder of this paper is organized as follows. Section 2 defines the economic dispatch problem. In Section 3, we describe our proposed algorithm. In Section 4, we present the test problems used to validate our proposed approach and parameters settings. In Section 5, we present our results and we discuss and compare them with respect to other approaches. Finally, in Section 6, we present our conclusions and some possible paths for future research.

2. Problem Formulation

The schedule has to minimize the total production cost and involves the satisfaction of both equality and inequality constraints.

2.1 Objective Function

Minimize

$$TC = \sum_{i=1}^{N} F_i(P_i)$$

where TC is the fuel cost, N is the number of generating units in the system, P_i is the power of i^{th} unit (in MW) and F_i is the total fuel cost for the i^{th} unit (in \$/h).

An EDP with a smooth cost function represents the simplest cost function. It can be expressed as a single quadratic function: $F_i(P_i) = a_i P_i^2 + b_i P_i + c_i$, where a_i , b_i and c_i are the fuel consumption cost coefficients of the i^{th} unit.

2.2 Constraints

1. Power Balance Constraint: the power generated has to be equal to the power demand required. It is defined as: $\sum_{i=1}^{N} P_i = P_D$
2. Operating Limit Constraints: thermal units have physical limits about the minimum and maximum power that can generate: $P_{mini} \leq P_i \leq P_{maxi}$, where P_{mini} and P_{maxi} are the minimum and maximum power output of the i^{th} unit, respectively.
3. Power Balance with Transmission Loss: some power systems include the transmission network loss, thus Power Balance Constraint equation is replaced by: $\sum_{i=1}^{N} P_i = P_D + P_L$. The P_L value is calculated with a function of unit power outputs that uses a loss coefficients matrix B, a vector B0 and a value B00: $\sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B0_i P_i + B00$.
4. Ramp Rate Limits: they restrict the operating range of all on-line units. Such limits indicate how quickly the unit's output can be changed: $\max(P_{minj}, P_j^0 - DR_j) \leq P_j \leq \min(P_{maxj}, P_j^0 + UR_j)$, where P_j^0

is the previous output power of the j^{th} unit (in MW) and, UR_j and DR_j are the up-ramp and down-ramp limits of the j^{th} unit (in MW/h), respectively.

5. Prohibited Operating Zones: they restrict the operation of the units due to steam valve operation conditions or to vibrations in the shaft bearing:

$$\begin{cases} P_{mini} \leq P_i \leq P_{i,1}^l \\ P_{i,j-1}^u \leq P_i \leq P_{i,j}^l \; j = 2,3,..,n_j \\ P_{i,nj}^u \leq P_i \leq P_{maxi} \end{cases}$$

where n_j is the number of prohibited zones of the i^{th} unit, $P_{i,j}^{\ l}$ and $P_{i,j}^{\ u}$ are the lower and upper bounds of the j^{th} prohibited zone.

3. Our Proposed Algorithm

In this paper, an adaptive immune system model based on the immune responses mediated by the T cells is presented. These cells present special receptors on their surface called T cell receptors (TCR: are responsible for recognizing antigens bound to major histocompatibility complex (MHC) molecules.) [6].

The model considers some processes that T cells suffer. These are proliferation (to clone a cell) and differentiation (to change the clones so that they acquire specialized functional properties); this is the so-called activation process.

IA_EDP_S (Immune Algorithm for Economic Dispatch Problem with Smooth Objective Function) is an adaptation of an algorithm inspired on the activation process [2], which is proposed to solve the EDP with Smooth Objective Function. IA_EDP_S operates on one population which is composed of a set of T cells.

For each cell, the following information is kept:

1. TCR: it identifies the decision variables of the problem (TCR $\in \Re^N$). Each thermal unit is represented by one decision variable.
2. objective: objective function value for TCR, (TC(TCR)).
3. prolif: it is the number of clones that will be assigned to the cell, it is N for all problems.
4. differ: it is the number of decision variables that will be changed when the differentiation process takes place (if applicable).
5. TP: it is the power generated by TCR ($\sum_{i=1}^{N} TCR_i$).
6. P_L: it is the transmission loss for TCR (if the problem does not consider transmission loss, then P_L =0).
7. ECV: it is the equality constraint violation for TCR (|TP - P_D -P_L|). If ECV > 0, then the power generated is bigger than the demanded

power, and if ECV < 0 then the power generated is lower than the required power.

8. ICS: it is the inequality constraints sum, $\sum_{i=1}^{nj} poz(TCR_i, i)$

$$poz(p,i) = \begin{cases} min(p - PZ_{lli}, PZ_{uli} - p) if\ p \in [PZ_{lli}, PZ_{uli}] \\ 0\ otherwise \end{cases}$$

where n_j is the number of prohibited operating zones and [PZ_{lli}, PZ_{uli}] is the prohibited range for the i^{th} thermal unit.

9. feasible: it indicates if the cell is feasible or not. A cell is considered as feasible if: 1) ECV=0 for problems without transmission network loss and $0 \leq ECV < \varepsilon$ for problems with transmission loss. This means that if a solution generates less than the demanded power, then it is considered as infeasible (ECV<0) and 2) ICS=0 for problems which consider prohibited operating zones.

Differentiation for feasible cells - Redistribution Process

The idea is to take a value (called d) from one unit (say i) and assign it to another unit (variable). i^{th} unit is modified according to: cell.TCRi = cell.TCRi - d, where d=U(prob*D, D), D= min(cell.TCR_i - ll_i, U(min, max))), U(w_1, w_2) refers to a random number with a uniform distribution in the range (w_1, w_2), max is the maximum power that can be generated by the other units according to their current outputs (i.e. max= $max^N_{n=1 \wedge n \neq i}$($ul_n$ - cell.TCR_n), min is the minimum power that can be generated by the other units according to their current outputs (i.e. min= $min^N_{n=1 \wedge n \neq i}$($ul_n$ - cell.TCR_n).

d was designed to avoid: 1) that the i^{th} unit falls below its lower limit and 2) to take from the i^{th} unit more power of what other units can generate. Next, d has to increase the power of another unit (say k). In a random way k is selected considering cell.TCR_k +d ≤ ul_k.

The main difference between IA_EDP_S and the algorithm proposed in [2] arises in the number of variables that are modified. This version just changes i and k while version [2] changes i and one o more variables. Note this operator only preserve the feasibility of solutions by taking into account the power balance constraints.

Differentiation for infeasible cells

For infeasible cells, the number of decision variables to be changed is determined by their differentiation level. This level is calculated as U(1, N) . Each variable to be changed is chosen in a random way and it is modified according to: cell.TCR'$_i$= cell.TCR_i ≠ m, where cell.TCR_i and cell.TCR_i' are the original and the mutated decision variables, respectively. m = U(0, 1) *|cell.ECV + cell.ICS|.

In a random way, it decides if m will be added or subtracted to cell.TCR_i . If the procedure cannot find a TCR'_i in the allowable range, then a random number with a uniform distribution is assigned to it (cell.TCR'_i= U(cell.TCR_i, ul_i) if m should be added or cell.TCR'_i= U(ll_i, cell.TCR_i), otherwise).

The algorithm works in the following way (see Algorithm 1). First, the TCRs are randomly initialized within the limits of the units (Step 1). Then, ECV and ICS are calculated for each cell (Step 2).
Only if a cell is feasible, its objective function value is calculated (Step 3). Next, while a predetermined number of objective function evaluations had not been reached or if after 50 iterations the best value does not improve (Steps 4-6) the cells are proliferated and differentiated considering if they are feasible or infeasible. Finally, statistics are calculated (Step 8).

Algorithm 1 IA_EDP_S Algorithm

```
1: Initialize_Population();
2: Evaluate_Constraints();
3: Evaluate_Objective_Function();
4: while(A predetermined number of evaluations has not been reached or Not
   improve) do
5:     Proliferation_Population();
6:     Differentiation_Population();
7: end while
8: Statistics();
```

4. Validation

IA_EDP_S performance was validated with five test problems, SYS_3U, SYS_6U, SYS_15U, SYS_18U and SYS_20U (see [2] for full description). Table 1 provides their most relevant characteristics and the maximum number of function evaluations. IA_EDP_S was implemented in Java (version 1.6.0_24) and the experiments were performed in an Intel Q9550 Quad Core processor running at 2.83GHz and with 4GB DDR3 1333Mz in RAM.

Table 1. Test Problems Characteristics

Problem	Thermal Units	P_L	Prohibited Zones	PD (MW)	Evaluations
SYS_3U	3	No	No	850.0	1000
SYS_6U	6	Yes	Yes	1263.0	3000
SYS_15U	15	Yes	Yes	2630.0	20000
SYS_18U	18	No	No	365.0	40000
SYS_20U	20	Yes	No	2500.0	20000

The required parameters by IA_EDP_S are: size of population, number of objective function evaluations, and probability for redistribution operator. To analyze the effect of the first and third parameters on IA_EDP_S's behavior, we tested it with different parameters settings. Some preliminary experiments were performed to discard some values for the population size parameter. Hence, the selected parameter levels were: a) Population size (C) has four levels: 1, 5, 10 and 20 cells and b) Probability has three levels: 0.01, 0.1 and 0.5.
Thus, we have 12 parameters settings for five problems. They are identified as C < size > -Pr < Prob >, where C and Pr indicate the population size and the probability, respectively. For each problem, 100 independent runs were performed.
The box plot method was selected to visualize the distribution of the objective function values for each power system. This allowed us to determine the robustness of our proposed algorithm with respect to its parameters. Figures 1 to 3 show in the x-axis the parameter combinations and the y-axis indicates the objective function values for each problem. We can see that better results are reached with the lowest probability value and the highest population size. So, C=5and Pr=0.01 were used to compare the results got by IA_EDP_S with those produced by other approaches.
Considering the lowest number of objective function evaluations used by the other approaches (see [2]) we take as maximum number of function evaluations, 1000, 40000, 3000, 20000 and 20000 for SYS_3U, SYS_18U, SYS_6U, SYS_15U and SYS_20U, respectively. Also, we set \epsilon =0.1 for those problems which consider loss transmission (e.d. SYS_6U, SYS_15U and SYS_20U).

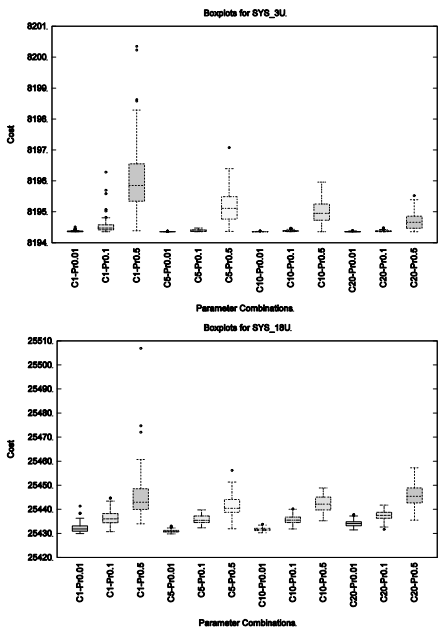


Fig. 1.*Box plots for the test problems with the best parameters combination*

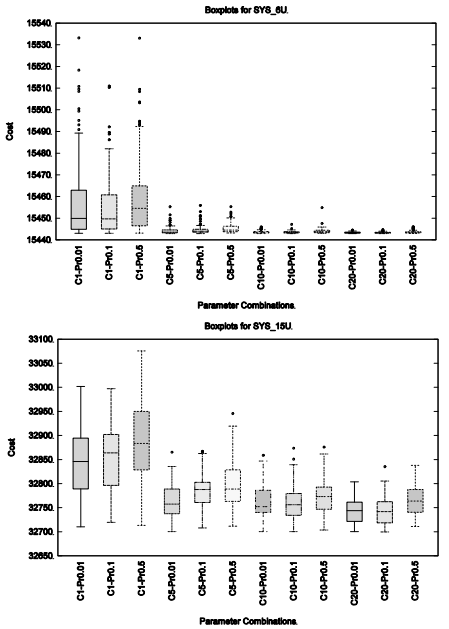


Fig. 2.*Box plots for the test problems with the best parameters combination*

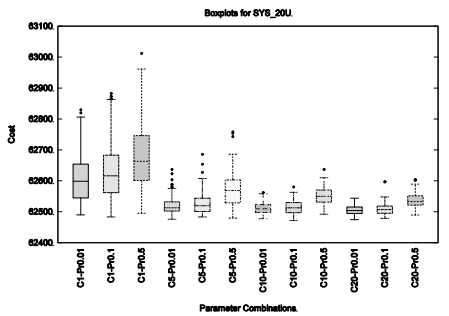


Fig. 3.*Box plots for the test problem with the best parameters combination*

5. Comparison of Results and Discussion

Table 2 shows: the best, worst, mean, median, standard deviation and number of function evaluations obtained by IA_EDP_S. Only four decimal digits are shown due to space restrictions. For all the test problems, our proposed IA_EDP_S found feasible solutions in all the runs performed.

Problems which do not consider transmission loss, rate ramp limits or prohibited zones, i.e., SYS_3U and SYS_18U, do not seem to be a challenge for IA_EDP_S. The standard deviations obtained by IA_EDP_S are lower

than 1. Additionally, the problem dimensionality does not seem to affect the performance of our proposed approach either.
For problems which consider transmission loss, rate ramp limits and prohibited zones, SYS_6U_a and SYS_15U, the standard deviations increase with the problem dimensionality.
For the only problem which considers transmission loss but not rate ramp limits or prohibited zones, SYS_20U, the standard deviation is lower than SYS_15U's standard deviation.

Problem	Best	Worst	Mean	Median	Std.	Ev.
SYS_3U	8194.3561	8194.3847	8194.3597	8194.3584	0.004	987.16
SYS_18U	25429.8005	25433.0655	25430.9312	25430.8415	0.614	35103.15
SYS_6U	15442.8962	15455.2466	15444.3082	15443.6071	1.877	1490.62
SYS_15U	32700.2971	32865.2657	32763.5364	32758.1897	35.765	18321.5
SYS_20U	62476.1186	62636.5875	62522.3703	62513.2753	30.371	8151.36

Eleven methods are compared with respect to IA_EDP_S. They are cited in Table 3 comparacion. The running time of each algorithm is affected by both the hardware environment and the software environment. That is the reason why the main comparison criterion that we adopted for assessing efficiency was the number of objective function evaluations performed by each approach. For having a fair comparison of the running times of all the algorithms considered in our study, they should all be run in the same software and hardware environment (something that was not possible in our case, since we do not have the source code of several of them). Clearly, in our case, the emphasis is to identify which approach requires the lowest number of objective function evaluations to find solutions of a certain acceptable quality.

However, the running times are also compared in an indirect manner, to give at least a rough idea of the complexities of the different algorithms considered in our comparative study. For all test problems IA_EDP_S found the best cost in the lowest time. Except for SYS_3U, where fast-PSO just required 0.01 second and IA_EDP_S spent 0.18 seconds to find the best solution.

Table 3 summarizes the performance IA_EDP_S with respect to that of the other methods. As shown in Table 3, considering the best cost found, IA_EDP_S outperforms all other approaches. Considering running times, IA_EDP_S requires less than one second to find solutions with an acceptable quality for SYS_3U and SYS_6U. It requires less than 1.4 second for SYS_15U and SYS_18U. And it requires less than 2.1 second for SYS_20U.

We could not found an approach that report feasible solutions for SYS_20U, so IA_EDP_S obtained the best results.

Table 3. Comparison of results. The best values are shown in **boldface**.

Problem/ Algorithm	Best	Worst	Mean	Std.	Time(s)	Ev.
SYS_3U						
IEP[10]	**8194.35**	-	-	-	-	-
MPSO[9]	**8194.35**	-	-	-	-	-
IPSO[11]	**8194.35**	-	-	-	0.42	3000
ModPSO[12]	8194.40	-	-	-	-	-
fast-CPSO[4]	**8194.35**	-	-	-	0.01	3000
IA_EDP_S	**8194.35**	**8194.37**	**8194.35**	0.004	0.18	987
SYS_18U						
ICA-PSO[13]	25430.16	25462.34	25440.89	-	18.585	40000
IA_EDP_S	**25429.80**	**25433.06**	**25430.93**	0.614	1.168	35103
SYS_6U						
IHS[7]	15444.30	-	15449.86	4.531	-	100000
BBO[3]	15443.09	**15443.09**	**15443.09**	-	-	50000
ICA-PSO[13]	15443.24	15444.33	15443.97	-	-	20000
IA_EDP_S	**15442.89**	15455.24	15444.30	1.877	0.828	1490
SYS_15U						
CCPSO[8]	32704.45	**32704.45**	**32704.45**	0.0	16.2	30000
MDE[1]	32704.9	32711.5	32708.1	-	-	160000
SA-PSO [5]	32708.00	32789.00	32732.00	18.025	12.79	20000
IA_EDP_S	**32700.29**	32865.26	32763.53	35.76	1.328	18321
SYS_20U						
IA_EDP_S	**62476.11**	62636.58	62522.37	30.371	2.016	8151

6. Conclusions and Future Work

This paper presented an adaptation of an algorithm inspired on the T-Cell model of the immune system, called IA_EDP_S, which was used to solve economic dispatch problems. IA_EDP_S is able to handle the five types of constraints that are involved in an economic dispatch problem: power balance constraint with and without transmission loss, operating limit constraints, ramp rate limit constraint and prohibited operating zones.

At the beginning, the search performed by IA_EDP_S is based on a simple differentiation operator which takes an infeasible solution and modifies some of its decision variables by taking into account their constraint violation. Once the algorithm finds a feasible solution, a redistribution power operator is applied. This operator modifies two decision variables at a time, it decreases the power in one unit, and it selects other unit to generate the power that has been taken.

The approach was validated with five test problems having different characteristics and comparisons were provided with respect to some approaches that have been reported in the specialized literature. Our results indicated that dimensionality increases standard deviations when the same types of constraints are considered but prohibited zones have more impact on the performance than dimensionality. Our proposed approach produced competitive results in all cases, being able to outperform the other approaches while performing
a lower number of objective function evaluations than the other approaches.

As part of our future work, we are interested in redesigning the redistribution operator in order to maintain the solutions' feasibility when a problem involves prohibited operating zones.

References

1. N. Amjady and H. Sharifzadeh. Solution of non-convex economic dispatch problem considering valve loading effect by a new modified differential evolution algorithm. International Journal of Electrical Power and Energy Systems, 32(8):893–903, 2010.
2. V.S. Aragon, S.C. Esquivel, and C.A. Coello Coello. An immune algorithm with power redistribution for solving economic dispatch problems. Information Sciences, 295(0):609 – 632, 2015.
3. A. Bhattacharya and P.K. Chattopadhyay. Biogeography-based optimization for different economic load dispatch problems. IEEE Transactions on Power Systems, 25(2):1064–1077, 2010.
4. Leticia Cecilia Cagnina, Susana Cecilia Esquivel, and Carlos A. Coello Coello. A fast particle swarm algorithm for solving smooth and non-smooth economic dispatch problems. Engineering Optimization, 43(5):485–505, 2011.
5. Cheng-Chien Kuo. A novel coding scheme for practical economic dispatch by modified particle swarm approach. IEEE Transactions on Power Systems, 23(4):1825–1835, 2008.
6. Leandro Nunes de Castro and Jonathan Timmis. Artificial Immune Systems: A New Computational Intelligence Approach. Springer-Verlag, New York, 2002.
7. V.R. Pandi, K.B. Panigrahi, M.K. Mallick, A. Abraham, and S. Das. Improved harmony search for economic power dispatch. In Hybrid Intelligent Systems, 2009. HIS ’09. Ninth International Conference on, volume 3, pages 403–408, 2009.
8. J.-B. Park, Y.-W. Jeong, J.-R. Shin, and K.Y. Lee. An improved particle swarm optimization for nonconvex economic dispatch problems. IEEE Transactions on Power Systems, 25(1):156–166, 2010.

9. Jong-Bae Park, Ki-Song Lee, Joong-Rin Shin, and K.Y. Lee. A particle swarm optimization for economic dispatch with nonsmooth cost functions. Power Systems, IEEE Transactions on, 20(1):34–42, 2005.
10. Y. M. Park, J. R. Won, and J. B. Park. A new approach to economic load dispatch based on improved evolutionary programming. Eng. Intell. Syst. Elect. Eng. Commun., 6(2):103–110, June 1998.
11. P. Sriyanyong, Y.H. Song, and P.J. Turner. Particle swarm optimisation for operational planning: Unit commitment and economic dispatch. In KeshavP. Dahal, KayChen Tan, and PeterI. Cowling, editors, Evolutionary Scheduling, volume 49 of Studies in Computational Intelligence, pages 313–347. Springer Berlin Heidelberg, 2007.
12. S. Siva Subramani and P. Raja Rajeswari. A modified particle swarm optimization for economic dispatch problems with non-smooth cost functions. International Journal of Soft Computing, 3(4):326–332, 2008.
13. J.G. Vlachogiannis and K.Y. Lee. Economic load dispatch a comparative study on heuristic optimization techniques with an improved coordinated aggregation-based pso. IEEE Transactions on Power Systems, 24(2):991–1001, 2009.
14. Ling Wang and Ling po Li. An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems. International Journal of Electrical Power & Energy systems, 44(1):832 – 843, 2013.

Prediction of Income Criminal Cases using Linear Genetic Programming

ALBERTO DAVID GARCETE RODRÍGUEZ[1] AND BENJAMIN BARÁN[2]

[1] adgr_x@hotmail.com – [2] bbaran@cba.com.py
[1] East National University - [2] National University of Asuncion
Campus Km. 8 Acaray, Street of the East National University and Republic of Paraguay
Ciudad del Este – Paraguay

Abstract. *This paper proposes a prediction methodology to estimate the income of criminal cases using* Linear Genetic Programming - *LGP. The study was based on monthly collected data for seven years (2007 to 2013), for the seven Guarantee Criminal Courts of Ciudad del Este - Paraguay. The verification of the proposed method was made by comparing the implemented LGP with well-known statistical alternatives, such as linear regression, moving average, exponential smoothing and exponential smoothing with trend, predicting values of time series, to compare average error of each prediction methodology. It was considered two error metrics: (1) root mean square error and (2) mean absolute error. Experimental results demonstrate the superiority of the implemented LGP over the statistical methods for the prediction of criminal income cases.*

Keywords: *Genetic Programming - GP, Linear Genetic Programming - LGP, criminal income, prediction.*

1. Introduction

The increasing common crime mobilizes the entire judicial structure, affecting the court numbers which may need to grow year by year [1, 2].
Investigating this increment, may be beneficial to improve the judicial system and to analyze the factors that can incise on these crimes. Consequently, this paper proposes for the first time the prediction of the numbers of crimes causes, using an evolutionary algorithm which later is compared to statistical methods such as linear regression, moving average, exponential smoothing and exponential smoothing with trend, to give a prediction tool to facilitate planning of resource at the Paraguayan Supreme Court. The work was based on monthly historical data, corresponding to a seven year period, between January/2007 and December/2013.

2. Previous Work

Azamathulla, Guven and Demir presented in [3] an LGP (*Linear Genetic Proigramming*) as an alternative tool for the prediction of submerged depth for pipes. The LGP model proposed a comparison to an adaptive *neuro-fuzzy* inference systems (ANFIS). Experimental results demonstrate that the proposed LGP obtain better results than the ANFIS system even on traditional regression equations v.

In [4], Guven and Kisi studied daily weather data to estimate evaporation using linear genetic programming. The estimates done using an LGP model was compared to the results obtained from other traditional prediction techniques, concluding that the LPG approach is appropriate for modeling evaporation processes.

In [5], Shavandi and Ramyani used LGP to predict the global solar radiation. Experimental results indicate that the LGP models give accurate estimates of the global solar radiation and significantly outperform other traditional models.

Alavi, Gandomi and Mollahasani presented in [6] a hybrid algorithm of search which combines the LGP algorithm with *Simulated Annealing* (SA), what they called LGP/SA. This algorithm demonstrated suitable performance characteristics for a stabilized soil. Nevertheless, the models based on *Linear Genetic Programming* found more accurately values than the hybrid models based in LGP/SA.

3. Genetic Programming

Evolutionary Computation (EC) proposes a set of computational models to automatically evolve the solution of a given problem, generation after generation, for solving certain problems [7] based on the *survival of the fittest* principle. In that context, *Genetic Programming* (GP) is a technique of automatic learning. It is used to optimize a population of programs utilizing an aptitude function called *fitness* [11]. The fitness is responsible for the quality evaluation of each candidate program.

The potential of genetic programming resides in its ability to automatically develop computer programs, once defined the objective function of the searched program and the metric of be used to compare different alternatives (or solutions). Its biological principle is the same than that one for a genetic algorithm, the principle of evolution of species proposed by Darwin [12]. The first proposals of GP used syntax trees applied to a programming language to represent individuals [10]. Later, GP implemented simpler versions, like the *Linear Genetic Programming - LGP* [3], treated in the next section.

4. Linear Genetic Programming

In the LGP methodology the programs are represented by a sequence of instructions of an imperative programming language, for example:

r[0] = r[2] + r[6]

where **r[i]** represents the register **i**. In this example, register **r[0]** is the destination register while registers **r[2]** and **r[6]** are the register where the operands reside for the operation to be performed, a sum in this case. For this work, instructions are composed as follows:

- A destination register where the result is stored.
- One or two operand registers where operand values are stored.
- An operation (a set of permitted operations are defined at the beginning).

Thus, a typical representation of a program (candidate or solution) obtained by an algorithm LGP could be as follows:

r[6] = r[3] + r[1]
r[1] = sin r[15]
r[0] = r[1] / r[2]

Algorithm 4.1 was designed to attend the needs of the specific prediction problem of criminal cases, studied in this work. For that purpose, this work developed an adaptation of the algorithm presented by Guven and Kisi in [4].

Algorithm 4.1: *Linear Genetic Programm implemented*

1: Randomly initialize a population of programs and calculate their fitness.
2: **Selection**. From the existing evolutionary population, select individuals (or solutions) *using a Roulette_Selector procedure based of the fitnessof each individual* [9].
3: Evolve programs (individuals) using one or more variation operators of following set:
Reproduction: copy an individual (program) without change.
Crossing: exchanges substructures (genes) between two programs (or individuals).
Mutation: two types of mutations are used: (1) micro mutation to modify a single element (register or operator) of a mutated instruction, and (2) macro mutation to insert or delete a complete instruction.

4: Build a new population with the varied programs obtained in step 3 and calculate the fitness *of each new program*.
5: If the stop criterion is not met, return to step 2.
6: Stop: the program with the best *fitness* in the last population represents the found solution.

The set of registers and the set of operations together are components from the LGP methodology, trying to build a computer program that solves a given problem. The number of registers is defined at the beginning [8]. 22 registers were chosen for this work. Register r[0] is the one that usually stores the output from an LGP algorithm and therefore, the same convention is used in this work. Registers r[10] to r[15] are input registers, containing information entered from the database, typically the last values of the studied time series. The chosen set of functions for this work includes 4 arithmetic functions, 3 exponential and 2 trigonometry functions.
In this work, the aptitude [9] of each program is estimated by weighting the error that occurs when predicting a future value using each given program proposed by the LGP algorithm (thus, a program aptitude is nothing else than an individual fitness). Consequently, the best individual (or program) will be the one with the smallest prediction error.
The most commonly used error function is the root mean square error [8], given in (1). Then, we can define an error "e" given in (1) as the average of the squared errors for "N" sample values, where "x_i" represents the real value from the sample "i" of the data while "p_i" represent the forecast (prediction or estimation) for the discreet time "i".

$$e = \frac{1}{N}\sum_{i=1}^{N}(x_i - p_i)^2 \qquad (1)$$

It should also be remembered that genetic operations are applied to a population evolving generation to generation. Consequently, parameters that control the execution of the algorithm LGP, should be defined. In this work we used the following parameters: population size (150 individuals), maximum length of an individual (200 instructions), probability of using crossover and mutation (20% and 30% respectively); among other parameters [3], whose details are not presented for lack of space.

5. Classic Prediction Methods

This work uses statistical methods for time serie prediction, to compare experimental results to the implemented LGP. These classical statistical methods are briefly presented below.

5.1 Linear regression

In a linear regression, it is assumed that there exists a linear relation between an independent variable **x** and another dependent variable **y**, which may be represented as a straight line [13].

$$\mathbf{y = mx + b} \qquad (2)$$

The values of the parameters "b" and "m" are chosen to minimize the average squared error given in (1).

5.2 Moving Average

It is a simple arithmetic average from the "N" recent observations [13], given by:

$$\mathbf{S_t = \frac{1}{N}\sum_{i=t}^{t-N+1} D_i} \qquad (3)$$

where "$\mathbf{D_i}$" represent the data available at discrete time "i". The main dtaw-back of this model is the loss of old data of the original series in the estimation [13].

5.3 Exponential smoothing

It is one of the most used methods [14]. It uses a correction mechanism that adjusts forecasts, where the weights decreases exponentially with time and it is given by:

$$\mathbf{F_t = \alpha\, D_{t-1} + (1 - \alpha) F_{t-1}} \qquad (4)$$

where α represent a smoothing constant which determines the relative weighting in the observation. Alternatively, if there is a clear trend in time series data, this method can be improved by estimating the trend, in what is known as *Exponential Smoothing with Trend,* presented below.

5.4 Exponential Smoothing with Trend

In addition to using the parameter α above presented, this method requires a second parameter β that soften a given trend. In most applications [14] the following relation is used:

$$\beta \leq \alpha \quad (5)$$

The equations usually used with the exponential smoothing method with trend are [14]:

$$\mathbf{S_t} = \alpha \mathbf{D_{t-1}} + (\mathbf{1} - \alpha)(\mathbf{S_{t-1}} + \mathbf{T_{t-1}}) \quad (6)$$

$$\mathbf{T_t} = \beta(\mathbf{S_t} - \mathbf{S_{t-1}}) + (\mathbf{1} - \beta)\mathbf{T_{t-1}} \quad (7)$$

$$\mathbf{F_t} = \mathbf{S_t} + \mathbf{T_t} \quad (8)$$

6. Experimental tests and Main Results

Experimental tests were performed comparing the proposed LGP method to the selected statistical methods (linear regression, moving average, exponential smoothing and exponential smoothing with trend) [13] and [14] using data available for the period between January/2007 to December/2013, extracted from the website of the Supreme Corte of Justice [1, 2]. The tests were conducted using available data in the following way:

Training / test: 72 months.
Validation / forecast: 12 months.
Total available data: 84 months (January/2007 to December/2013)

Comparisons of the implemented methodologies were performed considering two performance metrics: (1) the mean square error already presented and (2) the mean absolute error, given by:

$$e = \frac{1}{N}\sum_{i=1}^{N} |x_i - p_i| \quad (9)$$

Procedure used with the LGP method:

- The population of individuals is of constant size, where parents are replaced by the decedents at each generation.

- The algorithm performs the selection of individuals using the roulette method [9]. Evolutionary operators (selection, crossover and mutation), are applied to the selected individuals to generate the next generation of individuals (to update the evolutionary population).

- Given that the training of the LGP to find the best individuals (programs) was done using the first 72 months of available data, the evalu-

ation of the resulting prediction programs was made using the last 12 months, using data not known during training.

Rules obtained by the LGP method

Given the randomness of the proposed LGP method to obtain different solutions at each run, it may generate different solutions (programs) at each run. Experimental results proved that most solutions (programs) were adequate predictors. In fact, Table 1 shows three solutions (prediction programs) obtained by the implemented LGP algorithm which they were compared to the statistical methods presented above to probe the advantages of using Linear Genetic Programming to predict Income of Criminal Cases at the judiciary courts of Ciudad del Este - Paraguay.

Table 1. Table with 3 individuals (solutions or programs) proposed by the implemented LGP, showing some differences that may exist between two solutions of different runs

Individual A	*Individual B*	*Individual C*
r[9] = r[15] - r[21]	r[6] = |r[17]|	r[4] = r[14] + r[6]
r[0] = r [7] + r[0]	r[6] = r [18] ^ r[3]	r[9] = r[6] * r[4]
r[1] = r[15] / r[19]	r[3] = sen[12]	r[5] = r[7] + r[17]
r[7] = r[1] + r[1]	r[6] = r[16] * r[11]	r[3] = r[13] + r[10]
r[0] = |r[9]|	r[6] = r[17] - r[21]	r[3] = sen[15]
r[8] = r[21] + r[0]	r[2] = r[5] - r[7]	r[5] = r[7] + r[17]
r[8] = r[20] + r[0]	r[6] = cosin[9]	r[4] = |r[10]|
r[0] = |r[9]|	r[0] = |r[9]|	r[0] = r [13] ^ r[5]
r[0] = r [8] + r[1]	r[3] = sen[18]	r[8] = r[8] / r[7]
r[0] = r [7] + r[0]	r[3] = ln[14]	r[0] = r[7] * r[2]
r[8] = r[21] + r[0]	r[4] = ln[6]	r[0] = r[1] - r[0]
r[0] = r [7] + r[0]	r[9] = sin[13]	r[4] = cosin[5]
r[8] = r[21] + r[0]	r[9] = ln[16]	r[6] = r[13] - r[1]
r[0] = r [7] + r[0]	r[5] = r[10] * r[7]	r[7] = r [16] + r[16]
r[8] = r[21] + r[0]	r[5] = |r[16]|	r[3] = cosin[16]
r[0] = r [8] + r[1]	r[0] = cosin[11]	r[4] = r[4] * r[6]
r[7] = r[1] + r[1]	r[5] = cosin[19]	r[0] = r[7] * r[2]
r[0] = r [7] + r[0]	r[4] = |r[15]|	r[4] = r [14] + r[6]
r[8] = r[21] + r[0]	r[4] = |r[19]|	r[5] = r [18] + r[16]
r[0] = r[19] * r[8]	r[8] = r[14] * r[20]	r[7] = r [16] + r[16]
	r[2] = sin[21]	r[0] = r [19] + r[12]
	r[7] = r[3] / r[5]	
	r[4] = sen[14]	
	r[2] = r[7] / r[4]	
	r[7] = ln[3]	
	r[0] = r[2] - r[5]	
	r[3] = sin[0]	
	r[7] = r[8] + r[9]	
	r[0] = r[7] * r[20]	

Table 2 presents the prediction errors for 12 months of year 2013 using the mean square error. In the last three rows we can see: the total error, the mean error and the Ranking, verifying the excellent performance of individuals

LGP(A) and LGP (B) calculated by the implemented LGP. Note that LGP(C) is the worst solution, showing that not always the LGP calculates a good solution, due to its inherent randomness. Therefore, the use of a validation period (12 months in this work) is always recommended, even if a program seems especially good during training.

Table 2. Mean square error for each evaluated solution during the validation period, sorted in such a way that the best result is on the left and the worst result on the right (see row: *Ranking*)

Month	LGP(A)	LGP(B)	Exp. S.	Exp. S. Tend.	Regression	Moving A.	LGP(C)
January	14361,47	35628,71	67667,37	52617,80	59413,70	81035,11	47552,66
February	3764,75	6809,11	6573,60	316,19	28828,59	3927,11	15892,57
March	19565,15	9093,21	24134,67	47484,53	1100,22	35344,00	76139,73
April	10631,37	408,67	3506,06	7544,58	3,50	16129,00	27867,05
May	1002,25	7740,42	9307,00	6562,36	18201,12	7511,11	28583,22
June	4022,77	15009,00	7548,77	3351,48	27209,20	16469,44	26917,56
July	33926,80	65216,27	28944,29	17540,12	80651,84	32881,78	22519,72
August	17413,24	34560,86	2216,35	98,19	52915,29	1111,11	4364,68
September	32860,20	15342,73	82801,86	116682,34	7212,46	98177,78	135375,64
October	461,95	6413,29	747,92	658,03	13482,55	841,00	12754,15
November	4321,74	71,97	5415,65	6725,72	737,39	3844,00	12783,86
December	2118,48	8396,44	4204,92	4523,74	18822,61	13378,78	486,90
TOTAL Error:	144450,19	204690,68	243068,46	264105,08	308578,47	310650,22	411237,72
Mean Error:	12037,52	17057,56	20255,70	22008,76	25714,87	25887,52	34269,81
Ranking	1°	2°	3°	4°	5°	6°	7°

Using the mean absolute error metric to compare the seven studied alternatives in the same year 2013, Table 3 was obtained, where again LGP is the method which finds the best estimates. Once more, LGP (A) is the best predictor closely followed by LGP (B), consistently in the Ranking. Therefore, this work can recommend the use of LGP to make predictions on the number of court cases in the Courts of Ciudad del Este - Paraguay.

Finally, it is noted that according the metrics comparison (root mean square error or mean absolute error), the Exponential Smoothing and Exponential Smoothing with Trend methods, may be considered as the third and the fourth in the Ranking, respectively. Something similar happens with the Moving Average and Linear Regression, which can be fifth or sixth, but without any variation in the top of the ranking that consistently correspond to individuals LGP(A) and LGP(B). Likewise, the last in both rankings corresponds to the individual LGP(C), confirming that a LGP not always gives good predictions; therefore, it is advisable to perform several runs of the method before choosing the final program (solution) that will actually be used for future estimation.

Table 3. Mean Absolute Error for each evaluated solution during the validation period, sorted in such a way that the best result is on the left and the worst on the right (see row: *Ranking*)

Month	LGP(A)	LGP(B)	Exp. S. Tend.	Exp.S.	Moving A.	Regression	LGP(C)
January	119,84	188,76	229,39	260,13	284,67	243,75	218,07
February	61,36	82,52	17,78	81,08	62,67	169,79	126,07
March	139,88	95,36	217,91	155,35	188,00	33,17	275,93
April	103,11	20,22	86,86	59,21	127,00	1,87	166,93
May	31,66	87,98	81,01	96,47	86,67	134,91	169,07
June	63,43	122,51	57,89	86,88	128,33	164,95	164,07
July	184,19	255,37	132,44	170,13	181,33	283,99	150,07
August	131,96	185,91	9,91	47,08	33,33	230,03	66,07
September	181,27	123,87	341,59	287,75	313,33	84,93	367,93
October	21,49	80,08	25,65	27,35	29,00	116,11	112,93
November	65,74	8,48	82,01	73,59	62,00	27,15	113,07
December	46,03	91,63	67,26	64,85	115,67	137,20	22,07
TOTAL Error:	1149,95	1342,68	1349,69	1409,88	1612,00	1627,86	1952,26
Mean Error:	95,83	111,89	112,47	117,49	134,33	135,66	162,69
Ranking	1°	2°	3°	4°	5°	6°	7°

7. Contributions to the Judiciary

A relevant point determined by this study was the growing trend of criminal income (see Figure 1). The same algorithm LGP can in fact be used to analyze this worrying situation for the safety of the studied area.

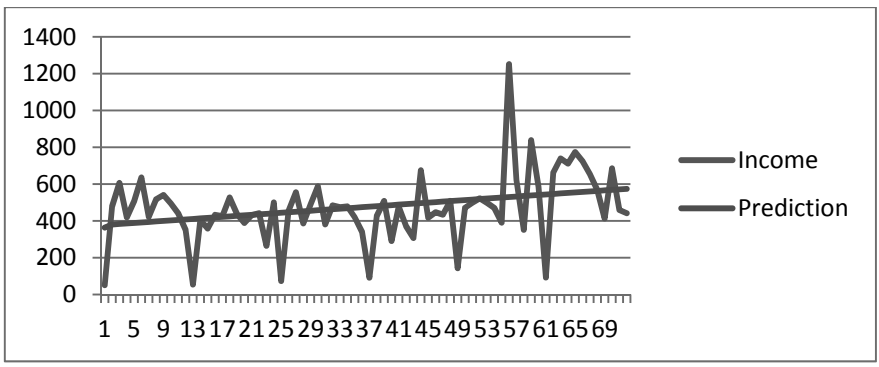


Figure 1. *Trend of income growth*

The number of cases studied during the 7 year was 38.305. The annual average of these causes, reveals that each court office, would be working with a total of approximately 5472 cases. The study data demonstrate more than

6000 criminal cases happening in recent years and the trend is clearly growing, which may imply that in short time, the authorities should enable more criminal courts in the region, which entails considerable expense, considering the high cost generated by each court. Consequently, this type of study brings relevant information to the Supreme Court with, to expand resources with planning necessary, minimizing setbacks and delays today in sight. The proposed tool will alert authorities of requirements and tendencies before reaching an extreme situation, projecting in time a suitable solution, for example by investing more in prevention.

8. Conclusions and Future Work

LGP algorithm was executed several times and in most executions it generates very good results when compared to other popular statistical methods. Observing these results and comparing to other popular models, it was noted that proposed solutions achieved adequate prediction values, outperforming classical statistical alternatives.

The implemented LGP algorithm proves to be suitable to be applied as a standard methodology for predicting income of criminal cases of Guarantees Courts of Ciudad del Este considering the efficiency of obtained results. Clearly, its application can be extended to other regions of the country. The success of the implemented LGP is based in its ability to predict good approximation values with pretty small errors, what explains the first two places in both presented rankings of alternatives (see Tables 2 and 3). Notably, as the error metric considered changes, the Ranking changes from Table 2 to Table 3, but anyhow the proposed LGP manages to stay in the top of both rankings, thanks to its ability to predict linear and nonlinear time series, while most statistical methods have difficulties with prediction of nonlinear values [8].

Finally, observing the increase of crimes and the growth in the human resource structure of the Supreme Court [1, 2], this kind of work could help the planning and the justice system and even can help for a complete reengineering of the judicial system, allocating the right amount of infrastructure and human resources where they are most needed, according to a reasonable forecast. Consequently, applying these techniques, we expect improvements in the justice area, where adjusting the processes can provide better services and more justice.

Possible future work to improve the presented work may include:

1. Analysis of optimal initialization parameters and input values to be used by the LGP algorithm.
2. Comparison to the LGP algorithm with non-linear prediction models, such as neural networks.

3. Complex problems with multiple outputs.
4. LGP algorithm implementation with multi-objective approaches.
5. Consider specific applications to other courts and different areas of the public sector where there is a need to know reasonable estimates of different items to improve its planning based on increasingly better predictions.

References

1. http://www.csj.gov.py/, Site of Supreme Corte, 2014.
2. http://www.pj.gov.py/, Site Judiciary, 2014.
3. Azamathulla, H. M., Guven, A., & Demir, Y. K. Linear genetic programming to scour below submerged pipeline. Ocean Engineering, 38(8), 995-1000, 2011.
4. Guven, A., & Kişi, Ö. Daily pan evaporation modeling using linear genetic programming technique. Irrigation science, 29(2), 135-145, 2011.
5. Shavandi, H., & Ramyani, S. S. A linear genetic programming approach for the prediction of solar global radiation. Neural Computing and Applications, 23(3-4), 1197-1204, 2013.
6. Alavi, A. H., Gandomi, A. H., & Mollahasani, A. A Genetic Programming-Based Approach for the Performance Characteristics Assessment of Stabilized Soil. In Variants of Evolutionary Algorithms for Real-World Applications (pp. 343-376). Springer Berlin Heidelberg. 2012.
7. Koza, J. R. Human-competitive results produced by genetic programming. Genetic Programming and Evolvable Machines, 11(3-4), 251-284, 2010.
8. R. Sánchez, J. Martinez y B. Barán. "Economic Time-Series Forecasting Using Linear Genetic Programming", Computational Intelligence in Economics and Finance - CIEF. 11th Joint Conference on In-formation Sciences JCIS' Kylin Villa – Shenzhen, China, 2008.
9. Koza, J. R. Herarchical Genetic Algorithms Operating on Populations of Computer Programs. N. S. Sridharan, Editor, Proceedings Of 11th International Joint Conference On Artificial Intelligence, San Mateo, Morgan Kaufmann, California, 1989.
10. Koza, J. R. Introduction to genetic programming tutorial: from the basics to human-competitive results. Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, 2137-2262, Julio 2010. ACM.
11. Poli, R., & Koza, J. Genetic Programming, 143-185. Springer US, 2014.
12. Darwin, Ch. The Origin of Species By Means Of Natural Selection or the Preservation of Favored Races in the Struggle for Life. Random House, New York, 1993.
13. Hernández, R. Statistics, Second Edition, Madrid Spain, 1997.
14. Triola, Mario F., Elementary Statistics, Seventh Edition, México 2000.

A Desiderata for Modeling and Reasoning with Social Knowledge

FABIO R. GALLO[1], NATALIA ABAD SANTOS[2], GERARDO I. SIMARI[1] AND MARCELO A. FALAPPA[1]

[1]Dept. of Comp. Sci. and Eng., Universidad Nacional del Sur (UNS), Argentina
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
Institute for Computer Science and Engineering UNS-CONICET, Argentina
[2] Dept. of Mathematics, Universidad Nacional del Sur (UNS), Argentina
{fabio.gallo, gis, mfalappa}@cs.uns.edu.ar, nasantos@uns.edu.ar

Abstract. *The ongoing surge in the amount of users that engage in online activities, as well as the expansion of the type of such activities, has recently made it clear that there is a widening gap between current knowledge representation and reasoning tools and the type of knowledge that is essentially up for grabs for whoever is willing (and has the tools) to extract it from social media sites. In this position paper, we propose the concept of Social Knowledge Base (SKB, for short) as an extension of traditional KBs with representation and reasoning capabilities that arise from the singular combination of characteristics that define this setting: (i) ontological knowledge, (ii) user preferences, (iii) reasoning under uncertainty, (iv) stream reasoning, and (v) representation of complex social networks. We propose a list of desirable properties—a desiderata—that next-generation KR formalisms for modeling and reasoning with SKBs should enjoy. The treatment is non-technical, focusing on building a road map of the formidable list of problems that must be solved in this complex setting rather than proposing a concrete solution, which would be impossible in a single article. We conclude by proposing some first steps towards achieving this goal.*

Keywords: *Social Web, Complex Networks, Reasoning Under Uncertainty, Preferences, Ontology Languages*

1. Introduction and Motivation

Recent times have seen a veritable explosion in the amount and kind of information that is available to anyone with a connection to the Internet. This explosion has its roots in the so-called World Wide Web [1], which revolutionized internet applications by allowing users to link resources with one another and easily organize the material that they wish to publish. The second revolution came with the advent of Web applications in which users produced their own material, such as in blogs or forums where users share

information ranging from plain text to photos, videos, and audio—this "new version" is often referred to as *Web 2.0* to highlight that a step was taken since the implementation of the original idea. Finally, in the last few years, the Web has once again taken an evolutionary step: in its current form, which many refer to as Social Web (or the *Web 3.0*), users and the relationships among them are the central participants. Another revolutionary aspect that appeared in Web 3.0 is that data is now also produced automatically by computers; examples of this are data output by the host of sensors now carried by most smart phones, or by the smart homes that are slowly becoming more and more present.

Unfortunately, research in Knowledge Representation and Reasoning formalisms has lagged behind this rapid evolution in how data is created and disseminated. The goal of this position paper is thus to explore a desiderata—a list of desirable characteristics—for the development of what we will call *social knowledge bases* (SKBs, for short). The idea behind this line of research is to derive a framework and methodology akin to Ontology Based Data Access (OBDA) [23] that is specialized for the unique social aspects discussed above. Our work is influenced by recent proposals in the complex networks literature [21, 20], which also establishes a set of criteria that is desirable for modeling cascades, a specific phenomenon that—as we will see—also plays an important role in our setting. Our desiderata are therefore inspired in this work, but necessarily go above and beyond their scope given the greater generality of the problems that need to be solved.

We now describe two settings that we will use as running examples to motivate our discussion. The first setting is an online matchmaking service.

Example 1 (Friendship/Dating site). Consider a web site where people register and complete a profile with the objective of meeting new people—the goal might be to establish a romantic relationship or simply make new friends. As a way to simplify the creation of profiles, the site offers the option to log in with the users' favorite social media site (like Facebook, Twitter, Google Plus, etc.), and optionally also link the profile with multiple such sites.

The main aim of the site is to match people who have compatible personalities; in order to have tools that can be leveraged to solve this (very difficult) problem, the site allows users to explicitly specify their preferences in different domains, such as music, literature, movies, and even relationships—these inputs are complemented by the information that is extracted (with the users' permission) from their linked social media profiles.

The second example setting is a comprehensive trip organization service.

Example 2 (Travel site). As a second example, suppose we have a web site (similar to TripAdvisor[1]) that is designed to help people choose a place to

[1] http://www.tripadvisor.com/

spend their next vacation; it includes information on destinations, transportation, hotels, tours, restaurants, best season to go, etc. Members publish reviews, including numeric scores for several different categories as well as free text where they can go into detail regarding their experience. There are rich social features available, such as tagging in posts or reviews, suggestions, and private messages.
As before, we assume that users are able to sign in with their social media accounts, which gives the system the possibility to extract relevant information— for instance, to suggest a destination, the system might use the fact that a user participates in Facebook groups for learning the German language to infer that they probably would like to travel to Germany.

In the following, we describe a list of desiderata to achieve the goal of designing a formalism to model and reason with social knowledge bases. As we will argue below, the problem essentially requires the combination of knowledge representation machinery for areas that have up to now largely been considered in relative isolation: (i) ontology languages, (ii) preference models, (iii) reasoning under uncertainty, (iv) stream reasoning, and (v) complex social networks.

2. Desiderata for Building and Querying Social KBs

In the previous section, we argued that it is necessary to develop novel KR tools to reason with social data; we will now offer further support for this argument by proposing a series of characteristics and capabilities that SKBs should have— developing such a desiderata has the additional value of acting as a road map for guiding future research efforts in this direction. For further discussion of literature related to each point, see Section 4,

(1) Model complex networks. In social knowledge bases, it should be possible for entities to be of different types: people, products, companies, books, movies, etc. Furthermore, it should be possible for there to be different kinds of relationships among them. It is thus necessary to be able to represent networks with different kinds of nodes, as well as multiple attributes and relationships for each one—such models are often referred to as *complex networks* [2].

Consider the setting from Example 1; in this case, it is clear that it would be useful for connections between users to contain additional information about the relationship they have. For instance, kind of tie (relative, classmate, work partner, etc.), how long they have known each other, how many social media sites they frequent, etc. Another important observation is that connections do not always need to be symmetric—in the dating example, person A can consider person B to be a good match, but B may not agree. Having rich information about entities and how they are related can thus be useful to improve users' experiences.

(2) Model atomic actions. A specific set of actions (by agents or exogenous factors) that can occur in the domain need to be identified.

Considering social media sites like Facebook or Google Plus, such actions could include posting, commenting, liking/+1 a post, friending/unfriending, messaging, etc. It is these actions that will be the building blocks for inferences about preferences or regarding reasoning under uncertainty, as we will discuss below. Also in connection with a point discussed in the following is the fact that an adequate selection of atomic actions to be modeled will have an impact on computational tractability. User data in social media suffers constant change, and a regular user could produce a large amount of data per day; depending on the way in which the SKB will be used, it may not be necessary to incorporate all of this data into the model. For instance, regarding the setting in Example 2, it may be a good idea to incorporate comments as actions since users may give information regarding preferences in their mode of travel (for example, that they are afraid to fly); on the other hand, this might be less relevant for Example 1. Hence, it is essential to characterize and prioritize atomic actions so that resources are not wasted by processing and storing unnecessary data.

(3) Model quantitative and qualitative preferences. Quantitative preferences are often useful when automatically learning from data, or in simple domains; on the other hand, qualitative preferences (defining strict partial orders) are often more naturally elicited from human beings but more difficult to extract automatically.

To illustrate this point, consider the travel setting from Example 2. Quantitative preferences could be obtained from users' explicit rankings of favorite cities, countries, museums, beaches, etc. On the other hand, reviews or polls could also provide less structured preferences, such as the fact that the user prefers beach destinations to mountain ones, or that hotels near the city center are preferred over those that are not.

(4) Reason about groups. Social knowledge is inherently related to groups of entities (where entities are not necessarily all people); groups sometimes function as higher-level entities with their own preferences, relationships, etc.

There are many ways in which groups can be important when leveraging social knowledge. In Example 1, a group may be defined with respect to people's age group and interests, and the general preferences of such groups can be used in order to supplement the preferences of the individual. On the other hand, in Example 2 one can take the users' closest friends as a source of suggestions for travel destinations or activities—in this case, the group of friends is used as the basis of a kind of crowdsourcing. Challenges in this respect involve identifying the best possible composition of groups (for instance, determining who the users' closest friends are by considering how long they have known each other, share interests, etc.), and what to do about group members with conflicting preferences. The latter has been recently addressed in [16].

(5) Reason about cascading processes. One of the main characteristics of social networks is that information "flows" through them—this kind of dynamic is often referred to as a "cascading process" [11].

A clear example of this kind of process can occur in the travel domain (Example 2), where a user might travel to a new destination and post a series of pictures with very positive comments about their experience. This might cause several of the user's connections to "like" that destination and even plan trips there—the process can of course continue, with the new converts' activities causing some of their connections to do the same. It is thus important to model how influence propagates; there is extensive work in this area, and the logic programming proposal of [20] is perhaps the closest in spirit to the general approach that is required for SKBs.

(6) Flexible characterization of consistency/inconsistency. Classical conceptions of consistency are not adequate for modeling the kind of information that occurs in social settings—a more flexible approach is required for handling conflicts.

In our example settings, simple inconsistency cases might occur, for instance, when users have accounts in several social media sites but focus more on one than the others. Since data is usually not shared between accounts, it can occur that a user who lives in city C_1 later moves to city C_2 and only updates their profile for one of the accounts. An SKB taking information from these profiles would thus encounter an inconsistency. A more challenging case of inconsistency, much more difficult to characterize, is the case of a user of the system in Example 2 who strongly prefers beach destinations but suddenly starts paying attention to mountain-related places and activities (such as with +1s, posts, comments). The classical way to deal with the above situations is to try to modify the information contained in the knowledge base as little as possible in order to reach a consistent state without losing unnecessary information [7]; this is closely related to the following point.

(7) Social network-based belief revision operators. In close connection to the previous point, belief revision operations need to be applied in response to different kinds of events that signal changes in the SKB. The difference with respect to the classical setting is in relation to other points on this list—in particular, consistency, cascades, and uncertainty.

Among these, the relationship between cascades and belief revision operators is, to the best of our knowledge, never been studied. As an example, consider our travel setting and suppose an influential individual changes their opinion with respect to a certain destination (for instance, they start to express negative opinions about it and "unlike" the relevant pages), causing others to follow suit; how should this cascading belief revision process evolve?

(8) Reason about uncertainty. Conflicting information and inherently uncertain data makes it necessary to have an explicit representation of uncertainty.

There are many examples of the need to reason with uncertain knowledge. In our example dating application, some user information is private, and so cannot be directly used and perhaps only approximations can be obtained. For instance, user location can be approximated by content-based methods leveraging features of posts, such as mentions of place names and use of local dialect—since these are prone to error, a measure of probability must be assigned that depends on the kind and amount of information that supports each inference. Approaches to reasoning with ontological knowledge and user preferences have recently been proposed in [17].

(9) Rich query answering. Social knowledge is rich, and access to such knowledge often requires queries that combine the basic relational database-style queries with the graph-based queries often used in linked data [4].

Consider the travel setting from Example 2; queries to an SKB in this case might involve complex requests such as "*hotels with free wi-fi connection that have been positively reviewed by people who share my views and that at least one connection recommends, in order of preference*". This involves reasoning under uncertainty (it is not always possible to determine if free wi-fi is available), reasoning about groups, and network structure, and preferences. Formalizing novel types of queries for SKBs, and obtaining effective algorithms to answer them, is therefore one of the main challenges ahead. Recent work [10] that can be leveraged towards this goal has proposed efficient algorithms for social networks under uncertainty.

(10) Time and space constraints: scalability and stream reasoning. Successful SKB formalisms must be able to cope with very large knowledge bases that are updated often with information that must be processed on the fly (or nearly so).

Micro-blogging is a clear example of how often new data is created: Twitter has about 100M active users who post over 230M tweets a day [3]. Processing such a high volume of data—much of which may not even be valuable [13] and that has a short life span—is a formidable challenge. An even greater challenge is to make the tools and processes that we propose in the previous points work adequately in such a setting. Considering the travel application from Example 2, a site with many active users must deal with a large volume of new comments, reviews, multimedia posts, and connections between users; an SKB that models even a portion of this activity must therefore be able to keep up with updates that, as we have seen, involve complex reasoning tasks.

3. Outlining a Framework for Social Knowledge Bases

Using the list of features discussed in Section 2 as a guide, we now briefly outline what a framework that integrates all of them might look like. A social knowledge base can be modeled as a 5-tuple of the form $SKB = (O, N, P, M, B)$, where:

– *O* is an *ontology* modeling the general knowledge about the domain. For instance, in the travel domain *O* would contain the database of hotels, flight routes, etc., as well as intensional knowledge such as *hostels are a kind of lodging,* or *wi-fi is a kind of internet connection*. This component could be modeled with the Datalog+/– family of ontology languages [5], which contains many different fragments focused on tractable query answering that generalize other well-known ontology languages such as the DL-Lite family of description logics.

– *N* is a model of the underlying *social network structure*. Since this is a kind of ontological knowledge, it could also be modeled using Datalog+/– ; however, we propose to model them as separate components so that other approaches that are more specific can be used, such as the MANCaLog language [20].

– *P* is a *preference model* over the consequences of ontology *O*. This kind of integration has already been proposed in [14] and later extended to preferences under uncertainty [17] and preferences over groups [16].

– *M* is a *probabilistic model* for ontology *O*. There are different ways in which probabilistic uncertainty can be integrated into ontological knowledge. For instance, in [8] annotations are added to both extensional and intensional knowledge, and the probabilistic model provides a probabilistic distribution over the annotations—this is an elegant way to allow for a separation of interests between the two models. Of course, other possibilities may be more appropriate depending on the domain of application.

– *B* is a set of *belief revision operators*. As was motivated before, revision operators that are informed by all the other components are needed in order to modify the knowledge base when new information needs to be incorporated. One approach in the logic-based probabilistic belief revision literature is the recent work of [22], which studies quantitative approaches to belief revision in a probabilistic structured argumentation language.

	D1	**D2**	**D3**	**D4**	**D5**	**D6**	**D7**	**D8**	**D9**	**D10**
O – Ontology	×	×	×	×	×	×	×	×	×	×
N – Network	×			×	×	×	×	×	×	×
P – Preferences			×	×		×		×	×	×
M – Probabilistic model	×				×	×		×	×	×
B – BR operators		×					×	×		×

Fig. 1. *Assuming an SKB of the form (*O, N, P, M, B*), this table shows an example of the involvement of each component in satisfying the desiderata from Section 2. Different application settings may require different setups.*

These components are coupled differently depending on the modeling or reasoning task that they are required to perform—the table in Figure 1 shows how each component might be typically involved in addressing each desideratum proposed above. For instance, desideratum D6 regarding consistency might involve components O, N, P , and M , since determining consistency may require ontological knowledge, access to network connections, user preferences, and probabilities. Of course, different applications may require different setups; returning to our example, in this case perhaps social connections are not considered for assessing consistency.

4. Discussion: Related Work and Challenges

Extracting, representing, and reasoning about the kind of information described above is a complex problem; although the examples may look simple, many issues arise when trying to combine all available data. There are some recent developments in the literature on ontological languages that are related to our present efforts in that they have already begun to investigate how some subsets of these areas can be adequately combined. The Datalog+/– family of ontology languages [5] has recently received a lot of attention given its flexibility and variety of available fragments that ensure tractable query answering. In [14], the authors explore an extension of Datalog+/– with preference models that allows to rank the answers to queries with respect to users' preferences; a further extension to this approach was proposed in [16], where group preferences are considered as well. A related approach, considering the problem from the somehow dual perspective of extending the general model of CP-theories for preference representation with ontological constraints, was recently proposed in [19]. Another recent approach is the Prob-EL formalism [9], which extends the EL description logic with probabilistic uncertainty over both assertional and terminological knowledge.

In a separate but closely related vein, Datalog+/– was also extended with probabilistic models in [8], where the authors study both algorithms for ranking answers with respect to their associated probabilities and query answering under inconsistency. These two lines were considered together in [17], where the authors explore the problem of ranking answers to queries with respect to both probabilistic uncertainty and user preferences. Several other ontology languages have been extended with probabilistic uncertainty—see [18] for a survey of earlier approaches.

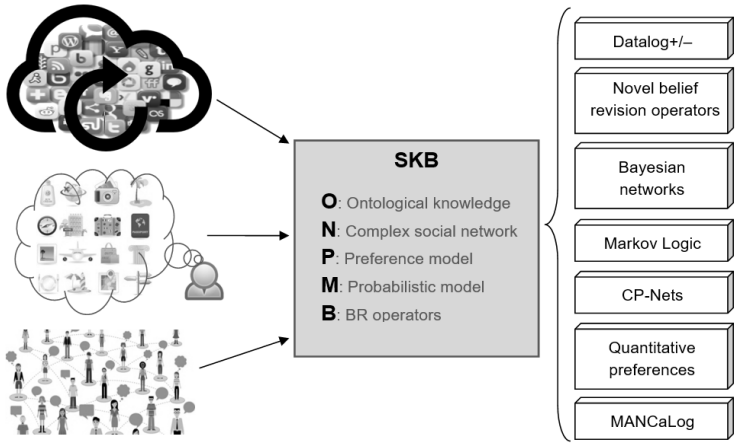


Fig. 2. *A high-level overview of the proposed process of modeling and reasoning with social knowledge. SKBs are built with information from the Web, individual users interacting with online services, and social media. Individual components of the SKB are modeled using different kinds of formalisms proposed in the literature for solving more specific sub-problems.*

Also related to this line of research is the study of probabilistic databases [12], where the ontological aspect is missing but the focus is rather on computational tractability. Another quite recent formalism for expressing preferences under uncertainty—also not ontology-based—was introduced in [15].

Stream reasoning [6] refers to the problem of processing information that continuously becomes available and cannot all be stored (a fixed window is generally assumed). From the point of view of making sense of data in social media, the recent work of [3] analyzes key research questions for mining data with semantic content from social media streams. Their work is perhaps the closest in spirit to our goal, though the main difference is that they are focused primarily on extracting information while we are focusing on the problems of adequately organizing and accessing the information that is already extracted.

Towards a general framework

We have thus far proposed a set of desirable properties and sketched the organization of a framework for modeling and reasoning with SKBs; however, there are many challenges towards materializing the general vision. Figure 2 shows a high-level outline of this vision—SKBs are populated by three general sets of sources: social media and general Web-based resources, users themselves, and users' interactions with others. A mix between learning, scraping, and elicitation techniques, as well as knowledge engineering in general, will help obtain not only the information necessary for the individual components of the SKB but also the relationships between

them. These components will be built by leveraging as much as possible existing tools (such as Bayesian networks, Datalog+/–, etc.). Even if we assume that all necessary information is available to populate these components, there are many challenges associated with bringing them together: scalability issues arising from the combination of individually tractable components, semantic issues arising from the combination of open-world and closed-world assumptions, alignment issues arising from different schemas used in different components, normalization issues arising from combining different quantitative preferences, and so on.

5. Conclusions

In this position paper, we have discussed the need to develop novel knowledge representation and reasoning tools and techniques that are adequate for tackling the challenges that come with modeling social knowledge. We proposed a set of desiderata to guide the development of such formalisms, and briefly outlined how a unifying model can be built by leveraging existing research and novel developments. The main contribution of such a discussion is the proposal of a road map to guide research efforts towards this goal, as well as the novel proposal of combining several research lines that up to now have been considered largely in isolation: ontologies, preferences, uncertainty, stream reasoning, and complex social networks.

Acknowledgments. This work was supported by funds provided by CONICET and Universidad Nacional del Sur, Argentina. Some of the authors of this work were also supported by the U.S. Department of the Navy, Office of Naval Research, grant N00014-15-1-2742. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Office of Naval Research.

References

1. Berners-Lee, T., Hendler, J., Lassila, O., et al.: The Semantic Web. Scientific American 284(5), 28–37 (2001)
2. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex Networks: Structure and Dynamics. Physics Reports 424(4), 175–308 (2006)
3. Bontcheva, K., Rout, D.: Making Sense of Social Media Streams through Semantics: A Survey. Semantic Web 1, 1–31 (2012)
4. Broecheler, M., Pugliese, A., Subrahmanian, V.S.: DOGMA: A Disk-oriented Graph Matching Algorithm for RDF Databases. In: Proc. of ISWC 2009, LNCS, vol. 5823, pp. 97–113. Springer (2009)
5. Calí, A., Gottlob, G., Lukasiewicz, T.: A General Datalog-based Framework for Tractable Query Answering over Ontologies. Journal of Web Semantics 14, 57–83 (2012)

6. Della Valle, E., Ceri, S., Van Harmelen, F., Fensel, D.: It's a Streaming World! Reasoning upon Rapidly Changing Information. IEEE Intelligent Systems (6), 83–89 (2009)
7. Fermé, E., Hansson, S.O.: AGM 25 years. Journal of Philosophical Logic 40(2), 295–331 (2011)
8. Gottlob, G., Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Query Answering under Probabilistic Uncertainty in Datalog+/– Ontologies. Annals of Mathematics and Artificial Intelligence 69(1), 37–72 (2013)
9. Gutiérrez-Basulto, V., Jung, J.C., Lutz, C., Schroder, L.: A Closer Look at the Probabilistic Description Logic Prob-EL. In: Proc. of AAAI (2011)
10. Kang, C., Pugliese, A., Grant, J., Subrahmanian, V.S.: STUN: Querying Spatio-temporal Uncertain (Social) Networks. Social Network Analysis and Mining 4(1), 1–19 (2014)
11. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the Spread of Influence through a Social Network. In: Proc. of KDD 2003. pp. 137–146. ACM (2003)
12. Koch, C., Olteanu, D., Re, C., Suciu, D.: Probabilistic Databases. Morgan-Claypool (2011)
13. Liu, Y., Kliman-Silver, C., Mislove, A.: The Tweets they are a-Changin: Evolution of Twitter Users and Behavior. In: Proc. of ICWSM. vol. 13, p. 55 (2014)
14. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Preference-based Query Answering in Datalog+/– Ontologies. In: Proc. of IJCAI. pp. 1017–1023. IJCAI/AAAI (2013)
15. Lukasiewicz, T., Martinez, M.V., Simari, G.I.: Probabilistic Preference Logic Networks. In: Proc. of ECAI 2014. pp. 561–566 (2014)
16. Lukasiewicz, T., Martinez, M.V., Simari, G.I., Tifrea-Marciuska, O.: Ontology-based Query Answering with Group Preferences. ACM Transactions on Internet Technology 14(4), 25: 1–25:24 (2014)
17. Lukasiewicz, T., Martinez, M.V., Simari, G.I., Tifrea-Marciuska, O.: Preference-based Query Answering in Probabilistic Datalog+/– Ontologies. Journal of Data Semantics 4(2), 81–101 (2015)
18. Lukasiewicz, T., Straccia, U.: Managing Uncertainty and Vagueness in Description Logics for the Semantic Web. Journal of Web Semantics 6(4): 291–308 (2008)
19. Noia, T.D., Lukasiewicz, T., Martinez, M.V., Simari, G.I., Tifrea-Marciuska, O.: Combining Existential Rules with the Power of CP-Theories. In: Proc. of IJCAI 2015. pp. 2918–2925 (2015)
20. Shakarian, P., Simari, G.I., Callahan, D.: Reasoning about Complex Networks: A Logic Programming Approach. Theory and Practice of Logic Programming 13(4–5-Online-Supplement) (2013)
21. Shakarian, P., Simari, G.I., Schroeder, R.: MANCaLog: A Logic for Multi-Attribute Network Cascades. In: Proc. of AAMAS 2013, pp. 1175–1176.
22. Simari, G.I., Shakarian, P., Falappa, M.A.: A Quantitative Approach to Belief Revision in Structured Probabilistic Argumentation. Annals of Mathematics and Artificial Intelligence 76(3–4): 375–408 (2016).
23. Spanos, D.E., Stavrou, P., Mitrou, N.: Bringing Relational Databases into the Semantic Web: A Survey. Semantic Web 3(2): 169–209 (2012)

Evaluation of two new algorithms for the design of wind farms

FABRICIO LOOR [1], GUILLERMO LEGUIZAMÓN [1] AND JAVIER APOLLONI [1]

[1]Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)
Departamento de Informática - FCFMyN
Universidad Nacional de San Luis, Argentina
{faloor,legui,javierma}@unsl.edu.ar

Abstract. *In the last years the growth in electricity consumption has been exorbitant, which has caused the necessity to use the wind as a promising resource to extract energy. The distribution of wind turbines within a wind farm, in order to optimize the energy captured, is a complex problem to resolve. This paper presents two different optimization techniques to solve the wind farm design problem. One of the techniques is based on the Grey Wolf Optimizer algorithm modified to deal with binary vectors. The other one,* DonQuijote*, is a novel method that includes the use of Differential Evolution and a deep analysis of the problem. The experimental study is conducted on five scenarios taken from the Wind Farm Layout Optimization Competition in the GECCO 2015 conference. The effectiveness of both proposals is compared with a well-known optimization technique in the area (Genetic Algorithm).The results show that* DonQuijote *outperforms the other two techniques.*

Keywords: *Wind energy, wind farm layout, optimization, differential evolution, metaheuristics.*

1. Introduction

The design of wind farms is a widely studied issue [1] and still states of being in force mainly for the following reasons:

- Renewable energies are highly requested resources with a promising future.
- They are safe and ecological sources to satisfy the current energy demand.
- Finding the optimum layout of the turbines in a reserved area for a wind farm is a very complex problem.

The excessively large number of possible positions to place wind turbines within a wind farm causes the design of wind farms to be an intractable problem. Clearly, the large size of the search space makes impossible to use an optimization algorithm including an exhaustive search. Therefore, another class of optimization method is necessary. In this sense, the wide variety of optimization methods must be analyzed in order to choose the best option to find a closely optimal distribution of the turbines.

The number of turbines and their distribution within the farm can lead to great economic benefits. On the contrary, a bad distribution may significantly decrease

the total amount of energy produced by the turbines. Therefore, it is appropriate to apply stochastic methods to find a profitable solution within an acceptable execution time, without a substantial decrease in the accuracy of the solution.
In the Wind Farm Layout Optimization (WFLO) competition carried out in GECCO 2015 congress [11], a simple Genetic Algorithm (GA) was given as a baseline method to test alternative methods. This baseline algorithm does not consider the information of the problem, hence, many improvements are certainly possible.
Several algorithms considered computationally efficient to solve this problem can be found in the literature, for example, GA [2], CMA-ES [3], TDA [4], DEVO-I [5], DEVO-II [6] and PSO[7]. In this paper, we have developed two competitive approaches that consider the influence among the turbines as information of the problem. One of them, namely *DonQuijote*, includes Differential Evolution (DE) to find the best setting according to the area of location of the turbines. Thus, a new approach to this problem is exhibited. The results of the simulation prove the effectiveness of our proposals.
The paper is organized in the following way. Section 2 provides an overview of the problem of distribution of turbines and the costs involved. Section 3 describes the proposed algorithms. Section 4 presents and analyzes the experimental results. Section 5 contains the conclusions and possible future works.

2. Description of the problem

Assuming a Cartesian plane to distribute turbines and considering that the plane allows coordinates with continuous values, the location of each turbine is indicated by a pair of variables establishing a position within the plane. The limits of the plane are given as parameters. In the context of the design of wind farms, this plane is also known as scenario. Then, a solution would be a sequence of points (x, y) satisfying the constraints of the limits, and security conditions of the scenario. Each point should be situated to a minimum fixed distance from any other point. This distance has been established as four times the size of the turbine rotor.
Following the guidelines proposed in the WFLO competition, the scenarios are represented as described in [8]. The wind is simulated by a homogeneous model, i.e., all regions of the scenario receive the same amount of wind, unless a turbine interferes with the uptake of energy. In case of interference, the behavior of the wind will be stochastically simulated by using the Weibull distribution (see more details in [8]).
Besides, the turbines are located so that the rotor is oriented perpendicularly to the wind direction.
The scenarios have prohibited regions, called obstacles, for the location of the turbines. The obstacles are modeled as rectangles, and a solution is considered invalid if has a turbine within an obstacle.

2.2. Cost Model

To determine the quality of a solution, we follow the established in the 2nd Edition of Wind Farm Layout Optimization Competition [11], where the fitness function takes into account two opposed variables: the number of turbines and the power produced by the configuration. Specifically, the fitness function is calculated following the Equation 1:

$$fitness(M) = \frac{c_t \cdot n + c_s \cdot \lfloor \frac{n}{m} \rfloor (\frac{2}{3} + \frac{1}{3} \cdot e^{-0,00174 \cdot n^2}) + c_{OM} \cdot n}{\frac{(1-(1+r)^{-y})}{r}} * \frac{1}{8760 \cdot P(M)} + \frac{0,1}{n}, \quad (1)$$

where c_t, c_s, m, r, c_{OM} and y are constants. n, M and P are the variables of the wind farm layout. The constant c_t is the cost of a turbine, from the creation until its installation. In our cost model, the value of c_t is US$ 750000. The constant c_s is the cost of the substation, which is US$ 8,000,000. m stands for the number of turbines included in the substation. r is the annual interest rate, in our case, r has a value of 0.03. The constant y refers to the lifetime of the wind farm. In this case, the average lifetime has been estimated to be 20 years. Finally, the constant c_{OM} is the annual operating cost per turbine, which, in our case, it is US$ 20,000.

The proposed number of turbines is represented by n and M is a matrix describing the positions of each turbine in the configuration. Function P defined by Equation 2 calculates the total energy produced by the farm according to the position of each of the turbines. The function is:

$$P(M) = \frac{e_c}{(w_f \cdot length(M))}, \quad (2)$$

where e_c is the energy captured by the configuration M, w_f is the wake free energy on the farm, and $length(M)$ corresponds to the number of turbines in the configuration.

3. The proposed algorithms

In this section, we will give an explanation of the developed approaches. Initially, we explain the structure of the Binary Genetic Algorithms, then we develop a binary version of the Grey Wolf Optimizer, and finally our proposed algorithm specially adapted to the wind farm design problem. Note that in all described algorithms, the solutions are binary vectors that indicate whether a turbine should or should not be put in a given position (x,y).

3.1. Binary Genetic Algorithm (BGA)

GA is a metaheuristic inspired by biological evolution with a high genetic-molecular base. In an analogous way to the natural process, the algorithm modifies the solutions to the problem, called individuals, by using the random actions of the crossover, mutation, and selection operators. The main structure of a GA is shown in Algorithm 1. The stop criterion is the maximum number of the evaluations (MAXEVAL). In our case, the value of MAXEVAL is 2000.

```
BEGIN
     1: GENERATE P(0); // Initialize a random population
     2: EVALUATE P(0); // Evaluate each solution
     3: FOR (E = 1 to MAXEVAL)
     4:        P'(E) = SELECT (P(E)); // Select parents
     5:        P''(E) = CROSSOVER(P'(E));// Cross parents and
   generate a new population
     6:        P'''(E) = MUTATE(P''(E));//Mutate the new
   individual
     7:        P(E+1)= SELECT(P'''(E));
     8:        EVALUATE P(E + 1);
END
```

Algorithm 1: Pseudocode for GA

3.2. A binary variant of the Grey Wolf Optimizer (GWO)

The GWO mimics the leadership hierarchy and hunting mechanism of grey wolves in nature. Within a wolf pack, there are different categories of wolves: *Alpha, Beta, Delta* and *Omega*. The simulation of the behavior follows a very simple mechanism. The *Alpha* wolf leads the pack and, therefore, it has a strong influence in the way of exploration of the search space. The pseudocode is presented in the Algorithm 2. At the end of the run, the best solution will be the one represented by an *Alpha* wolf.

```
BEGIN
     1: GENERATE X(0); // Initialize a random population
     2: INITPARAMETERS(a,A,C);
     3: EVALUATE X(0); // Evaluate each solution
     4: SELECTNEW(Alpha, Beta, Delta, X(0));
     5: FOR (E = 1 to MAXEVAL)
     6:        FOR(wolf w in Omega)
     7:               FOR(i = 0 to DIM)
```

```
    8:                    UPDATEPOSITION(w,i);
    9:        SETPARAMETERS(a, A, C);
    10:       EVALUATE P(E + 1);
    11:       SELECTNEW(Alpha, Beta, Delta, X(E+1));
END
```

Algorithm 2: Pseudocode for GWO algorithm

The parameter *a* is the factor of exploration which will be decreased from 2 until 0, as increasing the iteration number. Parameters *A* and *C* are vectors having three different random numbers and help explore the search space. The key of the algorithm is the function `updatePosition` (see line 8 in Algorithm 2) which is defined as:

$$\vec{X}_{l,i} = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{2}, \tag{3}$$

where $\vec{X}_1, \vec{X}_2, \vec{X}_3$ are obtained as follows:

$$\vec{X}_1 = |\vec{x}_\alpha - \vec{A}_1 * \vec{D}_\alpha|, \ \vec{X}_2 = |\vec{x}_\beta - \vec{A}_2 * \vec{D}_\beta|, \ \vec{X}_3 = |\vec{x}_\delta - \vec{A}_3 * \vec{D}_\delta|.$$

The values for $\vec{D}_\alpha, \vec{D}_\beta, \vec{D}_\delta$ are determined by applying the following formulas:

$$\vec{D}_\alpha = |\vec{C}_1 * \vec{X}_\alpha - \vec{X}|, \ \vec{D}_\beta = |\vec{C}_2 * \vec{X}_\beta - \vec{X}|, \ \vec{D}_\delta = |\vec{C}_3 - \vec{X}_\delta * \vec{X}|.$$

The GWO algorithm works with solutions based on real numbers. However, in this paper, the algorithms have been adapted to deal with to Boolean values. In order to achieve this purpose, the result of Equation 3 is discretized by the rule that if $\vec{X}_{l,i}$ is greater than 1, it returns 1; in another case, it returns 0.

3.3. A new model: *DonQuijote*

In this section, we introduce a novel algorithm for wind farm distribution. To do that, we divide the algorithm into three stages: initialization, preprocessing and iterative step.

Initialization

Several populations containing four random solutions are generated. Taking into account 30 turbines by substation, the amount of turbines per solution for each population is $(i * 30) - 1$, where the value of *i* is decreased whenever a new population is generated. The number of turbines is updated by a multiple of 30 units because the floor function (i.e., the largest integer less than or equal) in the numerator of Equation 1 divides the number of turbines by the

number of turbines in a substation (i.e., 30). This has an advantageous consequence in the manner that the search space is discretized.
Following the process, the population with the best average fitness value and / or the best individual is selected for the next stage.

Preprocessing

At this stage, several random individuals are added to the selected population in order to alleviate the pressure on certain areas of the search space. Initially, a solution is added, where the turbines are located in perpendicular lines regarding the direction in which greater strength is captured. Beside, new individuals with a fixed number of randomly located turbines are added.

Iterative step

As Binary Differential Evolution (BDE) operators are applied at this stage, it is appropriate a short description of that operator before presenting the iterative step itself.
Differential Evolution, like the genetic algorithm, forms part of the evolutionary computation and applies similar operators to GA (crossover, mutate and select) but they have a different behavior. DE was originally raised for continuous spaces. But in our case the solutions are binary vectors, which are modified in each generation or evaluation E. The vector of the binary Differential Evolution can be represented as:

$$X(i,e) = (x_{ie}(1), x_{ie}(2), \ldots, x_{ie}(D)),$$

where *i* represents the index of the vector in the population, also known as the number of individual, *D* means the number of possible places where turbines can be located [10].

The mutation operator generates a mutant vector from a base solution r_0 belonging to the population and the outcome of the Hamming distance over other two solutions. The result of the distance is multiplied by a factor of mutation F. This constant controls the speed and robustness of the search.
In the crossover operator, a solution of the current population is mixed with the mutated solution. The probability of crossover, Cr, determines how similar the outcoming vector should be to the mutated vector.
Finally, the selection operator chooses, deterministically, the best solution between the solution obtained by the crossover operator and a current member of the population.
Continuing with the *DonQuijote* algorithm, the main structure of the iterative step is outlined in Algorithm 3.

```
BEGIN
    1: FOR (E = 1 to MAXEVAL)
    2:        FOREACH(solution S in Population)
    3:               SWITCH (E mod 4)//mod is the rest of the
  integer division
    4:                    CASE 0: FirstOperation();
```

```
     5:                    CASE 1: SecondOperation();
     6:                    CASE 2: ThirdOperation();
     7:                    CASE 3: FourthOperation();
END
```

Algorithm 3: Overall structure of the iterative step

`FirstOperation` method selects a turbine to be moved to an empty position with the worst received input force. The `SecondOperation` method generates, in the first evaluations, new random distributions without changing the number of turbines. When the number of evaluations is large enough, the turbine with the worst average reception power is selected, and then a turbine found in the best direction of the selected turbine is relocate to the worst direction. The `ThirdOperation` method applies BDE, above named, with a crossover probability of 0.783 and mutation probability of 0.06. The `FourthOperation` method generates, for the first evaluations, random distributions decrementing by one the number of turbines. When the number of evaluations is large enough, the `FirstOperation` is applied.

4. Experimental results

The performance of the different algorithms will be showed on 5 scenarios taken from the 2nd WFLO competition [11]. The representation of the scenarios is detailed in Figure 1. The blue rectangles are obstacles, i.e., places where it is not possible to locate turbines.

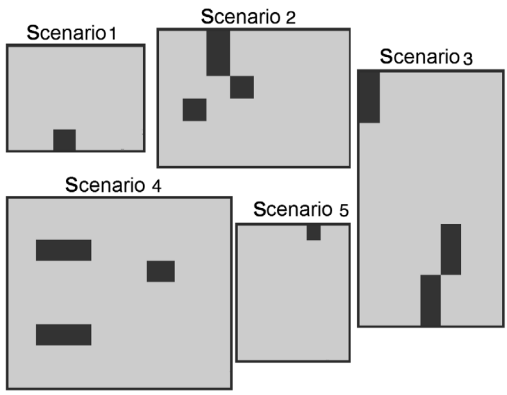


Figure 1. *Size of the scenarios and location of obstacles*

Table 1 helps us to compare the tested scenarios, showing the detail of the parameters used for each of them. The energy lost stated in Table 1 is a factor which influences the captured energy by the wind turbines and depends on ground conditions. This variable is defined in [8]. The maximum amount of turbines is the number of turbines that can be placed on the scenario, fit vertical rows and satisfying security restrictions between each pair of turbines.

Features	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Width	9240	6545	6930	10780	5390
Height	6545	5005	12320	9240	6545
Energy Lost	6148.648	8674.542	12344.639	11314.82	7441.038
Maximum amount of turbines	607	362	843	967	390

Table 1. *Parameters of the scenarios used in the experimental study.*

The compared algorithms was developed using JAVA language. The experimental results were conducted on a PC with a 2.20 GHz Intel (R) Core (TM) i3-2330M processor, 4 gigabytes of RAM memory running Windows 7 Home Basic 64-bits operating system.

The results of the executions are shown in Figure 2. The left-hand column of Figure 2 presents the results obtained by each algorithm in each scenario considering the production price of a kilowatt. The right-hand column of Figure 2 shows the number of turbines used in each configuration of the scenarios.

According the illustrated in Figure 2, *DonQuijote* is capable of providing very competitive results and obtain, in all scenarios, the best final fitness values. It is worth noting the performance of *DonQuijote* in the exploitation stage in relation to the number of turbines (graphs on the right in Figure 2). In those same graphs may be detected the initialization stage of *DonQuijote*. The stepped shape of the plots for *DonQuijote* (see right-hand column in Figure 2) is due to fact that the algorithm searches configurations with an amount of turbines multiple of *(i*30)-1*, such as stated in Section 3.3. Therefore, the algorithm shows a better performance as result of that particular manner of selecting the number of turbines. The improvement in the performance of the algorithm is due to a faster exploitation of the search space. This feature can be helpful in the design of wind farms using the fitness function shown in Equation 1.

The execution times for each algorithm on the 5 scenarios are as follows:

- Binary GA: 169 minutes and 45 seconds.
- Binary GWO: 173 minutes and 29 seconds.
- *DonQuijote*: 932 minutes and 21 seconds.

It is worth mentioning that the execution time can be negligible in the design of a wind farm. However, it can be a feature to take into account for future improving of the *DonQuijote* algorithm.

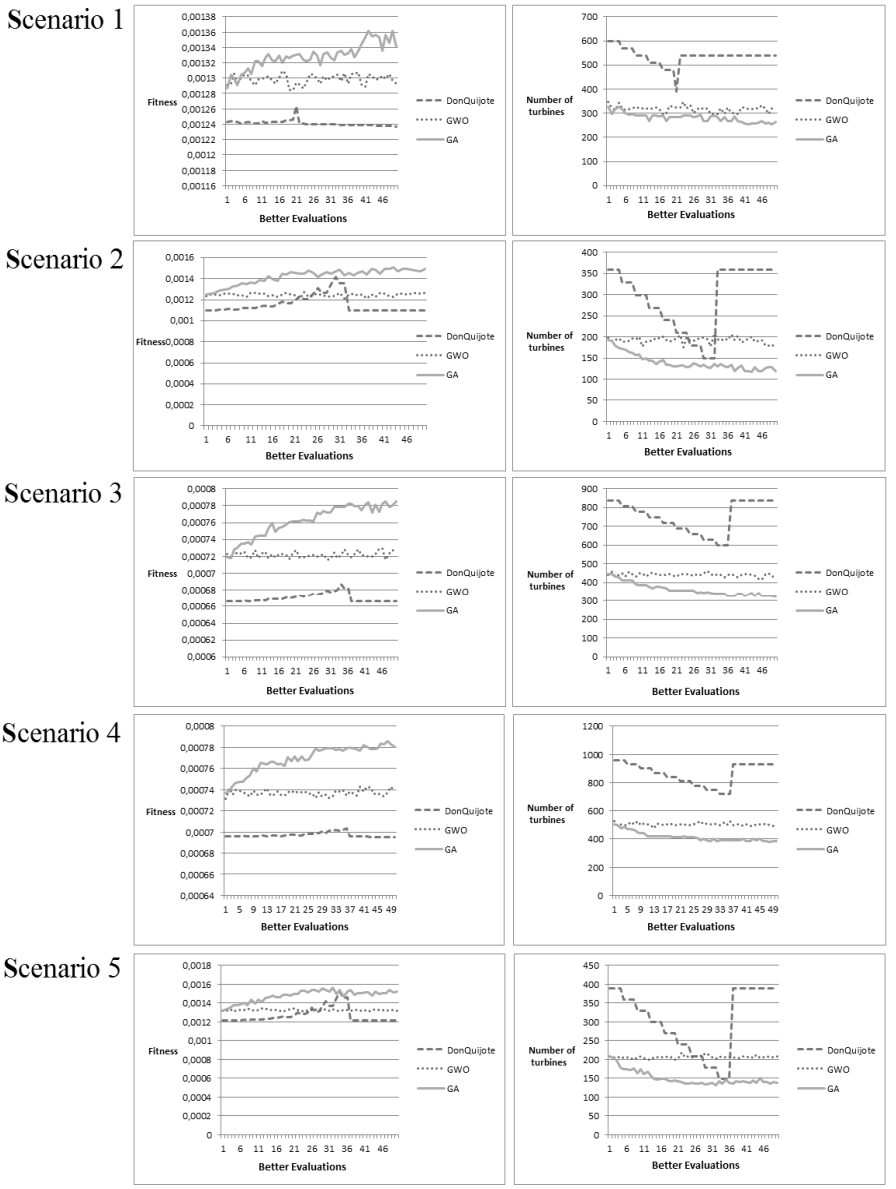


Figure 2. *Comparison between GA, GWO and* DonQuijote *algorithms in terms of the production price of a kilowatt and the number of turbines.*

5. Conclusions and Future Work

In this paper, we have introduced two new algorithms, GWO and *DonQuijote*, to solve the problem of wind farms layout optimization. The best performer algorithm, *DonQuijote*, combines simultaneously and cooperatively a metaheuristic together with a strong analysis of the problem in order to optimize the production price of a kilowatt. The initialization step of the algorithm is used to limit the search on the solution space, the preprocessing stage serves to exploit the areas of the search space found in the previous stage and, finally, evolutionary algorithm (binary DE) along with other own operations of the problem are used to improve the current solutions.

Preliminary results show that the proposed algorithm (*DonQuijote*) outperforms the results of the baseline algorithm (GA) for all evaluated scenarios. In consequence, we have experimentally confirmed the good results obtained by the evolutionary algorithms applied to optimize the design of a wind farm. However, the execution times of the *DonQuijote* algorithm should be improved.

For the future, we will try to locate the turbines anywhere, eliminating the static grid and keeping only constraints of security. Other future job is to improve the proposed algorithms in order to participate in the following editions of the WFLO Competition.

Acknowledgments

The authors are grateful to the LIDIC (Research Laboratory and Development in Computational Intelligence) and, also to the support of the funds of PROICO 330214 from the Universidad Nacional de San Luis.

References

1. R. Wiser and M. Bolinger, Annual report on U.S. wind power installation, cost, and performance trends: 2006. Available from. Golden, CO: NREL, US Department of Energy, http://www.nrel.gov/wind/pdfs/41435.pdf, (2007).
2. S. Grady, M. Hussaini and M. Abdullahet , Placement of wind turbines using genetic algorithms. Renewable Energy vol. 30:259--270, (2005).
3. M. Wagner, K. Veeramachaneni, K. Neumann, and U. O'Reilly. Optimizing the Layout of 1000 Wind Turbines.European Wind Energy Association Annual Event: 1--10, (2011).
4. M. Wagner , J. Day and F. Neumann. A Fast and Effective Local Search Algorithm for Optimizing the Placement of Wind Turbines, Renewable Energy vol. 51: 64--70, (2013).
5. D. Wilson, E. Awa, S. Cussat-Blanc, K. Veeramachaneni and U.-M. O'Reilly. On learning to generate wind farm layouts. Proceedings of the GECCO. 767--774, (2013).

6. D. Wilson , S. Cussat-Blanc, K. Veeramachaneni, , U. O'Reilly and H. Luga. A Continuous Developmental Model for Wind Farm Layout Optimization. Proceedings of the GECCO. 745--752, (2014).
7. Veeramachaneni, K.; Wagner, M.; O'Reilly and U.-M.; Neumann, F., Optimizing energy output and layout costs for large wind farms using particle swarm optimization, Evolutionary Computation (CEC), IEEE Congress on, 1--7, (2012).
8. A. Kusiak and Z. Song. Design of wind farm layout for maximum wind energy capture. Renewable Energy, vol. 35(3):685--694, (2010).
9. S. Mirjalili, S. M. Mirjalili, A. Lewis. Grey Wolf Optimizer, accepted in Advances in Engineering Software , vol. 69: 46--61, (2013).
10. T. Gong and A. L. Tuson. Differential Evolution for Binary Encoding, Soft Computing in Industrial Applications, ASC 39, 251--262, (2009).
11. 2nd Edition of the Wind Farm Layout Optimization Competition. GECCO. Avaliable in http://www.irit.fr/wind-competition/, source scenarios: https://github.com/d9w/WindFLO/tree/master/Wind%20Competition/2015/Scenarios, (2015).

XVI

Distributed and Parallel Processing Workshop

Characterizing a Detection Strategy for Transient Faults in HPC

DIEGO MONTEZANTI[1,4], DOLORES REXACHS[2], ENZO RUCCI[1,3], EMILIO LUQUE[2], MARCELO NAIOUF[1] AND ARMANDO DE GIUSTI[1,3]

[1] III-LIDI, Facultad de Informática, UNLP
Calle 50 y 120, 1900 La Plata (Buenos Aires), Argentina
{dmontezanti, erucci, mnaiouf, degiusti}@lidi.info.unlp.edu.ar
[2] Departamento de Arquitectura de Computadoras y Sistemas Operativos, UAB
Campus UAB, Edifici Q, 08193 Bellaterra (Barcelona), Spain
{dolores.rexachs, emilio.luque}@uab.es
[3] Consejo Nacional de Investigaciones Científicas y Técnicas
[4] Instituto de Ingeniería y Agronomía, UNAJ
Av. Calchaquí 6200, 1888 Florencio Varela (Buenos Aires), Argentina

Abstract. *Handling faults is a growing concern in HPC; greater varieties, higher error rates, larger detection intervals and silent faults are expected in the future. It is projected that, in exascale systems, errors will occur several times a day, and that they will propagate to generate errors that will range from process crashes to corrupted results, with undetected errors in applications that are still running. In this article, we analyze a methodology for transient fault detection (called SMCV) for MPI applications. The methodology is based on software replication, and it assumes that data corruption is made apparent producing different messages between replicas. SMCV allows obtaining reliable executions with correct results, or, at least, leading the system to a safe stop. This work presents a complete characterization, formally defining the behavior in the presence of faults and experimentally validating it in order to show its efficacy and viability to detect transient faults in HPC systems.*

Keywords: *transient faults, detection, scientific parallel applications, silent data corruption, HPC, fault injection.*

1. Introduction

Processor clock frequency stagnation has resulted in performance improvements being achieved through increasing the number of components. System escalation involves to the problem of a decrease in tension which, together with sub-micron miniaturization challenges, results in great increases in failure rates. Electromagnetic interferences generate current pulses that alter the values that are stored or in combinational logics. The higher variability in manufacturing processes

causes inconsistent behaviors, while aging results in permanent errors being more frequent and the likelihood of multiple failures has also increased [1,2]. Because all of this, system reliability has become critical, especially in the area of High-Performance Computing (HPC) with more than hundreds of thousands of cores. Recent studies in modern supercomputers show that Mean Time Between Failures (MTBF) are just a few hours [3], and it is estimated that they could even get to about 30 minutes in large parallel applications in exascale platforms. Consequently, these applications will not be able to progress efficiently without appropriate help [4,5]. The main concern is in relation to silent failures, namely Silent Data Corruption (SDC), with numerous reports and studies on their probabilities and impacts surfacing [2,6,7]. By potentially causing invalid results, SDCs create serious problems in science, which increasingly relies on large-scale simulations. For all these reasons, SDC mitigation is one of the major challenges for current and future resilience.

SDCs appear as bit-flips (change in the value of a bit) that affect the storage or the cores. To detect or correct them, manufacturers add more powerful Error Correcting Codes (ECC) in the memory, protect buses with parity bits, and add redundancy to the circuits of some logical units [8]. However, adding hardware redundancy to the registry and processor arithmetical logic units is too costly [9].

The small supercomputer market, which requires high reliability, can be satisfied with double- and triple-redundancy solutions to achieve detection and correction, respectively. Even though the cost of doing this is high, it is preferable to having corrupt results. SDCs remain latent until the altered data are used, and detection latencies depend on the application.

The standard, most commonly used method to handle errors in current parallel systems (particularly those that run MPI applications), is recording periodical checkpoints. In case of failure, the Checkpoint/Restart (C/R) method re-launches the application from the last checkpoint. Unfortunately, the overhead for using C/R increases with the number of cores. Taking into account the time required for C/R and re-launch, a significant amount of useful computation time could be wasted if the MTBF is very low. The situation gets worse if computation is strongly coupled, since an error in one node could be propagated to the others in micro-seconds [1,10].

The traditional model based on C/R assumes that detection is almost immediate. Additionally, if the stored checkpoint contains undetected failures, recovery will not be possible. The few general detection techniques currently available introduce high overheads in parallel applications [2,11]. Based on all this, detection latency ranges are expected to increase, making the problem even worse due to SDCs. There are no efficient containment mechanisms, either which means that a failure that affects one task can result in the application crash or in incorrect outputs that, in a best-case scenario, are only detected after execution is complete and which are very hard to correct.

Replication at process-level has proven to be a reliable alternative, but in order to make it appealing for HPC, there are some challenges still to be solved, such as minimizing time and resource utilization overheads, ensuring that the inner states of the replicas are equivalent to one another (which is not trivial, since non-deterministic operations could be run), and reducing energy consumption. Traditionally, SDC are detected by replicating executions and comparing the results obtained. RedMPI [2] does this at the level of the processes, but there are other methods that do it at the level of the threads [12]. Other solutions that require less resources and are less accurate have also been explored, such as approximate replication, which implements upper and lower limits for computation results [1].

In this context, the SMCV methodology [13,14] has been proposed in recent years. SMCV is designed to detect transient failures in HPC, specifically for scientific applications that use MPI on multicore clusters. SMCV allows obtaining reliable executions with correct results or, at the very least, report the occurrence of SDCs and taking the system to a safe stop after a limited detection latency, saving significant time, especially in long applications.

The remaining sections of this document are organized as follows: Section 2 reviews some basic concepts, while Section 3 describes related work. Section 4 details the strategy used in SMCV, in which the behavior in case of failure, its Sphere of Replication (SoR), and its vulnerabilities are formally defined. Section 5 describes the experiments carried out through a controlled fault injection, in order to validate the behavior defined and showing the efficacy and viability of SMCV to detect transient failures in HPC systems. Finally, in Section 6, the conclusions and future lines of work are presented.

2. Basic Concepts

Depending on the impact on application execution, transient faults can be classified as follows [13]:

- Latent Error (LE): it affects data that are not used afterwards, so it does not have an impact on results.
- Detected Unrecoverable Error (DUE): it causes an anomaly that the system software can detect and that is unrecoverable; it usually causes the application to end abruptly.
- Time Out Error (TO): the program does not end within a given period of time.
- Silent Data Corruption (SDC): it is not detected by any system software level, and its effects are propagated until the program ends with an incorrect output. In parallel applications with message passing, these can cause: Transmitted Data Corruption (TDC), which affects data

that are part of the contents of the messages to be transmitted (if undetected, it propagates to other processes), or Final Status Corruption (FSC), where the altered data are not transmitted, but they are propagated locally, corrupting the final status of the affected process.

3. Related Work

Current technologies cannot deal with frequent SDCs. Existing algorithmic solutions [15] can only be applied to specific kernels; hence, mechanisms that allow dealing with the errors that are beyond their scope should be assessed. On the other hand, compiler- or runtime software-based detection strategies can be applied to any code, but they are more complex in nature.

Contention aims to avoid the propagation to other nodes of the damage caused by the fault, or to prevent it from corrupting the data stored as a checkpoint, which would make recovery impossible [1]. In [16], the authors propose the use of redundancy in HPC systems, which allows increasing system availability and offers a trade-off between the number of components and their quality. In [17], the authors show that replication is more efficient than C/R in situations where MTBF is low and the time overhead of C/R is high. Software-redundancy solutions are focused on replication at the level of the threads [12], processes [9] and machine status to remove the need for expensive hardware.

MR-MPI [19] is another proposal for transparent redundancy in HPC - it offers partial replication (only some processes are replicated); it can be used in combination with C/R in non-replicated processes [20,21].

rMPI [18] is a protocol for the redundant execution of MPI applications, focused on failures that cause the system to stop; it used the profiling layer to interpose MPI functions. Each node has a replica so, in case of a permanent failure, the redundant node continues without interruptions; the application fails if two corresponding replicas fail. Redundancy scales, i.e., the probability of simultaneous failure of a node and its replica decreases when the number of nodes increases, at the cost of duplicating the amount of resources used and quadrupling the number of messages. RedMPI [2] is a MPI library that exploits rMPI's process replication to detect and correct SDC, comparing at the receiver the messages sent by replicated issuers. It implements an optimization based on hashing to avoid sending all messages and comparing their entire contents. It does not require application code modifications and it ensures that replicas are run deterministically. Results show that it can protect applications even with high failure rates with time overheads below 30%, so it can potentially be used on large-scale systems. The authors in [2] analyze the propagation of SDCs among nodes through MPI communications, and they show that even a single transient failure can have a deep effect on the application, causing a cascading corruption pattern towards all other processes.

The same as SMCV, by focusing on messages, RedMPI monitors the most critical data for the application; communication correction is necessary to output correction. Since SDC can affect data that are not communicated immediately, the failure is detected upon transmission. However, unlike SMCV, RedMPI performs its validation on the receiver side. This is because, on the side of the issuer, all replicas must communicate with the others to verify their contents internally before sending the message. This results in additional overhead and latency, since the receiver loses all that time before being able to continue. Since SMCV replicates at the level of the threads and not the processes, it does not need to send messages among issuers for validation. By sending just one message after the validation, it does not cause network congestion. The same as SMCV, with RedMPI corruption remains confined to a process, even without correction. It also allows customizing replica mapping on the same physical node as the native processes (or in their neighbors with lower network latency).

4. Characterizing SMCV

In this section, SMCV is characterized.

4.1 Brief Review of SMCV

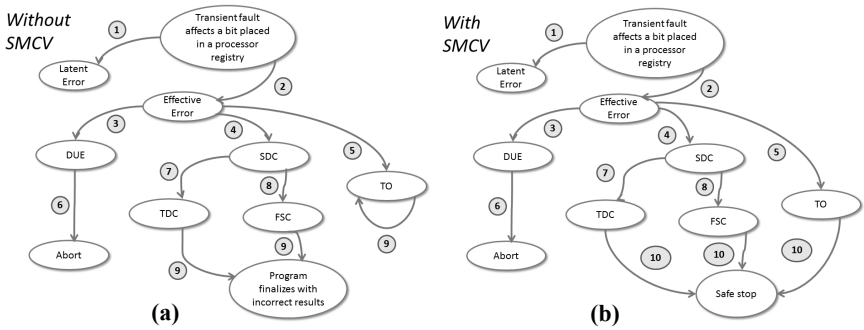
SMCV is a detection strategy that is based on validating the contents of the messages that are going to be sent among processes in deterministic parallel applications. It is designed to detect failures that cause SDCs (both variants) and TO. SMCV duplicates each application process in a thread, and it requires synchronization mechanisms between both concurrent replicas. When a communication is about to be established, the thread stops running and waits for its replica to catch up to it, and all message fields, calculated by both replicas, are compared in search for differences. If they match, only one of the threads sends the message, preventing errors to be propagated to other processes without using additional bandwidth. The receiver is synchronized with its replica, it receives the message and makes a copy for the replica, and then both replicas continue the execution. When they finish, results are verified to detect failures that may have been locally propagated to the end of the application.

4.2 Behavior in Case of Failure

In this section, the behavior of the detection methodology is described. Figure 1(a) shows a diagram with the possible status of an execution when there are no strategies implemented, while Figure 1(b) shows the same diagram when SMCV is applied. Ellipses represent statuses and arrows represent events that cause the transition from one status to another. Transitions are numbered, and each of them is described.

4.3 Sphere of Replication

Sphere of Replication [9] is a commonly accepted concept to describe the logical redundant execution domain of a given technique and specify the limits for failure detection. All data that enter the SoR are replicated, execution within its scope is redundant in some form, and output data are compared to ensure data correction before they leave it. Any execution outside the SoR is not covered for failures and should be protected by other means. The original concept of SoR was used to define reliability limits for redundant hardware designs, placing it around specific units. However, its application is not suitable for proposals implemented in software, despite which there are some solutions that use the compiler to insert redundant instructions that have tried imitating a SoR centered on hardware [22]. On the other hand, the failure detection paradigm centered on software places the SoR around software layers [9]. This shows that, even though failures affect hardware, only those that affect application accuracy are relevant, while it is safe to ignore those that remain latent. The disadvantage of this approach, however, is that detection is delayed until the error is confirmed through invalid data leaving the SoR, which means that a failure can remain latent indeterminately.



1. The affected bit is not used.
2. The affected bit is used by the application.
3. The altered bit affects data controlled by the operating system.
4. The altered bit affects user application data.
5. The altered bit causes the application to become unresponsive within a time limit.
6. The operating system detects the failure and aborts the application.
7. The affected data are transmitted to other process in the parallel application.
8. The affected data are only used by the local process.
9. Runtime.
10. SMCV detects the failure after some time and leads to a safe stop.

Fig. 1. *Diagram of execution statuses. (a) No failure detection strategy. (b) SMCV as detection strategy.*

SMCV is a software technique and, as such, adopts a SoR centered on software. Its objective is detecting failures that affect data that are handled inside processor registers, which are the most vulnerable part of the computer due to the difficulties involved with the implementation of hardware protection. As already explained, SMCV replicates in a thread the computations carried out by each process of the parallel application. Each thread operates on a local copy of the input data that is generated so that computation can be independent from that done on the replica. Therefore, the SoR is placed around the user application and its data, and it does not include the operating system or the communications library. Even though the memory is outside the SoR of SMCV, the use of global variables is not recommended, since they are centralized points of failure. If a failure that alters global variable occurs, both redundant threads would use the wrong value and, if no other failure occurred, SMCV would detect no errors.

4.4 Multiple Failures and Vulnerabilities

Most existing proposals can detect failures if it is assumed that a single bit-flip occurs during execution, but they are not as effective for failures that affect multiple bits. Fortunately, there are only two situations in which multiple failures can be combined to cause issues. The first of these situations is when the same bit is altered in both replicas, which results in a correct comparison and the failure is not detected. The second situation is when the failure affects one of the replicas, and the result of the verification is also altered, masking the original failure. However, the likelihood that any of these combinations occurs is very low, so they can be ignored without any serious risks. All other combinations of multiple failures are detected as simple failures as soon as the first difference is detected during verification [22]. SMCV can detect any simple transient failure that causes SDC or TO, but it does not support related multiple failures.

All failure tolerance techniques have vulnerabilities, i.e., circumstances under which they cannot detect the failures that effectively affect execution. The design characteristics of a strategy and the tests to which it is subjected (usually through failure injection) must allow making those vulnerabilities explicit.

Vulnerabilities are typically associated to failures that affect the detection mechanism itself [9], and SMCV is no exception. SMCV minimizes the delay between the time data are checked and the time when the validated values are used because verification is done when the data in a message are about to be used. This reduces the likelihood of failure in the time between both events (as in [22]); once the data are in the output buffer, they are outside of the SoR. On the other hand, checking the values to be sent is a centralized point of failure. If any error is detected when checking the data after a correct execution, a false positive has occurred and a safe stop is generated when the problem was in fact introduced by the detector itself. This vulnerability can be improved by a double comparison; however, even though it is not entirely reliable, partial redundancy in general is enough to meet user requirements [9]. Similarly, if validation is correct after a faulty execution, it means that the failure remained

hidden due to a second failure occurring. As already mentioned, SMCV cannot deal with this situation, but the likelihood of this happening is extremely low [22]. Additionally, the fact that SMCV can detect as TOs other failures that would be vulnerabilities if no such mechanism were available should also be considered. For instance, if an operation code is modified in such a manner that the resulting instruction is sending a message, or if a failure occurs while the tool is running, both replicas separate their execution flows. When one of them sends a message, synchronization is not successful, and the failure is detected after a period of time longer than the one established.

5. Validating Detection Efficacy

A number of tests were carried out to validate SMCV's detection efficacy. The application used was a parallel matrix multiplication MPI application (`C=A×B`) under the Master/Worker paradigm, where the Master participates in result computation [13]. The application operates as follows:

- The Master process divides matrix A among all Worker nodes and, using the function `MPI_Scatter`, sends a piece of the matrix to each one of them, keeping a piece for itself to calculate its portion of the resulting matrix.
- The Master sends a complete copy of matrix B to each Worker using the function `MPI_Broadcast`.
- All processes compute their respective pieces of matrix C, and then send their results to the Master process using function `MPI_Gather`.
- The Master builds matrix C using the pieces sent back by the Workers and its own results.

For the validation step, the application was adapted for integration with the functionality offered by SMCV as described in [14]. To do this, the source code of the application has to be modified, with the subsequent recompilation. The experiment consisted in injecting faults in a controlled manner at several points of the application using the GDB debugging tool[1]. To do this, a breakpoint is inserted on one of the running processes, the value of a variable is modified, and execution is resumed. Thus, a bit-flip is simulated in a processor register, since data corruption manifests itself if there is an observable difference between replica memory statuses. Even though transient faults can occur at any place and time during the execution, significant points were selected for this controlled injection process, both in relation to the computation done by the Master and that done by the Workers.

[1] GDB is available at www.gnu.org/software/gdb/

```
diego@Lidi137:~/Dropbox/diego/Para trabajo de Especialización/Experimentos$ mpirun -np 5 mm-SMCV 10
PID 4583 on 0 ready for attach
PID 4586 on 3 ready for attach
PID 4584 on 1 ready for attach
PID 4587 on 4 ready for attach
Restan 10 segundos...
PID 4585 on 2 ready for attach
Restan 9 segundos...
Restan 8 segundos...
Restan 7 segundos...
Restan 6 segundos...
Restan 5 segundos...
Restan 4 segundos...
Restan 3 segundos...
Restan 2 segundos...
Restan 1 segundos...
MM-SMCV;5;10;11.065258;11.049720;0.015538
```

Fig. 2. *Output of a run with no faults. The time to attach the debugger is shown.*

```
diego@Lidi137:~/Dropbox/diego/Para trabajo de Especialización/Experimentos$ sudo gdb -q -pid=4746
Adjuntando a process 4746
Leyendo símbolos desde /home/diego/Dropbox/diego/Para trabajo de Especialización/Experimentos/mm-SMCV...hecho.
```

Fig. 3. *Example showing how to attach the debugger to inject faults.*

```
(gdb) b 121
Punto de interrupción 1 at 0x401cfc: file mm-SMCV.c, line 121.
(gdb) c
Continuando.
[Nuevo Thread 0x7f1885118700 (LWP 4813)]

Breakpoint 1, master (ptr=0x85baf0) at mm-SMCV.c:121
121             multiplicarMatricesFilCol(a, b, c, n, n/cantProc);
(gdb) p a[14]
$1 = 1
(gdb) set var a[14]=3
(gdb) p a[14]
$2 = 3
(gdb) d 1
(gdb) c
Continuando.
[Thread 0x7f1885118700 (LWP 4813) terminado]
[Inferior 1 (process 4799) exited with code 01]
```

Fig. 4. *Injection of a fault that causes FSC.*

```
diego@Lidi137:~/Dropbox/diego/Para trabajo de Especialización/Experimentos$ mpirun -np 5 mm-SMCV 10
PID 4799 on 0 ready for attach
PID 4801 on 2 ready for attach
PID 4800 on 1 ready for attach
Restan 10 segundos...
PID 4802 on 3 ready for attach
PID 4803 on 4 ready for attach
Restan 9 segundos...
Restan 8 segundos...
Restan 7 segundos...
Restan 6 segundos...
Restan 5 segundos...
Restan 4 segundos...
Restan 3 segundos...
Restan 2 segundos...
Restan 1 segundos...

SMCV_Error: Los resultados finales difieren en el Byte 40. Ejecute nuevamente la aplicación---------
----------------------
mpirun has exited due to process rank 0 with PID 4799 on
node Lidi137 exiting improperly. There are two reasons this could occur:
```

Fig. 5. *Output when a FSC occurred using SMCV as detection strategy.*

For the experiments, five processes were used (one Master and four Workers) and 10x10 square matrixes, so each of the five processes calculates two rows of matrix C. Even though this size does not really require parallel execution, it is used solely to show the consequences of failure injection and SMCV's detection capabilities. The experimental platform is an Intel Core i5-2310 2.9Ghz CPU with 6MB L3 cache memory and 8GB RAM, and the operating system is GNU/Linux Ubuntu 14.04.

Figure 2 shows a normal run of the application, with no fault injection. The initial count corresponds to the time used to attach the debugger to one of the processes, in order to simulate a fault that affects data used by that process. Figure 3 shows how the debugger is attached to perform the injection experiments.

Figure 4 shows the procedure carried out to inject a fault during the execution of the Master process in one of the first 20 elements in matrix A (those kept for local computation), after executing function `MPI_Scatter` but before the multiplication operation. This situation simulates the occurrence of a failure that corrupts a datum that is used for computing the result, but is never transmitted to other process in the application, causing FSC. Figure 5 shows the output of the application, with error detection and safe stop.

Figure 6 shows the injection of a fault during the operation of a Worker process in an element of matrix B after the execution of `MPI_Broadcast` but before the multiplication operation. This allows simulating the corruption of a datum that is part of the calculation carried out by that Worker. The results of these calculations are transmitted to the Master in the subsequent `MPI_Gather`, so the incorrect result (calculated using the altered value) is detected as TDC. Figure 7 shows the output of the application, with error detection and safe stop. Since the fault caused TDC, the output message is different from that of the previous case.

```
(gdb) b 150
Punto de interrupción 1 at 0x401e85: file mm-SMCV.c, line 150.
(gdb) c
Continuando.
[Nuevo Thread 0x7f79642e7700 (LWP 4875)]

Breakpoint 1, worker (ptr=0xc05af0) at mm-SMCV.c:150
150             multiplicarMatricesFilCol(a, b, c, n, n/cantProc);
(gdb) p b[71]
$1 = 1
(gdb) set var b[71]=8
(gdb) d 1
(gdb) c
Continuando.
[Thread 0x7f796e489700 (LWP 4862) terminado]
[Inferior 1 (process 4862) exited with code 01]
```

Fig. 6. *Injection of a fault that causes TDC.*

```
diego@Lidi137:~/Dropbox/diego/Para trabajo de Especialización/Experimentos$ mpirun -np 5 mm-SMCV 10
PID 4862 on 2 ready for attach
PID 4861 on 1 ready for attach
PID 4860 on 0 ready for attach
Restan 10 segundos...
PID 4863 on 3 ready for attach
PID 4864 on 4 ready for attach
Restan 9 segundos...
Restan 8 segundos...
Restan 7 segundos...
Restan 6 segundos...
Restan 5 segundos...
Restan 4 segundos...
Restan 3 segundos...
Restan 2 segundos...
Restan 1 segundos...
SMCV_Error: Los mensajes a enviar difieren en el byte 28. No se enviara el mensaje

               Emisor: 2        Receptor: 0    Tag: 0--------------------------------------------
mpirun has exited due to process rank 2 with PID 4862 on
node Lidi137 exiting improperly. There are two reasons this could occur:
```

Fig. 7. *Output when TDC occurred using SMCV as detection strategy.*

Figure 8 shows the injection of a fault on an element of matrix C for one of the Workers. The subsequent multiplication operation overwrites the altered value, so the failure results in a LE. Consequently, Figure 9 shows that the output is normal and correct.

Finally, Figure 10 shows the application output when a fault that causes TO has occurred; both detection and the safe stop can be seen. In this case, the failure is injected on a variable that acts as index, making one of the Worker replicas to restart its computation after it has already done part of its task. This causes a difference in time between the progress of both redundant threads, which is detected as a TO error. The ideal consequence of a failure that causes TO is that the process enters an infinite loop, but this behavior cannot be forced in the selected application with a simple failure.

It should be noted that the time at which the failure is assumed to have occurred is configurable. There is no optimal value; it depends on each particular application. To clarify this, detection through TO is based on the premise that, in an application that is run on a dedicated homogeneous system, the execution times of two replicas that carry out the same computation should be similar [14]. Therefore, a notorious difference in processing times assumes that both replicas have separated their flows due to a silent fault. Thus, TO time should be configured based on what is to be expected for the application: if this value is too high, detection latency will increase; if it is too low, a small difference in computation times will result in the detection of a false positive. In the previous test, the injected failure only causes an abnormal delay in synchronization. A short time was deliberately configured to show that the mechanism can react to this event. However, if one of the processes went into an infinite loop, SMCV would be effective in detecting an error.

```
(gdb) b 150
Punto de interrupción 1 at 0x401e85: file mm-SMCV.c, line 150.
(gdb) c
Continuando.
[Nuevo Thread 0x7fbf841b8700 (LWP 4980)]

Breakpoint 1, worker (ptr=0x1523af0) at mm-SMCV.c:150
150             multiplicarMatricesFilCol(a, b, c, n, n/cantProc);
(gdb) p c[18]
$1 = 0
(gdb) set var c[18]=7
(gdb) p c[18]
$2 = 7
(gdb) d 1
(gdb) c
Continuando.
[Thread 0x7fbf841b8700 (LWP 4980) terminado]
[Inferior 1 (process 4968) exited normally]
```

Fig. 8. *Injection of a fault that causes LE.*

```
diego@Lidi137:~/Dropbox/diego/Para trabajo de Especialización/Experimentos$ mpirun -np 5 mm-SMCV 10
PID 4966 on 0 ready for attach
Restan 10 segundos...
PID 4968 on 2 ready for attach
PID 4970 on 4 ready for attach
PID 4967 on 1 ready for attach
PID 4969 on 3 ready for attach
Restan 9 segundos...
Restan 8 segundos...
Restan 7 segundos...
Restan 6 segundos...
Restan 5 segundos...
Restan 4 segundos...
Restan 3 segundos...
Restan 2 segundos...
Restan 1 segundos...
MM-SMCV;5;10;104.154512;11.043658;93.110854
```

Fig. 9. *Output of the execution when LE occured.*

```
diego@Lidi137:~/Dropbox/diego/Para trabajo de Especialización/Experimentos$ mpirun -np 5 mm-SMCV 10
PID 5116 on 1 ready for attach
PID 5119 on 4 ready for attach
PID 5117 on 2 ready for attach
PID 5118 on 3 ready for attach
PID 5115 on 0 ready for attach
Restan 10 segundos...
Restan 9 segundos...
Restan 8 segundos...
Restan 7 segundos...
Restan 6 segundos...
Restan 5 segundos...
Restan 4 segundos...
Restan 3 segundos...
Restan 2 segundos...
Restan 1 segundos...
SMCV_Error: Timeout.    Emisor: 0        Receptor: 1    Tag: 0--------------------------------------
mpirun has exited due to process rank 0 with PID 5115 on
node Lidi137 exiting improperly. There are two reasons this could occur:
```

Fig. 10. *Output when TO occurred using SMCV as detection strategy.*

5. Conclusions and Future Work

As HPC systems are scale and the likelihood of node failures and SDC increases, the need to protect the data and obtaining availability at low cost becomes even more critical. Redundancy is a viable solution for detecting SDCs in the context of HPC. The fact that a single SDC causes deep effects on all processes that communicate, it can be concluded that protecting the applications at the level of the MPI messages is a feasible and effective method for detecting, isolating and preventing subsequent data corruption.

Based on the tests carried out, it is concluded that SMCV is capable of detecting failures that affect message contents, notifying the user and leading the application to a safe step so that the data corruption does not propagate. On the other hand, in the case of failures that affect data that are kept for local computation, and those that occur during the final phase (corresponding to the FSC fraction), are detected when comparing the results. Finally, those failures that result in considerable asymmetries in the computation times of the replicas are detected through a TO mechanism.

Our future work will include completing a transient failure-tolerant methodology that incorporates a recovery mechanism that is based on multiple incremental distributed checkpoints, so that a process can store information about the failure that occurred in another process, and thus determine if the last checkpoint is valid or if a previous one should be used for recovery [10].

References

1. Cappello, F., Geist, A., Gropp, W., Kale, S., Kramer, B., & Snir, M.: Toward exascale resilience: 2014 update. Supercomputing frontiers and innovations, 1(1) (2014).
2. Fiala, D., Mueller, F., Engelmann, C., Riesen, R., Ferreira, K., & Brightwell, R.: Detection and correction of silent data corruption for large-scale high-performance computing. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (p. 78). IEEE Computer Society Press (2012).
3. Zheng, Z., Yu, L., Tang, W., Lan, Z., Gupta, R., Desai, N., & Buettner, D.: Co-analysis of RAS log and job log on Blue Gene/P. In Parallel & Distributed Processing Symposium (IPDPS), IEEE International (pp. 840-851) (2011).
4. Borkar, S., & Chien, A.: The future of microprocessors.Communications of the ACM, 54(5), 67-77 (2011).
5. Moody, A., Bronevetsky, G., Mohror, K., & De Supinski, B. R.: Design, modeling, and evaluation of a scalable multi-level checkpointing system. In High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for (pp. 1-11). IEEE (2010).
6. Elliott, J., Hoemmen, M., & Mueller, F.: Evaluating the impact of SDC on the GMRES iterative solver. In Parallel and Distributed Processing Symposium, 2014 IEEE 28th International (pp. 1193-1202). IEEE (2014).
7. Li, D., Vetter, J. S., & Yu, W.: Classifying soft error vulnerabilities in extreme-scale scientific applications using a binary instrumentation tool. In Proceedings of

the International Conference on High Performance Computing, Networking, Storage and Analysis (p. 57). IEEE Computer Society Press (2012).
8. Snir, M., Wisniewski, R. W., Abraham, J. A., Adve, S. V., Bagchi, S., Balaji, P., ... & Van Hensbergen, E.: Addressing failures in exascale computing. International Journal of High Performance Computing Applications (2014).
9. Shye, A., Blomstedt, J., Moseley, T., Reddi, V. J., Connors, D. A.: PLR: A software approach to transient fault tolerance for multicore architectures; IEEE Transactions on Dependable and Secure Computing. 6(2), pp. 135-148 (2009).
10. Lu, G., Zheng, Z., & Chien, A.: When is multi-version checkpointing needed? In Proceedings of the 3rd Workshop on Fault-tolerance for HPC at extreme scale (pp. 49-56). ACM (2013).
11. Hari, S. K. S., Adve, S. V., & Naeimi, H.: Low-cost program-level detectors for reducing silent data corruptions. In Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on (pp. 1-12). IEEE (2012).
12. Yalcin, G., Unsal, O. S., & Cristal, A.: Fault tolerance for multi-threaded applications by leveraging hardware transactional memory. InProceedings of the ACM International Conference on Computing Frontiers (p. 4). ACM (2013).
13. Montezanti, D., Frati, F.E., Rexachs, D., Luque, E., Naiouf, M.R., De Giusti, A.: SMCV: a Methodology for Detecting Transient Faults in Multicore Clusters.; CLEI Electron. J. 15(3), pp. 1-11 (2012).
14. Montezanti, D., Rucci, E., Rexachs, D., Luque, E., Naiouf, M.R., De Giusti, A.: A tool for detecting transient faults in execution of parallel scientific applications on multicore clusters; Journal of Computer Science & Technology , 14(1), pp. 32-38 (2014).
15. Chen, Z.: Algorithm-based recovery for iterative methods without checkpointing. In Proceedings of the 20th international symposium on High performance distributed computing (pp. 73-84). ACM (2011).
16. Engelmann, C., Ong, H., & Scott, S. L.: The case for modular redundancy in large-scale high performance computing systems. In Proceedings of the IASTED International Conference (Vol. 641, p. 046) (2009).
17. Ferreira, K., Stearley, J., Laros III, J. H., Oldfield, R., Pedretti, K., Brightwell, R., ... & Arnold, D.: Evaluating the viability of process replication reliability for exascale systems. In Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (p. 44). ACM (2011).
18. Ferreira, K., Riesen, R., Oldfield, R., Stearley, J., Laros, J., Pedretti, K., & Brightwell, T.: rMPI: increasing fault resiliency in a message-passing environment. Sandia National Laboratories, Albuquerque, NM, Tech. Rep. SAND2011-2488 (2011).
19. Engelmann, C., & Böhm, S.: Redundant execution of HPC applications with MR-MPI. In Proceedings of the 10th IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN) (pp. 15-17) (2011).
20. Elliott, J., Kharbas, K., Fiala, D., Mueller, F., Ferreira, K., & Engelmann, C.: Combining partial redundancy and checkpointing for HPC. InDistributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on (pp. 615-626). IEEE (2012).
21. Ni, X., Meneses, E., Jain, N., & Kalé, L. V.: ACR: automatic checkpoint/restart for soft and hard error protection. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (p. 7). ACM (2013).
22. Reis, G. A., Chang, J., Vachharajani, N., Rangan, R., August, D. I.: SWIFT: Software Implemented Fault Tolerance. In: Proceedings of the International Symposium on Code generation and optimization, pp. 243–254. IEEE Press, Washington DC (2005).

Including accurate user estimates in HPC schedulers: an empirical analysis

NESTOR ROCCHETTI, SANTIAGO ITURRIAGA AND SERGIO NESMACHNOW

[1] Universidad de la República
Montevideo, Uruguay
{nrocchetti, siturria, sergion}@fing.edu.uy

Abstract. *This article focuses on the problem of dealing with low accuracy of job runtime estimates provided by users of high performance computing systems. The main goal of the study is to evaluate the benefits on the system utilization of providing accurate estimations, in order to motivate users to make an effort to provide better estimates. We propose the Penalty Scheduling Policy for including information about user estimates. The experimental evaluation is performed over realistic workload and scenarios, and validated by the use of a job scheduler simulator. We simulated different static and dynamic scenarios, which emulate diverse user behavior regarding the estimation of jobs runtime. Results demonstrate that the accuracy of users runtime estimates influences the waiting time of jobs. Under our proposed policy, in a scenario where users improve their estimates, waiting time of users with high accuracy can be up to* 2.43 times *lower than users with the lowest accuracy.*

Keywords: *high performance computing, scheduling, execution time estimation, quality of service.*

1. Introduction

Parallel supercomputers are high-end machines designed to support the execution of parallel jobs [1]. Nowadays, supercomputers have become a common commodity in scientific oriented companies and research institutions, especially those working on High Performance Computing (HPC). Along with the development of HPC infrastructures, the main trend has been using commercial cluster management software suites. These software suites offer a wide variety of features, which include queue management, process prioritization, and scheduling algorithms [2].

Due to the increasing usage of supercomputers, job scheduling has become a critical task, where small differences in policies can result in great changes in resource utilization, and in performance [3]. The most popular scheduling policy used in batch schedulers is first-come, first-served

(FCFS) [2]. This scheduling policy often comes in combination with a backfilling method called EASY-Backfilling. The idea of this method is to select small jobs (i.e., jobs with low number of requested cores or walltime) for execution before the time they were supposed to, whenever holes of idle resources appear [4]. Backfill systems relay on users job runtime estimates to accomplish their task.
Execution time estimation has a significant impact on how a scheduler treats different jobs, and on general performance [4]. The inaccuracy of user estimates worsens the overall performance of the parallel system [5]. For this reason, many studies have been performed in order to improve runtime estimates, to make a positive impact on both system-related and user-related performance metrics.
This article focuses on the problem of dealing with low accuracy of job runtime estimates provided by users. The main goal of the study is to evaluate the benefits of providing accurate estimation, in order to motivate users to make an effort to better estimate the system utilization.
The main contributions of this article are: i) the study of the impact of user runtime estimations in the system utilization for current HPC infrastructures; ii) the design and implementation of a novel scheduling strategy, named *Penalty Scheduling Policy* (PSP), which prioritizes jobs from users that provide good estimates on jobs runtime; and iii) the experimental evaluation of PSP using realistic workloads, on both static and dynamic scenarios, which emulate diverse user behavior regarding the estimation of jobs runtime.
We present an empirical evaluation of PSP under five scenarios that represent different user behaviors regarding the estimated runtime of jobs. For each scenario, four simulations are performed considering different workload patterns that models the real situation of our HPC infrastructure, Cluster FING. Then, we analyze the impact of their accuracy on the queuing time of their jobs.
The paper is organized as follows. Section 2 introduces some general concepts about scheduling. A review of related work is presented in Section 3. Section 4 describes the proposed PSP algorithm. Section 5 presents the workload analysis and the problem instances characteristics. Then, the experimental evaluation of PSP is presented in section 6. Finally, section 7 presents the conclusions and formulates the main lines for future work.

2. Background

This section presents a brief description of Cluster FING at Facultad de Ingeniería and the SLURM simulator [6], the tool used to perform the scheduling evaluation.

2.1 Cluster FING

Cluster description. Cluster FING [7] is the HPC infrastructure at Facultad de Ingeniería, Universidad de la República, Uruguay. It is an heterogeneous

cluster of computing resources with 1672 cores, which has been operational since 2008, with a steady growth in components. It is used mostly for the batch execution of scientific and engineering computing jobs.
Job scheduling. Cluster FING uses Maui [8] for job administration. Maui is a policy engine to manage resources (such as processors, memory, and disk) that are assigned to jobs. It also provides other features like mechanisms for resource usage optimization, monitor system performance, help diagnose problems, and general system manage. The default behavior of Maui is defined by a first-come, first-served (FCFS) batch scheduler, plus EASY Backfilling [8].
FCFS is a queue policy where the jobs are attended in the same order that they arrive: the first job to arrive is the first to get access to the requested resources. Backfilling is a policy that requires users to estimate the runtime of their jobs. Provided this information of runtime, short (runtime) jobs are allowed to execute before a larger job at front of the queue [9]. The EASY Backfilling algorithm only moves ahead jobs that do not delay the job at the head of the queue.

2.2 The SLURM workload manger

SLURM (Simple Linux Utility for Resource Management) [10] is an open-source workload manager designed for clusters running Linux. SLURM provides the basic workload manager tasks for allocating resources to users for a requested amount of time. It also provides tools for starting, executing, and monitoring jobs on a set of allocated nodes. Besides that, it manages a queue of pending work that is configured by the administrators of the application/infrastructure.
SLURM design is modular, including many optional built-in plugins. Two relevant plugins that are used in this work are *SLURM Priority Plugin API* and *SLURM Accounting Storage Plugin API*. SLURM Priority Plugin API allows computing the priority of the queued jobs in every iteration. The default configuration of this plugin is the basic implementation, which provides a basic FIFO job priority. It also comes with a multifactor job priority plugin that can be configured easily. SLURM Accounting Storage Plugin API allows the storage of accounting data collected during the execution of the scheduler, it can be configured to use a MySQL database in order to store accounting data for future processing. We use SLURM Accounting Storage Plugin API to store the accounting data in the simulations performed to evaluate the priority scheduler considering user runtime estimates proposed in this work.
In this work, we have adapted SLURM Priority Plugin API to implement our proposed priority policy. It is important to state that, in SLURM, the larger the priority number, the higher the job will be positioned in the queue, and the sooner the job will be executed.

2.3 The SLURM simulator

The SLURM simulator [6] is a job trace simulator that uses the SLURM scheduler as the simulation tool with minor SLURM code changes. The implementation of the simulator was left outside the SLURM source code; this way the simulation mode can be used with future releases of the scheduler. The simulator contains two programs, external to SLURM: *sim_mgr*, the simulation manager, which keeps control of the simulation time and *sim_lib*, the simulation library, which captures time-related calls and synchronizes with *sim_mgr* for sleep calls or getting simulation time.
A workload generator for SLURM is provided with the simulator. This workload generator, with slight changes in its source code, is used in this article to create the synthetic workloads used in the experimental evaluation of the proposed scheduler. The workload generated is based on real workload registered on Cluster FING. The hardware infrastructure used on the simulations is also based on Cluster FING (see details about the problem instances on Section 5).

3. Related work

This section describes the related work about analyzing user runtime estimates, its impact on job scheduling, and proposed techniques to improve the accuracy of the estimations.
Several relevant related works reported that user runtime estimates of jobs are usually inaccurate. For example, Cirne and Berman [1] showed that in four traces of different supercomputers, 50% to 60% of jobs made use of less than 20% of their requested time. Other features were also reported, for example the relation between failed jobs and accuracy, and between job length and accuracy.
The impact of user runtime estimates has been a matter of study in many articles. As stated by Tsafrir [5], some of the studies performed gave surprising, counterintuitive results. While some researchers found that inaccurate estimates are usually preferable over accurate ones, other studies show that performance is insensitive to accuracy of users runtime estimates [3,11 –14]. Tsafrir reported results showing that performance is affected by the quality of users runtime estimates.
The empirical study by Tang et al. [3] showed that FCFS is not sensitive to user runtime estimates. However, using accurate runtime estimates improve performance on scheduling policies that give precedence to short jobs, like Shortest Job First. It is also presented a scheme that uses historical information about the quality of estimates of both user and project scopes, to redefine the runtime estimate of a given job. The proposed adjusting scheme is transparent to users and easy to deploy.
In Iturriaga et al. [15], we studied the problem of energy consumption in heterogeneous computing scenarios proposing novel scheduling algorithms and reporting their experimental evaluation performed over realistic workloads and scenarios. We analyzed three real-world task workloads and

proposed a workload generation model considering uncertainties. We computed improvements of up to 32% in computing performance and up to 18% in energy consumption.
In this line of work, this article focuses on analyzing the impact of users improving their runtime estimates when using the proposed PSP in HPC clusters. PSP is based on lowering the priority of jobs submitted by users whose runtime estimates have been inaccurate in the past, as it is described in the following section.

4. The proposed Penalty Scheduling Policy

The penalty policy applied in PSP consists in affecting the priority of jobs according to the historical precision of runtime estimates of the users.
We define the accuracy of a users job runtime estimate as $A = \frac{t_{run}}{t_{req}}$, where t_{run} is the real runtime of the job, and t_{req} is the requested time. Accuracy can take values between 0.0 and 1.0, thus the average accuracy is also between that interval. The bigger the average accuracy, the better the user is when estimating runtime, and the PSP method will assign higher priority to the users newly submitted jobs.
To affect the priority in the PSP scheduler, the accuracy of users estimates is used. We used a dynamic update scheme for estimating the accuracy of users, by computing the average deviations (i.e., ratio) between estimated time and real execution time for the last ten completed jobs for each user.
Table 1 shows the intervals used to assign priority to jobs. The priority is a number between 1 and 5, a higher number means that the jobs is closer to the head of the queue. For example, a job whose user has an accuracy of 0.35 will have a priority of 2. This priority is first calculated when the job is submitted, and it is updated every time a new job is submitted or when releasing resources (i.e., a job ends).

Table 1. Intervals for accuracy of estimates and priorities for each tag names.

tag name	*accuracy interval*	*priority*
a1	[0.0,0.2)	1
a2	[0.2,0.4)	2
a3	[0.4,0.6)	3
a4	[0.6,0.8)	4
a5	[0.8,1.0]	5

We consider that a job runtime estimate is "good" when its accuracy is over 0.6, under that it is considered a poor quality estimate. That consideration is based on the study of workload trace at Cluster FING, in which the users with coefficient of accuracy of estimates higher than 0.6 is just 4% of the total platform users.

Algorithm 1 presents a pseudocode for the implementation of the proposed scheduler into SLURM.

Algorithm 1. PSP implementation in SLURM

```
priority_thread_tasks()
  while (true)
    waitEvent(job_completion, job_submission, time_lap, ...);
    jobs_list.computeNewPriority();
  end;
end;
scheduling_thread_tasks()
  while(true)
  //scheduling thread tasks
  end;
end;
main()
…
scheduling_thread.create();
priority_thread.create();
…
end;
```

We included our code in the Multifactor implementation of SLURM Scheduler Priority Plugin API. This priority API is used by the *Job Manager,* which is the component that accepts jobs requests and includes pending jobs in a priority ordered queue. The function computeNewPriority() called by the priority_thread communicates with that API and updates the priority of all jobs in pending state based on data retrieved from the database in which job accounting information is stored. This function is called periodically and when there is a change in a job state that may permit another job to begin execution.

5. Workload analysis and problem instances

The design of realistic problem instances is a very relevant issue when dealing with the evaluation the new approaches for scheduling and managing HPC infrastructures. We analyzed the workload of Cluster FING in order to gather real information for creating realistic instances of the scheduling problem (including workloads and user behavior when estimating jobs runtime). This section summarizes the main findings about workload analysis and users job runtime estimates and describes the problem instances generated.

5.1 Workload analysis

We analyzed the complete trace of jobs submitted to Cluster FING between April 2010 and March 2015, containing a total of 276803 jobs. As the main

results of the statistical analysis of jobs, we found that almost half (49.3%) were small jobs, with less than a minute of execution time, sequential jobs were 44.2%, and parallel jobs were 6.5%. We found a predominance of power of two number of cores requested in parallel jobs (85.1%).
We computed the average accuracy of users runtime estimates by applying the model described in the previous section. Regarding this average, we divided the users in six groups (a bigger group number means a higher accuracy of estimates): *g1*–0 to 0.05, *g2*–0.05 to 0.15, *g3*–0.15 to 0.25, *g4*–0.25 to 0.35, *g5*–0.35 to 0.60, and *g6*–0.60 to 1.0. These groups have 21%, 21%, 18%, 17%, 19%, and 4% of the 117 regular users of the cluster respectively.

5.2 Problem instances

Using the information gathered in the workload analysis, we created problem instances to evaluate the PSP scheduling algorithm under different scenarios.
The simulated infrastructure consists of 37 machines with 12 cores each (a total number of 444 cores). We also configured an execution queue that accepts serial, and parallel jobs requesting up to 16 cores, and up to ten days of execution time.
Regarding the task workload generation, we used the software included with the SLURM simulator, and customized its source code to generate specific instances for the problem to study. The changes include adding constraints on the number of cores requested and maximum requested job runtime, so the jobs generated fulfill the constraints of the execution queue configured.
Each generated workload has 1000 jobs, from 20 users. Each job demands a number of cores that is a power of two between 1 to 16, and up to 10 days of runtime execution. The distribution of the number of cores and runtime execution is representative of the workload at Cluster FING.
In order to test different accuracy of estimates situations, six scenarios simulating different user behavior were generated, the main characteristics of those scenarios are shown on Table 2. Four scenarios are *static* regarding the accuracy on runtime estimates, while the other two emulate users that learn and improve the accuracy of their execution time estimation.

Table 2. Scenarios generated to simulate different user behavior.

scenario	*type*	*accuracy*	*learning schema*
BE	static	group a1 (0.0–0.2)	none
GE	static	group a6 (0.8–1.0)	none
CF	static	groups g1 to g6	none
CFG	static	groups g1 to g6	none
DI	dynamic	incremental improving	all users
HDI	dynamic	incremental improving	half of the users

The first scenario is BE (*Bad Estimates*), in which all users have the worst level of job runtime estimate accuracy (group a1, defined in Section 4), with accuracy between 0.0 and 0.2. In the second scenario, GE (*Good Estimates*),

every user has the highest level of accuracy of job runtime estimates (group a6), with an accuracy between 0.8 and 1.0. The third scenario is CF (Cluster FING), where the users accuracy was generated so it is representative of the one accounted at Cluster FING. We divided the accuracy in the six groups, g1 to g6, defined in the previous subsection.
The fourth scenario is CFG (Cluster FING with good estimates), introducing changes in CF scenario to model a situation where almost half the user estimations are in group a6. In order to keep coherence we changed the weight of the groups as follows: a1: 12%, a2: 12%, a3: 9%, a4: 8%, a5: 10%, and a6: 49%.
The last two are dynamic scenarios, which emulate a rational behavior of users that gradually learn how to estimate job execution times. The improvement of estimations is calculated with a frequency of 5 jobs (i.e. every 5 jobs submitted by the user) as follows: if $accuracy \leq 0.5$, then it is increased by $(1-accuracy)\times 0.1$; else $accuracy$ is increased by $accuracy\times 0.1$.
In the fifth scenario, DI (Dynamic Improvement), all users improve their estimations as described. On the other hand, in the last scenario HDI (Half Dynamic Improvement), only half of the users were modeled to correctly learn how to improvement on their estimates.
The source code of the SLURM synthetic workload generator was modified in order to emulate this six different scenarios of user behavior, and generate them at once, using the exact same workload trace.

6. Experimental analysis

This section reports the experimental analysis of the proposed PSP algorithm over 24 scenarios defined from the combinations of infrastructure, workload, and estimations. All simulations were performed in a virtual machine running Ubuntu v14.04. The results of each simulation were stored in a MySQL database, a functionality provided by SLURM scheduler.
In order to get information about each class of accuracy of users job runtime estimates, we studied the average waiting time per user for the five classes. Table 3 reports, for each of the six scenarios evaluated, the average waiting time (in minutes) of each of the accuracy classes defined. We identify the empty classes with a "–".

Table 3. Average waiting time for each scenario divided by accuracy class.

scenario	*average waiting time (minutes)*				
	a1	***a2***	***a3***	***a4***	***a5***
BE	820.05	–	–	–	–
GE	–	–	–	–	796.18
CF	1093.12	637.53	519.30	450.30	–
CFG	1488.60	986.33	759.15	706.35	517.35
DI	–	1030.25	804.93	534.15	424.43
HDI	1091.58	837.51	781.38	509.70	484.20

We compared the waiting time for users that did not increase their accuracy, and the users that did. First we discuss the results of static scenarios (CF and CFG), then we continue with dynamic scenarios (DI and HDI).

CF is a static scenario in which the highest accuracy class with users is *a4* (0.6 to 0.8). This class has an average waiting time of *450.3 minutes,* which is *2.43, 1.82, and 1.77 times* lower than the average waiting time of class a1*,* BE*,* and GE respectively. Only one user was on class *a4,* and his jobs were of the highest priority on the simulation.

Scenario CFG is quite different from CF: there are 6 users on the higher class (*a5*). The average waiting time per user in a5 is *517.35 minutes*, which is *2.88, 1.59,* and *1.54 times* lower than the average waiting time of class a1, BE, and GE respectively. The waiting time in CFG is also higher than the one in CF; the reason is that more users are on class a5, and they compete with each other for computing resources.

DI and HDI are dynamic scenarios, with improvements on the accuracy of users runtime estimates. Due to the dynamism, we decided to include the average waiting time of each user in the class where he belongs *at the end of the simulation*. In DI, the highest accuracy class is *a5*, having an average waiting time of *424.23 minutes*, which is *2.43*, *1.93*, and *1.88 times* lower than the average waiting time of class a2, BE, and GE respectively. The highest accuracy class in scenario HDI is *a5*, in which the average waiting time is *484.2 minutes*, this value is *2.25*, *1.69*, and *1.64* times lower than the average waiting time of class a1, BE, and GE respectively.

Scenario DI has the lower waiting time of all 6 scenarios for its highest populated class, which is *a5*. DI also has an overall average waiting time of 698.44 minutes, lower than the overall average waiting time for HDI (740.87 minutes). In a scenario where all users improve their estimates, the waiting time improves for all users, even for the ones belonging to the higher class.

Fig. 1 summarizes the average waiting time (in minutes) for jobs submitted for the scenarios CF, CFG, DI, and HDI. The average waiting times are grouped by class, from a1 to a5. For a particular scenario, results show that the waiting time for a job significantly reduces when moving to a higher accuracy class. In a scenario where all users improve their accuracy of runtime estimates, every class experience a drop in their average waiting time, which means a general improvement in the quality of service of the HPC infrastructure. For example, class a2 in DI (the lowest class of the scenario) has lower waiting time than class a1 in all other scenarios in the figure.

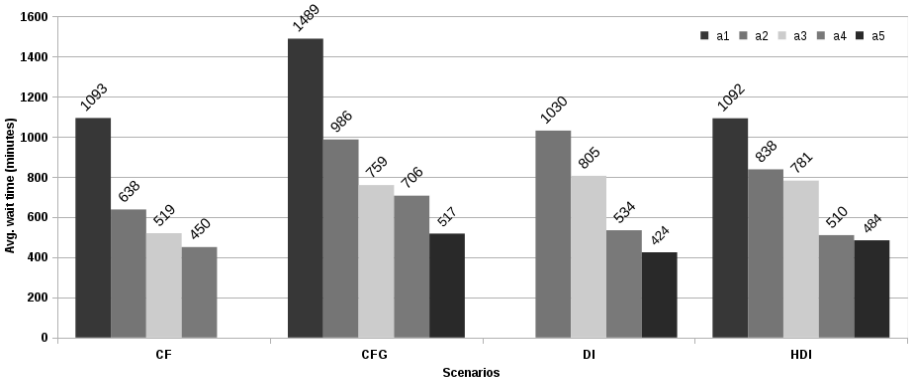


Fig. 1. *Average waiting time of each accuracy group, grouped by scenario.*

7. Conclusions and future work

This article presented a scheduling policy called *Penalty Scheduling Policy (PSP),* which focuses on the problem of dealing with low accuracy of job runtime estimates provided by users. The goals of the study are twofold. First, promoting users to accurately estimate the execution time of their jobs. Second, to evaluate the benefits of including a policy for a wise planning of computing resources by prioritizing requests from those users that provide accurate estimate for job execution times.

An experimental evaluation of the use of Penalty Scheduling Policy in a simulated computer system environment, developed using the SLURM simulator, was presented. The empirical study analyzed the PSP performance on six different scenarios regarding the user behavior when estimating job time execution. The main results of the experimental analysis show that in an environment where all users improve their estimates, every users experience a improvement on their quality of service. The proposed strategy was included in SLURM, but it can be easily included in other popular resource management systems such as Maui.

The main lines for future work are related to extend the experimental evaluation of the proposed scheduler using different workloads and statistics. The performance of PSP could be studied over synthetic workloads, created with the generator implemented by Tsafrir [5] using realistic (modal) job runtime estimates. This study will help to evaluate the real difference between the impact of bad user job runtime estimates, and good ones. We also plan to test the PSP method in a real environment, for example on Cluster FING, in order to test users acceptance of this policy.

References

1. Cirne, W., Berman, F.: A comprehensive model of the supercomputer workload. IEEE International Workshop on Workload Characterization, pp. 140–148 (2001).
2. Etsion, Y., Tsafrir, D.: A short survey of commercial cluster batch schedulers. Technical Report 2005-13. School of Computer Science and Engineering, The Hebrew University of Jerusalem (2005).
3. Tang, W., Desai, N., Buettner, D., Lan, Z.: Analyzing and adjusting user runtime estimates to improve job scheduling on the Blue Gene/P. IEEE International Symposium on Parallel & Distributed Processing, pp. 1–11 (2010).
4. Tsafrir, D., Etsion, Y., Feitelson, D.: Modeling user runtime estimates. In: 11th international conference on Job Scheduling Strategies for Parallel Processing, pp. 1–35 (2005).
5. Tsafrir, D.: Using inaccurate estimates accurately. In: 15th international conference on Job Scheduling Strategies for Parallel Processing, pp. 208-221 (2010).
6. Lucero, A.: Simulation of batch scheduling using real production-ready software tools. In: 5th Iberian Grid Infrastructure Conference, pp. 345–356 (2011).
7. Nesmachnow, S. Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República, Revista de la Asociación de Ingenieros del Uruguay 61:12–15, 2010 (text in Spanish).
8. Jackson, D., Snell, Q., Clement, M.: Core algorithms of the Maui scheduler. In: 7th international conference on Job Scheduling Strategies for Parallel Processing, pp. 87–102 (2001).
9. Mu'alem, A. W., Feitelson, D. G.: Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. IEEE Transactions on Parallel and Distributed Systems 12(6):529–543, 2001.
10. Yoo, A. B., Morris, A. J., Grondona, M.: Slurm: Simple linux utility for resource management. In: 9th international conference on Job Scheduling Strategies for Parallel Processing, pp 44–60 (2003).
11. Zotkin, D., Keleher, P. J.: Job-length estimation and performance in backfilling schedulers. In: 8th International Symposium on High Performance Distributed Computing, pp. 236–243 (1999).
12. Zhang, Y., Franke, H., Moreira, J., Sivasubramaniam, A.: Improving parallel job scheduling by combining gang scheduling and backfilling techniques. In: 14th IEEE International Parallel and Distributed Processing Symposium, pp. 133–142 (2000).
13. England, D., Weissman, J., Sadago-pan, J. : A new metric for robustness with application to job scheduling. In: 14th IEEE International Symposium on High Performance Distributed Computing, pp. 135–143 (2005).
14. Guim, F., Corbaĺan, J., Labarta, J.: Prediction f based models for evaluating backfilling scheduling policies. In: 8th IEEE International Conference on Parallel and Distributed Computing, Applications & Technologies, pp. 9–17 (2007).
15. Iturriaga, S., García, S., Nesmachnow, S.: An Empirical Study of the Robustness of Energy-Aware Schedulers for High Performance Computing Systems under Uncertainty. In High Performance Computing, pp. 143-157 (2014).

XIV

Information Technology Applied to Education Workshop

Personalized Recommendations for Ubiquitous Learning Applications

MARGARITA M. ÁLVAREZ, SILVINA I. ÚNZAGA AND ELENA B. DURÁN

Instituto de Investigación en Informática y Sistemas de Información (IIISI)
Facultad de Ciencias Exactas y Tecnologías (FCEyT)
Universidad Nacional de Santiago del Estero (UNSE)
Avenida Belgrano (S) 1912, Santiago del Estero, 4200, Argentina
{alvarez; sunzaga; eduran}@unse.edu.ar

Summary. *Ubiquitous learning is a new educational model characterized by students' mobility and by the use of wireless devices to access learning resources. Additionally, the educative processes relate to the circumstances affecting the student since it is possible to adapt the educational resources to the learning profile and context. To offer personalized services to the students in ubiquitous contexts, we have developed an Ontological Models-Driven Architecture. In this article, we introduce a strategy to be implemented within such architecture to personalize recommendations addressing students together with a case study where this strategy is applied. We conclude that both the frame provided by the Architecture and the strategy introduced here are suitable for guiding the personalization of recommendations when ubiquitous learning applications are developed.*

Keywords: *Ubiquitous learning, Personalization, Recommendations, Models-Driven Architecture, Ontologies.*

1. Introduction

Last years the various mobile devices, the technology of radio-transmission and that of sensors are developing rapidly and increasingly being used. Mobile devices such as personal digital assistants (PDA), smartphones, internet terminals and smart labels supported on this technology are shaping a new environment referred to as "ubiquitous environment" where users may easily access to wideband nets and other services [13]

Such a considerable growth of communication technology as well as the rising of new paradigms on the web with their subsequent applications on the education field allow together for a great variety of educational resources become available to students. It also permits that new and diverse formative environments can be created, learning can be customized and a set of formative activities can be completed from anywhere on any device. These developments have given place to the appearing of ubiquitous learning (u-learning) which is a new educational paradigm taking place in a ubiquitous

computing environment that let the correct content be learned in the more appropriate place at the proper moment and in the correct way.
Ubiquitous learning environments exceed the drawbacks of a traditional class or environment; extend learning making the idea of learning everywhere and at any time become true and allow people to access to better learning experiences in their daily life environments. The use of such devices as mobile phones and PDAs generates new opportunities for the students to be connected intensely. Therefore, educational contents may be accessed and interactions can be achieved wherever students need them, in different fields of their daily life with no restrictions of space or time [7].
In ubiquitous environments, providing personalized education is essential. In the context of Informatics, personalization refers to the ability of a system or application to adapt itself and meet the needs of every user by taking into account, for example, their knowledge level, learning styles, cognitive abilities, present location, motivation, interests, language preferred and so on. So doing, learning environments allow the student to have a more efficient, beneficial and successful learning experience [7]. On the other hand, recommender systems are tools that generate recommendations on a specific study object out of the preferences and opinions given by the users. The use of these systems are increasingly fashionable on the Internet since they are quite useful to evaluate and filter the great amount of information available on the web aiming at assisting the users in their information searching and retrieving processes [8]. However, most existing recommender systems, only used the information on user preferences to provide personalized services, leaving aside the contextual information [14], which is highly relevant in the ubiquitous environments.
We have been investigating how ubiquitous computing and the techniques of customization might be used to enhance learning and how to develop u-learning environments efficiently. To achieve this objective we proposed an Ontological Models-Driven Architecture to ubiquitous learning applications [5]. This architecture is made up of six software modules (*Registration Module; User Interface Module; Ubiquitous Context Acquisition Module; User Petition Analysis Module; Personalized Module; Maintenance Module of the Ontological Models*) interacting with various ontologies that represent and integrate student, domain and context information, which are referred to as ONTO-AU [1].
In this article, we introduce, in particular, a description of the strategy designed to generate personalized recommendations in the personalization module, that is part of the already mentioned architecture.
Along the following sections, we revise selected works related to the personalization of ubiquitous learning systems; describe the strategy to generate personalized recommendations and its application on a given case to eventually draw conclusions and outline further works.

2. Related Works

Some projects investigate the use of ubiquitous computing to provide new perspectives of learning. In this section, we present certain background information that suggests the application of personalization techniques in ubiquitous learning environments.

In [10] is describe a ubiquitous learning log system called SCROLL. This study primarily exploits a personalized learning and context-aware method supporting ubiquitous learning log system. Its aims lie in helping learners recall what they have logged (learned) making use of the contexts and learners' learning habits. The method contains three main measures, which are to recommend learning objects in accordance with both learners' needs and contexts, to detect their learning habits using the context history and to prompt them to review what they have learned regarding their learning habits. What's more, by monitoring learners' reaction on the recommendation or prompting, the method can improve its prediction. The system can recommend learning objects to a specific student taking into account both the context and their study needs.

Chia-Chen Chen and Tien-Chi Huang [4] propose a context-sensitive ubiquitous learning system based on the identification through radio frequency, wireless net, embedded hand device, and the database technologies to detect and examine students' learning behavior in the real world. This system provides a customized learning process that employs the curricular sequence to generate a customized learning course for each student by means of the dynamic selection of optimal didactic materials. Such an approach based upon a formative examination to collect the student's incorrect learning concepts through computerized adaptation tests drawn from a questions bank. To construct an almost optimal learning path, they use the patterns of incorrect responses to the formative examination and search for the corresponding teaching materials out of the database of the curriculum in terms of its difficulty level.

Won-Ik Park et al. [14] propose an efficient context-aware personalized technique taking into account the situation and preferences of the users in ubiquitous computing environments. The method bases on the use of an hybrid personalization technique that applies context-aware personalization and user's profile-based personalization using ontologies, rules and multi-criteria analysis. The context-aware personalization recommends a list of candidates using the preferences defined in the ontology and rules. Then, through the user's profile-based personalization, it recommends a final list of candidates using multi-criteria analysis.

Shu-Lin Wang and Chun-Yi Wu [12] propose applying context technology and recommendation algorithms to develop a ubiquitous learning system that helps students, in a lifelong learning, become aware of personalized learning objectives in a context-aware way. They take into account the learning needs of the students in a real context as well as

the differences in the personal preferences. To develop the system, they used collaborative filtering and association rules. For recommend the materials, they use an association rules mining model of aiming at enhancing students' learning motivation and efficacy.
In Ovalle et al. [11] a personalized recommendation model of educational resources is proposed for intelligent agents-based virtual adaptive courses. The functionalities of the prototype are the planning of virtual courses, online evaluation, learning objects searching and retrieval, and awareness services. The built prototype shows proactive and deliberative intelligent agents that allow for the search and recommendation of information adapted to the student's profile. They employ ontologies to describe the concepts needed to define the user's profile structure and the rules to select adaptively the contents to plan virtual course and the adaptation of the learning objects to the user´s features.

3. Strategy for the generation of personalized recommendation

Out of considering different theoretical approaches [2, 3, 6, 9] concerning the types of adaptation or personalization in e-learning, we have determined the following types of adaptation to use in this work (Fig. 1)

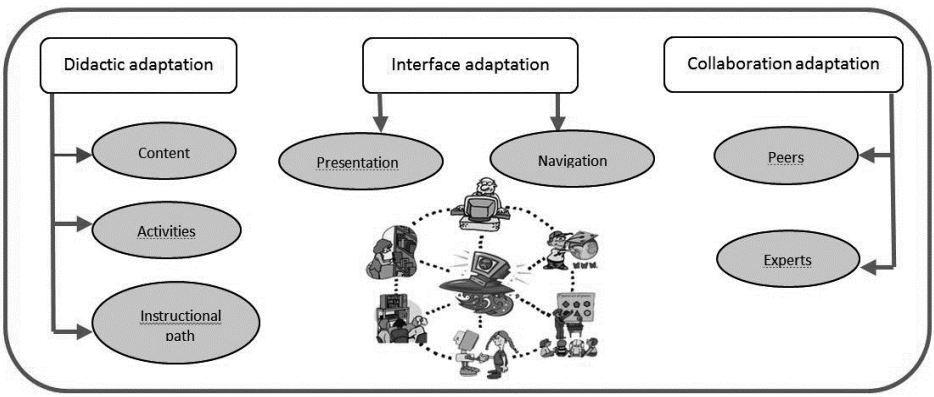


Figure 1: *Types of adaptation*

Didactic Adaptation: it consists in adapting the instructional design out of a rigorous planning in which criteria are established in relation to: the *contents* comprising the learning units of the knowledge domain, the set of *activities* and the *instructional path* to be proposed to the student according to its characteristics and contextual conditions.
Interface Adaptation: it is founded on the way the system is arranged for the students with the adaptations in: *Presentation* as to the arrangement by which the materials and activities are provided, with an appearance and interaction

in accordance with the conditions of each student; and that of *Navigation* which includes web-based link structure or inter-objects relationships for surfing on the system.

Collaborative Adaptation: it consists in adapting recommendations to the student's collaborative activities comprising mostly recommendation on collaborators according to the students' own personal and contextual characteristics and those of potential collaborators (*peers* or *experts*).

On the other hand, in e-learning, there exist several methods that can be implemented for personalization [11]. To personalize in u-learning, we adopt in this work a strategy based on a **hybrid approach.** It combines *user´s profiles-based personalization,* that allows the adaptation of the system to the relevant characteristics of the students, with the *context-aware personalization* that performs the adaptation based on those aspects characterizing a given learning situation, the environment where it occurs and the means and devices used. We also use a semantic approach by using ontologies to modelling both the profile data and the context. In both cases, the rules are the technique applied to filter the relevant aspects to be recommended. Figure 2 gives a global view of the approach used in the Personalization Module of the Ontological Models-Driven Architecture for Ubiquitous Learning Applications [5].

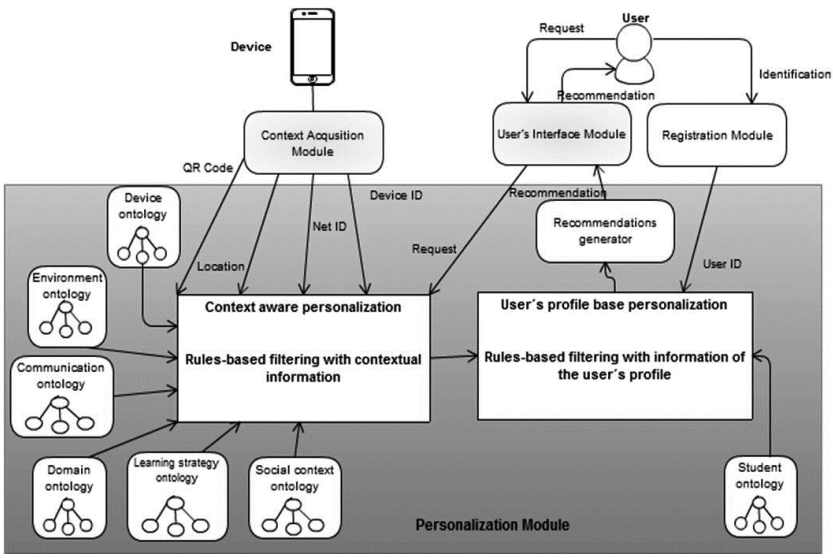


Figure 2: *A global view of the approach applied to the Personalization Module of the Ontological Models-Driven Architecture for Ubiquitous Learning Applications.*

4. Applying the strategy to a case study

Next, the use of the strategy, described above, is shown. It is use to design personalized recommendations in a ubiquitous learning application that provides five different kinds of services. They are:

1. *Service 1*: to recommend Learning Objects for interested points selected by the student.
2. *Service 2*: to recommend a learning path out of a learning objective selected by the student.
3. *Service 3*: to recommend interested points close to the student location.
4. *Service 4*: to recommend experts to advise the student when he need complete a task.
5. *Service 5*: to recommend peers to advise the student when he need complete a task.

Personalized Recommendations for Service 1. The student wants to learn about a given Interested Point (IP), scans the QR code associated to the IP using their mobile device and the system recommends appropriate Learning Objects (LO) represented in the Domain Ontology. To perform this recommendation we applied context-aware personalization to recommending a list of LO candidates related to both the IP and the technical features of the student's mobile device represented in the Device Ontology. Then, by applying the user's profile-based personalization technique, the system recommends a final list of LO candidates in accordance with the student's learning style represented by the Student's Ontology. The following example shows the rules that generate the list of LO candidates.

Example 1
(Defrule Service1)
L1 = (Select **LO** with Id_LO = Id_Interested-Point AND Format_LO = "video/mpeg" AND Size_LO < "4200" AND Type_Platform_LO = SO AND Name_Platform = "android" AND Type_Net_LO = "wifi" AND Type_Hardware_LO = "tablet 10")
L2 = (Select from L1 **LOs** with Type_Resouce_Educational_LO = "simulation" OR "diagram" OR "figure" OR "graphic" OR "slide" AND Type_interactivity_LO = "active" AND Level_Interactivity_LO = "high" OR "very high" AND Difficulty_LO = "easy")
This rule filters the LO meeting the following conditions:

A) They are related to the IP whose QR code was sensed by the student's mobile device.
B) Their technical conditions coincide with those of the student´s (10"tablet, ANDROID OS, software for .mpeg videos.
C) Their pedagogical characteristics coincide with the student's learning style (visual-active) and knowledge level (low)

Recommendations of a learning path from a learning object selected by the student. The student selects a learning object and is guided by the system on the basis on their present location, user profile and learning history (modelled on the Student Ontology) and the data collected by the sensors of their mobile device.
To perform this service we will use a heuristic algorithm to determine a personalized learning path that consists in determining the PI to be visited by

the student in looking for the best learning path. The following is an example of the rules that generate the customized learning path.

Example 2
(Defrule Service 2)
L1=(Select Theme-EA with Id-Objetive-EA = selected target)
For each issue of L1 apply
L2 = (Select Theme-D with Id_Theme_D = Id_Theme L1 AND Select PI with Id_IP_Theme_D = Id_IP)
L3 = (Select FROM L2 IP with State_IP_ES = "unrealized")
L4 = (L3 sort by:
{Calculate distance IP location in relation to student distances
Sort ascending })

This rule selects a sequence of topics (L1) from the Learning Strategy (LS) Ontology that corresponds to the objective selected by the student. Then for each topic of the sequence, the IP associated to the topic (L2) are retrieved from the Domain (D) ontology. Continuing, the rule selects from L2 those unvisited IP by the student (L3). For each IP, the rule calculates the distance in terms of the student's present position and generates a crescent ordered list of these IP.

Personalized Recommendations for Interested Points close to the student's location. The system recommends the closest to the student's location IP in terms of their learning history modelled in the Student's Ontology. What follows is an example of the rule that generates the list of IP candidates.

Example 3:
(Defrule Service 3)
L1 = (Select Interested_Point with (DIFERENCE BETWEEN (Student-Location, Interested_Point_Location) <= 10 mts.))
L2 = (Select FROM L1 IP with State_II_ES = "unrealized")
This rule filters the IP which distance to the student's location is less than 10 m (L1). From there, it selects the unvisited IP by the student (L2)

Recommendations of experts to advise the student to fulfil a task. In this service, the student requires expert advice or collaboration for the task to be carried out. The system recommends experts on the topic that are on line and physically close to the student's present location. To do so, we consult to the Social Context Ontology. An example of the rule that generates the lists of expert candidates is shown below.

Example 4:
(Defrule Service 4)
L1 = (Select Experts with Theme-SC = Theme-selected)
L2 = (Select FROM L1 Experts with State-SC = "connected")
L3 = (Select FROM L1 Experts with (DIFERENCE BETWEEN ((Student-Location, Expert-Location) <= 10 mts.))
This rule generates a first role with the Experts of the Social Context (SC) whose expertise corresponds with the topic with which the student is working. From this list those experts meeting the being on line condition are filtered to L2. Those experts in L1 that are less than 10 m distant from the student's present location are filtered to L3.

Recommendation of peers to advise the student to fulfil a task. In this service, the student requires the advice or collaboration of a peer that have completed the task that they want to complete. To do so, once the student has identified themselves and the task to be completed has been recognized by applying a user's profile-based technique and the learning history, the system recommends a list of peer candidates that are on line at that moment and/or physically close to the present student's location. An example of the rule generating the lists of peer candidates is given below.

Example 5:
(Defrule Service 5)
L1 = (Select Student with Id-Student-Course= Id-Course AND State-task-ES = "approved")
L2 = (Select FROM L1 Student with State-ES = "connected")
L3 = (Select FROM L1 Student with (DIFERENCE BETWEEN ((Student_Location, Pair_Location) <= 10 mts.))
This rule generates an initial role (L1) including the Peers belonging to the same course of the student that have passed the task the student has selected to complete. From L1 those peers connected to the system at that moment are filtered into L2. In turn, L3 is made up of the peers being less than 10m far from the student's present location.

Table 1 summarizes the kinds of adaptation applied in terms of what was defined at the beginning of this section for each of the services.

Table 1: Summary of adaptations

SERVICE	KIND OF ADAPTATION	PERSONALIZATION TECHNIQUE
1	Didactic adaptation of content Interface adaptation: presentation	Hybrid technique, that combines context-aware personalization with user´s profile-based personalization
2	Didactic adaptation of sequence	Heuristic algorithm to determine the learning path
3	Didactic adaptation of content.	Hybrid technique, that combines context-aware personalization with user´s profile-based personalization
4	Adaptation of collaboration	Hybrid technique, that combines context-aware personalization with user´s profile-based personalization
5	Adaptation of collaboration	Hybrid technique that combines context-aware personalization and collaborative filtering.

5. Conclusions and Further Works

In a ubiquitous learning environment is essential to consider the elements that involve students in the real world since these determine the content to be learned, the learning activities and its sequencing. Consequently, one of the

main problems to consider when designing applications supporting this kind of learning is the perception of the student's context.

In this article, we have shown how the ubiquitous computing technology and the customizing technologies can be useful in these new learning settings. Particularly, we have introduced a strategy to generate recommendations in a ubiquitous learning supporting system, implementable in the Personalization Module of the Ontological Models-Driven Architecture.

We conclude that concerning the strategy introduced in this work and its application to a case study, the following can be highlighted:

- The Ontological Models-based Architecture provides an appropriate framework for applying the proposed strategy;
- The strategy to generate recommendations in based on a hybrid personalization approach was suitable when applied to a particular case. It allowed to adapt contents, learning path, interface and collaboration to the students' relevant characteristics, the aspects characterizing a given learning situation, the environment where it occurs as well as the devices and means used.
- The semantic approach applied using the ontologies for modelling both profile and context data resulted appropriate since it allowed to instantiate the general aspects of each ontology in a specific case.
- The rules-based technique utilized to filter relevant aspects to recommend became efficient since it allowed for personalizing conditions of both the context and the student profile were clearly specified.

At present, we are working on the implementation of the particular case on a ubiquitous application serving the student learning along the university entrance course using the Protégé software for the development of ontologies.

References

1. Alvarez, M. M; Duran, Elena B. and Unzaga, S.I. ONTO-AU: Una ontología para sistemas de apoyo al aprendizaje ubicuo. VIII Jornadas de Ciencia y Tecnología de Facultades de Ingeniería del NOA. V, 143-149. Tucumán (2012)
2. Brusilovsky P. Methods and techniques of adaptive hypermedia. User Modeling and User Adapted Interaction, 1996, v 6, n 2-3, pp 87-129
3. Brusilovsky, P. Adaptive Hypermedia, User Modeling and User-Adapted Interaction, vol. 11, pp. 87-110, 2001.
4. Chia-Chen Chen and Tien-Chi Huang. Learning in a u-Museum: Developing a context-aware ubiquitous learning environment. Computers & Education 59, 873–883 (2012)
5. Durán, E.B.; Alvarez, M. M. and Unzaga, S.I. Ontological Model-driven Architecture for Ubiquitous Learning Applications. 7th Euro American Association on Telematic and Information Systems (EATIS 2014), Valparaiso, Chile, April 2-4, 2014. Proceedings published by ACM Digital Library within its International Conference Proceedings Series, ISBN 978-1-4503-2435-9. Available in: http://dl.acm.org/citation.cfm?id=2590776&dl=ACM&coll=DL&CFID=356022841&CFTOKEN=38496822 (2014)

6. García, V. Personalization in Adaptive E-Learning Systems. A Service Oriented Solution Approach for Multi-Purpose User Modelling Systems, Computer Science, Institute for Information Systems and Computer Media (IICM). Graz University of Technology, Graz, 2007.
7. Graf Sabine and Kinshuk. Adaptivity and Personalization in Ubiquitous Learning Systems. A. Holzinger (Ed.): USAB 2008, LNCS 5298, pp. 331–338, 2008. Springer-Verlag Berlin Heidelberg (2008)
8. Herrera-Viedma E., Porcel C. and L. Hidalgo L. Sistemas de recomendaciones: herramientas para el filtrado de información en Internet [on line]. "Hipertext.net", núm. 2, 2004. http://www.hipertext.net
9. Kobsa A., Koenemann J. and Pohl G. Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships, The Knowledge Engineering Review, vol. 16, pp. 111-155, 2001.
10. Mengmeng Li, Hiroaki Ogata, Bin Hou, Noriko Uosaki, Yoneo Yano Hwang, Tsai, and Yang. Personalization in Context-aware Ubiquitous Learning-Log System. Seventh IEEE International Conference on Wireless, Mobile and Ubiquitous Technology in Education. 978-0-7695-4662-9/12. IEEE DOI 10.1109/WMUTE.2012.14. pp 41-48 (2012)
11. Ovalle, D.; Salazar, O. & Duque, N. Modelo de Recomendación Personalizada en Cursos Virtuales basado en Computación Ubicua y Agentes Inteligentes. Información Tecnológica Vol. 25(6), 131-142 (2014)
12. Shu-Lin Wang and Chun-Yi Wu. Application of context-aware and personalized recommendation to implement an adaptive ubiquitous learning system. Expert Systems with Applications 38 10831–10838 (2011)
13. Weiser, M. The computer for the 21st century. Scientific American, 265(3), 94–104 (1991).
14. Won-Ik Park, Jong-Hyun Park, Young-Kuk Kim and Ji-Hoon Kang. An Efficient Context-Aware Personalization Technique in Ubiquitous Environments. ICUIMC '10 Proceedings of the 4th International Conference on Uniquitous Information Management and Communication. Article No. 60 ACM. New York, USA. ISBN: 978-1-60558-893-3 (2010).

XIII

Graphic Computation, Images and Visualization Workshop

AnArU, a Virtual Reality Framework for Physical Human Interactions

MATÍAS SELZER[1,2] AND MARTÍN LARREA[1,2]

[1]Departamento de Ciencias e Ingeniería de la Computación
[2]Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica
Universidad Nacional del Sur
Av. Alem 1253, Bahía Blanca
Buenos Aires, Argentina
{matias.selzer, mll}@cs.uns.edu.ar

Abstract. *Virtual Reality has become, once again, a popular and interesting topic, both as a research and commercial field. This trend has its origin in the use of mobile devices as computational core and displays for Virtual Reality. Android is one of the most used platform in this context and Unity3d is a suitable graphic engine for such platform. In order to improve the immersive experience, some electronic devices, Arduino especially, are used to gather information, such as the movement of the user's arms or legs. Although these three elements are often used in Virtual Reality, few studies use all of them in combination. Those who do, do not develop a reusable framework for their implementations. In this work we present AnArU, a framework for physical human interaction in Virtual Reality. The goal of AnArU is to allow an easy, efficient and extensible communication between electronic devices and the Virtual Reality system.*

Keywords. *Virtual Reality, Arduino, Android, Unity3d, Human Computer Interaction.*

1. Introduction

In recent years, there has been a spate of interest in Virtual Reality and its interactions, especially regarding the creation of Head Mounted Displays (HMD) using mobile devices as the computational core ([1, 2, 3, 4, 5]). We have seen mobile phones and small tablets become an ideal platform for Virtual Reality (VR). The current generation of these devices have full color displays, integrated cameras, fast processors and even dedicated 3D graphics chips.
From a Human Computer Interaction perspective, the majority of studies have focused on the visual aspects of a VR experience ([4, 5, 6]), even the interactions with the VR world are solved through visual elements. For instance, in [5] when users want to touch a virtual button, they must first look at it inside the HMD and then click a physical fix positioned one. No matter

where users are looking, they always used the same physical button. This, of course, decreases the immersive experience resulting in unpleasant results.
The use of Game Engines for the creation of VR content has been extensively studied in the past years ([7, 8, 9]). They provide developers a fast way of develop virtual content without the necessity of program directly in low-level languages. They also allow the application to process input from many different sources, including keyboards, cameras and microphones. For this project we decided to use Unity3d, not only for the creation of virtual content, but also because it has a plugin architecture that allows developers to extend the core functionality. Furthermore, Unity3d allows the exportation to many platforms, including Android.
There have been many publications addressing the problem of creating a more natural interaction between the user and the virtual environment, most of them include electronic devices ([15, 16]). However, each of these investigations has solved one particular problem and none of them has considered the creation of a practical and extensible framework for the communication of these electronic devices to the VR system.
The aim of this paper is to present a framework called AnArU (Android, Arduino, Unity3d) which allows an easy, efficient and extensible communication between electronic devices and the VR system by combining Android, Arduino and Unity3d.
In the next section we provide a brief introduction to the background concepts related to the AnArU Framework, and then discuss some relevant work done in this topic. Next we describe the AnArU Framework and its architecture, follow by the case study used to test it. Finally, we conclude with some remarks on the framework and directions for future research.

2. Background

In this section, we provide a brief introduction to the three main component of the AnArU Framework: Android, Arduino and Unity3d.

2.1. Android

Android[1] is a mobile operating system (OS) based on the Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers.
Its source code is released by Google under open source licenses. This has encouraged a large community of developers and enthusiasts to use the open-source code as a foundation for community-driven projects, which add new features for advanced users. What is more, that brings the possibility of installing Android to devices with other operating systems.

[1] www.android.com

Taking advantage of these benefits, in a similar approach as [2, 3, 5], we developed a Head Mounted Display by using the power of a smartphone running Android.

2.2 Unity3d

Unity3d[2] is a cross-platform game engine developed by Unity Technologies and it is widely used to develop video games for PC, consoles, mobile devices and websites. Unity3d was developed with an emphasis on portability, thus it is capable of porting to Windows, Xbox 360, Mac, Android and iOS.
We chose Unity3d for AnArU Framework because it is available for free, works very well with Android and it is easy to use.

2.3 Arduino

Arduino is an open-source computer hardware and Software Company[3] that designs and manufactures microcontroller-based kits for building digital devices and interactive objects that can sense and control the physical world. Their products are commonly known as Arduino and they are available commercially in preassembled form. The hardware design specifications are openly available, allowing the Arduino boards to be manufactured by anyone. It is estimated that in mid-2011 more than 300,000 official Arduinos were commercially produced and in 2013 more than 700,000 official boards were in users' hands.
We decided to use Arduino UNO for this project because it is one of the most popular, powerful and cheap microcontrollers nowadays, and also there have been many investigations which use it to create intuitive Human Computer Interaction devices ([11, 15]).

3. Previous Work

The combination of Android, Arduino and Unity3d as a platform for VR is not very common. There are very few and recent works published about the three of them associated.
In many investigations, virtual worlds are created in the context of cultural heritage; that is the case of [10], which works with archaeological sites. The virtual constructions are done using Unity3d and the platform for visualization is Android. An Arduino is used to gather information such as orientation of the viewer, physical location, tilt, pan and other movements of the tablet.

[2] www.unity3d.com
[3] www.arduino.cc

Lyons et al. ([11]) developed Loupe, a handheld near-eye display. Although it is not a HMD, it is very similar. In this case Unity3d is used for the Loupe's GUI, Android is the computational core and Arduino provided sensor information.
In contrast to what happens in VR, there are several investigations about Augmented Reality (AR) using Android, Arduino and Unity3d. AR is defined as a live direct or indirect view of a physical, real-world environment whose elements are augmented (or enhanced) by computer-generated sensory input, such as sound, video or graphics ([17]).
We can found several research papers that used this set of elements ([12, 13, 14]) in AR, but all these works, as well as the ones about VR, developed a solution for just a particularly problem. None of them thought about implementing a reusable framework.

4. AnArU Framework

AnArU consists of three main modules: a Unity3d Module running on an Android device; an Arduino Uno Module, responsible for controlling any electronic device attached to it; and a Java Plugin responsible for the communication between the other two. An overview of the framework is shown in Figure 1.

4.1. Arduino Module

The main purpose of the Arduino Module is to obtain information from any connected electronic device and send it to the rest of the system via a Bluetooth communication. Thus, those electronic devices can be used as interactive devices in the Virtual Reality system. The essential components of this module are an Arduino Uno board and a Bluetooth shield, responsible for the communication between the Arduino Module and the rest of the system.
The application running on Arduino is in charge of two simple tasks: waiting for the arrival of any message, and sending new messages whenever necessary. Thus, a new level of abstraction is introduced because any user can communicate by using such interface regardless of the connected devices. The communication protocol will be explained later on this section.
Once users know how to communicate, they can connect their particular devices or peripherals to the Arduino Module and start sending messages to the application running on the other modules of the framework. Furthermore, they can prepare the system to do specific tasks when any special message is received.

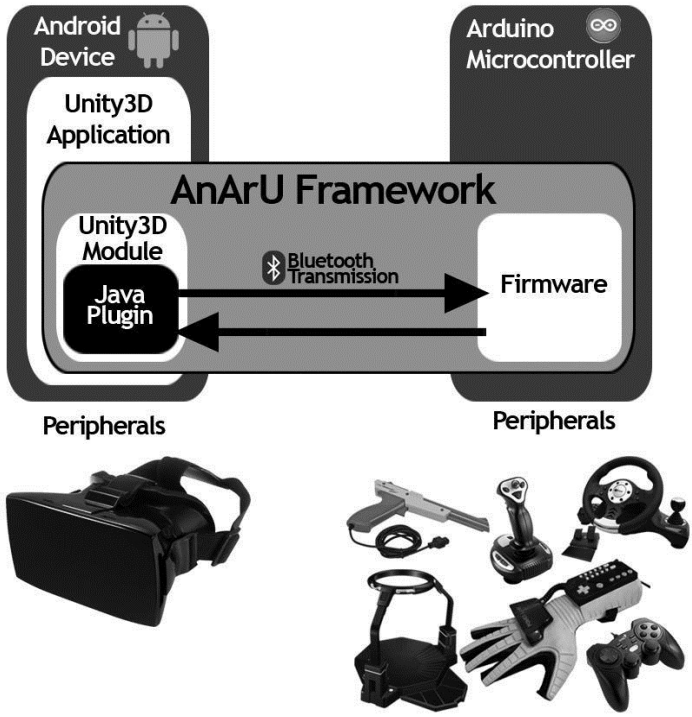


Fig. 1. *AnArU Framework overview*

4.2 Android Java Plugin

A Java Plugin is another part of AnArU Framework. It establishes a bridge between the Arduino Module and the Unity3d Module. Besides, as this plugin runs on Android devices, it has access to native Android libraries and properties, such as Bluetooth or gyroscope information, which are not available for a normal Java plugin.

The Plugin first task is connecting to the Arduino Module by using the Bluetooth libraries. To accomplish this, the Arduino Bluetooth shield MAC is required. Thus, once the connection is established, a thread is executed in order to wait for any received message. A function is provided to send messages from and to the Arduino Module.

As any electronic device works with many different values, a protocol was implemented in order to get a generic message format. No matter what device is connected to the Arduino, the user has to package the respective values into a single string of text, surrounded by special characters. Then, when needed, this string can be sent. On the other side of the framework, the Java Plugin automatically gets any new message but, in order to save memory, only keeps a backup of the last message received. Note that the protocol is Asynchronous in order to save time. However, a retransmission system was

implemented to detect and recover broken or lost messages. On the other hand, when the Java Plugin wants to send a message to the Arduino Module, an analog communication takes place.

4.3 Unity3d Module

The Unity3d Module consists on a collection of scripts that allow the initialization and communication of the Java Plugin. Hence, users can make any application in Unity3d, communicate to the Arduino Module by using the provided methods, and do anything they need with the received values. Note that the values come in form of a string of text. However, as the user knows the specific structure of that string, it can be parsed in an easily manner and the values can be retrieved.

5. Study Case

In recent years, numerous investigations has focused on the ability of walking in Virtual Reality environments, leading to an increment of users' immersion ([18]). Our current investigation involved building an omnidirectional walking platform controlled by Arduino which communicates to a VR application running on an Android mobile device, by using AnArU Framework.
The platform consists of a circular wood base and iron pipes with a ring in the middle in order to hold the users inside of it. In addition, elastic ropes are used to hold users exactly in the middle of the platform. They are also equipped with a pair of rollers so they can walk freely inside the ring while maintaining their position.
In order to sense users' movements, an Arduino UNO Microcontroller and a gyroscope are attached to one of their legs. Thus, by measuring the leg angle respecting to the vertical, the system knows whether users are walking or not, and the corresponding direction. Users are also wearing a Head Mounted Display running a Unity3d application which recreates a complete model of them in a virtual environment. That is, a virtual representation of the user's body and environment in an appropriate scale. By using AnArU Framework, a communication between the Arduino on the leg of the user and the HMD application is performed, giving users the sensation that they are really walking inside the virtual environment, increasing the immersion level. In Figure 2 a user with all mentioned components is shown.

Fig. 2. *User on the omnidirectional platform. He is wearing a HMD running a Unity3d application on an Android tablet, and the Arduino Module is attached to his leg.*

Baraka et al. ([19]) showed that in most cases the developer has to be very engaged in the designing of the communication protocol between Android and Arduino modules. In our study, the communication between the different modules is transparent. Hence, there is no need to configure a new communication each time a new way of interaction between these technologies is done. Lai et al. ([20]) used a commercial Plugin to perform a similar connectivity. However, this Plugin is not open source and just a few aspects of the microcontroller are available.

During the first experimentations, we observed some problems when the communication went from Arduino Module to Unity Module and the other

way around, at the same time. A retransmission mechanism was implemented to solve this problem, and hence, we have not observed any more similar issues since then.
In an ideal VR system, the minimal delay should exist between users' movements and the corresponding visualization ([21]). In general, a maximum accepted delay is in the order of milliseconds. Hereby, a quantitative speed analysis to measure the communication time, based on the transmission time of a specific size message**,** was applied. As the average speed of a Bluetooth 2.0 communication is 3Mbit/sec, we tested the communication time of a specific message containing three float values corresponding to the values provided by the gyroscope. For this message, with a size of 9 Bytes, after a few tests, we obtained an overall transmission time of 0.04ms, compared to an ideal transmission time of 0.024ms. In case longer messages are needed, we tested the communication time of messages with a size of 23 Bytes. This experiment showed an overall transmission time of 0.07ms. Hence, these results suggests that the communication time of AnArU Framework is fast enough to fulfill its objectives.

6. Conclusions

Prior work has documented some interactivity between Arduino, Android and Unity3d technologies. However, none of them define a transparent and extensible method of communicating these technologies. In this study we created and tested a framework capable of interconnecting these technologies in a straightforward way. Furthermore, we found that the communication is fast enough to satisfy the necessities of VR interactions.
AnArU Framework can be used for any VR interaction, helping developers to save time as they would not need to design any new communication protocol. However, some limitations are worth noting. Although Arduino UNO is a powerful tool for prototyping, it has little memory and computational power, so that, in some cases, it would not be functional.
Future work should therefore consider using a different microcontroller in case that more memory or computational power is needed. Other ways of communication should be considered too. USB or WiFi connectivity may be better in other contexts.

References

1. Amer, A., & Peralez, P. (2014). Affordable altered perspectives: Making augmented and virtual reality technology accessible. In *Global Humanitarian Technology Conference (GHTC), 2014 IEEE* (pp. 603-608). IEEE.
2. Olson, J. L., Krum, D. M., Suma, E., & Bolas, M. (2011, March). A design for a smartphone-based head mounted display. In *Virtual Reality Conference (VR), 2011 IEEE* (pp. 233-234). IEEE.

3. Petry, B., & Huber, J. (2015, March). Towards effective interaction with omnidirectional videos using immersive virtual reality headsets. In *Proceedings of the 6th Augmented Human International Conference* (pp. 217-218). ACM.
4. Hürst, W., & Helder, M. (2011, November). Mobile 3D graphics and virtual reality interaction. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (p. 28). ACM.
5. Steed, A., & Julier, S. (2013, March). Design and implementation of an immersive virtual reality system based on a smartphone platform. In *3D User Interfaces (3DUI), 2013 IEEE Symposium on* (pp. 43-46). IEEE.
6. Pujol-Tost, L. (2011). Realism in Virtual Reality applications for Cultural Heritage. *International Journal of Virtual Reality*, *10*(3), 41.
7. Wang, S., Mao, Z., Zeng, C., Gong, H., Li, S., & Chen, B. (2010, June). A new method of virtual reality based on Unity3D. In *Geoinformatics, 2010 18th International Conference on* (pp. 1-5). IEEE.Shiratuddin, M. F., & Thabet, W. (2011). Utilizing a 3D game engine to develop a virtual design review system. *Journal of Information Technology in Construction–ITcon*, *16*, 39-68.
8. Shiratuddin, M. F., & Thabet, W. (2011). Utilizing a 3D game engine to develop a virtual design review system. *Journal of Information Technology in Construction–ITcon*, *16*, 39-68.
9. Jacobson, J., & Lewis, M. (2005). Game engine virtual reality with CaveUT. *Computer*, *38*(4), 79-82.
10. Davies, C. J., Miller, A., & Allison, C. (2012). Virtual Time Windows: Applying cross reality to cultural heritage. In *Proceedings of the Postgraduate Conference on the Convergence of Networking and Telecomunications*.
11. Lyons, K., Kim, S. W., Seko, S., Nguyen, D., Desjardins, A., Vidal, M., ... & Rubin, J. (2014, October). Loupe: a handheld near-eye display. In *Proceedings of the 27th annual ACM symposium on User interface software and technology* (pp. 351-354). ACM.
12. Olmedo, H., & Augusto, J. (2013). Towards the Commodification of Augmented Reality: Tools and Platforms. In *New Trends in Interaction, Virtual Reality and Modeling* (pp. 63-72). Springer London.
13. Lin, C. F., Pa, P. S., & Fuh, C. S. (2013, October). Mobile application of interactive remote toys with augmented reality. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific* (pp. 1-6). IEEE.
14. Lin, C. F., Pa, P. S., & Fuh, C. S. (2014). A MAR Game Design via a Remote Control Module. In *Augmented and Virtual Reality* (pp. 3-18). Springer International Publishing.
15. Schmidt, D., Kovacs, R., Mehta, V., Umapathi, U., Köhler, S., Cheng, L. P., & Baudisch, P. (2015, April). Level-Ups: Motorized Stilts that Simulate Stair Steps in Virtual Reality. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (pp. 2157-2160). ACM.
16. Blake, J., & Gurocak, H. B. (2009). Haptic glove with MR brakes for virtual reality. *Mechatronics, IEEE/ASME Transactions on*, *14*(5), 606-615.
17. Azuma, R. T. (1997). A survey of augmented reality. *Presence*, *6*(4), 355-385.
18. Cakmak, T., & Hager, H. (2014, July). Cyberith virtualizer: a locomotion device for virtual reality. In *ACM SIGGRAPH 2014 Emerging Technologies* (p. 6). ACM.
19. Baraka, K., Ghobril, M., Malek, S., Kanj, R., & Kayssi, A. (2013, June). Low cost arduino/android-based energy-efficient home automation system with smart task scheduling. In *Computational Intelligence, Communication Systems and Networks (CICSyN), 2013 Fifth International Conference on* (pp. 296-301). IEEE.

20. Lai, A. S., & Leung, S. Y. (2013, December). Mobile Bluetooth-Based Game Development Using Arduino on Android Platform. In *Applied Mechanics and Materials* (Vol. 427, pp. 2192-2196).
21. Earnshaw, R. A. (Ed.). (2014). *Virtual reality systems*. Academic press.

A Serious Game based on Crowdsourcing

NICOLÁS JOFRÉ, GRACIELA RODRIGUEZ, YOSELIE ALVARADO, JACQUELINE FERNÁNDEZ AND ROBERTO GUERRERO

Laboratorio de Computación Gráfica (LCG) - Universidad Nacional de San Luis,
Ejército de los Andes, 950
Tel: 02664 420823, San Luis, Argentina
{npasinetti, gbrodriguez, ymalvarado, jmfer, rag}@unsl.edu.ar

Abstract. *Nowadays serious games topic is one of the biggest existing industries and it is still growing steadily in many sectors. As a major subset of serious games, designing and developing virtual reality applications to support education or promote social behavior has become a promising frontier, because games technology is inexpensive, widely available, fun and entertaining for people of all ages, with several health conditions and different sensory, motor, and cognitive capabilities.*

In this paper, we provide an overview about a serious game with a perspective of virtual reality for social behavior. The work uses a serious game in an immersive learning environment for recycling learning. In order to improve the user experience the game was developed to work in a cave-like immersive environment, with natural interaction selective alternative.

The game includes static and dynamic 3D environments, allowing sharing the experience of scenario navigation among users, even geographically distributed.

Keywords: *Serious Games, Virtual Reality (VR), CAVE, Computer.*

1. Introduction

For some time now, virtual reality has allowed the generation of interaction environments that facilitate new contexts of exchange and communication of information. More specifically, the employment of Virtual Reality is a natural idea to improve the impression of living in a simulated reality, so this tool is largely used in many areas such as medicine, industries, education and entertainment [1].

In entertainment industry, in particular, the creation of computer games using different technologies, rules and goals among others, has grown considerably. Today, playing computer games has become a popular activity for people of different cultures and ages. This habit motivated game developers, educators and domain experts to create other kind of applications for computer game technologies [2, 3].

These new applications which aim to address a specific problem or to teach a certain skill are called *Serious Games* and mainly relates to interactive computer-based game software that intentionally produces games outside of entertainment, i.e. games with serious purposes [3].

A serious game is designed based on different educational, training, informational and learning motivations. Specifically, game-based learning involving educational, cognitive and affective aspects induces learners/gamers to higher motivation and enhance their learning success. While there are inherent tensions between contemporary youth culture and traditional education, researches show that new learning game developments promise to help shortcutting the bridge of that growing generational divide. Besides as another positive effect of learning games is to allow the learning of knowledge and problem solving skills with better performance and long-lasting attributes [4]. At the moment, serious games have allowed to solve lot of problems from technological, medical, educational and environmental areas.

One of current problems in the world is the increasing recognition of the need to sustain an ecologically-balanced environment. A helping action to this problem is to reduce and avoid negative impacts of waste on the environment, being the diversion of biodegradable waste from landfills an important contribution to limiting greenhouse as emissions. In this context, serious games of learning using virtual reality are tools that add entertainment to teaching and training for adoption of sustainable development practices [5, 6].

Adopting these practices begins on bringing about behavior change. The designing process to enhance behaviors strongly dependent of intention, i.e. the commitment to a certain action. Many times are deliberate acts based on the beliefs of the individual and the norms imposed by society [7]. When an individual is positively predisposed toward a particular behavior, and additionally perceives support for this from people around them, then it will form a positive behavioral intention towards that behavior [8]. Such collective behavior is needed on issues such as recycling, i.e. a model of social behavior to enable this kind of "contagion" in social and sustainable problems such as waste recycling. In business world there is a new web-based model that harnesses the creative solutions of a distributed network of individuals through what amounts to an open call for proposals. This model is called *crowdsourcing* and its name is formed from two words, *crowd*, making reference to the people who participate in the initiatives, and the word *sourcing*, which refers to a number of procurement practices aimed at finding, evaluating, and engaging suppliers of goods and services. Literally, *crowdsourcing* means to outsource an activity to the crowd and for that it quickly began to be used in other areas such as entertainment, sociology, psychology and others [9, 10].
Particularly, this property of outsourcing it to an undefined (and generally large) network of people in the form of an open call allow to developers and

researchers of serious games to use it in games that allow solve problems either collectively or competitively [11].
This work presents *Recycle Now!*, a serious-game-based virtual reality for enhancing recycling behaviors and environment awareness. Essentially, the idea is to develop a game for motivate and teach the basic principles of recycling and training about different types of recycling using the crowdsourcing concept to create a collective behavior.

2. The developed Serious Game concepts

Currently, even though in world there are many campaigns on recycling (*Wecycle*, *Plastic Hero*, among others) still exists a lack of awareness among people. Some people raise than this activity needs an extra effort since separating the garbage of their homes and putting it on their home's trash container it is a hard work, besides they don't want to dedicate more place to garbage in their homes. Certainly there are major problems like climate change, lack of protection of wildlife, landfills among others, which require the support of the whole society. Therefore people should adopt recycling good habits firstly at home and then apply them in public places [12, 13].

Finding a way to make from recycling a daily activity is not easy but as it was mentioned, serious games are an interesting tool to making recycling a fun and natural activity. Thus, a person could learn at his home the recycling's basic principles and then out into the world and unconsciously apply it. This behavior implies analyze several theoretical aspects which will be used for the development of this game.

2.1 Learning

Some time ago, new technologies have been incorporated to education as learning tools, particularly, people are finding ease increasingly in learning games environments. This game is presented as a game where the user can collect, identify and place trash. At the same time, other players may be doing the same task and also correct each other in order to earn more points. Apparently, the game can be categorized as competitive, but the learning is not; it is expected that at the end of every play each player ends up learning a little more about recycling practices [14].

Therefore, this game is based in a learning method called *Cooperative Learning*; which is an instructional approach in which learners work together in small groups to achieve shared learning goals. This approach invites group members to reach outcomes by setting and working towards a common goal, putting emphasis on cooperative evaluation of these outcomes. While learners are all on equal footing, great emphasis is placed on the responsibility of individuals [15]. Accordingly, the players of this serious game have one goal in common (recycle) and also each is responsible for the

moves they perform. Finally, players are somehow cooperating with each other to adopt good practices for recycling.

2.2 Crowdsourcing

Crowdsourcing is evolving as a distributed problem-solving and business production model in recent years. In crowdsourcing paradigm, tasks are distributed to networked people to be completed such that cost and time can be greatly reduced.

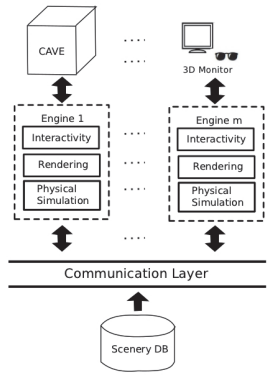
Nowadays, many tasks that are trivial for humans continue to challenge even the most sophisticated computer programs, such as image annotation. These tasks cannot be computerized [16]. Current research in crowdsourcing often focuses on micro-tasking, however, participants are people with rich capabilities including learning and collaboration, suggesting the need for more nuanced approaches that place special emphasis on participants. There are no recent studies using learning among these approaches, so crowdsourcing efforts based on learning through a game is a good objective [17, 18].
As it was mentioned the *Cooperative Learning* model allows users to achieve shared learning using interaction between them. Specific situations where the user will be benefited from this concept are related to game dynamics. Some examples where the user learns by interacting crowdsourcing are: when it is a witness of another player's mistake, it corrects another player, or it is corrected by another player on its own mistakes. Clearly, all learning situations depend on the existence of the cross-interaction via network.
Recycle Now! is a serious game that combine crowdsourcing mechanism for learning purposes. In the following subsection we describe a framework that supports the learning mechanism with the mentioned crowdsourcing concept.

2.3 Platform

The game was developed to work on a computing platform for immersive collaborative 3D virtual world visualization (See Figure 1 (a)). It allows the use of geographically distributed VR media, called a multi-VRmedia. Remote players can interact into a 3D scenario through different multi-VRmedia. During navigation, players can exchange information in order to cooperatively solve the observed problems [19].
Each multi-VRmedia comprises the hardware and software necessary to gather the information obtained during interaction between user and game: via keyboard, mouse, data gloves, sound system, 3D active glasses, screen/projection surfaces, projectors, among others.
A computing platform includes a hardware architecture and a software framework (including application frameworks), where the combination allows software, particularly application software, to run. Typical platforms include a computer's architecture, operating system, programming languages and related user interface (run-time system libraries or graphical user interface). A system to visualize scenarios in a multi-virtual reality media environment has been defined.

Such system will provide the necessary structure for attributes definition, rendering and collaborative multi-visualizations, as well as the needed interactive resources. Figure 1 (b) shows an overview of the work.
The system uses a client/server architecture similarly to a traditional network game, this allows user's interaction with others gamers distributed geographically.

(a) Immersive environment



(b) Computing Platform

Fig. 1: *Virtual Reality System.*

3. GamePlay

According to Prensky, "*Gameplay is all the activities and strategies game designers employ to get and keep the player engaged and motivated to complete each level and an entire game*" [20]. This serious game offers a good gameplay combining a familiar and innovative **game design** and an educational and motivating **players experience**.

3.1 Game Design

Serious games must fulfill all the necessary and sufficient conditions to become a game. There are several aspects used in design of traditional games which pretends to provide a good gaming experience to players. The most important aspects of the game are explained below.

First-Person game. Player perspective is one of the important design choices made when creating a digital game. Traditional camera options include audience, isometric, bird's eye, trailing camera, third person and first person [21]. These views support distinctive experiences of immersion for video game play and different perception of the game space. First person (FP) allows the player to perceive the game through the eyes of the character, observing the world around them up close, giving a clear view of the scenario in front of them. As mentioned, this game is a game-based virtual reality and

one of the pillars of VR is immersion. Therefore this game features a FP view to increase the feeling of being immersed inside game, for example, when the player needs to put waste in the correct place using haptic devices, requires a vision close to the container. Thus this FP feature stablishes a "player-character" relationship to provide the most immersive feel for the player and improving learning abilities stimulating his visual and auditory capabilities.

Multiplayer game. In games world, most players want to share the same experience with other players, i.e. seeing and feeling like they are playing the same game (being connected). A game meeting this feature is said to be a multiplayer game and *Recycle Now!* is not the exception [22]. The mainly reason making this multiplayer game is because the game's theme required the use of crowdsourcing's concept, namely allowing another players to give a solution to an specific problem of the game. The mentioned game platform was built around a client-server architecture where each client connects to a single server resulting in the illusion of a shared experience but really each player is playing a separate game, each with its own game state. This feature allows meet an expected functionality for players and moreover a collective solution.

Affordance. Game engine has increasingly developed to include aspects like emotion, joy of use, user experience, or motivation. Therefore a concept has been sporadically applied to games for several years, this concept is known as "*affordance*" and refers to perception mapping what the external world affords the perceiver [23]. However games researchers try to explain how people discover the functionality of features in game applications. Particularly, virtual reality games developers have focused primarily on what players perceive they can do, as opposed to what players can actually do in an interactive virtual environment [24].
In this case in addition to offering a game with a serious purpose we wanted to give players a way to play to help their perception, granting them different means to traditional desktop game such as mouse and keyboard. It is for this reason that the game platform built allows some players to make use of tools such as 3D glasses, data gloves, body sensors among others, and so to increase its capacity and reduce perceptual cognitive effort. This way affordance is a powerful tool for understanding the relationship between player and system.

Environment. To achieve the proposed goals in the game, a scenario was developed as environment. The game was situated at the central square of a city, where the user can navigate for the square and streets around of it. Stage was set with ambient sounds, inanimated objects (benches, lights, trees, garbage, trash cans) and animated objects (people) allowing social interaction, making the obtained learning through the game were similar to real world. Particularly, people on stage are avatars (See Figure 2).

Fig. 2: *Scenario.*

Physical realism. Making an object looks real to the user more than visual realism is required. In a realistic game, objects also should behave as in reality including their physical characteristics, so it is necessary to simulate aspects like gravity and collision avoiding crossing between solid objects, achieving then a behavior like the impact of a ball on the ground and soft objects deformations, among others.
In addition user's movement basic physics, the game requires to perform collision detection for activities such as grasping and releasing waste. *Collision Shapes* are used for that like envelopes that allow sensing the world surrounding the object and making possible to visualize collisions between objects.

Real time. The virtual reality enables users to simultaneously experience real-time and interactive simulations. Real-time factors considered here are diverse: visualization, realistic audio, media interaction and user response times.
As was described in platform's section, real time is enhanced by haptic devices and others virtual reality devices. Because grasping and releasing waste is a key type interaction, a real-time hand gesture interface to manipulate objects in the game has been implemented. As an example, a user may see a simulated virtual representation of themselves (an Avatar) or a part of themselves (hands) that reflects real-time movements (e.g., lifting a finger, close the hand, etc.).
The developed game has the ability to interact with users in real time and receive feedback. The game allows users from around the world communicating, playing, learning and networking in real time.

3.2 Player's Experience

A good gaming experience involves keeping players motivated. To achieve this motivation both the game and opposing players must be a challenge for all players. Regarding players, an experienced player will not get the expected challenge if it is playing with beginners. These possible differences

in experiences must be balanced before starting to play. Consequently *Recycle Now!* provides different tools and documentation to help players who need it and reports on all relevant regulations about game.

Rules. Before starting the game, players must join to the same play, one player will be the server that created the play and others are guess. Initially each player has a set of different garbage containers labeled: *Organic*, *Plastic*, *Metal* and *Paper*, and they must decide where will place each of them on stage. To achieve a uniform distribution, players are prohibited placing containers near each other (See Figure 3). Once containers of all players are located, they can not be relocated. After it, game starts. Players must find the trash, collect, analyze and throw it in their containers. Players compete against each other for garbage pick up quickly and increase their scores. After a certain time, the game ends and the player with the highest score wins the game. All players can see the scores of other players while they are playing.

Fig. 3: *Garbage Containers Distribution.*

Skills. The game allows to see the score of players, showing who is the winning player so far. It also offers players the opportunity to discredit other players using an ability called "*Let me check!*". All players can execute this skill a certain number of times as they see fit. When a disbelieving player (player A) wants to discredit a refuted player (player B), player A must execute "*Let me check!*" action. With this ability, player A can adjust player B garbage classification, and consequently player A score, increasing it for each right classification, and decrementing it for every wrong classification. The score of player B will be decremented only for each player A correct classification (See Figure 4). After checking up Player B collected garbage, both players return to play and collect garbage.

Fig. 4: *Let me check*

4. Conclusions and Future Works

New technology has great potential to benefit education. From this example, it should be clear how important games can be for stimulating rapid re-mixing for educational examples.
This paper involved the development of a multiplayer serious game with several components: virtual reality, learning, crowdsourcing and affordance, among others. We described a serious game that was designed to motivate players to recycle properly so they can use it on his daily life at real spaces with waste sorting such as squares, public building, etc. The game was designed to be played by any player even without experience with games giving an interactive visualization experience and multi-RV interaction.
In this work we had focused on crowdsourcing learning and how to do it more natural and intuitive to users allowing proficient user-interactivity in real-time, meaningful feedback and learning through an interface.
The effectiveness of the game as a pedagogical vehicle can be highlighted by some influencing factors: social interaction, immersion, ambient noise level, among others [25]. By the moment, some game's aspects are beta version.
Future works will be oriented to improve game engine annoyances. Some environment issues, such as hardware platform, limited the scenario's resolution of the game. This constraint often forced a trade-off in the amount and nature of special effects and the number of textures used to create the game environment.

Acknowledgments. This work is supported by the European Community, Alfa III - GAVIOTA, Contract N: EuropeAid/129-877/C/ACT/RAL-1.

References

1. Williamson Ben. Computer games, schools and young people, 2009.
2. Minhua Ma, Andreas Oikonomou, and Lakhmi C. Jain.Serious Games and Edutainment Applications. Springer Publishing Company, Incorporated, 2014.
3. Y. Cai and S.L. Goei. Simulations, Serious Games and Their Applications. Gaming Media and Social Effects.Springer, 2013.
4. Ashok Ranchhod, Clin Guru, Euripides Loukis, and RohitTrivedi. Evaluating the educational effectiveness of simulation games: A value generation model. Inf. Sci., 264:75–90, April 2014.
5. K. Katsaliaki and N. Mustafee. A survey of serious games on sustainable development. In *Simulation Conference (WSC)*, *Proceedings of the 2012 Winter*, pages 1–13, Dec 2012.
6. Pascal Lessel, Maximilian Altmeyer, and Antonio Kr¨uger. Analysis of recycling capabilities of individuals and crowds to encourage and educate people to separate their garbage playfully. In Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI ’15, pages 1095–1104, New York, NY, USA, 2015. ACM.
7. I. Ajzen and M. Fishbein. *Understanding attitudes and predicting social behaviour*. Prentice-Hall, Englewood Cliffs, New Jersey, 1980.

8. Michele Tonglet, Paul S Phillips, and Adam D Read. Using the theory of planned behaviour to investigate the determinants of recycling behaviour: a case study from brixworth, uk. *Resources*, *Conservation and Recycling*, 41(3):191–214, June 2004.
9. Daren C. Brabham. Crowdsourcing as a Model for Problem Solving. *Convergence: The International Journal of Research into New Media Technologies*, 14(1):75–90,February 2008.
10. Enrique Estellés-Arolas and Fernando González-Ladrón-De-Guevara. Towards anintegrated crowdsourcing definition. *J. Inf. Sci*., 38(2):189–200, April 2012.
11. SigalSina, Avi Rosenfeld, and Sarit Kraus. Generating content for scenario-based serious games using crowdsourcing. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. AAAI Press, 2014.
12. M. Martin, I.D. Williams, and M. Clark. Social, cultural and structural influences on household waste recycling: A case study. *Resources, Conservation and Recycling*, 48(4):357–395, 2006.
13. W. Kip Viscusi, Joel Huber, Jason Bell, and Caroline Cecot. Discontinuous behavioral responses to recycling laws and plastic water bottle deposits. Working Paper 15585, National Bureau of Economic Research, December 2009.
14. J. McGonigal. Reality Is Broken: *Why Games Make Us Better and How They Can Change the World*. Penguin Publishing Group, 2011.
15. D.F. Salisbury. *Five Technologies for Educational Change: Systems Thinking, Systems Design, Quality Science, Change Management, Instructional Technology*. Educational Technology Publications, 1996.
16. Man-Ching Yuen, I. King, and Kwong-Sak Leung. A survey of crowdsourcing systems. In *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third International Conference* on *Social Computing (SocialCom), 2011 IEEE Third International Conference* on, pages 766–773, Oct 2011.
17. M.J.-Y. Chung, M. Forbes, M. Cakmak, and R.P.N. Rao. Accelerating imitation learning through crowdsourcing. In *Robotics and Automation (ICRA), 2014 IEEE International Conference* on, pages 4777–4784, May 2014.
18. Jean-Claude Bradley, RobertJ Lancashire, AndrewSID Lang, and AntonyJWilliams. Thespectral game: leveraging open data and crowdsourcing for education. *Journal of Cheminformatics*, 1(1), 2009.
19. Y. Alvarado, N. Moyano, D. Quiroga, J. Fernández, and R. Guerrero. *Augmented Virtual Realities for Social Developments. Experiences between Europe and Latin America*, chapter A Virtual Reality Computing Platform for Real Time 3D Visualization, pages 214–231. Universidad de Belgrano, 2014.
20. Marc Prensky. The Motivation of Gameplay: or, the REAL 21st century learning revolution. *On the Horizon*, 10(1), 2002.
21. Francois Dominic Laramee. *Game Design Perspectives*. Charles River Media, Inc.,Rockland, MA, USA, 2002.
22. Yanna Vogiazou. Presence based massively multiplayer games: Exploration of a new concept. 2002.
23. Gibson J J. *"The theory of affordances," in Perceiving, Acting, and Knowing. Towards an Ecological Psychology*. Number eds Shaw R., Bransford J. Hoboken,NJ: John Wiley & Sons Inc., 1977.
24. B. Dalgarno and M. J. W. Lee. What are the learning affordances of 3-d virtual environments? *British Journal of Educational Technology*, 41(1):10–32, 2010.
25. B.P. Bergeron. *Developing Serious Games*. Charles River Media game development series. Charles River Media, 2006.

XII

Software Engineering Workshop

Enhancing a Lexicon Model by Concept Mapping

ALBERTO SEBASTIÁN[1] AND GRACIELA D.S. HADAD[1,2]

[1] Facultad de Ingeniería y Tecnología Informática, Universidad de Belgrano
[2] Escuela de Informática, Universidad Nacional del Oeste
alberto.sebastian@comunidad.ub.edu.ar, ghadad@uno.edu.ar

Abstract. *Starting with good requirements specifications is an essential key to increase the likelihood of software project success. Defects persisting in specifications will inevitable be transferred to subsequent models and software components damaging client expectations through the software product. Therefore, early verifications of requirements models should be a regular activity in a software process. Most verification techniques aim to detect defects about wrong facts, inconsistencies and to a lesser extent omissions. Models produced in a Requirements Engineering process are frequently written in natural language, thus enabling the appearance of ambiguities. This article presents a preliminary proposal for verifying a lexicon model driven by concept maps, which is mainly focused on detecting omissions and ambiguities, and it recommends corrections to the model. This proposal was applied to a model produced for a real software project, detecting a reasonable number of defects.*

Keywords: *Requirements Engineering, Requirements Verification, Completeness, Ambiguity, Concept Maps, Language Extended Lexicon.*

1. Introduction

Incompleteness is one of the concerns most difficult to detect and most harmful in Requirements Engineering, since it threatens the quality of the produced models [1, 2]. It is well known that completeness is impossible to achieve in complex problems, therefore, the goal is to reach an acceptable level of completeness [3]. In case of models written in natural language, another type of defect usually occurs: ambiguity, which leads to more than one interpretation [4]. The standard IEEE 2948 [5] points out that completeness and non-ambiguity are two of the main properties that every requirement and the entire requirements specification itself must accomplished.

Since requirements specifications are the basis for building the software, it is mandatory to carry out activities to check the quality of those specifications in well-established software processes. Verification is one of these activities, providing different techniques depending on the type of model to be checked. Requirements review is a typical technique used to verify models written in natural language [1, 6, 7, 8].

The purpose of this paper is to present a preliminary proposal for the verification of a lexicon model created as the first activity of a client-oriented Requirements Engineering process [9]. This verification process mainly identifies omissions and ambiguities by means of an intermediate artifact: concept maps [10], which are constructed from the lexical model. It also includes steps suggesting the correction of the identified defects. These concept maps allow studying concepts and their semantic by establishing relationships between concepts; this property makes easier to identify omissions and ambiguities, since both types of defects mainly obstructs the understanding of the map. This preliminary proposal was applied to a real case, allowing the identification of different types of defects and achieving a more understandable lexicon model after defect resolution.

In next section, a brief about the lexicon model, target of the present proposal, is provided, and verification techniques applied to natural language models are discussed. In section 3, concept maps are introduced, while in section 4 the designed verification process is detailed and the results of its application in a real case are exposed. Finally, conclusions about the proposal are presented, along with future steps to extend the verification process in order to allow the detection of another type of defects.

2. Verification of Requirements in Natural Language Models

In the Requirements Engineering field it is a frequent practice the use of models written in natural language, such as use cases and scenarios [9]. This practice is encouraged due to the ease of comprehension by the clients. Comprehension is further improved when those models are accompanied by glossaries, in particular, a specific glossary, called Language Extended Lexicon (LEL) [11], which describes the vocabulary used in a given application context. This model is part of a client-oriented Requirements Engineering process [9], which focuses on using the terms included in the LEL while creating other models.

Every LEL entry, named *symbol*, has a name (or names in case of synonyms), a notion (denotation of the symbol) and a behavioral response (connotation of the symbol). Both, the notion and the behavioral response, are described by means of one or more sentences driven by two principles: i) circularity principle (to maximize the use of LEL symbols while describing other LEL symbols), and ii) minimum vocabulary (to minimize the use of terms not belonging to the LEL). As a consequence of the first principle, every symbol must mention at least another symbol. This reference is implemented using a hyperlink pointing to the definition of the mentioned symbol. The minimal vocabulary principle needs the existence of a list of terms that allow describing general ideas in any domain.

LEL symbols are classified in four types: subject, object, verb and state. The type of a symbol indicates what kind of information should be located in the notion and in the behavioral response; this helps to gain homogeneous descriptions along the LEL model [11].

Fig. 1 shows two LEL symbols described for an auditing management system in a health care organization. The goal of this software system was to control the relationship among the organization and external health service providers. This LEL was created to improve the communication among physicians, service providers and the software developer team.

(a) LEL Symbol – Verb Type	**(b)** LEL Symbol – Object Type
MANAGE THE APPLIED FILTERS **Notion:** • Process by which the health provider may export the applied filters or import the applied filters. • The health provider executed it when analyzing the issued monthly billing. **Behavioral Response:** • The health provider stores the applied filters for later use or transmission.	**BROWSER OF INSURANCE HEALTH** **Notion:** • A web application that helps the health providers to manage the issued monthly billing of ICBA. **Behavioral Response:** • It allows managing the applied filters.

Fig. 1. *Examples of LEL symbols. Underlined terms represent hyperlinks to their corresponding definition. These symbols have not undergone a verification process.*

The use of natural language in requirements models introduces some problems, such as incompleteness, ambiguity and poor information structuring [3, 4, 12, 13]. An empirical study, carried out by Ben Achour et al. [14], concluded that 50% of the use cases contained ambiguities. These authors reduced the proportion of ambiguities providing style guides to describe use cases. Verifying the syntax and semantics of sentences contained in models is another approach to deal with ambiguity. Since a linguistic viewpoint, a text could be disambiguated by changing constructions, accentuation and punctuation, rewriting and adding words [15]. Therefore, Requirements Engineering should take into account linguistic aspects as part of the discipline.

Besides, studies about completeness based on statistical comparisons among LEL models created by different requirements engineers for the same organization, have shown a notorious dispersion between what different engineers believe about which symbols should be included [16]. The best LEL in that study had a completeness estimation of 59%, which is far from desirable. These results encourage the search for strategies to improve as much as possible the LEL creation heuristics. One of them is the use of a verification approach oriented to identify omissions.

Reviews are a well known and much used technique to verify models written in natural language [1, 7, 8, 17]. The review technique may be performed in different ways [6]: an ad-hoc fashion, using a checklist, driven by predefined procedures or by creating an auxiliary model to help defect detection. The last approach is not so extensively used, due to the additional task of creating another artifact [18]. However, it may promote better defect detections due to the creation task itself. This

article proposes a verification technique based on this approach.
There exists in the literature a strategy to verify the LEL model using an inspection technique driven by procedures that involve filling forms and analyzing them in order to detect defects [1]. This technique, even useful, is syntactically oriented. It detects poorly written sentences, empty components, repetitions and ambiguities. It has also the inconvenience of having a temporal cost of $O(n^2)$, being n the number of sentences, in order to detect some type of defects, such as symbol omissions.

3. Use of Concept Maps

The misunderstandings that appear in the requirements product, and finally are transferred to the software, basically come from the long distance between what the engineers think or understand the software system should provide, and what the clients think or understand they need by means of a software system. A learning activity about the clients' environment, their activities and their vocabulary is a way to deal with this problem.
Therefore, the initial activities of a Requirements Engineering process consist mainly in acquiring knowledge about the context in which the future software system will operate. These tasks have many coincidences with the Knowledge Management discipline [19]. In fact, they share several strategies and techniques. One of them is concept maps, seldom used in Requirements Engineering but extended in Knowledge Management. Concept maps organize a set of conceptual meanings by means of a graphic schema oriented to the visualization of propositions [10].
Constructing concept maps [20] is a very easy task, which includes the following steps: i) define the scope of the concept map in a restricted domain of knowledge; ii) identified the concepts in that domain, writing down them avoiding the repetition of concepts; iii) rank these concepts from the more general to the more specific ones, being the root concept (the most general) the one under study; iv) establish relations among concepts by means of linking words or phrases, such as articles, verbs, prepositions and/or conjunctions; v) identify possible cross-links, which relate concepts from different segments or domains of the concept map; and vi) review the map keeping simplicity and clarity in mind. The graphic syntax of the concept map uses ovals to hold concepts and labeled arrows to depict the links. Colors and other visual effects may be used to enhance the understanding of the concept map.
There are three distinctive features in concept map construction: hierarchical structure, selection and visual impact [21]. The first one establishes a rank of importance or inclusion where concepts more important or more inclusive should be in the upper zone of the map or at the same level. Selection establishes that a map is a synthesis containing the most significant concepts. Visual impact establishes the consistency of the map in terms of location and coloring of concepts and their relations.

4. Verification of the LEL using concept maps

The proposed process of improving the model LEL consists of two phases: i) construction of concept maps from LEL symbols, and ii) detection and correction of defects. The first phase involves the creation of a concept map for each LEL symbol by extracting the information contained in the symbol definition and its relationship with other symbols. The second phase focuses on detecting mainly omissions and ambiguities by studying each concept map and then repairing the identified defects.

4.1 Phase: Construction of Concept Maps from LEL Symbols

An adaptation of the concept mapping technique is used as a means of verification. Considering the recommendation of constructing concept maps in a bounded domain [20], one concept map is derived from each LEL symbol. The root concept of each map represents the LEL symbol under study, called RCL (*Root Concept that is an LEL symbol*). Each sentence in the notion and the behavioral response of the LEL symbol is considered a proposition. Therefore, each sentence is transformed into concepts and linking phrases in the concept map. LEL symbols mentioned in the sentences are directly considered concepts; other concepts may be identified when analyzing the sentences. Summing up, besides the root concept, each map contains two types of concepts: those that represent other LEL symbols, called SCL (*Secondary Concept that is an LEL symbol*) and those that do not represent LEL symbols, called SCnoL (*Secondary Concept that is not an LEL symbol*).
Additionally, the map is divided into two sections: the *map space*, which contains the concept map of the RCL itself, and the *mention space*, which includes secondary symbol concepts (SCL) that mention the root symbol RCL in their respective maps. Each LEL symbol concept (RCL or SCL) is drawn within an oval with its name and type (s: subject, o: object, v: verb, e: state). Ovals are painted in a different color depending on the type of concept that represents: RCL, SCL or SCnoL. Relations between concepts coming from the RCL notion are represented by solid arrows and those coming from the behavioral response are depicted by dashed arrows. The linking phrases are extracted from the symbol definition and they label the arrows. Fig. 2 shows the concept map of an LEL symbol created for the health care organization case. The definition of that LEL symbol is shown in Fig.1 (a).

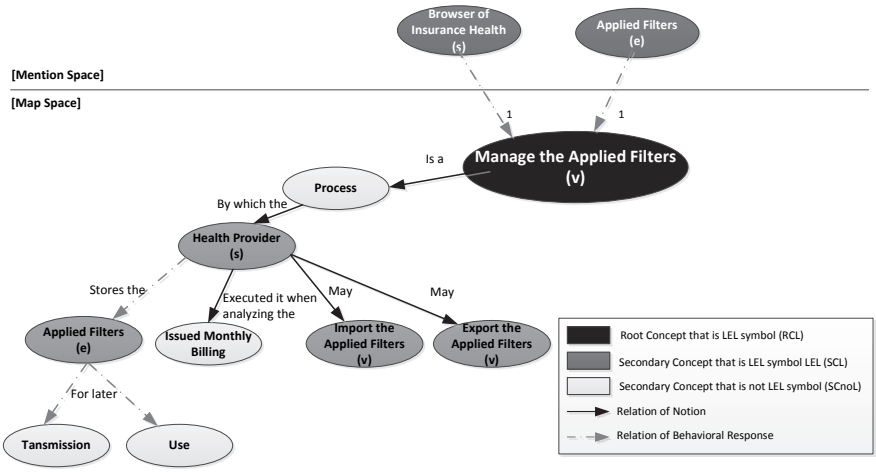


Fig. 2. *Example of a concept map representing an LEL symbol, distinguishing the* map space *and the* mention space

4.2 Phase: Defects Detection and Correction

The purpose of this phase is to detect and correct defects in the LEL model, through the analysis of the constructed concept maps. It focuses on defects of the following types: omissions, ambiguities and, in a minor degree, errors. The phase includes the analysis of concepts that are not LEL symbols and the analysis of relationships within each map and among maps.

Sub-phase: Analysis of Non LEL Symbol Concepts. SCnoL of every constructed concept maps are studied in detail. Each SCnoL concept belongs to the minimum vocabulary. The following steps are performed:

1) Create an alphabetically sorted list of the SCnoL concepts obtained from each conceptual map, recording the symbol RCL of the map where it appears. The SCnoL may be repeated if it appears in more than one map.

2) Establish the presence of synonyms among SCnoL from the created list (see Table 1) in order to use only one name for the same concept. Thus, the application of the minimum vocabulary principle is improved, reducing ambiguity in the LEL. Type of defect: Ambiguity. Correction:

a) Unify the name of the SCnoL concept and replace this name in all concept maps where the SCnoL is mentioned.

b) Replace this name in all appearance in the LEL model.

c) Reduce the minimum vocabulary by leaving only the unified name.

Table 1. Example of a partial list of SCnoL concepts, where the concept *Action* is replaced by its synonym *Process*. Both terms are SCnoL.

SCnoL	RCL whose map includes the SCnoL	Detected Synonyms of SCnoL
Action	Export the applied filters	Process
Action	Import the applied filters	Process
Process	Visualize the issued monthly billing	
Process	Filter the issued monthly billing	

3) Identify the SCnoL concepts that are synonyms of symbol concepts (RCL) in order to use only the symbol name. This helps to reduce ambiguity in the definitions of the LEL, by improving the application of the principles of circularity and minimum vocabulary. In addition, an omission of a synonym of an LEL symbol may be detected.
Type of defect: Ambiguity and possible Omission. Correction:
   a) Replace each occurrence of the SCnoL by the name of the corresponding LEL symbol in the concept maps and identify each replacement as an SCL concept.
   b) Replace in the LEL each occurrence of the term SCnoL by its corresponding symbol name, including the hyperlink to the symbol definition.
   c) Delete the SCnoL from the minimum vocabulary.
   d) Elicit information to establish if the SCnoL name is also used in the application context. In that case, add the name as a synonym of the corresponding LEL symbol.

4) Calculate the frequency of occurrence of each remaining SCnoL in all the concept maps.

5) Perform a Pareto analysis on the SCnoL concept list, considering that 20% of the SCnoL concepts that have the higher frequency will constitute a set of *candidate* concepts to become LEL symbols (see Table 2).

Table 2. Example of Pareto analysis over SCnoL concepts

SCnoL	Frequency	% Frequency	Cumulative Frequency	
Process	8	14%	14%	} *Candidate Symbols*
Medium	4	7%	21%	
File	3	5%	26%	
Patient	3	5%	32%	
Billing Sector	2	4%	35%	

6) Analyze the meaning of each *candidate* SCnoL to establish if it provides new semantics that is characteristic of the application context in order to include it as an LEL symbol. This allows the detection of missing symbols in the LEL. Type of defect: Omission. Correction:
   a) Elicit information to establish if the SCnoL is used in the application context.
   b) In this case, add the SCnoL concept as a new symbol with its definition in the LEL.

c) Include the hyperlink to the definition of the new symbol in each mention of the SCnoL concept in other LEL symbols.
d) Construct a concept map for the SCnoL in order to later analyze its relationships.
e) Mark the SCnoL concept as an SCL concept in the other concept maps that mention it.
f) Delete the SCnoL from the minimum vocabulary.

Sub-phase: Analysis of Relationships in Concept Maps. The analysis of the relationships of each map allows finding problems in the propositional semantics of the notion or the behavioral response of the corresponding LEL symbol. Analyzing relationships between concept maps allows finding violations to the circularity principle and missing information in notion or behavioral response of LEL symbols. The following steps are performed:

1) Detect unconnected sub-graphs in a concept map. Omitted relationships in the notion or behavioral response of the symbol are identified. This correction allows a better understanding of the symbol (see Fig.3). Type of defect: Omission. Correction:
a) Establish a relationship between the RCL symbol and the unconnected concept, being either SCL or SCnoL. If necessary, rewrite the relationships derived from the SCL or SCnoL.
b) Rewrite the sentence in the notion or behavioral response of the symbol RCL in the LEL.
c)

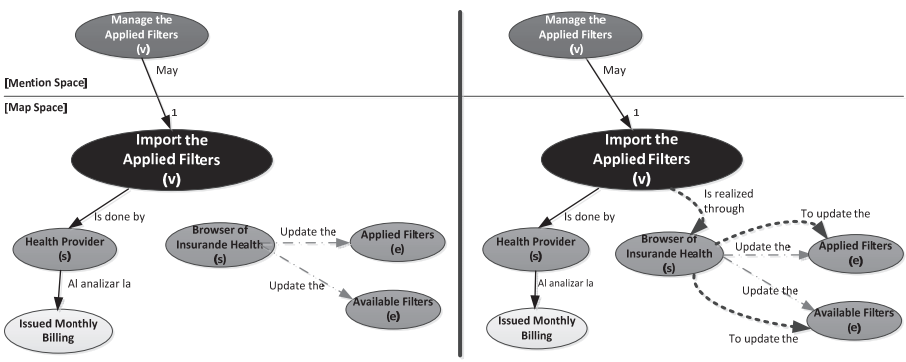


Fig. 3. *Example of an unconnected sub-graph. The disconnection of the sub-graph is observed in the left concept map. On the right, the correction is presented using descending dotted curved arrows.*

2) Detect the absence of relationships of notion or behavioral response of the RCL with any SCL in the *map space*. This allows detection of LEL symbols without references to any other symbols in its definition, violating the principle of circularity. Adding an omitted relation improves the understanding of the LEL symbol definition.

Eventually, it may be established that the symbol is not relevant and should be deleted.

a) Search the use of every SCnoL of the RCL concept map under study in other concept maps, and analyze if those concept maps have any semantic relationship with the RCL.

2.1) Case where a semantic relationship between the two concept maps can be established through the SCnoL. Type of defect: Omission. Correction:

b) Add the SCL or RCL symbol detected in other concept map as a new SCL in the *map space* of the concept map under study. Two alternatives may occur from here:

c) Create an indirect relationship in the RCL concept map under study between the existent ScnoL and the new added SCL; or

d) Copy the existent relationship in the other concept map to the RCL concept map under study linking the existent ScnoL and the new added SCL.

e) In both alternatives, add the new proposition (relationship plus added SCL concept) as a new sentence or part of an existent sentence in the notion or behavioral response of the RCL symbol in the LEL, including the hyperlink to the new added SCL symbol.

2.2) Case where non semantic relationship with any other concept map can be established through any SCnoL of the RCL concept map under study. Analyze two possibilities: i) missing information to set the relationship and ii) the RCL symbol is not a relevant one in the application context. Type of defect: Omission or Error. Correction:

f) Elicit more information about the RCL symbol in the application context.

g) Alternative where a relationship with other LEL symbol is obtained: Include in the RCL concept map the existent SCL and its relationship with the RCL. Add the proposition (relationship plus SCL concept) in the definition of the RCL symbol as a new sentence or part of a sentence in the notion or in the behavioral response of the LEL model, including the hyperlink to the definition of the SCL symbol (see Fig. 4); or

h) Alternative where the irrelevance of the symbol is detected: Remove the RCL concept map. Mark this irrelevant concept as an SCnoL in other concept maps where it figures as an SCL. Delete the symbol from the LEL model. Remove references to this symbol from other symbols. Add this concept in the minimum vocabulary.

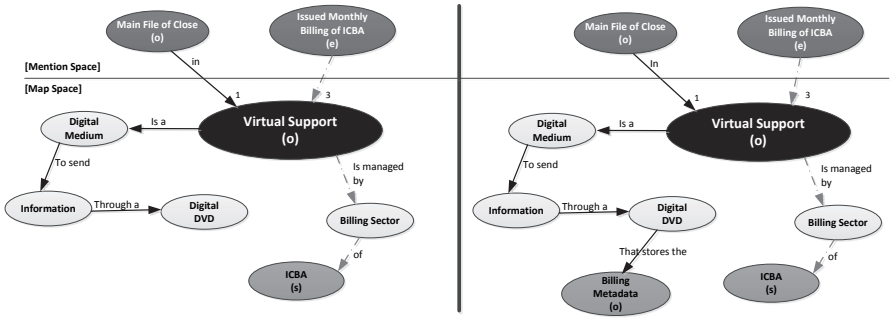


Fig. 4. *Example of absence of relationship between RCL concept and any SCL. On the left map, the RCL named* Virtual Support *has no relationship of notion type with an SCL. The right map represents the correction made by adding an SCL symbol named* Metadata of Billing *and its relation*.

3) Detect RCL symbols with the *mention space* empty. This means the symbol is not referenced by any other symbol of the LEL model, disregarding the principle of circularity.

a) Search for any SCnoL of the RCL concept map under study in other concept maps, and find a possible semantic relationship with the RCL in those other maps.

3.1) Case where a semantic relationship can be establish in other concept maps. Type of defect: Omission. Correction:

b) Add the RCL under study in the other concept map and marked it as an SCL.

c) Create a relationship between the SCnoL and the RCL under study (new SCL) in the other concept map.

d) Add the created relationship in the *mention space* of the RCL concept map under study.

e) Add the new proposition as a new sentence or part of an existent sentence in the notion or behavioral response of the LEL symbol whose concept map received this new proposition, including the hyperlink to the definition of the RCL symbol.

3.2) Case where no semantic relationship could be established with other concept maps. Analyze the possibility of missing information to set up the relationship or eventually the omission of an LEL symbol that allows the relation with other symbols. Type of defect: Omission. Correction:

f) Elicit missing information about the RCL symbol in the application context.

g) If a new semantic relation is detected with the RCL under study in any existent map, then add the relationship in that map with the RCL symbol as an SCL. Add the RCL symbol of the concept map, where the RCL under study was added as a new SCL in the *mention space* of the RCL map under study. Add the new proposition as a new sentence or part of an existent sentence in the notion or behavioral response of the symbol whose concept map

received the new SCL, including the hyperlink to the definition of the RCL symbol under study; or

h) If the existence of a new LEL symbol is elicited from the application domain, then construct its concept map including the RCL symbol under study as an SCL and establishing the corresponding relationship. Add the new symbol (RCL of the new concept map) in the *mention space* of the RCL map under study. Add the definition of the new symbol in the LEL model, including the hyperlink to the RCL symbol under study. Be aware that the new symbol should be mentioned by any other symbol to satisfy the circularity principle.

4.3 Verification Results

The lexicon model created in the context of a health care organization has consisted of 22 symbols, 28 sentences in the notions and 34 sentences in the behavioral responses. The set of constructed conceptual maps has contained 49 concepts that were not LEL symbols, which were reduced to 37 after verification, since 14 concepts were identified as synonyms among them, leaving only 6 of those 14 concepts, and additionally, 4 concepts were detected as synonyms of 3 LEL symbols. The decrease of concepts that were synonyms among them introduced an improvement in the use of the minimum vocabulary, by minimizing the level of ambiguity that is characteristic of models written in natural language. The 4 concepts detected as synonyms of LEL symbols were replaced by the corresponding symbols, improving the understandability of the glossary for clients and developers.
Summing up, the proposed verification process applied to the lexicon model identified 34 defects, which were: 15 omissions, 18 ambiguities and only 1 error. Table 3 details the number and type of defects detected in each step of the process.

Table 3. Number of detected defects

Analysis of Non LEL Symbol Concepts			Analysis of Relationships in Concept Maps		
Step	**Type of Defect**	**Number of Defects**	**Step**	**Type of Defect**	**Number of Defects**
2	Ambiguity	14	1	Omission	3
3	Ambiguity	3	2.1	Omission	2
3	Omission	1	2.2	Omission	1
6	Ambiguity	1	2.2	Error	1
6	Omission	7	3.1	Omission	0
			3.2	Omission	1

Additionally to the complementary tables that are filled during verification (see Tables 1 and 2), another form is filled recording every defect. The form contains columns for detailing the detected defect, the concept map containing the defect, comments, suggested corrections and doubts to be treated with the authors LEL.

5. Conclusions

A verification process of the Language Extended Lexicon has been presented with the purpose of improving the LEL quality. Good quality of this model is necessary since some models are created by the derivation from the LEL in the requirements process [9, 22, 23] and, additionally, models written in natural language are enhanced by using the terminology defined in the LEL in order to reduce ambiguity [9]. Furthermore, the present proposal was motivated by the results obtained from statistical studies about the completeness of this model, which have reported a high level of omissions [3, 16].

Unlike the inspection technique of the LEL model based on filling forms presented in [1], the verification process based on concept mapping focuses on the detection of omissions and ambiguities. The application of this proposed approach allows mainly the identification of omissions of symbols, omissions of notions and behavioral responses, ambiguities non-compliance with principles of circularity and minimum vocabulary, and errors due to including irrelevant symbols.

Despite the good results obtained in the health care case, the proposed verification process requires to be tested in more cases. In next steps, a deeper semantic study will also be conducted, including new steps and qualifying the severity of the detected defects. A future work will consist in creating semantic trees complementary to concept maps in order to detect other type of defects.

References

1. Kaplan, G., Hadad, G., Doorn, J., Leite, J.: Inspección del Léxico Extendido del Lenguaje. In: 3rd Workshop on Requirements Engineering, pp.70--91. Rio de Janeiro, Brazil (2000)
2. Firesmith, D.: Are Your Requirements Complete? Journal of Object Technology, vol.4, nº1, pp.27--43 (2005)
3. Hadad, G., Litvak, C., Doorn, J., Ridao M.: Dealing with Completeness in Requirements Engineering. In: Mehdi Khosrow-Pour (ed), Encyclopedia of Information Science and Technology, 3rd ed., pp.2854--2863. IGI Global, Information Science Reference (2015)
4. Berry, D.M., Kamsties, E.: Ambiguity in Requirements Specification. In: J. Leite, J. Doorn (eds.) Perspectives on Software Requirements, pp.7--44. Kluwer Academic Publishers. Springer US (2004)
5. IEEE 29148-2011, IEEE Systems and software engineering - Life cycle processes - Requirements engineering. IEEE, Nueva York (2011)
6. Regnell, B., Runesom, P., Thelin, T.: Are the perspectives really different? Further experimentation on scenario-based reading of requirements. Requirements engineering with use cases – a basis for software development. Technical Report 132, Lund University, pp.141--180 (1999)
7. Porter, A.A., Votta Jr., L.G.: Comparing Detection Methods for Software Requirements Inspections: A Replication Using Professional Subjects. Empirical Software Engineering, vol.3, nº4, pp.355--380 (1998)
8. Cheng, B., Jeffrey, R.: Comparing Inspection Strategies for Software

Requirements Specifications. In: Australian Software Engineering Conference, pp.203--211 (1996)
9. Leite, J.C.S.P, Doorn, J.H., Kaplan, G.N., Hadad, G.D.S., Ridao, M.N.: Defining System Context using Scenarios. In: J.C.S.P. Leite, J.H. Doorn (eds.) Perspectives on Software Requirements, pp.169--199. Kluwer Academic Publishers. Springer US (2004)
10. Novak, J., Gowin, D.B.: Aprendiendo a aprender. Ediciones Martínez Roca, Barcelona, Spain (1988)
11. Hadad, G.D.S., Doorn, J.H., Kaplan, G.N.: Creating Software System Context Glossaries. In: Mehdi Khosrow-Pour (ed) Encyclopedia of Information Science and Technology, 2ºed., pp.789--794. IGI Global, Information Science Reference, USA (2008)
12. Zowghi, D., Gervasi, V.: The Three Cs of Requirements: Consistency, Completeness, and Correctness. In: Proceedings of 8th International Workshop on Requirements Engineering: Foundation for Software Quality, Essen, Germany: Essener Informatik Beitiage, pp.155--164 (2002).
13. Doorn, J.H., Ridao, M.N.: Completeness Concerns in Requirement Engineering. In: Mehdi Khosrow-Pour (Ed.), Encyclopedia of Information Science and Technology, Second Edition, Hershey, PA: IGI Global, Information Science Reference, pp.619--624 (2009)
14. Ben Achour, C., Rolland, C., Maiden, N.A.M., Souveyet, C.: Guiding Use Case Authoring: Results of an Empirical Study. In: International Symposium On Requirements Engineering (RE´99), Limerick, Ireland, IEEE Computer Society Press, pp.36--43 (1999)
15. García Negroni, M.M.: Escribir en español: claves para una corrección de estilo, updated 2nd ed., Santiago Arcos, Buenos Aires, Argentina (2011)
16. Litvak, C.S., Hadad, G.D.S., Doorn, J.H.: Correcciones semánticas en métodos de estimación de completitud de modelos en lenguaje natural. In: 16th Workshop on Requirements Engineering, pp.105--117. Montevideo, Uruguay (2013)
17. Laitenberger, O., Debaud, J.M.: An Encompassing Life-Cycle Centric Survey of Software Inspection. Journal of Systems and Software, vol.50, Nº1, pp.5--31 (2000)
18. Dyer, M.: Verification-based inspection. In: 26th Annual Hawaii International Conference on System Sciences, pp.418--427 (1992)
19. Gallego, D., Ongallo, C.: Conocimiento y Gestión. Pearson Prentice-Hall, Madrid, Spain (2004)
20. Novak, J., Cañas, A.: The Theory Underlying Concept Maps and How to Construct and Use Them. Technical Report, Florida Institute for Human and Machine Cognition (2008)
21. Betancourt, A.M., Díaz Garzón, N.: Mapas Conceptuales: Elaboración y aplicación, Editorial Magisterio, 1ra ed., Bogotá, Colombia (2002)
22. Leonardi, M.C., Mauco, M.V.; Integrating Natural Language Oriented Requirements Models into MDA. In: 7th Workshop on Requirements Engineering, Tandil, Argentina, pp.65--76 (2004)
23. Antonelli, L., Rossi, G., Leite, J.C.S.P., Oliveros, A.: Deriving requirements specifications from the application domain language captured by Language Extended Lexicon. In: 15th Workshop on Requirements Engineering, Buenos Aires, Argentina (2012)

Adaptability-based Service Behavioral Assessment

DIEGO ANABALÓN[1;3], MARTIN GARRIGA[1;3], ANDRES FLORES[1;3], ALEJANDRA CECHICH[1] AND ALEJANDRO ZUNINO[2;3]

[1] GIISCo Research Group, Facultad de Informática,
Universidad Nacional del Comahue, Neuquen, Argentina.
[diego.anabalon, martin.garriga, andres.flores,
alejandra.cechich]@fi.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. *Building Service-oriented Applications implies the selection of adequate services to fulfill required functionality. Even a reduced set of candidate services involves an overwhelming assessment effort. In a previous work we have presented an approach to assist developers in the selection of Web Services. In this paper we detail its behavioral assessment procedure, which is based on testing and adaptation. This is done by using black-box testing criteria to explore services behavior. In addition, helpful information takes shape to build the needed adaptation logic to safely integrate the selected candidate into a Service-oriented Application. A concise case study shows the potential of this approach for both selection and integration of a candidate Web Service.*

1. Introduction

Service-oriented Applications imply a business facing solution that consumes services from one or more providers and integrates them into the business process [13]. Although developers do not need to know the underlying model and rules of a third-party service, its proper reuse still implies quite a big effort. Yet searching for candidate services is mainly a manual exploration of Web catalogs usually showing poorly relevant information [12]. Even a favorable search result requires skillful developers to deduce the most appropriate service to be selected for subsequent integration tasks. The effort on assessing candidate services could be overwhelming. Not only services interfaces must be assessed, but also their operational behavior as key feature of a service contract. Besides, correct adaptations must be identified so client applications may safely consume services while enabling loose coupling for maintainability.

To ease the development of Service-oriented Applications we presented in previous work [3,6] a proposal to assist developers on service selection by means of testing and adaptation. This approach complements the conventional compatibility assessment by using black-box testing criteria to

explore services behavior. The aim is to fulfill the *observability* testing metric [8,1] that observes a service operational behavior by analyzing its functional mapping of data transformations (input/output). In addition, a helpful information takes shape concerning the adaptation logic to integrate a service into a client application. Hence, a wrapping algorithm was defined based on *mutation testing* [4,9], to identify the right adapter configuration. However, mutation testing carries a high effort (cost) both on generation and execution.
In this work, we improve the wrapping algorithm based on a set of adaptability factors recently defined [3]. In this way, we were able both to be more accurate on setting the best adapter and to highly reduce the involved costs on mutation testing. A concise case study shows the potential of improvements implemented into our approach.
The rest of the paper is organized as follows. Section 1 presents an overview of the *Selection Method*. Section 3 explains the steps to build a *Behavioral TS*. Section 4 briefly describes the *Interface Compatibility* procedure. Section 5 describes the *Behavioral Compatibility* procedure. Section 6 presents related work. Conclusions and future work are presented afterwards.

2. Service Selection Method

During development of Service-Oriented Applications, specific parts of the system may be implemented in the form of in-house components. Besides, some of the comprising software pieces could be fulfilled by the connection to Web Services. A set of candidate services could be obtained by making use of any service discovery registry. Even with a wieldy candidates' list, a developer must be skillful enough to determine the most appropriate service for the consumer application. Figure 1 shows our proposal to assist developers in the selection of Web Services, which is briefly described as follows:
As an initial step, a simple specification is needed, in the form of a required interface I_R, as input for the three comprising procedures.

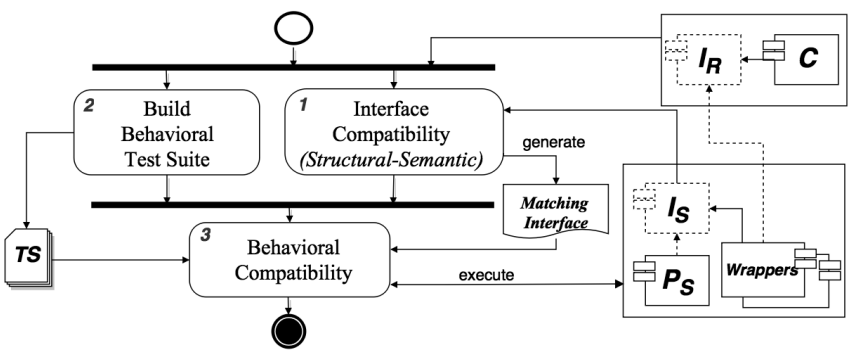


Fig. 1: *Service Selection Method*

The *Interface Compatibility* procedure (step 1) matches the required interface (I_R) and the interface (I_S) provided by a candidate service *S*. A structural-semantic analysis is performed to characterize operation signatures (return, name, parameters, exceptions) at four compatibility levels: *exact*, *near-exact*, *soft*, *near-soft*. This analysis also considers adaptability factors to reduce the integration effort. The outcome of this step is an *Interface Matching* list where each operation from I_R may have a matching with one or more operations from I_S [3]. Particularly, operations from I_R with multiple matchings are considered as "*conflictive operations*" in this approach – i.e., they must be disambiguated yet.

When a functional requirement (I_R) from an application can be fulfilled by a potential candidate Web Service, a *Behavioral Test Suite* (TS) is built (step 2) [6]. This TS describes the required messages interchange from/to a third-party service, upon a selected testing coverage criteria [8,1], to fulfill the *observability* testing metric.

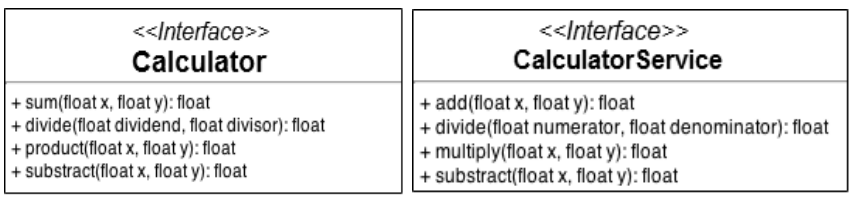
The *Behavioral Compatibility* procedure (step 3) evaluates the required behavior of candidate Web Services by executing the *Behavioral TS*. For this the *Interface Matching* list is processed to generate a set of wrappers *W* (adapters) – based on the identified *conflictive operations* – allowing to run the TS against the candidate service *S*.

After exercising the TS against each wrapper $w \in W$, at least one wrapper must successfully pass most of the tests to confirm both the proper matching of conflictive operations and the behavioral compatibility of the candidate service *S*. Besides, such successful wrapper allows an in-house component to safely call service S once integrated into the client application.

Next sections provide detailed information particularly related to the aforementioned procedures. A simple example will be used to illustrate the usefulness of the *Selection Method*.

2.1 Proof-of-Concept

To illustrate our proposal, we assume a simple example of a calculator application, with the four basic arithmetic operations. Figure 2(a) shows the required interface (I_R) called `Calculator` and Figure 2(b) shows the interface (I_S) of a candidate Web Service named `CalculatorService`.



(a) Required Interface Calculator (b) Service Interface CalculatorService

Fig. 2: *Case Study of Calculator service*

3. Behavioral Test Suite

In order to build a TS as a behavioral representation of services, specific coverage criteria for component testing has been selected [6]. The goal of this TS is to check that a candidate service *S* with interface I_S coincides on behavior with a given specification described by a required interface I_R. Therefore, each test case in TS will consist of a set of calls to I_R's operations, from where the expected results were specified to determine acceptance or refusal when the TS is exercised against *S* (through I_S).
The *Behavioral TS* is based on the *all-context-dependence* criterion [8,1], where synchronous events (e.g., invocations to operations) and asynchronous (e.g., exceptions) may have sequential dependencies on each other, causing distinct behaviors according to the order in which operations and exceptions are called. The criterion requires traversing each operational sequence at least once. In our approach, this is called "*interaction protocol*" [2], formalized by using *regular expressions*, which allows to automatize test case generation. The alphabet for regular expressions comprise the signature of service operations.
In addition, an imperative specification must be built to describe the expected behavior of the interface I_R, with a set of representative test data. This is called shadow class and takes the same name as I_R. Hence, each test case uses these test data as input for parameters on each call to operations of the I_R's interface. This means a black box relationship or input/output functional mapping.

3.1 TS for Calculator

For the interface (I_R) `Calculator`, a shadow class was defined using the values `0` and `1` as test data to the four arithmetic operations. Then, the interaction protocol (in the form of a regular expression) is defined as follows:

```
Calculator (sum | subtract | product | divide)
```

This regular expression implies operational sequences limited to an only operation to be invoked, since `Calculator` is a *stateless* service without dependencies between operations. A set of *test templates* is generated from the regular expression, representing each operational sequence. In this case, `4` test templates are derived, each one composed of the constructor operation and one arithmetic operation.
Then, the selected test data is combined with the `4` test templates to generate a TS in a specific format: based on the MuJava framework [10]. From this combination, `8` test cases were generated in the form of methods into a test file called `MujavaCalculator`. Code Listing 1.1 shows the test case `testTS_3_1`, which invokes the sum operation.

Listing 1.1: MuJava test case for Calculator´

```
Public String testTS_3_1(){
       calc.calculator obtained = null;
       obtained = new calc.calculator();
       float arg1 = (float) 0;
       float arg2 = (float) 1;
       float result0 = obtained.sum(arg1, arg2);
       return result0.toString();
}
```

4. Interface Compatibility

In the *Interface Compatibility* procedure is determined the level of compatibility between the operations of the interface I_R and the operations of the interface I_S of a candidate service S [3]. A structural-semantic analysis is performed to operation signatures.

Structural aspects consider signatures and data types, while semantic aspects consider identifiers and terms in the names of operations and parameters. Information Retrieval (IR) techniques and the WordNet (*https://wordnet.princeton.edu/*) dictionary are used for semantic aspects.

A scheme of constraints allows to characterize pairs of operations *($op_R \in I_R$, $op_S \in I_S$)* in four compatibility levels: *exact*, *near-exact*, *soft* and *near-soft*. Such constraints describe similarity cases based on adaptability (structural and/or semantic) conditions for each element of an operation signature (return, name, parameters, exceptions). As a result an *Interface Matching* list is generated, where each operation $op_R \in I_R$ may have a match to one or more operations $op_S \in I_S$, with likely one o more matchings in the parameters list.

In some cases, certain required operations ($op_R \in I_R$) could obtain multiple matchings (with the same compatibility) – at level of operations and/or parameters – to the candidate service interface (I_S). At operation level: an op_R has matching to several op_S.

At parameters level: an op_R has several matchings in the parameters list – i.e., a set of all possible permutations of arguments. These operations need a disambiguation and they are called "conflicting operations" in this approach.

For non-conflictive operations it is possible to assume a high reliability in the operation matching – i.e., they may confirm their compatibility through the *Behavioral Compatibility* procedure.

4.1 Calculator-CalculatorService Interface Matching

Table 1 shows the interface matching result for `Calculator` and `CalculatorService`. Operations `sum` and `product` of `Calculator` are identified as *conflictive operations* at operation level. They obtained three matchings with operations `add`, `subtract` and `multiply` of `CalculatorService`, with the same level of compatibility *near-soft* (n_soft_55). Operations `subtract` and `divide` of `Calculator` are non-

conflictive operations. They obtained a unique correspondence of higher compatibility level to their homonyms from `CalculatorService` – i.e., *exact* match for `subtract` operation and *near-exact* (n_exact_3) match for `divide` operation.
Moreover, all operations obtained a unique matching at parameters level. Parameters `(float x; float y)` of operations `sum`, `subtract` and `product` of `Calculator` are identical (in name and type) to their counterparts of `CalculatorService`. For `divide` operation of `Calculator`, its parameters have identical types and equivalent (synonyms) names –`dividend` with `numerator` and `divisor` with `denominator` – with the operation of `CalculatorService`.

Table 1. Interface Compatibility for Calculator-CalculatorService

Calculator	CalculatorService		
float subtract (float x, float y)	[1, exact, float subtract (float x, float y)] {(x:float-x:float),(y:float-y:float)}	[109, n_soft_55, float add (float x, float y)]	[109, n_soft_55, float multiply (float x, float y)]
float sum (float x, float y)	[109, n_soft_55, float add (float x, float y)] {(x:float-x:float),(y:float-y:float)}	[109, n_soft_55, float subtract (float x, float y)] {(x:float-x:float), (y:float-y:float)}	[109, n_soft_55, float multiply (float x, float y)] {(x:float-x:float), (y:float-y:float)}
float divide (float dividend, float divisor)	[4, n_exact_3, float divide (float numerator, float denominator)] {(dividend:float-numerator:float), (divisor:float-denominator:float)}	[116, n_soft_62, float add (float x, float y)]	[116, n_soft_62, float subtract (float x, float y)]
float product (float x, float y)	[109, n_soft_55, float add (float x, float y)] {(x:float-x:float),(y:float-y:float)}	[109, n_soft_55, float subtract (float x, float y)] {(x:float-x:float), (y:float-y:float)}	[109, n_soft_55, float multiply (float x, float y)] {(x:float-x:float), (y:float-y:float)}

5. Behavior Compatibility

To carry out the *Behavior Compatibility* evaluation for a candidate service *S*, a set of wrappers (adapters) *W* needs to be built to allow executing the *Behavioral TS* and compare their results with those specified in the interface I_R. The wrappers set is generated by processing the *Interface Matching* list, according to the multiple correspondences from the *conflictive operations* identified – both at operation and parameters levels. Hence, those multiples correspondences could be disambiguated so to identify proper univocal correspondences.
Wrappers generation can be seen as applying the *Interface Mutation* technique [4,9], by using a mutation operator to change invocations to operations and to change arguments in the parameters list. Thus, each

wrapper is considered a faulty version (or mutant) regarding the wrapper that contains the proper matchings of operations and parameters.
Previously [6], our approach was only based on structural aspects (signatures and data types) to generate wrappers, producing a larger set of wrappers *W*. This is because usually a larger number of conflictive operations were identified – both at operation and parameters levels.
A major improvement in this work involves to consider the semantic aspects provided by the *Interface Matching* list, in which a less number of conflictive operations is identified, effectively reducing the *W* set.

5.1 Wrappers Generation

A tree structure is built to generate wrappers, where each path from the root to a leaf node represents a specific matching between operations of I_R and I_S (i.e., a wrapper to be generated). Thus, the number of leaf nodes determines the size of the wrappers set *W*. Each conflictive operation produces several branches on the tree. On the contrary, a non-conflictive operation (implying a univocal match) does not involve additional branches in the tree.
In the case of a conflictive operation at operation level, a new branch is added for each matching to a service operation. At parameters level, a new branch is added for each arguments matching from the set of permutations – even though there could be a univocal operation matching.
Particularly, in this work was updated the algorithm that implements the mutation operator to change arguments into the wrappers generation. Thereby, the new algorithm to treat parameters matchings considers the following cases:

1. *Without any matching*: if any matching was identified at all (structural and/or semantic), parameters will be permuted between each other, producing branches for each arguments combination.
2. *Only structural matching*: If a semantic matching was not identified, parameters are related only through the structural information (data types). If multiple matchings were identified, for each of them a branch is produced. For the remaining parameters the *case 1* is applied.
3. *Structural-semantic matching*: Parameters are related through the structural and semantic information. If multiple matchings were identified, for each of them a branch is produced. For the remaining parameters the *case 2* is applied.
4. *Service's extra parameters*: If a service operation contains more parameters than the required operation, then some parameters are left outside of the matchings. For them, a test value is required when invoking the service operation. Hence, in this approach a default value is assigned according to each parameter data type – `""` (quotes) for strings, `' '` (space character) for characters, `true` value for booleans, and `0` (zero) for numerical types.

5.2 Wrappers for Calculator-CalculatorService

Figure 3 shows the wrapper generation tree for `Calculator` and `CalculatorService`. Branches were only produced at operation level according to the *conflictive operations* identified: `sum` and `product` of `Calculator` with respect to `add`, `subtract` and `multiply` of `CalculatorService`. Regarding to parameters matching, the *case 3* was applied since a structural-semantic matching was identified for all parameters.

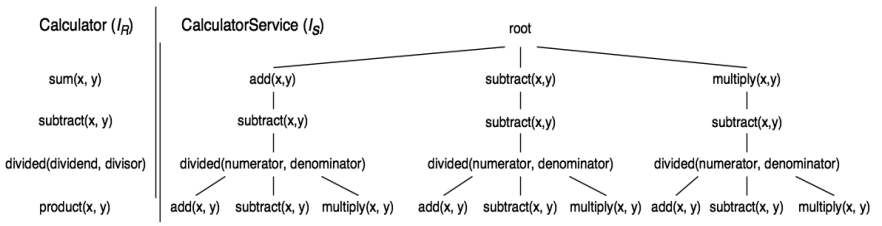


Fig. 3: *Wrapper generation tree to CalculatorService*

The total number of wrappers (size of *W*) to be generated is `9`, which is the number of leaves on the tree. Notice that without considering semantic aspects, particularly for parameters, a major number of permutations there had been generated. Since all parameters are of the same type, multiple structural matchings there had been identified, making the size of *W* scaling to `144` wrappers.
Listing Code 1.2 and 1.3 show a fragment of the code from `wrapper2` and `wrapper3` respectively. Where `wrapper2` represents both the tree path down-to the third leaf node and the most appropriate matchings. Likewise, `wrapper3` represents the path down-to the fourth leaf node – being a faulty (mutant) version.

Listing 1.2: Wrapper2 for Calculator-CalculatorService

```
public class Calculator{
       protected katze. ... .Calculator Service proxy =
null;
       public Calculator(){
              this.proxy = new katze. .. .Calculator
       Service();
       }
       public float sum(float arg1, float arg2){
              float ret0;
              try{ ret0 = candidate.add(arg1, arg2);
              } catch(exception ex){
                     ex.printStackTrace();
                     throw new RuntimeException(ex);
              }
              return ret0;
       }
       //...
       public float product(float arg1, float arg2) {
              float ret0;
              try{ ret0 = candidate.multiply(arg1,arg2);
              } catch(exception ex){
```

```
                    ex.printStackTrace();
                    throw new RuntimeException(ex);
                }
                return ret0;
        }
}
```

Listing 1.3: Wrapper3 for Calculator-CalculatorService

```
public class Calculator{
//...
        public float sum(float arg1, float arg2){
                float ret0;
                try{ ret0 = candidate.subtract(arg1, arg2);
                } catch(exception ex){
                        ex.printStackTrace();
                        throw new RuntimeException(ex);
                }
                return ret0;
        }
        //...
        public float product(float arg1, float arg2) {
                float ret0;
                try{ ret0 = candidate.add(arg1,arg2);
                } catch(exception ex){
                        ex.printStackTrace();
                        throw new RuntimeException(ex);
                }
                return ret0;
        }
}
```

5.3 Wrappers Evaluation

Once generated the set of wrappers *W*, the *Behavior TS* is executed against each wrapper $w \in W$ to assess the behavior of the candidate service *S*. Using our tool based on the MuJava framework, the TS is exercised against the I_R and iterating over the list of wrappers. After that, results are compared to determine for each wrapper the number of test cases that failed – which produced a result different from the one expected. A wrapper may survive (as mutation case) when most of the test cases are successful. A successful wrapper allows to disambiguate the conflictive operations, confirming the right matchings both at operation and parameters levels. In addition, this wrapper may be used as integration artifact allowing a safe communication to the candidate service *S*.

5.4 Behavioral Evaluation for Calculator-CalculatorService

The TS called `MujavaCalculator` was executed against `Calculator` (I_R) and the `9` wrappers generated for `CalculatorService`. Table 2 shows the

execution results, where wrapper2 passed successfully 100% allowing to confirm the behavioral compatibility of CalculatorService. In addition, this wrapper contains the right matchings of operations (sum-add, subtract-subtract, divide-divide, product-multiply). Finally, wrapper2 can be used as an adapter for the safe integration of CalculatorService in the client application.

6. Related work

Due to lack of space this section briefly presents related work without a detailed comparison with our approach.

Table 2. Execution results of TS for Calculator-CalculatorService

Wrappers	Test Cases		
	Successful	Failed	Success rate
wrapper3,wrapper4,wrapper6,wrapper7	0	4	0
wrapper0,wrapper1,wrapper5,wrapper8	2	2	50
wrapper2	4	0	100

In [7] we survey current approaches on selection, testing and adaptation of services with focus on composition. Service selection approaches are closely related to discovery, in which IR techniques and/or a semantic basis (e.g., ontologies) are generally used.
Service evaluation mainly use WSDL documents and/or XML schemes of data types, or even WSDL-based ad-hoc enriched specifications. Service implementation may also affect its evaluation: contract-first services are designed prior to code, improving their WSDL descriptions; code-first services use automatic tools to derive WSDL documents from source code, reducing their description quality.
Regarding service testing, the work in [2] presents a survey of approaches that use strategies of verification and software testing. Some of them evaluate individual operations of atomic services, others also use a semantic basis such as OWL-S, and others evaluate a group of services that could interact in a composition.
The work in [5] presents an overview on service adaptation, at service interface and business protocol levels. This is required even though the Web Service standardization reduces the heterogeneity and simplifies interaction. At interface level adaptations deal with operation signatures, that implies perform message transformations or data mapping. At business protocol level, services behavior is affected on the order constraints of the message exchange sequences – such as deadlock and non-specified reception.

7. Conclusions and Future Work

In this paper we have presented an approach to assist developers in the selection of services, when developing a Service-oriented Application. Particularly, our approach addresses two main aspects. On the one side, confirming the suitability of a candidate service by a dynamic behavioral evaluation (execution behavior), in which the applied testing criteria increase the reliability level. On the other side, effectively building the right adaptation logic for a selected Web Service, while reducing the adaptation and integration effort.

Currently, we are working on service compositions [7]. This is particularly useful when a single service cannot provide all the required functionality. In this context, it is necessary to generate software artifacts (e.g., tests and adapters) according to specifications in business process languages such as BPEL and BPML [14]. Finally, another interesting extension of this work is to automatically derive software artifacts from system models – for example from models described in SoaML [11], a UML profile for modeling Service-oriented Applications.

Acknowledgments

This work is supported by projects: ANPCyT PICT 2012-0045 and UNCo-SPU Reuse (04-F001).

References

1. X. Bai, W. Dong, W-T. Tsai, and Y. Chen. WSDL-based Automatic Test Case Generation for Web Services Testing. In IEEE International Workshop Service-Oriented System Engineering, SOSE'05, pp. 207–212, 2005.
2. M. Bozkurt, M. Harman, and Y. Hassoun. Testing and Verification in Service-Oriented Architecture: A Survey. Software Testing, Verification and Reliability, 23(4):261–313, 2013.
3. A. De Renzis, M. Garriga, A. Flores, A. Zunino, and A. Cechich. Semantic-Structural Assessment Scheme for Integrability in Service-Oriented Applications. In Latin-american Symposium of Enterprise Computing, held during CLEI'14, pp. 637-647. September 2014.
4. M. Delamaro, J. Maidonado, and A. Mathur. Interface Mutation: An Approach for Integration Testing. IEEE Transactions on Software Engineering, 27(3):228–247, 2001.
5. M. Eslamichalandar, K. Barkaoui, and H. Motahari-Nezhad. Service Composition Adaptation: An Overview. 2nd IEEE IWAISE, pp. 20-27, 2012.
6. M. Garriga, A. Flores, A. Cechich, and A. Zunino. Behavior Assessment based Selection Method for Service-Oriented Applications Integrability. In 41st Argentine Symposium on Software Engineering, ASSE'12, SADIO, pp. 339–353, La Plata, BA, Argentina. 2012.

7. M. Garriga, A. Flores, A. Cechich, and A. Zunino. Web Services Composition Mechanisms: A Review. IETE Technical Review, 32(5): 376-383, 2015.
8. M. Jaffar-Ur Rehman, F. Jabeen, A. Bertolino, and A. Polini. Testing Software Components for Integration: a Survey of Issues and Techniques. Software Testing, Verification and Reliability, 17(2):95–133, June 2007.
9. Y. Jia and M. Harman. An Analysis and Survey of the Development of Mutation Testing. IEEE Transactions on Software Engineering, 37(5):649–678, 2011.
10. μJava Home Page. Mutation System for Java Programs, 2008. http://www.cs.gmu.edu/ offutt/mujava/.
11. OMG. Service Oriented Architecture Modeling Language (SoaML) Specification. Technical report, Object Management Group, Inc., 2012. http://www.omg.org/ spec/SoaML/1.0.1/PDF/.
12. M. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. Service-Oriented Computing: A Research Roadmap. International Journal of Cooperative Information Systems, 17(02):223–255, 2008.
13. D. Sprott and L. Wilkes. Understanding Service-Oriented Architecture. The Architecture Journal. MSDN Library. Microsoft Corporation, 1:13, January 2004. http://msdn.microsoft.com/en-us/library/aa480021.aspx.
14. S.Weerawarana, F. Curbera, F. Leymann, T. Storey, and D. Ferguson. Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More. Prentice Hall PTR, 2005.

XII

Database and Data Mining Workshop

Capturing relational NEXPTIME with a Fragment of Existential Third Order Logic

JOSÉ MARIA TURULL-TORRES [1,2]

[1] Depto. de Ingeniería e Investigaciones Tecnológicas. Universidad Nacional de La Matanza
[2] Massey University, New Zealand
J.M.Turull@massey.ac.nz

Abstract. *We prove that the existential fragment of the third order logic* TO_ω *captures the* relational *complexity class* non deterministic exponential time*. As a Corollary we have that relational machines can simulate third order relational machines.*

1. Introduction

Relational machines (RM) were introduced in [AV,91] (there called loosely coupled generic machines) as abstract machines that compute queries to (finite) relational structures, or relational database instances (dbis) as functions from such structures to relations, that are generic (i.e., that preserve isomorphisms), and hence are more appropriate than Turing machines (TM) for query com- putation. RMs are TMs endowed with a relational store that hold the input structure, as well as work relations, and that can be accessed through first order logic (FO) queries (sentences) and updates (formulas with free variables). As the set of those FO formulas for a given machine is fixed, an RM can only distinguish between tuples (i.e., sequences of elements in the domain of the dbi) when the differences between them can be expressed with FO formulas with k variables, where k is the maximum number of variables in any formula in the finite control of the given RM. Note that the same is true for FO queries (i.e., relational calculus), or equivalently relational algebra queries.

On the other hand, it has been proved that RMs have the same computation, or expressive power, as the (effective fragment of the) well known infinitary logic with finitely many variables ([AVV,95]), (in the context of Finite Model Theory, i.e., with sentences interpreted by finite relational structures or database instances - dbi's). This logic extends FO with conjunctions and disjunctions of sets of formulas of arbitrary (infinite) cardinality, while restricting the number of variables in each (infinitary) formula to be finite. This is a very important logic in descriptive complexity theory, in which among other properties, equivalence is characterized by pebble (Ehrenfeucht-

Fraissee) games, and on ordered dbis it can express all computable queries (see [Lib,04], among others). Hence, a nice characterization of the discerning power of RMs is also given by those games.
Consequently, k-ary RMs are incapable of computing the size of the input structure though, however, they can compute its $size_k$. A k-ary RM, for a positive integer k, is an RM in which the FO formulas in its finite control have at most k different variables, and the $size_k$ of a structure (or dbi) is the number of equivalence classes in the relation $\equiv_k$ of equality of FO_k types in the set of k-tuples of the structure, for $1 \leq k$. Then, it was a natural consequence to define a new notion of complexity suit- able for RMs. *Relational complexity* was introduced in [AV,91] as a complexity theory where the (finite relational) input structure A to an algorithm is measured as its $size_k$, for some $k \geq 1$, instead of the size of its encoding, as in computational complexity. Roughly, two k-tuples in A have the same FO_k types if they both satisfy in A exactly the same FO formulas with up to k variables, r of them being free, for all $0 \leq r \leq k$. That is, if the two tuples have the same properties in the structure A, considering only the properties that can be expressed in FO_k. In that way, relational complexity classes mirroring classical complexity classes like *P*, *NP*, *PSPACE*, *EXPTIME* and *NEXPTIME*, etc., have been defined ([AV,91], [AVV,97]), and denoted as P_r, NP_r, $PSPACE_r$, $EXPTIME_r$ and $NEXPTIME_r$, respectively (the class $NEXPTIME_r$ is actually defined later in this article).
Beyond the study of RM's as a model of computation for queries to relational databases, relational complexity turned out to be a theoretical framework in which we can characterize exactly the expressive power of the well known fixed point *quantifiers* (FP) of a wide range of types. Those quantifiers are typically added to first order logic, thus forming the so called *fixed point logics*, where the different types of fixed point quantifiers add to FO different kinds of iterations of first-order operators ([Lib,04], [AVV,97]). In [AVV,97], S. Abiteboul, M. Vardi and V. Vianu introduced new fixed point quantifiers, and organized a wide range of them as either deterministic (det), non deterministic (ndet), or alternating (alt), and either inflationary (inf) or non inflationary (ninf), according to the type of iteration implied by the semantics of each such quantifier. In the same article they proved the following equivalences: det-inf-FP = P_r, ndet-inf-FP = NP_r, alt-inf-FP = det-ninf-FP = ndet-ninf-FP = $PSPACE_r$, and alt-ninf-FP = $EXPTIME_r$ (in the case of ndet FP no negation affecting an FP quantifier is allowed). Those characterizations of relational complexity classes are actually very interesting and meaningful, given that it was already known that if we restrict the input to only ordered structures, the following equivalences with computational complexity classes hold: det-inf-FP = *P*, ndet-inf-FP = *NP* , det-ninf-FP = ndet- ninf-FP = alt-inf-FP = *PSPACE*, and alt-ninf-FP = *EXPTIME* ([Lib,04], [AVV,97]). Regarding the characterization of relational complexity classes with other logics, A. Dawar introduced in [Daw,98] the logic SO_ω, defining it as a semantic restriction of second order logic (SO) where the valuating relations for the quantified second order variables are "unions" of complete

FO_k types for r-tuples for some constants $k \geq r \geq 1$, that depend on the quantifiers[1]. That is, the relations are closed under the relation $\equiv_k$ of equality of FO_k types in the set of r-tuples of the structure.
In [Daw,98] it was also proved that the existential fragment of SO_ω, $\Sigma_{1,\omega}$ characterizes exactly the non determinestic fixed point logic (FO + *NFP*), and hence, by the equivalences mentioned above, it turned out that $\Sigma_{1,\omega}$ captured NP_r, analogously to the well known relationship $\Sigma_1 = NP$ ([Fag,74]). Continuing the analogy, the characterization of the relational polynomial time hierarchy PH_r with full SO_ω was stated without proof in [Daw,98], and later proved by the second author jointly with F. Ferrarotti in [FT,08].
In [AT,14], aiming to characterize higher relational complexity classes, and as a natural continuation of the study of the logic SO_ω , we defined a variation of third order logic (TO) denoted as TO_ω , under finite interpretations. We defined it as a semantic restriction of TO where the (second order) relations which form the tuples in the third order relations that valuate the quantified third order variables are *closed* under the relation $\equiv_k$ as above. In [AT,14] we also introduced a variation of the non deterministic relational machine, which we denoted 3-NRM (for third order NRM), where we allow TO relations in the relational store of the machine. We defined the class $NEXPTIME_{3,r}$ as the class of 3-NRMs that work in time exponential in the $size_k$ (see above) of the input dbi. We then proved that the existential fragment of TO_ω, denoted $\Sigma_{2,\omega}$, captures $NEXPTIME_{3,r}$.
In the present article, we prove a stronger result: we show that the existential fragment of TO_ω also captures the relational complexity class $NEXPTIME_r$. Then, adding the result proved in this article, we have the following picture regarding the known characterizations of relational complexity classes up to now: P_r = (FO + det-inf-FP), NP_r = (FO + ndet-inf-FP) = $\Sigma_{1,\omega}$, PH_r = SO_ω , $PSPACE_r$ = (FO + alt-inf-FP) = (FO + det-ninf-FP) = (FO + ndet-ninf-FP), $EXPTIME_r$ = (FO + alt-ninf-FP), and $NEXPTIME_r = \Sigma_{2,\omega}$.Then, as it turned out that $NEXPTIME_r = NEXPTIME_{3,r}$, an interesting consequence of our result is that RM's in their original formulation are strong enough as to simulate the existence of TO relations in their relational store and, hence, to also simulate the existence of TO_ω formulas in their finite control (without TO_ω or SO_ω quantifiers, as in 3-NRM's in [AT,14], see below). That is, for every 3-NRM that works in time $NEXPTIME_{3,r}$, i.e., relational *third order* exponential time, in the $size_k$ of their input, there is an NRM that computes the same query, and that works in time $NEXPTIME_r$, i.e., relational exponential time in the $size_k$ of their input.
Note: This article has been selected in CACIC 2015 for its publication in JCST. We refer the reader who is interested in following the technical details to [Tur,16], since the use of Latex in that version provides a much clearer notation.

[1] in the sense of [FPT,10] these relations are redundant relations

2. Preliminaries

We assume a basic knowledge of Logic and Model Theory (refer to [Lib,04]). We only consider vocabularies of the form σ = (R1,…, Rs) (i.e., purely relational), where the arities of the relation symbols are $r1, \ldots, rs \geq 1$, respectively. We assume that they also contain equality. And we consider only *finite* σ-structures, denoted as **A** = (A,R1*,…, Rs*), where A is the domain, also denoted *dom*(**A**), and R1*, …, Rs* are (second order) relations of the proper arity in A. If $\gamma(x1, \ldots, xl)$ is a formula of some logic with free FO variables $\{x1, \ldots, xl\}$, for some $l \geq 1$, with γ**A** we denote the l-ary relation defined by γ in **A**, i.e., the set $\{(a1, \ldots, al) : a1, \ldots, al \in A \wedge \mathbf{A} \models \gamma(x1, \ldots, xl)\ [a1, \ldots, al]\}$. For any l-tuple $\bar{a} = (a1, \ldots, al)$ of elements in A, with $1 \leq l \leq k$, we define the *FOk* type of $\bar{a}$, denoted *Typek*(**A**, $\bar{a}$), to be the set of formulas $\phi \in FOk$ with free variables among $x1, \ldots, xl$, such that **A** $\models \phi[a1, \ldots, al]$. If τ is an *FOk*type, we say that the tuple $\bar{a}$ realizes τ in **A**, if and only if, τ = *Typek*(**A**, $\bar{a}$). Let **A** and **B** be σ-structures and let $\bar{a}$ and $\bar{b}$ be two l-tuples on **A** and **B** respectively, we write (**A**, $\bar{a}$) $\equiv k$ (**B**, $\bar{b}$), to denote that *Typek*(**A**, $\bar{a}$) = *Typek*(**B**, $\bar{b}$), or $\bar{a} \equiv k\ \bar{b}$ if **A** = **B.** *sizek*(**A**) is the number of equivalence classes in $\equiv k$ in **A.** An l-ary relation R in **A** is closed under $\equiv k$ if for any l-tuples a, b in A, $\bar{a} \in R \wedge \bar{a} \equiv k\ \bar{b} \Rightarrow \bar{b} \in R$. Let S be a set, a binary relation R is a pre-order on S if it satisfies: 1) $\forall a \in S\ (a, a) \in R$ (reflexive), 2) $\forall a, b, c \in S\ (a, b) \in R \wedge (b, c) \in R \Rightarrow (a, c) \in R$ (transitive), 3) $\forall a, b \in S\ (a, b) \in R \vee (b, a) \in R$ (conex). A pre-order $\leqq$ on S induces an equivalence relation $\equiv$ on S (i.e., $a \equiv b \Leftrightarrow a \leqq b \wedge b \leqq a$), and also induces a total order over the set of equivalence classes of $\equiv$. When the classes induced by a pre-order on k-tuples from some structure **A** agree with the classes of $\equiv k$, then the pre-order establishes a total order over the FOk types for k-tuples which are realized on **A**.

With $\Sigma 1,\omega\ [\sigma]$ we denote the class of formulas with m alternated blocks of SO quantifiers of the form $\exists kY\ r,k$ or $\forall kY\ r,k$, and then an FO formula φ of the vocabulary σ augmented with all such variables Y where $r \leq k$. We then define SOω as the union of all such classes of formulas for every m. The second order quantifier $\exists k$ has the following semantics: let **I** be a σ-structure; then **I** $\models \exists kY\ r,k\phi$ if there is an r-ary (SO) relation Rr,k on I that is closed under the relation $\equiv k$ in **I**, and (**I**, R) $\models \phi$.

3. The Restricted Third-Order Logic TOw and 3-NRM's

A third order relation type is a w-tuple $\tau = (r1, \ldots, rw)$ where $w, r1, \ldots, rw \geq 1$. In addition to the symbols of SOω, the alphabet of TOω ([AT,14]) contains for every $k \geq 1$, a TO quantifier $\exists k$, and for every relation type τ such that $r1, \ldots, rw \leq k$ a countably infinite set of third order variables, denoted χ with parameters τ, k and called TO variables. We use upper case Roman letters for SOω variables (in this article we will often drop the superindex k, when it is clear from the context), where r is their arity, and

lower case Roman letters for individual (i.e., FO) variables. Let σ be a relational vocabulary. A TOω atomic formula of vocabulary σ, on the TOω variable χ with parameters τ, k is a formula of the form $\chi(V1, \ldots, Vw)$, where $V1, \ldots Vw$ are either SO variables of the form Xri,k, or relation symbols in σ, and whose arities are respectively $r1, \ldots, rw \leq k$. Note that all the relations that form a σ-structure are closed under $\equiv k$, since k is $\geq$ than all the arities in σ (see above, and Fact 9 in [FT,08]). Let $m \geq 1$. We denote by $\Sigma 2,\omega$ m[σ] the class of formulas with m alternated blocks of TO quantifiers of the form $\exists kY\ r,k$ or $\forall kY\ r,k$ and then an SOω formula ψ with the addition of TOω atomic formulas. We define TOω as the union of all such classes of formulas for every m. A TOω *relation θ of type τ and closed under $\equiv k$* on a σ structure **I** is a set of w tuples (R1,…, Rs) of SO relations on I with respective arities $r1, \ldots, rw \leq k$, closed under $\equiv k$. The TO quantifier $\exists k$ has the following semantics: let **I** be a σ-structure; then **I** $\models \exists k\ \chi(\phi)$ if there is a TOω relation θ of type τ on I closed under the relation $\equiv k$ in **I**, such that (**I**, θ) $\models \phi$.

Here (**I**, θ) is the *third order* ($\sigma \cup \{\chi\}$) structure expanding **I**, in which $\chi$is interpreted as θ. Note that a valuation in this setting also assigns to each SO variable X an SO relation on I of arity r that is closed under $\equiv k$ in **I**, and to each TO variable $\chi$with parameters $\tau, k,$ a TO relation θ on I of type τ , closed under $\equiv k$ in **I**. We do not allow free SO or TO variables in the logics SOω and TOω . Note that allowing elements (from the domain of the structure) in a TO relation type would change the semantics, since we could use a TO relation of such type to simulate an SO relation not closed under $\equiv k$. See [AT,14] for an example of a non trivial query in TOω. A third order non-deterministic relational machine ([AT,14]), noted as 3-NRM, of arity , for $k \geq 1$, is a 11-tuple $(Q, \Sigma, \delta, q0, b, F, \sigma, \tau, T, \Omega, \Phi)$ where: Q is the finite set of internal states; $q0 \in Q$ is the initial state; Σ is the finite tape alphabet; $b \in \Sigma$ is the symbol denoting blank; $F \subseteq Q$ is the set of accepting states; τ is the finite vocabulary of the *rs* (its *relational store*), with finitely many *TOω* relation symbols θ of any arbitrary type $\tau i = (ri1, \ldots, riw)$, with $1 \leq ri1, \ldots, riw \leq kl = k$, and finitely many SO$\omega$ relation symbols Rj with parameters *ri , kll* of arities $ri \leq kll = k$; $T \in \tau$ is the output relation; σ is the vocabulary of the input structure; Ω is a finite set of TOω formulas with up to k *FO* variables, with *no* SOω *or* TOω *quantifiers*, and with no free variables of any order (i.e., all the *SOω* and *TOω* relation symbols are in τ); Φ is a finite set of TOω formulas with up to k *FO* variables, that are not sentences, with *no* SOω or TOω *quantifiers*, and where the free variables are either *all FO* variables, or *all SO ω* variables; $\delta : Q \times \Sigma \times \Omega \rightarrow P(\Sigma \times Q \times \{R, L\} \times \Phi \times \tau)$ is the transition function. In any pair in δ, if ϕ, S occur in the 5-tuple of its 2nd component, for Φ and τ, then either S is a TOω relation symbol $R\tau i,k$ in rs and ϕ has $|\tau i|$ SOω free variables X1, …, X$|\tau i|$ with parameters *rj, kll* and arities according to τi, and $1 \leq r1, \ldots, r|\tau i| \leq k'' = kl = k$, or S is an SO$\omega$ relation symbol Ri with parameters ri , k'' in rs and ϕ has $1 \leq r_i \leq k'' = k$ FO free variables. At any stage of the computation of a 3-NRM on an input σ-structure **I**, there is one relation in

its *rs* of the corresponding relation type (or arity) in **I** for each relation symbol in τ, so that in each transition there is a (finite) τ-structure **A** in the rs, which we can *query* and/or *update* through the formulas in Ω and Φ, respectively, and a finite Σ string in its tape, which we can access as in Turing machines. The concept of computation is analogous to that in the Turing machine. We define the complexity class $NEXPTIME_{3,r}$ as the class of the *relational languages* or Boolean queries (i.e., sets of finite structures of a given relational vocabulary, closed under isomorphisms) that are decidable by 3-NRM machines of *some* arity k_1, that work in non deterministic exponential time in the number of equivalence classes in $\equiv_k$, of the input structure. A non-deterministic relational machine, i.e., an NRM in its classical formulation, denoted as NRM, of *arity k*, for $k \geq 1$, is a 11-tuple as above, where all the formulas in Ω and Φ are *FO* formulas with up to *k FO* variables, in the vocabulary τ, and where all the relations in the rs are SO relations of arity at most k. The relational complexity class NEXPTIME_r is the class of *relational languages* or *Boolean queries* that are decidable by NRM machines of some arity k_1, that work in non deterministic exponential time in the number of equivalence classes in $\equiv_{k1}$ of the input structure. In [AT,14] we proved the following results:

Theorem 1: ([AT,14]) Given a 3-NRM M in $NTIME_{3,r}(2^{c\cdot(size_k)})$, for some positive integer c, and with input vocabulary σ that computes a Boolean query q we can build a formula $\phi_M \in \Sigma_{2,\omega}$ such that, for every σ-structure ***I***, *M* accepts ***I*** iff $\boldsymbol{I} \models \phi_M$.

Theorem 2: ([AT,14]) Every class of relational structures definable in $\Sigma_{2,\omega}$ is in $NTIME_{3,r}(2^{c\cdot(size_k)})$.

4. Existential TOω captures NEXPTIMEr

Corollary 3: Given an NRM M that works in $NTIMEr(2c\cdot(sizek))$, for some positive integer c, and with input vocabulary σ that computes a Boolean query q we can build a formula $\phi M \in \Sigma 2,\omega$ such that, for every σ-structure ***I***, M accepts ***I*** *iff* $\boldsymbol{I} \models \phi M$.

Proof. This is a consequence of Theorem 1 by the following two immediate facts: 1) an NRM is a special case of a 3-NRM, with no third order relations in its rs, and 2) an NRM *M* is in *NEXPTIMEr* iff *M*, as a 3-NRM, it is in *NEXPTIME*3,*r*.

Theorem 4: Every class of rel. structures definable in $\Sigma 2,\omega$ is in $NTIMEr(2c\cdot(sizek))$.

Proof. Let σ be a relational vocabulary, let φ be a $\Sigma 2,\omega\ [\sigma]$ sentence of the form $\exists k3,1\ \chi 1. \ldots \exists k3,s\ \chi s\ (\psi)$ where ψ is a $\Sigma 1,\omega t$ formula, for some $t \geq 1$, with atomic $TO\omega$ formulas formed with the $TO\omega$ variables $\chi 1, \ldots, \chi s$. For the sake of a simpler presentation we assume w.l.o.g. that for $1 \leq i \leq s$ the type of the relation χi is $\tau i = (r3,i, \ldots, r3,i)$ of cardinality $r3,i$, with $r3,i \leq k3,i$. Suppose the formula ψ is as defined above, on SO variables

Y*ij*, with parameters *k2,11, . . . k2,1l1,..., k2,t1 ,..., k2,tlt* and *r2,11, . . . r2,1l1, ..., r2,t1 ,..., r2,tlt* , respectively, where ϕ is an FO formula of the vocabulary σ augmented with all such variables Y*ij*, with atomic *TOω* formulas, and $r2,11 \leq k2,1l1 , . . ., r2,tlt \leq k2,tlt$, respectively. We now build an NRM $M\varphi$ which accepts a given σ structure **I** iff **I** $\models \varphi$. It is known that for every σ, and every $k \geq 1$, a formula $\gamma(\bar{x}, \bar{y})$ with $kll \geq 2k$ variables of the fixed point logic (*FO + LFP*) can be built s. t. on any σ structure **J**, γ defines a pre-order $\leqq k$ in the set of *k*-tuples of **J**, whose induced equivalence relation is $\equiv k$ (see T.11.20 in [Lib,04]). On the other hand, it is known that (*FO +LFP*) captures relational polynomial time *Pr* ([AVV,97]). Hence, an *RM Mk* of some arity $kl \geq 2k$ can be built, that constructs, on input **J**, the pre-order $\leqq k$ in **J**, in time polynomial in *sizek'* (**J**). We define the arity of $M\varphi$ as k = max (*{k'3,1 ,..., k'3,s , k'2,11 ,..., k'2,tlt }*), where the *k'ij* are the arities of the RMs that build the pre-orders $\leqq$for *k3,1 ,..., k3,s , k2,11 ,..., k2,tlt*, respectively.

Let **I** be the input structure. $M\varphi$ works as follows: **1):** $M\phi$ simulates the RMs that build the pre-orders $\leqq$for *k3,1,..., k3,s , k2,11 ,..., k2,tlt*. $M\phi$ builds the pre-orders in time polynomial in all the different *sizek* of **I** corresponding to *k'3,1,..., k'2,tlt*. As all these arities are $\leq k$ (see above), that time is also polynomial in *sizek*(**I**) (see [FT,08]). **2):** By stepping through the equivalence classes of the relation $\equiv k3,1$ in the order given by $\leqq k3,1$, $M\varphi$ computes the *sizek*3,*1* (**I**), and the same process is followed to compute all the different *sizek* of **I** corresponding to *k3,2 ,..., k2,tlt* by using the corresponding equivalence relations (recall that all those pre-orders induce total orders in the equivalence classes of the corresponding equivalence relations). Note that by the choice of k, all these computations are done by $M\phi$ in time polynomial in *sizek* (**I**). **3):** $M\varphi$ needs to guess the *TOω* relations $\theta 1, ..., \theta s$ of types $\tau 1, \ldots, \tau s$, as interpretations of the *TOω* variables $\chi 1, . . . , \chi s$, respectively. Each θi is a set of *r*3,*i*-tuples of *r*3,*i*-ary (*SO*) relations closed under $\equiv k3,i$. To represent θi we use we use three sorts of bit strings as follows: **a)** each bit string of sort b3 (for an SO relation R with parameters *r3,i* , *k3,i*) of size *sizek*3,*i* (**I**), represents one of the possible *r*3,*i*-ary (*SO*) relations on I, closed under $\equiv k3,i$; note that each bit represents one equivalence class in $\equiv k3,i$, following from left to right the total order induced by $\leqq k3,i$; **b)** each bit string of sort b2 (for an *r*3,*i*-tuple of SO relations R with parameters *r3,i , k3,i*) of size *r*3,*i* * *sizek*3,*i* (**I**), represents one of the possible *r*3,*i*-tuples of *r*3,*i*-ary (SO) relations on I, closed under $\equiv k3,i$; **c)** each bit string of sort b1 (for a TO relation θi of type τi, and parameter *k3,i*) of size 2 raised to the exponent *r*3,*i* * *sizek*3,*i* (**I**), represents one of the possible sets of *r*3,*i*-tuples of *r*3,*i*-ary (SO) relations on I, closed under $\equiv k3,i$, i.e., one of the possible *TOω* relations on I, of type τi, closed under $\equiv k3,i$. Let b be a bit string of sort b1. Each bit in b represents one of the possible bit strings of sort b2 of size *sizek*3,*i* (**I**). The leftmost bit in b represents a bit string of type b2 that has all its bits 0, i.e., it is the bit string that corresponds to the *r*3,*i*-tuple formed by *r*3,*i* empty *r*3,*i*-ary (SO) relations. The following bits in

b represent the bit strings of sort b2 that correspond to the order in all the possible bit strings of sort b2 according to their binary value. And so on, up to the rightmost bit in b, which represents a bit string of sort b2 that has all its bits 1 (i.e., it is the bit string that corresponds to the $r3,i$-tuple formed by $r3,i$ copies of the $r3,i$-ary relation that has the $r3,i$-tuples in all the equivalence classes in the relation $\equiv k3,i$). Then, $M\varphi$ guesses s bit strings of sort b1, one for each one of the $TO\omega$ relations θi. Note that this is done in time $2c \cdot sizek3,i$ (**I**), and hence also in time $2d \cdot sizek$ (**I**), since $k3,i \leq k$ (see above), for some constants c, d. **4):** Regarding the $SO\omega$ variables quantified in the $\Sigma 1$, ω formula ψ, to interpret each of them we *build all* the possible $SO\omega$ relations of the corresponding arity and closed under the corresponding equivalence class in the the rs of $M\varphi$. We build those relations by stepping in the equivalence classes of tuples $\equiv k2,ij$ according to the total orders induced by the corresponding pre-orders $\leqq k2,ij$. The details on how to do that are equal to the algorithm used in [FT,08] to prove $\Sigma 1, \omega \subseteq NTIMEr((sizek)c)$. Note that we can afford to do that because for each SO variable Yij the number of such relations is bounded by $2d \cdot sizek$ (**I**), corresponding to $k2,ij$ and hence also by $2d \cdot sizek$ (**I**), since $k2,ij \leq k$ (see above), for some constant d that depends on the arity. Then, for each SOω variable Yij we will require that either *for all* the generated relations, or for *at least one* of them, depending on the corresponding quantifier being $\forall$ or $\exists$, respectively, the formula ϕ is true. **5): Evaluation of ϕ:** Recall that ϕ is an FO formula with atomic TOω formulas. To evaluate ϕ we consider the syntax tree of ϕ, $T\phi$, and evaluate one node of it at a time in the finite control of $M\varphi$, in a bottom up direction. To that end, for every node α in $T\phi$ that represents a sub-formula with $r \geq 1$ free FO variables, we define in the rs an r-ary relation variable $R\alpha$. And for every node α in $T\phi$ that represents a sub-formula with *no* free FO variables, we define in the rs a 1-ary relation variable $B\alpha$ that represents a Boolean variable, which we interpret as True if $B\alpha = dom(\mathbf{I})$, and as False if $B\alpha = \emptyset$. Note that all the SOω relations that appear in the nodes in $T\phi$ are in the rs of $M\varphi$. Every node in $T\phi$ is of one of the following kinds: i) an atomic FO formula with a relation symbol either in σ or quantified by an SOω quantifier in ψ, ii) a $\vee$ connective, iii) a $\wedge$ connective, iv) a $\neg$ connective, v) an existential FO quantifier, or vi) an atomic TOω formula with a relation symbol quantified by a TOω quantifier in φ. We omit the details on how to evaluate the nodes of the first 5 kinds, since they are straightforward, and focus on the nodes that correspond to atomic TOω formulas. Suppose a given node α in $T\phi$ corresponds to the sub-formula χi ($V1, \ldots, Vr3,i$) with parameters τi, $k3,I$ for χi , *and* $r3,i$,$k3,i$ for $V1, \ldots, Vr3,i$ with $\tau i = (r3,i, \ldots, r3,i)$, of cardinality $r3,i$ with $r3,i \leq k3,i$ as stated in the beginning of the proof, and where $V1, \ldots, Vr3,i$ are either relation symbols inσor quantified by an SOω quantifier in ψ. We check whether or not the $r3,i$-tuple of relations ($V1, \ldots, Vr3,i$) is in the TOω relation θi guessed above for the variable χi , using the (guessed) bit string b1 that represents θi , with the following algorithm, that clearly runs in time

$2c \cdot sizek$ (**I**), corresponding to $k3,i$ and hence also by $2d \cdot sizek$ (**I**), for some constants c, d:

- $B\alpha \leftarrow \emptyset$ (*i.e.*, $B\alpha \leftarrow FALSE$);
- **for all** bit strings of sort b2, counting in binary with index n (i.e., for all $r_{3,i}$-tuples of $r_{3,i}$-ary (SO) relations closed under $\equiv_{k3,i}$);
  - **for** j = 1 through $r_{3,i}$ (i.e., the j-th component in the tuple of (SO) rel.);
    - $S_{i,j} \leftarrow \emptyset$ (with parameters $r_{3,i}$,$k_{3,i}$);
    - **for** l = 1 through $size_{k3,i}$ (**I**), (i.e., bit l in bit substring of sort b3 for the SO relation $S_{i,j}$);
      - **if** bit m of bit string a_n is 1, where $m = (j - 1) \cdot size_{k3,i}(\mathbf{I}) + l$, (i.e., bit m in a bit string of sort b2);
        - add to $S_{i,j}$ the l-th equivalence class in $\equiv_{k3,i}$, according to pre-order $\leqq_{k3,i}$ (i.e., all the $r_{3,i}$-tuples of elements in that class);
      - **end** l;
    - **end** j;
  - **if** bit n in bit string b1 = 1
- **if** ($V_1 = S_{i,1} \wedge \ldots \wedge V_{r3,i} = S_{i,r3,i}$)
  - $B\alpha \leftarrow dom(\mathbf{I})$ (*i.e.*, $B\alpha \leftarrow TRUE$);
- **end all**;

5. Conclusions

From Theorems 1, 2, 4, and Corollary 3, we have the following result:

Corollary 5 Let M3 be a 3-NRM that works in $NTIME3,r(2c \cdot (sizek))$, for some positive integer c, that computes a Boolean query q. Then, there is an *NRM M*2 that works in $NTIMEr(2d \cdot (sizek))$, for some positive integer d, that also computes q.

This is very interesting, since in the general case it is *much easier* to define an NRM using TO relations in its rs, and TO formulas to access them, than restricting the machine to SO relations in its rs, and SO formulas. Then, to prove that a given query is computable by an NRM it is enough with showing that it can be computed by a 3-NRM. Note however, that we think that we still *need* 3-NRMs as well as the third order relational complexity class *NEXPTIME*3,*r*, if we need to work with *oracle* NRMs with third order relations, since as the oracle cannot access the tape of the base machine (see [FT,08]), there seems to be no way to pass the bit strings that represent TO relations from the base to the oracle.

Recall that it has been proved that RMs have the same computation, or expressive power, as the (effective fragment of the) well known infinitary logic with finitely many variables $L\omega\infty\omega$ ([AVV,95]). On the other hand, analogously to the well known result that states that the computation power of deterministic and non deterministic Turing machines is the same, it is straightforward to see that any NRM *M*N can be simulated by a (deterministic) RM *M*D working in relational time exponentially higher, just by checking in *M*D all possible transitions

instead of guessing one in each non deterministic step of the transition relation of MN. Then, the following is immediate:

Corollary 6: $\Sigma_{2,\omega} \subseteq$ (effective fragment of) $L\omega\infty\omega$.

Finally, in [GT,10], the logic SOF was introduced and defined as a semantic restriction of SO where the valuating r-ary relations for the quantified SO variables are closed under the relation ≡F of equality of FO types in the set of r-tuples of the structure. It was shown there that its existential fragment Σ1,F is not included in Lω∞ω, as opposite to Σ1,ω which is. Then we have:

Corollary 7: $\Sigma_{1,F}$NOT $\subseteq \Sigma_{2,\omega}$.

References

1. [AT,14] J. Arroyuelo, J. M. Turull-Torres, "The Existential Fragment of Third Order Logic and Third Order Relational Machines", in "Proceedings of the XIX CACIC", ISBN 978-987-3806-05-6, Buenos Aires, October 20-24, p. 324-333, 2014.
2. [AV,91] Abiteboul, S., Vianu, V., "Generic Computation and its Complexity", STOC 1991.
3. [AVV,95] Abiteboul, S., Vardi, M., Vianu, V., "Computing with Infinitary Logic", Theoretical Computer Science 149, 1, pp. 101-128, 1995.
4. [AVV,97] Abiteboul, S., Vardi, M. Y., Vianu, V., "Fixpoint logics, relational machines, and computational complexity", JACM 44 (1997) 30-56.
5. [Daw,98] Dawar, A., "A restricted second order logic for finite structures". Informa- tion and Computation 143 (1998) 154-174.
6. [Fag,74] Fagin, R., "Generalized First-Order Spectra and Polynomial-Time Recognizable Sets", in "Complexity of Computations", edited by R. Karp, SIAM-AMS Proc., American Mathematical Society, Providence, RI, pp. 27–41, 1974.
7. [FPT,10] F. A. Ferrarotti, A. L. Paoletti, J. M. Turull-Torres, "Redundant Relations in Relational Databases: A Model Theoretic Perspective", Journal of Universal Computer Science, Vol. 16, No. 20, pp. 2934-2955, 2010. http://www.jucs.org/jucs 16 20/redundant relations in relational
8. [FT,08] Ferrarotti, F. A., Turull-Torres, J. M., "The Relational Polynomial-Time Hierarchy and Second-Order Logic", invited for "Semantics in Databases", edited by K-D. Schewe and B. Thalheim, Springer LNCS 4925, 29 pages, 2008.
9. [GT,10] Grosso, A. L., Turul-Torres J. M., "A Second-Order Logic in which Variables Range over Relations with Complete First-Order Types", 2010 XXIX International Conference of the Chilean Computer Science Society (SCCC) IEEE, p. 270-279, 2010.
10. [Lib,04] Libkin, L., "Elements of Finite Model Theory", Springer, 2004.
11. [Tur,06] J. M. Turull-Torres, "Relational Databases and Homogeneity in Logics with Counting", Acta Cybernetica, Vol 17, number 3, pp. 485-511, 2006.
12. [Tur,16] J. M. Turull-Torres, "Capturing relational NEXPTIME with a Fragment of Existential Third Order Logic", in "Journal of Computer Science and Technology", selected in CACIC 2015, for its publication in JCST, 6 p., Vol. 15, N. 2, November 2015. http://journal.info.unlp.edu.ar/?page_id=1469

Keyword Identification in Spanish Documents using Neural Networks

GERMÁN AQUINO[1, 2] AND LAURA LANZARINI[1]

[1]Instituto de Investigación en Informática – III–LIDI Facultad de Informática – Universidad Nacional de La Plata - Argentina
[2] CONICET– Consejo Nacional de Investigaciones Científicas y Técnicas
{gaquino, laural}@lidi.info.unlp.edu.ar

Abstract. *The large amount of textual information digitally available today gives rise to the need for effective means of indexing, searching and retrieving this information. Keywords are used to describe briefly and precisely the contents of a textual document. In this paper we present an algorithm for keyword extraction from documents written in Spanish.This algorithm combines autoencoders, which are adequate for highly unbalanced classification problems, with the discriminative power of conventional binary classifiers. In order to improve its performance on larger and more diverse datasets, our algorithm trains several models of each kind through* bagging.

Keywords: *Keyword Extraction, Neural Networks, Autoencoders.*

1. Introduction

The large amount of textual information digitally available today gives rise to the need for effective means of indexing, searching and retrieving text documents quickly and without having a user to read them entirely, which in many cases is not feasible. Keywords are used to describe briefly and precisely the contents of a text document, so that a user can find documents relevant to him/her without having to read them beforehand. Keywords are widely used in search engines as they help in the process of searching, indexing, and retrieving information [1]. However, there are many documents without keywords and the task of manually assigning keywords to them is slow, difficult and highly subjective. For this reason it is beneficial to have tools that assist professional indexers by providing a list of terms candidates to be keywords [2].

In this paper a new algorithm for keyword extraction from text documents written in Spanish language is presented. This algorithm is based on a classification model capable of learning the structural features of the terms considered keywords, and to recognize terms having these features in unseen documents. A combination of discriminant classifiers and autoencoders is used to build a classification model that assigns a score to each term of a document. This score is used to construct a ranking of the terms considered most informative for a given document.

This paper is organized as follows. Some algorithms for keyword extraction are described in Section 2. The proposed algorithm is explained in detail in Section 3. The results of the experiments carried out are presented in Section 4, and Section 5 summarizes the obtained conclusions and future work.

2. Related Work

The problem of keyword extraction has been treated from the machine learning discipline since a few decades ago [2][3][4]. This approach aims to transform text data into a structured representation suitable for learning algorithms. Such algorithms work with a feature set computed for each term of a document and consider keyword extraction as a classification problem, determining whether each term is a keyword or not. Supervised learning methods usually use the terms designated as keywords by the authors of the training documents as examples of one class, and the rest of the terms as examples of the other class. The class of the terms that are not keywords is naturally much more numerous than the other class. This imbalance in the number of elements of each class and the inherent ambiguity of natural language makes keyword extraction a very difficult problem to solve. Many of the mistakes made by the keyword extraction algorithms, specially those which apply supervised classification schemes, are due to redundancy (in the case of several semantically-equivalent terms are selected) and over-generalization (in the case of selection of terms that contain important terms but are not keywords themselves). The flexibility of the vocabulary used and the ambiguity of the human language makes very difficult for automatic classifiers to distinguish between two seemingly equivalent terms, and to see a relation between subtly related terms [5].

In order to find a suitable representation for learning algorithms, many keyword extraction methods apply *stemming*, which consists of reducing each term to its morphological root, and filter terms using a *stoplist*, which is a list of terms with low semantic value (*stopwords*) such as articles, prepositions, conjunctions and pronouns.

One of the first advances in considering keyword extraction as a classification problem to be solved through machine learning was reported by Peter Turney [2]. Turney developed an algorithm called GenEx that applies a set of rules whose parameters are tuned in a first stage using a genetic algorithm. These rules are used to rank terms and select the ones that have the highest score in the second stage. GenEx has a pre-processing step in which *stemming* is applied to terms and *stopwords* are filtered.

Among the most recent algorithms for keyword extraction there is Maui, developed by Olena Medelyan [6][7]. Maui is also a supervised classification algorithm that computes a set of features for the candidate terms. Maui uses a *stemmer* and a *stoplist* of the given language and it is built on top of the machine learning platform Weka [8] and uses bagged decision trees to classify terms.

In a previous work [9] we introduced a keyword extraction algorithm that relies on auto-associative neural networks or autoencoders [10] to identify keywords. This algorithm uses only the elements belonging to the minority class, the class of the keywords, to build a recognition model as opposed to discriminative models obtained using conventional neural networks and other machine learning algorithms. The autoencoder approach has the advantage that it handles naturally the imbalance inherently present in the keyword extraction problem, and also it enables to control the number of keywords extracted from each document and to rank them. Also, it is much faster than other algorithms as it processes only the examples of the minority class.
The algorithm presented in this paper is also a supervised machine learning algorithm, and it is an improvement over our previous approach as it combines qualities of both discrimination-based (supervised) and recognition-based (unsupervised) classifiers in order to improve performance on larger and less regular datasets. The potentially large variance present in the training and testing examples is handled through the use of *bagging* [11] in order to average the classification decisions of different classifiers. As its predecessor, the proposed algorithm does not use *stoplists* to rule out insignificant or malformed terms but instead it applies *part-of-speech (POS) tagging* to allow the correct identification of noun phrases present in the text.

3. Description of the Algorithm

In this work autoencoders are used to classify terms in two classes, 'keyword' and 'non-keyword'. Autoencoders are adequate for unbalanced classification problems and one-class recognition problems [12]. To enhance their recognition capabilities, several autoencoders are combined by the use of bagging and also a set of discriminant classifiers is used.
An autoencoder processes examples of only one class. Autoencoders try to find an approximation of the training set to itself, finding in the process an approximation to the identity function of such training set. This allows them to assign a reconstruction error that characterizes the similarity between a new element and the training set. On the other hand, discriminant classifiers attempt to find a possibly non-linear boundary in the feature space of the training examples in order to define regions in such space for each class. Here, the decision of discriminant classifiers is used to weight the reconstruction error assigned to the examples by the autoencoders. Both kinds of classifiers made decisions through a voting scheme, which will be explained further in Section 3.3.

3.1 Pre-processing

The first step of the proposed algorithm consists in splitting the text in sentences and words using two list of delimiters provided as parameters. These delimiters can be any character sequence and will not be part of

extracted terms. Once the sentences and words are obtained the algorithm proceeds to compute the features for the terms.
Terms are represented by N-grams, which are sequences of N consecutive words in the same sentence, and for each one we compute a set of features relative to position and frequency of the term in the document. In this work we will use 'term' and 'N-gram' interchangeably. In the N-grams extraction task the Fürnkranz algorithm [13] is applied for avoiding the generation of every possible N-gram from the text and increasing the efficiency in the generation of N-grams. This algorithm requires the specification of the maximum length of the terms considered and the minimum frequency such terms must have in a document to be eligible as keywords.
In order to further reduce the number of terms to be processed, after the feature calculation phase we apply a filter which discards N-grams that do not start or end with nouns or adjectives. This filtering discards sequences of words that are not eligible as keywords, for example 'de forma que'. This process is similar to the application of a *stoplist*, with the difference that we do not use an exhaustive list of terms to rule out but instead we assign *POS* tags to each word of the document based on its use. To this end we apply a *maximum entropy* model trained with the tool OpenNLP [14] using a tagged corpus as training set. This filtering greatly reduces the required processing time, since it discards an important number of terms that should not be considered as keywords.
The POS tagging model for Spanish was trained using the tagged corpus Conll-2002 [15] and the grammatical tags defined by the EAGLES group [16]. The corpus was provided in the 2002 *Conference on Computational Natural Language Learning* to be used to train and evaluate algorithms of Named Entitity Recognition (NER), which is the problem of finding person names, places, organizations and similar information in the text.

3.2 Term characterization

The features computed for each N-gram consist of several frequential and positional quantities extracted from the text. Most of these features are computed using only the information present in each document, but some of them require the processing of the entire training corpus for their computation. The features are:

1. **Term length:** the number of individual words composing the N-gram.
2. **Term Frequency (TF):** the rate between the frequency of the term and the number of words in a document.
3. **Inverse Document Frequency:** it measures how common is a given term by counting how different documents in the corpus contain it.
4. **Term Frequency – Inverse Document Frequency (TF-IDF) [17]:** consists in weighting the term frequency with the inverse document frequency. TF-IDF favors terms that are infrequent in the corpus but frequent in the given document.
5. **First Occurrence:** the relative position of the first occurrence of the term in the text. It is calculated as the ratio between the number of words

that appear before the first occurrence of the given term and the number of words of the document.

6. **Position in Sentence:** a measure of the relative position of a term in the sentences it appears in. For each sentence *s* that contains term *t*, we count the number of words that appear in *s* before *t*, and we average these values.
7. **Occurrence in Title:** this attribute is set to 1 if the term appears literally in the document title and 0 otherwise. It represents the notion that terms appearing in the title are important and hence are candidates to be keywords.
8. **Occurrence of Members in Title:** this attribute, like the previous one, relates the importance of a term with its appearance in the title. The difference is that this attribute considers occurrences in the title of the individual words of the term. This allows considering terms whose occurrences in the title are not literal, such as when the words are in a different order or that have more or less lexical words. It is the ratio between the number of words of a term *t* that appear in the title and the length of *t*.
9. **Normalized Sentence Length:** it is a measure of the length of the sentences in which a given term appears in, calculated by averaging the lengths of these sentences. Such lengths are also normalized by dividing them by the length of the longest sentence in the document.
10. **Normalized Frequency (Z-Score) [18]:** consists in normalizing the term frequency using its mean frequency in the training corpus and its standard deviation. It measures the difference between the frequency of a term and its mean frequency in the corpus.
11. **Last occurrence:** the last position in the text in which the term appears.
12. **Spread:** the difference between first and last occurrences.
13. **Normalized frequency:** the frequency of the term normalized by the highest frequency of any term in the document.
14. **Lowest position in sentence:** considering all the positions a term occupied in each of its sentences, this is the closest to the beginning of the sentence, normalized using the sentence length.
15. **Highest position in sentence:** similar to the previous one, but considering the position closest to the end of the sentence.
16. **Shortest sentence length:** the length of the shortest sentence a term appears in, normalized by the highest length of any sentence.
17. **Longest sentence length:** similar to the previous one, but considering the longest sentence a term appears in.
18. **Log frequency:** a non-linear monotonic function is applied to the term frequency in order to reduce the impact of its absolute value but at the same time to keep its magnitude.
19. **Condition of being a named entity:** this is a boolean feature that indicates if the term is a named entity or not. To identify named entities in the document a NER OpenNLP model is applied.
20. **Keyphraseness [3]:** the number of times a given term was chosen as a keyword in the training set. It makes sense if the testing documents belong to the same domain as the training documents, which should be the case to obtain a reasonable performance.

3.3 Keyword Identification

As mentioned earlier, the proposed method is a supervised classification algorithm. It uses the feature vectors of the terms of the training document set in order to build a classification model to be applied to the feature vectors of a testing document set.

In the proposed method three ensembles of classifiers are used. The first ensemble is composed of conventional bagged *multi-layer perceptrons*, trained using sampling with replacement from the training set. In order to cope with the imbalance problem, the number of elements that are sampled from the majority class is proportional to the sampled number of elements in the minority class. As all of these sampled smaller training sets are different, the resulting classifiers will yield different views on the original feature space. Given the large variance present in the problem domain and the intrinsic non-deterministic nature of neural networks, *bagging* helps to improve the performance of the obtained models, giving more consistent and more robust predictions. These classifiers are trained to distinguish important terms from non-important ones.

The other two ensembles are composed of autoencoders. The first of these two ensembles attempts to characterize the set of elements belonging to the minority class (the positive set), which in our case are the feature vectors of the terms designed as keywords in the training set. The other ensemble attempts to characterize the set of elements belonging to the majority class (the negative set), which is naturally much more diverse. Both ensembles are also trained applying *bagging*, and the autoencoders of the majority class are trained with larger samples in order to provide more accurate estimates of the complete set.

Autoencoders are neural networks that have as many output units as they have input units, so given an input vector X they can produce an approximate vector X'. The difference between the original vector and the approximate vector can be characterized by the reconstruction error, which is the sum of the squared differences between both vectors. As training is carried out using the elements of the class of interest it is expected that new elements that are similar to the ones in the training set have a lower reconstruction error than those that are not.

The autoencoders are trained in the same way as conventional neural networks. In this work we used *Resilient Backpropagation* [19] as training algorithm, both for the autoencoders and the multi-layer perceptrons. This algorithm allows a faster convergence, providing better results, and at the same time it eliminates the need to specify a learning rate.

As we mentioned earlier, the autoencoder assigns a reconstruction error to each element of a testing set, which represents the similarity between the element and those of the training set. Instead of determining a cutoff threshold to accept or reject a term as keyword we opted to select the R terms with lowest reconstruction error from each document of the testing set. As we are using two sets of autoencoders, one for the positive class and one for the negative class, we have two scores for each term of the testing set. Let P_d be the

reconstruction error of the term in respect to the positive set, and Neg_e the reconstruction error in respect to the negative set. An informative term should minimize Pos_e, as it should be similar to the elements in the positive set, and at the same time it should maximize Neg_e, its dissimilarity to the negative set. Hence, an informative term should minimize $Pos_e - Neg_e$, and this is the score used to construct the term ranking. The selection scheme employed gives preference to the terms chosen by the discriminant classifiers as informative terms, and then their reconstruction error is considered.

The use of the reconstruction error as a selection mechanism provides two benefits: first, we obtain a *ranking* of the extracted terms, and second, it is guaranteed that each document of the testing set will have terms to represent it, which does not necessarily hold with the use of a global threshold or a discriminant classifier. Besides, R is a parameter of the algorithm which gives more control and allows the user to adjust the output of the algorithm when more precision or more recall is preferred. By default, the number of terms to extract is the average number of keywords of the documents of the training set.

4. Experimental Results

Some experiments were carried out to assess the performance of the proposed method. A dataset formed by a set of scientific articles published between 2005 and 2013 in Argentine Congress of Computer Science (CACIC) [20] was used in these experiments. The dataset includes 888 documents written in Spanish language and contains 130792 terms from which 1683 are labeled as keywords, giving an imbalance rate of 1.28%, that is, less than 2% of all terms belong to the minority class. We also used a dataset composed of 166 scientific articles from the Workshop of Researchers in Computer Science (WICC) [21]. This dataset was used to measure the performance of the previous version of our method [9], and it is used here to assess that the new version is indeed superior.

The metrics used were precision, recall and f_1-measure calculated for each of the four algorithms.These metrics were applied considering as a hit the match between a term selected by an algorithm and a term designated as keyword by the authors of the given document. Thus, a false positive occurs when a method identifies as keyword a terms that is not included in the list of keywords by the author, and a false negative when the method fails to extract a keyword contained in that list. In our case precision measures the proportion of extracted terms that match assigned keywords, and recallmeasures the proportion of keywords correctly identified by the method. F_1-measure is the harmonic mean between precision and recall, and therefore it is a good measure of the global performance of a given method.

The evaluation methodology we applied is 10-fold cross validation. This evaluation process was repeated 30 times to obtain a significative sample over which we can average the results. We configured both algorithms to extract 5 keywords as this is the average number of keywords per document on the dataset.

In our experiments we used 15 multi-layer perceptrons as discriminant classifiers, 5 autoencoders for the positive set, and 10 autoencoders for the negative set. All these neural networks were trained using 20 hidden neurons, a maximum of 50 epochs, and the logistic function as activation function in the hidden and output layers. The implementation used of Maui is the one developed by its authors. For Maui we applied the Spanish stemmers and stoplists provided with the implementations. For the previous version of our method, the autoencoder was configured to use 15 hidden neurons, a maximum of 100 epochs, and the same activation functions as the new version. In these experiments the terms extracted by all methods have a maximum length of 4 words and a minimum frequency of 3 occurrences in their respective documents.

The results of the 30 runs of the cross-validation for each algorithm on each dataset are shown in the Figure 1, identifying the proposed algorithm as AE*, for autoencoder. The previous version of our method is simply denoted as AE.

The tests results show that the proposed algorithm outperforms Maui on these datasets. It can be seen also that it handles properly larger and more diverse datasets than its predecessor. One of the main goals of our algorithm is to capture the largest possible number of descriptive terms, and this goal is quantified by the recall metric. A high recall is important because it allows capturing the maximum possible of eligible terms, which in turn gives the possibility of suggesting descriptive terms that were not chosen by the authors. However, getting a high recall at the expense of precision is not beneficial, since the quality of the extracted terms will be inferior. Therefore it is necessary to find a balance between precision and recall.

In order to verify that these differences are statistically significant, we ran a Kolmogorov-Smirnov test on the results of the precision, recall and f-measure obtained from the cross-validation procedure for both methods, and we ran a t-test on the difference of the means of the samples for the three metrics. The tests showed that the mean for the three metrics obtained by our method are higher than the ones obtained by Maui with a significance level of 0.05, as the obtained p-values are 1.3669e-30, 3.7699e-40 and 4.2676e-35 respectively.

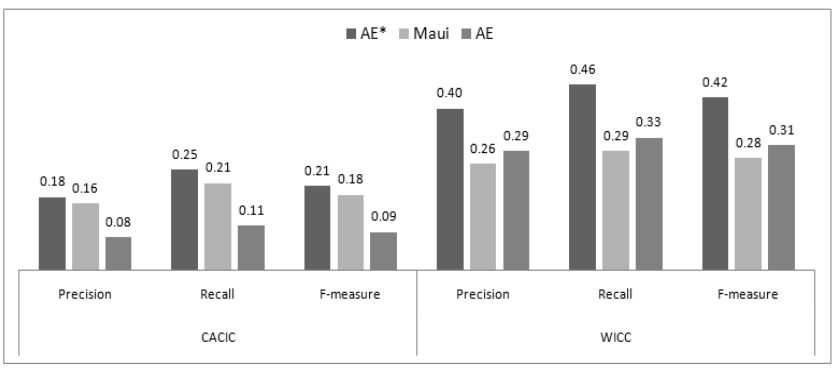


Figure 1. *Average precision, recall and f_1-measure of the three methods on the CACIC dataset.*

In the Table 2 there are shown the lists of keywords extracted of both methods for a set of documents from the CACIC dataset, and these keywords are compared to the real keywords assigned by the authors of the respective documents. The matches between an extracted keyword and a real one are highlighted in bold. It is important to notice that some of these documents have fewer keywords than the specified number of keywords to extract. This necessarily means that the methods will have false positives errors, despite the selected terms may be considered descriptive by a human observer. It is also noteworthy that some terms are semantically equivalent to the true keywords, but as they are not exact matches are hence considered false positives too. The high variability of the keyword assignment criteria of the authors, combined with the ambiguity of the human language contributes to the high difficulty of the keyword extraction problem. These issues could be addressed by the use of semantic knowledge bases that could map related terms to the same concept, and by the definition of more advanced scoring criteria for performance assessment than exact matching.

Table 2. Comparative results of the keyword extraction methods performance on some sample cases.

Documents in dataset	Keywords assigned by authors	Keywords extracted by AE*	Keywords extracted by Maui
Una implementación paralela de las Transformadas DCT y DST en GPU.	-procesamiento paralelo **-GPU** **-CUDA** **-procesamiento de señales** **-DCT**	-transformadas **-GPU** **-CUDA** -GPU CUDA **-procesamiento de señales**	-MPI **-DCT** -transformadas -DST **-CUDA**
Programación híbrida en clusters de multicore.	-arquitecturas paralelas **-programación híbrida** **-cluster** **-multicore** **-jerarquía de memoria**	**-cluster** **-multicore** -programación **-programación híbrida** **-jerarquía de memoria**	**-jerarquía de memoria** **-cluster** **-multicore** -pasaje de mensajes -caso de estudio
Evaluación de variantes en modelo destinado a anticipar la conveniencia de trazar proyectos de software.	**-ingeniería de software** **-análisis ROC** **-trazabilidad de requerimientos**	-trazabilidad -métricas **-análisis ROC** **-ingeniería de software** **-trazabilidad de requerimientos**	-ROC -trazabilidad -métricas -variantes -factores
Autorregulación del aprendizaje en entornos mediados por TIC.	**-autorregulación** **-TIC** **-aprendizaje**	**-autorregulación** **-TIC** **-aprendizaje** -intervención -autorregulación del aprendizaje	**-aprendizaje** **-TIC** -propuesta de intervención **-autorregulación** -intervención
Integración segura de MANETs con limitaciones de energía a redes de infraestructura.	-MANET **-bluetooth** **-IPSec** **-energía** **-seguridad**	**-bluetooth** **-IPSec** -MANETs **-energía** -ad hoc	**-seguridad** **-Bluetooth** **-IPSec** -consumo -consumo de energía

5 Conclusions and Future Work

In this paper we presented a new algorithm for keyword extraction from Spanish documents. The main feature of our proposal is the use of autoencoders to capture the properties of important terms, yielding comparable or even better results than other well known keyword extraction algorithms. Autoencoders classification decisions are further reinforced by the use of discriminant classifiers. We consider important to achieve a high recall so that the algorithm can capture more terms eligible by different human observers, with the goal to act as a recommendation system of possible keywords. The only language-dependent of our method are the POS tagging and NER models, thus replacing these models with models trained with documents in another language would allow us to apply our method in such language.

Given that the number of terms to extract is a parameter of the algorithm the user can adjust the expected level of precision or recall from the terms suggested by the system.

We are currently working on the term representation to include features related to the grammatical structure of a given language, as the use of parsing trees in order to find head noun phrases in sentences. We are also interested in incorporating the use of knowledge bases in order to find semantic relations between pairs of terms and to identify their degree of generality or specificity in a given domain.

References

1. Gutwin, C., Paynter, G., Witten, I., Nevill-Manning, C., Frank, E.: Improving Browsing in Digital Libraries with Keyphrase Indexes. Journal of Decision Support Systems, Vol.27, no 1-2, pp.81--104. (1999)
2. Turney, P.D.: Learning Algorithms for Keyphrase Extraction. Information Retrieval, vol. 2,303--336 (2000).
3. Witten, I. H., Paynter, G. W., Frank, E., Gutwin C., Neville-Manning, C. G.: KEA: Practical Automatic Keyphrase Extraction. In Proceedings of the 4th ACM Conference on Digital Libraries, pp. 254--255 (1998).
4. Hulth, A.: Improved automatic keyword extraction given more linguistic knowledge. In Proceedings of the 2003 Conference on Empirical Methods in NLP, pp. 216--223 (2003).
5. Hasan, K. S., Ng V.: Automatic Keyphrase Extraction: A Survey of the State of the Art. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1262--1273 (2014).
6. Medelyan, O.: Human-competitive automatic topic indexing. Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 3, pp. 1318--1327, Association for Computational Linguistics (2009).
7. Kim, S. N., Medelyan, O., Kan, M., Baldwin, T. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In Proceedings of the 5th International Workshop on Semantic Evaluation. pp. 21--26 (2010).
8. WEKA, http://www.cs.waikato.ac.nz/ml/weka/, accessed in July 2015.

9. Aquino, G, Hasperué, W, Lanzarini, L. Keyword Extraction using Auto-associative Neural Networks. XX CongresoArgentino en Ciencias de la Computación (2014).
10. Japkowicz, N, Myers, C, Gluck, M.: A Novelty Detection Approach to Classification. Proceedings of the Fourteenth Joint Conference on Artificial Intelligence, pp. 518--523 (1995).
11. Breiman, L.: Bagging Predictors. Machine Learning, pp. 123--140 (1996).
12. Japkowicz, N.: The Class Imbalance Problem: Significance and Strategies. Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI), pp. 111--117 (2000).
13. Fürnkranz, J.: A Study Using n-gram Features for Text Categorization (1998).
14. OpenNLP, http://opennlp.apache.org/, accessed in July 2015.
15. Conference on Computational Natural Language Learning (CoNLL-2002), http://www.clips.ua.ac.be/conll2002/ner/, accessed in July 2015.
16. Expert Advisory Group on Language Engineering Standards (EAGLES), http://www.ilc.cnr.it/EAGLES96/home.html, accessed in July 2015.
17. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management, pp. 513--523 (1988).
18. Andrade, M.A., Valencia, A.: Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. Bioinformatics, vol. 14, no. 7, pp. 600--607 (1998).
19. Riedmiller, M.: Advanced Supervised Learning in Multi-layer Perceptrons - From Backpropagation to Adaptive Learning Algorithms (1994).
20. Congreso Argentino en Ciencias de la Computación, http://redunci.info.unlp.edu.ar/cacic.html, accessed in July 2015.
21. Workshop de Investigadores en Ciencia de la Computación, http://redunci.info.unlp.edu.ar/wicc.html, accessed in July 2015.

An experimental study for the Cross Domain Author Profiling classification

MARÍA JOSÉ GARCIARENA UCELAY, MARÍA PAULA VILLEGAS, LETICIA CECILIA CAGNINA AND MARCELO LUIS ERRECALDE

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Facultad de Ciencias Físico, Matemáticas y Naturales,
Universidad Nacional de San Luis – Ejército de los Andes 950
(D5700HHW) – San Luis – Argentina, Tel.: (0266) 4420823 / Fax: (0266) 4430224
emails: {mjgarciarenaucelay, villegasmariapaula74, lcagnina, merrecalde}@gmail.com

Abstract. Author Profiling *is the task of predicting characteristics of the author of a text, such as age, gender, personality, native language, etc. This is a task of growing importance due to the potential applications in security, crime detection and marketing, among others. An interesting point is to study the robustness of a classifier when it is trained with a dataset and tested with others containing different characteristics. Commonly this is called* cross domain *experimentation. Although different cross domain studies have been done for datasets in English language, for Spanish it has recently begun. In this context, this work presents a study of cross domain classification for the author profiling task in Spanish. The experimental results showed that using corpora with different levels of formality we can obtain robust classifiers for the author profiling task in Spanish language.*

Keywords: *Author Profiling, Natural Processing Language, Cross Domain Classification*

1. Introduction

The evolution of the World Wide Web sites to the Web 2.0 has mainly implied a proliferation of contents created and shared from all kinds of users in different social networks. Also, it has facilitated the increment of falsification of identity, plagiarism and a significant increase in the traffic of spam data through the Internet. For this reason, automatic methods are needed to detect if a given text belongs to a specific author, if the gender and age stated by a user of social media is compatible with his/her writing style, etc. In this context, the Author Profiling task refers to the identification of different demographic aspects like gender [1], age [2, 3], native language [4], emotional state [5, 6] or personality [5, 7] of an anonymous author of a text [8].
A particular problem concerned with the author profiling task in Spanish language is the lack of data for experimentation. For that, it is important to take in advantage of all the available data in order to obtain good and

enough general classifiers for the task and then, to use those for new data that can be collected.

Traditional machine learning methods construct reliable and accurate models using available labeled data. These models are generally tested with data drawn from the underlying distribution or domain. Then, a classification model working well in one domain could not work as well in another one [9]. *Cross domain* classification is used to tackle that problem.

For *domain* we can consider the source of the documents (Twitter, blogs, chats, magazines, news, etc) [9], topics (places, politic, food) [10], products (books, furniture, movies) [11], research areas (computer science, biology, physics) [11], etc. In the present work we define the domain such as the level of "informality" of a text.

In PAN-2014 competition an extra experiment of cross domain was held, for both English and Spanish languages, which served as a previous work [12]. Thus, here we perform several experiments in order to determine the corpus we can obtain a general classifier with.

In this paper we present the results obtained from carrying out cross domain experiments. Such tests have not been previously performed in Spanish due to the lack of resources in this language and because training with a corpus and then testing with another is a recently studied approach. However, cross domain experimentation becomes an interesting field for researchers working in actual classification tasks as author profiling is. We have used available corpora provided for PAN competitions (2013 and 2014) which present a high level of informality in the texts contained. Also we have considered a formal corpus named SpanText [13] with similar characteristics with respect to those of PAN competitions, in terms of genre and age of people who wrote the texts. The results obtained with the experimentation with cross domain in terms of informality of the texts demonstrate that reliable classifiers can be obtained for the author profiling classification task.

The cross domain experiments may be helpful for other tasks, besides contributing to the author profiling itself. For example, it could be used to generate a classifier from a large collection of different types of texts, properly selected. Then, that classifier could be used to analyze texts hard to obtain or for analyzing online data.

The rest of the paper is organized as follows. In Section 2, we briefly introduce the author profiling task and some concepts related to cross domain experiments. In Section 3, the main characteristics of the different data collections used in the experimentation are presented. Section 4 describes an experimental study about cross domain among the available corpora in Spanish language. Finally, in Section 5 some conclusions are drawn and future works are proposed.

2. Author Profiling Task

Nowadays, the evolution of the Web sites on the Internet and the increasing use of social networks like Facebook and Twitter have made available a huge

amount of information. A large part of this information is in plain text and it can be used to infer about the writer. The Author Profiling Task (APT) consists in knowing as much as possible about an unknown author, just by analyzing the given text [5]. In this regard, profiling tries to determine the author's gender, age, level of education, geographic origin, native language and personality type [1-8].

The APT has mainly focused on documents written in English but, according to our knowledge, this situation has started to change with papers presented at PAN-2013 competition [14], when the organizers considered the gender and age aspects of the author profiling problem, both in English and Spanish.

Now, we have several collections in Spanish with different kind of "formality". That is, a corpus is "informal" if the texts have noise like typos, images, hyper-links, emoticons, contractions, etc. This noise becomes the corpus in a very challenging dataset for any classifier. However, from the results of the competition PAN-2013 it can be seen that some approaches like the one used in [15] (the winner of the competition), can obtain interesting results even when the nature of the documents makes very difficult the classification. Unfortunately, it is unclear how these techniques work when these are trained with some corpora and tested with data with a different distribution. This was the reason that motivated us to study the cross domain approach.

When we talk about a *domain*, in the data mining field, it could be loosely defined as a specialized area of interest for which we can develop ontologies, dictionaries and taxonomies of information. We can refer to the different scopes (very broad or more narrowly specialized domains), or also the type of source from which the texts come from (like blogs, forums, etc.), or simply, a domain could be considered as the writing style (formal, informal, scientific, etc.). Thus, cross domain can be interpreted in several ways.

However, this paper simply assumed that a *domain* is a texts collection with a particular level of informality. Therefore, a *cross domain* experiment indicates a classification where you train with a corpus with certain level of informality and test with other with a different level of informality. Cross domain tests are also called by others authors as Domain Transfer experiments [16]. These consist in generating a classifier from texts that belongs to a source domain (training set) to apply it to a different target domain (test set). In other words, the underlying purpose of this concept is to check how well the trained classifier generalizes when it run on a different collection of documents.

3. Data Collections

We consider three different corpora in Spanish for the experimental study: *SpanText* and others two which were provided by the PAN-CLEF competition in the years 2013 [14] and 2014 [12]. These latter are called *PAN-2013* corpus and *PAN-2014* corpus. Also, we use a sub-corpus of PAN-2013 which we have proposed for this experimen-tation. The characteristics of each one are presented below.

SpanText is a set of "formal" documents written in Spanish extracted from the Web [13]. In this context, we use the term "formal" (as opposed to "informal") to refer to those documents whose content has a low percentage of "non-dictionary" words, abbreviations, contractions, emoticons, slang expressions, etc. that are typical in messaging and the social Web. This dataset consists of a variety of texts that one supposes to find in newspapers, students' reports, books and so on. These "speak" about different topics and they were written by Spanish speakers from Spain and Latin American countries. Besides, there are only one document (file) per author.

Two versions of this collection were presented in [13]. They are called "balanced" and "unbalanced" versions. However, there is another one called "semi-balanced", in which we are interested. Spantext (like PAN-2013) considers age and gender as the basic demographic information for the authors. All the documents are labeled with both characteristics. For age detection, it contemplates three classes: 10s, 20s and 30s. In the semi-balanced version, the number of documents per class is proportional to the amount of PAN-2013's documents. These are only uniformly distributed with respect to gender.

Regarding the PAN-2013 collection, it was built automatically with texts from blogs and other social networks [14]. The organizers of the competition provided two corpora: one in English and other in Spanish language. The dataset was divided into the following sub-sets: training, early bird evaluation and final testing. In this work, PAN-2013 will refer to the training and test sets of the Spanish language. Documents in PAN-2013 considered a wide spectrum of topics and they include "informal" text. The posts were grouped by author selecting those authors with at least one post and chunking in different files with more than 1000 words in their posts. But it also included some authors with few and shorter posts. For age classification, this collection considers the three same classes as SpanText and it is balanced by gender and imbalanced by age group, having more texts in class 20s than in 30s, and more in 30s than in 10s.

However, due to the difference between the sizes of SpanText and PAN-2013, it was needed to separate a sub-corpus of the latter (called sub-PAN2013), so it has the same number of documents per category as the semi-balanced version of the former. Thus, the results of diverse experiments can be fairly compared and the difference in the results will be limited to other variables, such as the quality of the texts.

The collection of texts written in Spanish in the PAN-2014 corpus was collected semi-automatically from four different sources: social media, blogs, Twitter and hotel reviews (the last only provided in the English corpus). In the competition of the year 2014, the PAN-CLEF organization opted for modeling age in a more fine-grained way and considered the following ranges (classes): 18-24, 25-34, 35-49, 50-64 and 65+ years old. The full collection was also divided into training, early bird evaluation and final testing parts. It is worth noting that we could access only to the training set and we use that part in the experimental study because the test corpus is not available at the time of writing this article.

Table 1. Vocabulary (number of words without repetition), number of terms (words), number of files and average number of terms for each collection.

Collection	SpanText	PAN-2013	PAN-2013 sub-corpus	PAN-2014
#Vocabulary	31 504	342 068	29 616	306 809
#Terms	294 434	22 868 586	294 596	17 686 634
#Files	1 000	84 060	1 000	1 500
Average Terms	294	301	294	11 806

Table 1 shows the statistics for each full collection. We can observe that PAN-2013 corpus presents the biggest numbers except in average number of words. This is because of its structure, it has more files (or authors) and more wealth in terms of writing styles but, the texts are not too long.
If we compare SpanText with PAN-2014, the latter is 50% bigger than the former, but SpanText only has a 10% of vocabulary than its counterpart. PAN-2014 prioritized the amount of texts from the same author, rather than the number of authors. It was probably because these are often short texts due to the source from which they came from (e.g. Twitter). This is verified in the amount of average terms for document that overcomes highly the other two corpora.
However, we must emphasize that in this regard SpanText is not far from the PAN-2013 collection. Perhaps if we could increase the number of documents of SpanText, maintaining its characteristics, this corpus would become the most useful. Since the proportion between, the amount of repeated words and the vocabulary is 10% for SpanText and 1% for both PAN-2013 and PAN-2014.

4. Experimental Study

In this section, we describe the cross domain experiments performed using the software WEKA [17]. Basically we performed two kinds of studies: APT as a classification of documents by gender, and then, considering both together age and gender. This is because, as we previously mentioned, the corpus PAN-2014 considered different age ranges from the ones defined in PAN-2013; in such way that we cannot make a join or separation of categories in order to consider the same ranges of age for both corpora.
Table 2(a) shows the information about the cross domain experiments: name of the corpus used for training and amount of documents considered, and name of the corpus used for testing with the corresponding amount of documents for performing the classification only by gender. The same information for the classification by gender and age considered together is shown in Table 2(b). From now on, to refer to a particular experiment, first we will mention the name of the corpus that was used to train, followed by the name of the collection employed to test

(short forms of the original names of the corpora). For example, SPAN-PAN13 corresponds to the experiment which uses SpanText to generate the model and PAN-2013 to validate it.

Table 2. List of the cross domain experiments carried out.

(a) Classifications only by gender

Training	Docs	Test	Docs
PAN-2014	1 500	SpanText	1 000
SpanText	1 000	PAN-2014	1 500
PAN-2014	1 500	PAN-2013	84 060
PAN-2013	84 060	PAN-2014	1 500
SpanText	1 000	PAN-2013	84 060
PAN-2013	84 060	SpanText	1 000

(b) Classifications by age and gender

Training	Docs	Test	Docs
PAN-2013	84 060	SpanText	1 000
SpanText	1 000	PAN-2013	84 060
Sub-PAN13	1 000	SpanText	1 000
SpanText	1 000	Sub-PAN13	1 000

We used two traditional models of representation of documents: *bag of words* (BoW) [18] and *character trigrams* [19]. Regarding the weighting schema, we employed: *Boolean* [18] and *tf-idf* [20]. We also considered the *Second Order Attributes* (SOA) representation [13] because it has been demonstrated to be effective for this task. We have constructed the models and performed the classification using N*aïve Bayes* [21] and *LibLINEAR* [22] methods. Besides those, we considered an interesting approach *Sistema de Perfiles* (SP) [23] which generates its own model (profiles) using the most frequent character trigrams of the texts (L value) and then evaluates the belonging of the test documents in the profiles. It is important to note that due to the characteristics of its functioning, we could not use SP for those experiments which required to train with the PAN-2014 collection, because it was not able to generate the required profiles for the classification. The values for the L parameter of SP mentioned in the tables were chosen from carrying out prior executions for different values of this, choosing the one with we obtained the best accuracy. All approaches were evaluated considering the accuracy as metric.

4.1 Classifications only by gender

The percentages of correctly classified instances (accuracy) obtained in the cross domain classification only by gender are shown in Table 3. The table is divided into three sub-tables (a), (b) and (c) considering three different cross domain experiments. The highest accuracy values obtained are highlighted in boldface. The first value is the accuracy obtained with Naïve Bayes algorithm and the one after the slash corresponds to the accuracy obtained with the LibLINEAR algorithm.

Table 3. Accuracy obtained in cross domain classifications only by gender with "Naïve Bayes / LibLINEAR" algorithms.

	(a) PAN-2014 and SpanText		(b) PAN-2013 and SpanText		(c) PAN-2014 and PAN-2013	
	PAN14-SPAN	SPAN-PAN14	PAN13-SPAN	SPAN-PAN13	PAN13-SPAN	SPAN-PAN13
Boolean words	50,0 / 48,2	54,1 / 52,6	53,0 / 58,1	**53,1** / 52,1	50,3 / 52,4	57,5 / **67,5**
TF-IDF words	51,7 / 53,6	52,7 / 52,9	51,6 / **60,9**	51,1 / 51,9	52,4 / 53,3	58,3 / 64,9
SOA words	**61,2** / 59,4	54,9 / **55,4**	60,1 / 53,0	50,1 / 50,0	**59,1** / 58,5	61,1 / 62,6
Boolean 3grams	50,0 / 49,8	48,9 / 51,5	51,0 / 55,8	**57,7** / 51,6	50,1 / 49,5	50,7 / 58,8
TF-IDF 3grams	50,1 / **53,7**	51,3 / 51,3	54,8 / **60,3**	50,6 / 50,1	**54,9** / 54,6	56,3 / **62,2**
SP 3grams	-	**54,3**	58,7	51,3	-	57,8

The baseline used by PAN-CLEF Lab competition to determine if a two-class classifier is acceptable is 50%. Table 3 shows that almost all percentages exceeded or equaled this value (48,2; 49,8; 48,9 and 49,5 are the exception). Note that with PAN13-SPAN it was not obtained percentages lower than the 50%.

Figure 1 provides a visual summary of Table 3. The bars with no plot at the left correspond to the representation of documents and the bars with plot (dots and rhombus) at the right with classifiers. The accuracy shown is the average of all the accuracies obtained for each approach for every training corpus used. Furthermore, results are shown from the baseline so that it would highlight better the differences obtained. It is important to note that words strategies dominate character trigrams approaches.

If we analyze the document representations, in general SOA accomplish the best performing, which precisely works with words. Next it follows the SP with character trigrams, then, in third and fourth places are the tf-idf representation with words and character trigrams respectively. Certainly, with a little more elaborated approaches than the simple use of frequencies, it achieves better results.

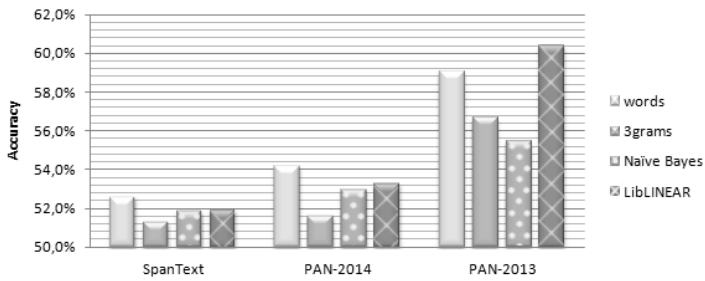
Regarding the classifiers, it can be concluded that LibLINEAR is superior. Out of the eleven best results in bold, six were obtained with it. Moreover, if an average of executions is calculated grouping them by classifier, it results that using LibLINEAR the average accuracy is around 60%, while with Naïve Bayes reaches only 56%.

The highest results were achieved when we trained with PAN-2013 and tested with PAN-2014, 67.5% for words and 62.2% for character trigrams. If we make an average of all executions in which this corpus was used to train the model, we found that this obtained the best percentage. This is also exhibited in Figure 1. Therefore, with 57.8% against 52.9% training with PAN-2014 and 51.9% with SpanText, we can say that the PAN-2013 collection is the one that generates a more general classifier.

At the PAN-CLEF competition in 2014, they tested the approaches of the participants who participated in 2013 (the approaches were trained with PAN-

2013 corpus) using the 2014 collection (testing with PAN-2014). The SP achieved 69.4% of accuracy taking the first position in the final ranking [12]. Observing the results obtained we conclude that with the PAN-2013 collection we can get a general model able to classify documents from different corpora. Additionally, the results of the experiments accomplished in this work, at least for classifications only by gender are promising and overcome at least in a 3% the experiments performed on a single domain.

Figure 1. Summary of the results obtained for the cross domain classification only by gender distinguished by representations and classifiers.



4.2 Joint classifications by age and gender

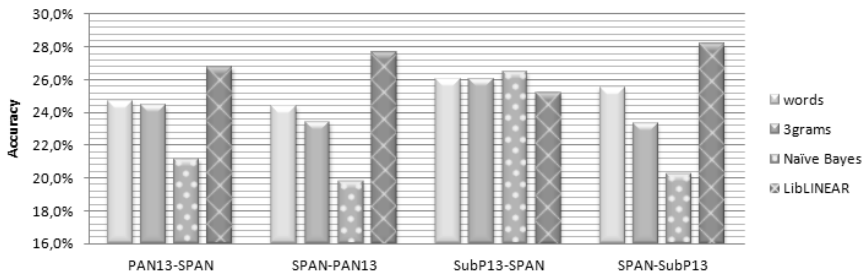
The results obtained for the cross domain classification considering age and gender are shown in Table 4. The best result of each section is highlighted in boldface. The baseline for this case is 16% because there are six categories (the combination of female and male with the three ranges of age). In Table 4 there are three cases in which the percentage does not reach the baseline. The first correspond to SPAN-PAN13 combination employing words-tf-idf representation and Naïve Bayes. Then, the second and third cases use character trigrams-Boolean representation with Naïve Bayes classifier, SpanText to train and PAN-2013 (or its sub-corpus) to test. However, when an average of the results is calculated, for example based on the classifiers, we can say that all the values are over the baseline.

Table 4. Accuracy obtained in cross domain classifications by age and gender with "Naïve Bayes / LibLINEAR" algorithms.

	(a) SpanText and PAN-2013		(b) SpanText and sub-PAN2013	
	PAN13-SPAN	SPAN-PAN13	subP13-SPAN	SPAN-subP13
Boolean words	19,7 / 26,5	20,1 / 28,1	20,0 / 25,7	19,8 / 28,3
TF-IDF words	19,3 / **29,5**	14,0 / 28,3	24,0 / 24,8	17,5 / **29,6**
SOA words	25,6 / 27,4	**28,7** / 27,1	**35,1** / 26,4	29,2 / 28,8
Boolean 3grams	20,5 / 22,8	11,5 / **28,0**	25,0 / 23,7	15,1 / 26,9
TF-IDF 3grams	20,7 / 27,7	24,8 / 26,9	**28,4** / 25,5	19,8 / 27,3
SP 3grams	**30,5**	25,9	27,5	**27,6**

Figure 2 summarizes the information of Table 4. In the bars the different models of representation (words and character trigrams) are at the left and they do not have a plot. Whereas the bars with dots and rhombus that are at the right, represent the behavior of the classifiers. As compared to the cross domain classifications only by gender, where words always predominated, here the bars exhibited are more similar among them. So it seems that the character trigrams help to distinguish better out the six categories.

Figure 2. Summary of the results obtained for cross domain classifications by age and gender distinguished by representations and classifiers.



If we analyze the traditional representations, i.e. Boolean and tf-idf, we obtained better results using words when we trained with the complete PAN-2013 corpus (Table 4 (a)). In particular, the combination of the tf-idf representation with the classifier LibLINEAR has worked considerably well. However, when it is trained with SpanText, the character trigrams strategy achieves a higher percentage on average. Now if we consider slightly more elaborated approaches in SPAN-PAN13 combination, the SOA representation is the best at discriminating the different classes. Nevertheless, the best overall result for the joint classification by gender and age is reached in PAN13-SPAN with the SP.

Table 4 (b) shows the results obtained with the sub-corpus of PAN-2013 which are different than those obtained with the complete corpus of PAN-2013. Even though, this case is a specific one thereof.

In general, regarding the classifiers, Naïve Bayes obtained poor results, highlighting even more the difference in performance respect to its counterpart. As we mentioned above, LibLINEAR with tf-idf representation using words obtained the second best result for cross domain classification by gender and age using the whole corpora.

Thus, in these experiments the same behavior is observed as in the classifications only by gender in which the approaches that use words are better. This is evidenced by the 35.1% obtained with the SOA representation in PAN13-SPAN combination. In addition, the highest percentage is accomplished again using the sub-corpus PAN-2013 to train the model.

5. Conclusions and Future Work

Cross domain experimentation has started to raise the interest of researchers turning their attention to the possibility of building a general enough classifier to classify any type of text documents. Hence its importance in the APT in which it is difficult to find properly labeled and lesser noise collections of texts, particularly for the Spanish language, is significant. For example, to detect pedophiles on the network or other kind of tasks that require a real-time response, and where the previous training with information which is not necessarily of the same type of the task to evaluate, is limited or non-existent.

In this paper we present a preliminary study considering cross domain author profiling classification. We made different experiments considering some corpora for training and testing using others considering different level of formality.

We analyzed the corpora available for APT in Spanish language using different representations and classification algorithms. Aiming not only to see how well a corpus generalizes a model, but also to evaluate the desirable characteristics that should have them, we conclude that the PAN-2013 collection is the one which better serves for that purpose. The highest accuracies were obtained with more elaborate representations such as SOA and approaches such as SP. Therefore, the results of the cross domain experiments obtained in this study turn to be promising, since they get close and even exceed the values obtained in experiments conducted in a single domain (or inter-domain).

Finally, it would be interesting to verify how the SP approach would behave when it trained with the PAN-2014 collection, and instead of using character trigrams, using words or more sophisticated representations.

References

1. Koppel, M., Argamon, S., and Shimoni, A. R. Automatically Categorizing Written Texts by Author Gender. Literary and Linguistic Computing. Vol. 17, no 4, pp. 401–412, 2002.
2. Argamon, S., Koppel, M., Fine, J., and Shimoni, A. R. Gender, Genre, and Writing Style in Formal Written Texts. TEXT. Vol. 23, pp. 321–346, 2003.
3. Schler, J., Koppel, M., Argamon, S., and Pennebaker, J. W. Effects of Age and Gender on Blogging. In AAAI Spring Symposium: Comp. Approaches to Analyzing Weblogs. Vol. 6, pp. 199–205, 2006.
4. Koppel, M., Schler, J., and Zigdon, K. Determining an Author's Native Language by Mining a Text for Errors. In Proc. of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining. ACM, pp. 624–628, 2005.
5. Rangel, F. Author Profile in Social Media: Identifying Information about Gender, Age, Emotions and beyond. Proc of the 5th BCS IRSG Symposium on Future Directions in Information Access, pp.58–60, 2013.
6. Bo, P., and Lee, L. Opinion Mining and Sentiment Analysis. Foundations and Trends in Information Retrieval. Vol. 2, issue 1-2, pp. 1–135, 2008.

7. Pennebaker, J.W., Mehl, M.R., and Niederhoffer, K. Psychological Aspects of Natural Language Use: Our Words, Our Selves. Annual Review of Psychology. Vol. 54, pp. 547–577, 2003.
8. Argamon, S., Koppel, M., Pennebaker, J. W., and Schler, J. Automatically Profiling the Author of An Anonymous Text. Communications of the ACM. Vol. 52, pp. 119–123, 2009.
9. Ramakrishna Murty, M., Murthy, J.V.R., Prasad Reddy, P.V.G.D., and Satapathy, S.C. A survey of cross-domain text categorization techniques. In Recent Advances in Information Technology (RAIT), 1st International Conference, pp. 499–504, 2012.
10. Li, L., Jin, X., and Long, M. Topic Correlation Analysis for Cross-Domain Text Classification. In Proc. AAAI, 2012.
11. Pan, S. J., Ni, X., Sun, J., Yang, Q., and Chen, Z. Cross-Domain Sentiment Classification via Spectral Feature Alignment. In proc. of the 19th international conference on World wide web, pp. 751–760, 2010.
12. Rangel, F., Rosso, P., Chugur, I., Potthast, M., Trenkmann, M., Stein, B., Verhoeven, B., and Daelemans, W. Overview of the 2nd Author Profiling Task at PAN 2014. Proc. of the Conference and Labs of the Evaluation Forum (Working Notes), 2014.
13. Villegas, M. P., Garciarena Ucelay, M. J., Errecalde, M. L., and Cagnina, L. C. A Spanish text corpus for the author profiling task. In Proc. of XX CACIC. San Justo, Buenos Aires, Argentina, pp 1-10, 2014.
14. Rangel, F., Rosso, P., Koppel, M., Stamatatos, E., and Inches, G. Overview of the Author Profiling Task at PAN 2013. Notebook Papers of CLEF, pp. 23–26, 2013.
15. López-Monroy, A. P., Montes-y-Gómez, M., Escalante, H. J., Villaseñor-Pineda, L., and Villatoro-Tello, E. Inaoe's Participation at PAN'13: Author Profiling Task. Notebook PAN at CLEF 2013, 2013.
16. Finn, A., Kushmerick, N., and Smyth, B. Genre classification and domain transfer for information filtering. In Advances in information retrieval, Springer, pp. 353–362, 2002.
17. WEKA. Data Mining Software in Java. http://www.cs.waikato.ac.nz/ml/weka/
18. Feldman, R., and Sanger, J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstruc-tured Data. Cambridge University Press, 2007.
19. Cavnar, W. B., and Trenkle, J. M. N-gram-based text categorization. Ann Arbor MI, 48113(2):161–175, 1994.
20. Manning, C. D., Raghavan, P., and Schütze, H. Introduction to Information Retrieval. Cambridge Uni-versity Press, New York, USA, 2008.
21. Lin, J. Automatic author profiling of online chat logs. Doctoral thesis, Monterey, California. Naval Postgraduate School, 2007.
22. Fan, R.-E., Chang, K.-W., Hsieh, C.-J.,Wang, X.-R., and Lin, C.J. LIBLINEAR: A library for large linear classification. The Journal of Machine Learning Research, 9:1871–1874, 2008.
23. Fúnez, D., Cagnina, L., and Errecalde, M. Determinación de Género y Edad en Blogs en Español Mediante Enfoques Basados en Perfil. In XVIII CACIC, 2013.

Dynamic List of Clustered Permutations on Disk

KARINA FIGUEROA[1], CINTIA MARTÍNEZ[2], RODRIGO PAREDES[3], NORA REYES[2] AND PATRICIA ROGGERO[2]

[1] Facultad de Ciencias Físico-Matemáticas, Universidad Michoacana, México
karina@fismat.umich.mx
[2] Departamento de Informática, Universidad Nacional de San Luis, Argentina
{proggero,nreyes}@unsl.edu.ar, cintiavmartinez@hotmail.com
[3] Departamento de Ciencias de la Computación, Universidad de Talca, Chile
raparede@utalca.cl

Abstract. *Similarity searching is a difficult problem and several indexing strategies have been defined to process similarity queries efficiently in many applications, including multimedia databases and other repositories handling complex objects. Metric indices support efficient similarity searches, however most of them are designed for main and static memory. Thus, they can only handle small datasets, while suffering serious performance degradations when the objects reside on disk and do not support insertions of new elements. Most real-life database applications require indices able to work on secondary memory and dynamism.*
Among a plethora of indices, the List of Clustered Permutations (LCP), Permutation-based algorithm (PBA) and Approximating and Eliminating Search Algorithm (AESA) have shown to be competitive in main memory. We introduce a dynamic and secondary-memory combination of AESA, PBA and LCP, which maintains the low number of distance evaluations, and also needs a low number of I/O operations at construction and searching.

Keywords: *metric spaces, permutation-based algorithm, secondary memory*

1. Introduction

"Proximity" or "similarity" searching is the problem of looking for objects in a dataset, that are "close" or "similar enough" to a given query object, under a certain (expensive to compute) distance. Similarity searching has become a very important operation in applications that deal with unstructured data sources (for example, multimedia databases that manage objects without any kind of structure, such as images, fingerprints or audio clips). This approximation has applications in a vast number of fields. Some examples are non–traditional databases, text searching, information retrieval, machine learning and classification, image quantization and compression, computational biology, and function prediction. These problems can be model as *metric spaces* [3]. That is, there is a universe X of objects, and a non negative real valued distance function $d: X \times X \rightarrow \mathbb{R}^+$ defined among them.

This distance satisfies the three axioms that make the set a *metric space*: *strict positiveness*, *symmetry*, and *triangle inequality*.
The smaller the distance between two objects, the more "similar" they are. We have a finite database $U \subseteq X$, $|U| = n$, which is a subset of the universe and can be preprocessed to build an index. Later, given a new object from the universe (a query $q \in X$), we must retrieve all similar elements in the database. There are two typical similarity queries:

- ***Range query*** (q, r)*:* retrieve all elements within distance r to q in U.
- ***k-Nearest Neighbor query*** *(k-NN):* retrieve the k closest elements to q in U.

Our focus is on *approximate proximity searching*, where accuracy can be traded off for efficiency, as opposed to exact similarity search algorithms. However, there are generic techniques to convert any exact algorithm into approximate by using a form of aggressive pruning, as described for example, in [1].
For general metric spaces, there exist a number of methods to preprocess the database in order to reduce the number of distance evaluations [2], [3], [4]. In general metric spaces, the (black-box) distance function is the only way to distinguish between objects, and usually, the function of distance is expensive to compute (in time and/or resources), compared to the CPU time to traverse the index and decide which elements are relevant. However, when the index is located in secondary memory the I/O operations are also very significant [5]. Therefore, the goal of similarity search algorithms for metric spaces in secondary memory is to solve queries using the minimum number of distances computations and I/O operations.
Since this kind of datasets lacks of total order, it is necessary to preprocess the database to build an index in order to avoid a full linear scan, which allows answering queries with less effort. The *List of Clusters* (LC) and *Approximating and Eliminating Search Algorithm* (AESA) [2]–[4], [6] are ones of the most efficient algorithms. However, $O(n^2)$ distance calculations are required to build both indices. On the other hand, the *Permutation Based Algorithm* (PBA) [7], [8], [9] is an approximate method that has been showed unbeatable in practice, but only works well in high dimensions, as the authors claim. Once the index is built by calculating the "permutation" of each database object, during searching time the permutation of the query object q is computed and compared against all permutations of database objects, to stablish the order to review permutations. This takes at least $O(|P|)$ distance calculations, where $|P|$ is the permutation size, and $O(n)$ evaluations of the "permutation distance". There have been several proposals to avoid the sequential scan in PBA, however all of them lost accuracy regarding the original technique [10], [11]. In [9], a combination of the main ideas of LC and PBA is presented as a new metric index to answer approximate similarity search. This new index, called as *List of Clustered Permutations* (LCP), achieves a good search performance and beats both LC and PBA.
However, when we want to answer approximate similarity queries on large volumes of data, working in secondary memory and considering distance and I/O

costs is necessary. The I/O time is composed of the number of disk pages read and written; we call B the size of the disk page in bytes. Given a dataset of $|U| = n$ objects of total size N bytes and disk page size B, queries can be trivially answered by performing n distance evaluations and N/B I/Os. The goal of a secondary-memory index is to preprocess the dataset and to answer queries with as few distance evaluations and I/Os as possible.

Therefore, in this article we use the idea of LCP [9], built on LC and PBA, but considering the index has to be located in secondary memory. So, the idea is to keep each cluster of the list on a disk page in secondary memory. Hence, the cluster size must consider the disk page size. Besides, in order to accelerate searches the information of centers (permutants) and some few more data are also maintained in main memory. In addition, we are interested that the index can support the insertion of new elements.

The rest of this paper is organized as follows. In Section 2 we describe the previous works and some basic concepts. Next, in Section 3 we detail the List of Clustered Permutations (LCP) and in Section 4 we present our dynamic, secondary-memory variant of LCP. In Section 5 we show the experimental evaluation of our proposal. Finally, we draw some conclusions and future work directions in Section 6.

2. Previous Works

In order to introduce our secondary-memory index, we describe briefly the main aspects of the previous works used as basis.

Approximating and Eliminating Search Algorithm The Approximating and Eliminating Search Algorithm (AESA) [2]–[4], [6], allows very fast queries for small databases at the expense of quadratic memory usage, being indeed a lower bound for pivot based indices (basically, pivots are points of reference). The structure is simply a matrix of n^2 distances between all pairs of n objects. Particularly, due to symmetry property of metric function only a half of the matrix below the diagonal is stored, that is, $n(n-1)/2$ distances. In AESA, every object plays the role of a pivot. At the beginning, the search operation for range query (q, r) picks at random an object p in the set of n objects and uses it as a pivot; next, it evaluates the distance from q to p and uses it to either discard objects or to lower bound the distances to other objects. Later, it chooses the object with the minimum lower bound as the next pivot, and repeats the process.

Permutation-Based Algorithm In [8] the authors introduce the permutation based algorithm (PBA), a novel technique that shows a different way to sort the space. At preprocessing time, a subset of objects $P = \{p_1, p_2, \ldots, p_s\} \subseteq U$, called the *permutants,* is selected out of the database. Each object $u \in U$ computes its distance to all the permutants (i.e., computes $d(u, p)$ for all $p \in P$) and sorts them increasingly by distance. Then, for each $u \in U$, just the order of the permutants (not the distances) is stored in the index.

If we define Π_u as the permutation of (1, …, P) for the object u, so $\Pi_u(i)$ is the i-th cell in the permutation of u and $p_{\Pi u(i)}$ denotes the i-th permutant. For example, if $\Pi_u = (5, 2, 1, 3, 4)$ then $p_{\Pi u(3)} = p_1$. Within the permutation, for all $1 \leq i \leq |P|$, it holds either $d(p_{\Pi u(i)}, u) < d(p_{\Pi u(i+1)}, u)$ or, if there is a tie $d(p_{\Pi u(i)}, u) = d(p_{\Pi u(i+1)}, u)$, then the permutant with the lowest index appears first in Π_u. We call the i-th permutant $\Pi_u(i)$, the inverse permutation Π_u^{-1}, and the position of i-th permutant $\Pi_u^{-1}(p_i)$. The set of all the permutations stored in the index needs just $O(n|P|)$ memory cells. During searching time, we compute the distance from the query $q \in X$ to all the permutants in P and obtain the query permutation Π_q. Next, Π_q is compared against all the permutations stored in the index, which takes $O(n)$ permutation distances. The order induced by the permutation of q (i.e. Π_q) is very promising and reviewing a small database fraction is enough to get a good answer.

The permutation distance is calculated as follows: let Π_u and Π_q be permutations of (1, …, $|P|$). We compute how different is a permutation from the other one using *Spearman Rho* (S_ρ) metric [12]: $S_\rho(\Pi_u, \Pi_q) = \sqrt{\sum_{1 \leq i \leq |P|} (\Pi_u(i) - \Pi_q(i))^2}$.

The main disadvantage of the PBA is that its memory requirement could be prohibitive in some scenarios, especially where n is actually huge and it has an effect on how long is the fraction to consider when solving the approximate similarity query.

List of Clusters There are many indices for metric spaces [2], [7], [13]. One of the most economical in space used and rather efficient is the *List of Clusters* (LC) [13], because it needs $O(n)$ space and has an excellent search performance in high dimension. Regrettably, its construction requires $O(n^2)$ distance evaluations, which is very expensive. The LC index is built recursively. The LC has two variants, one with a fixed cluster size and the other with a fixed cluster radius. We describe here the variant of fixed cluster size that sets the maximum number of elements that fits in a disk page included into a cluster.

Firstly, a center c is selected from the database and a bucket size b is given. c chooses its b-closest elements of the database and build the subset I, which is the answer of a b-nearest neighbor query of c in U. Let cr_c be the distance from c to its farthest neighbor in I. The tuple (c, I, cr_c) is called a *cluster*. This process is recursively repeated with the rest of the non-clustered objects. Finally, we have a set of centers C with their cluster elements and their covering radii, organized as a list. To answer queries, the query object q is compared with all the cluster centers in C. During a range search (q, r), for each cluster with center c_i, if the distance from its c_i to the query q is larger than its covering radius cr_{c_i} plus the query radius r we can discard its whole bucket, otherwise we review it exhaustively. Formally, if $d(q, c_i) > cr_{c_i} + r$ the cluster of c_i can be completely discarded.

3. List of Clustered Permutations

As it is aforementioned, LC is a good search index but is costly to build and PBA gives a way to answer approximate similarity queries, while trades accuracy or determinism for faster searches. There are two possibilities to reduce the construction time of LC: a bigger bucket size, or using another, cheaper, way to build the index. Following the second strategy, in [9], they propose to combine the PBA with the LC. A set of permutants is chosen, where each one within this set has a double role, as permutant and as a cluster center. Besides, only the cluster centers store their permutation. This index is called *List of Clustered Permutations* (LCP) [9].

As we mention previously, when we solve a similarity query q with the standard PBA, we need to spend $|P|$ evaluations of the distance d to compute the query permutation Π_q, n evaluations of the permutation distance S_ρ to compute the order induced by Π_q on U, and O(fn) distance evaluations (of d) to compare q with the fraction of f the dataset objects that are the most promising to be relevant for the query. With the LCP index, only |P| ($\ll n$) evaluations of the permutation distance S_ρ are needed to compare Π_q with the permutation of each cluster center. Then, some distances are needed to review non-discarded clusters.

The building process of the index is done as follows: a set $P = \{c_1, \ldots, c_s\}$ of centers (permutants) is randomly selected, and for each database object $u \in U$, $d(u, c_i)$, for all $c_i \in P$ is calculated. Hence, we can compute the permutations for all the objects u in the dataset U. Then, the first center is chosen and grouped its b most similar objects according to the permutation distance S_ρ (excluding all the cluster centers, so that no center can be inside the bucket of another center). The process continues iteratively with the rest of elements in P until every element in $U \setminus P$ is clustered. Every center c_i maintains its covering radius cr_{c_i} and its permutation. All the permutations of elements in $U \setminus P$ are discarded; that is, the permutations of all the objects within a bucket will not be stored. Fig. 1 shows an example of LCP index for a little set of points in $\mathbb{R}^2$, where the set $P = \{c_1, c_2, c_3, c_4\}$ of centers (permutants) is selected and $b = 2$ (only two points belong to each cluster). We also show the covering radii and the permutations of centers.

Therefore, the space used for the index is $n + |P|^2$ cells, and the construction time is O($n|P|$) evaluations of both the space distance d and the permutation distance S_ρ. It can be noticed that the whole LCP index can be packed using only $(n + |P|^2) \log_2|P|$ bits.

As it is mentioned, the standard LC discards clusters during a range search (q, r) by using the covering radii criterion. Let $d(q, c)$ be the distance between the query q and the center of the cluster c and cr_c the covering radius of center c. So, if $d(q,c) > r + cr_c$, the cluster whose center is c can be discarded.

Since the centers of LCP have permutations, a heuristic method can be introduced to discard clusters, modifying the criteria explained in [13]. In [9],

authors mention that their preliminary experimental results have shown that if an object (for instance, a cluster center), and its permutation have (just) one permutant that moved far away with respect to its position inside query permutation, then this object is not relevant, so it can be discarded (and also its bucket). For example, if the permutation of the query is (1, 2, 3, 4) and the permutation of the center is (4, 1, 2, 3), even though most of both permutations are similar, the position shifting of permutant 4 suggests that the object can be discarded.

Basically, it is necessary to know how much could a permutant move away inside the permutation of an object. So, by using the query permutation Π_q and the range query radius r, it can be estimated how far a permutant could shift. To do that, for a pair of permutants c_i, c_j, where c_i is closer to the query q than c_j, and $d(c_j, q) - d(c_i, q) \leq r$, the method does not discard an object whose permutation has an inversion of these permutants; this is, it does not discard an object that is closer to c_j than to c_i. But, if the distance difference is larger, although permutant inversion is possible there as a big chance that the object were irrelevant, so the object can be discarded. During query process, a cluster center (and its bucket) is discarded when a permutant shifts more than tolerated.

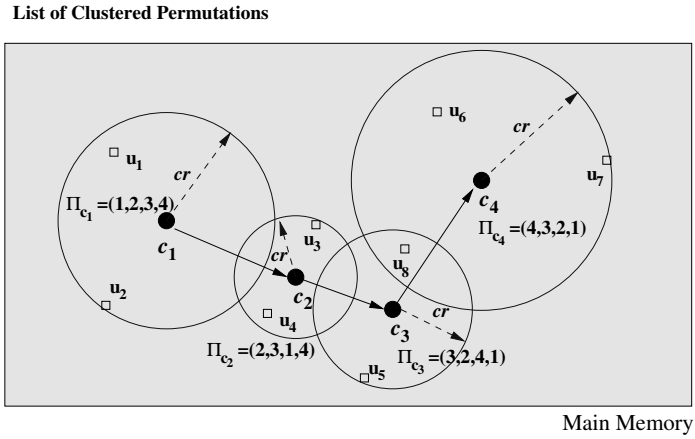


Fig. 1. *An example of LCP.*

4. Dynamic List of Clustered Permutations

In order to obtain an efficient dynamic variant of LCP for secondary memory, we have to consider some important aspects of using a disk as storage. An I/O operation on disk involves three main times: the time of head positioning, lattency, and transfer time. The transfer unit of a disk is called a *disk page*. Therefore, a way of reducing times is to read/write few disk pages. One key aspect for this objective is to use as few disk pages as possible, and other is to read/write disk pages when it is strictly necessary.

Therefore, our proposal consider the design of a dynamic and secondary-memory variant of LCP that occupies the smallest possible amount of disk pages and it only reads/writes a disk page when it is actually necessary, and that supports insertions. While the dataset does not have enough elements, the elements are maintained in main memory and they are indexed with AESA. When a new element is inserted, it calculates all the distances to the other elements which have arrived previously. Then, when the number of elements exceeds the memory capacity, we use the distances within AESA index to obtain the element permutations, next we discard the AESA matrix, and build the index setting the size of the clusters as how many database elements fits in a disk page of size B. In addition, we can take advantage of main memory to store some information of the new index, to reduce the number of I/O operations at searches. We called our variant as *DLCP*.

The build process of *DLCP* is almost the same used for LCP. As we mentioned, we set the cluster size as a function of disk page size and the size of the representation of an object. Therefore, at this point *DLCP* is different from LCP, because b is a fixed value defined mainly by B and not as a function of the number of centers selected. On the other hand, the number of centers needed is determined as a function of the resulting size b of a cluster; that is, $b = (n/|P|) - 1$. Thereby, we force that each cluster fits completely in a disk page and, because of that, when we need to review the elements of a cluster we only have to read only one disk page. In each cluster only the real objects are stored. As LCP does, all the permutations of elements in $U \setminus P$ are discarded. Furthermore, taking advantage of main memory storage, we replicate some information of *DLCP* in main memory, in order to avoid reading unnecessarily a page (cluster) only to compare the query object q with the center c of a cluster and then determine its cluster is non relevant. Hence, we maintain in main memory the list of selected centers P. For each center $c \in P$ we store its covering radius, its permutation, the actual number of elements in its cluster, and the number of disk page where is stored its cluster. Then, when we process a range query (q, r), we can determine without reading any disk page the set of candidate clusters that can be relevant to the query. This stage needs $|P|$ distance computations to obtain the permutation Π_q in addition to $|P|$ calculations of S_ρ distance to compare Π_q with the permutation of each center and determine which clusters have to be reviewed. Next, in order to optimize the necessary time to retrieve all the candidate clusters, we sort the number of disk pages that will be read, because it is cheaper to read disk pages in a sequential way. Then, we order the elements retrieved from the clusters read and we compare q with the ordered set of elements as LCP does.

Fig. 2 depicts the same example of Fig. 1, but simplified because we want to show mainly the two parts of our index when it is on disk. As in LCP, we can use the parameter f to limit the fraction of more promising database objects that will be compared, via d, with the query q. Besides, if it appears as necessary we can add another parameter s to *DLCP* that limits the number of disk pages that we will read. In this case, among the list of candidate clusters we select the s more promising. Therefore, it is possible to trade accuracy

with distance evaluations and I/O operations as we need, and limit to f the number of distance evaluations and/or to m the number of I/O operations.

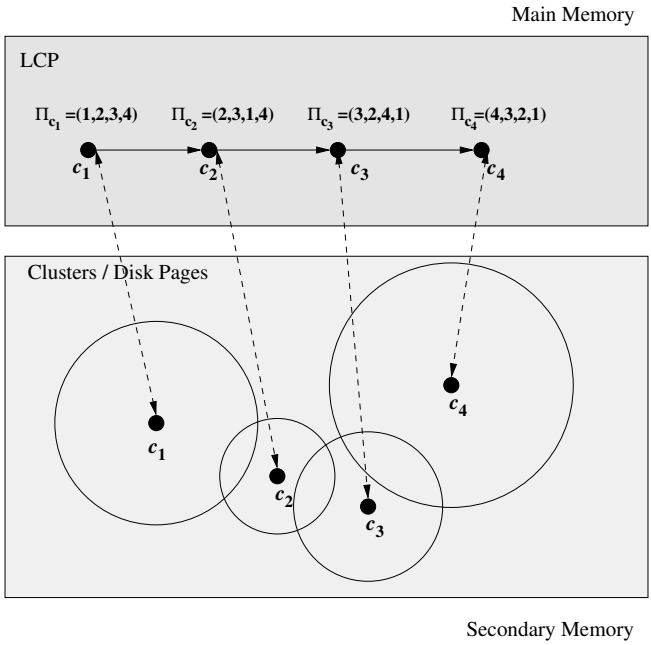


Fig. 2. *A simplified example of DLCP.*

5. Experimental Results

In order to evaluate the performance of our *DLCP*, we select a sample of different metric spaces from SISAP [14]: sets of synthetic vectors on the unitary cube and a real-life database. For lack of space, we only show the results obtained with the real-life metric space. Since our *DLCP* is an approximated method when it is on disk, we can relax the discarding criteria by accepting bigger shifts. We tabulate these results.

The metric space used is a dictionary of 69,069 words in English with the *edit distance*; that is the minimum number of character insertions, deletions, and substitutions needed to make two strings equal. It is a representative example of a real-life database. In all cases, we build the index with the 90% of the database elements and we use the remaining 10%, randomly selected, as queries. So, the elements used as query objects are not in the index. We average the search costs of all queries. We evaluate the effect of using different page sizes of 4KB and 8KB, that produce different clusters sizes and number of centers.

In Fig. 3 we show the average search cost per element, for different maximum database sizes in the AESA stage of the index. We consider range queries with radii from 1 to 4.

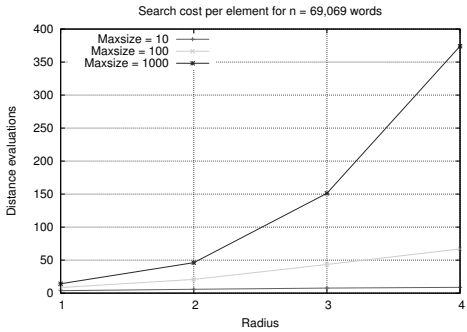


Fig. 3. *Search costs of AESA stage of* DLCP*, considering different maximum sizes.*

The Fig. 4 illustrates the search costs of query, measured in distance evaluations (left) and number of pages read (right), as the shifting criterion is relaxed. We also evaluate the effect of page sizes for the search performance: 4KB and 8KB. As it can be noticed, the number of distance evaluations grows as dimension increases, but sublinearly. Besides, the number of pages read is very low, between 1 and 35 for all cases. Surprisingly, the number of pages read does not decrease as page size increases. This odd behavior can be because as page size increases cluster sizes grows, but the clusters are so big that they can not be discarded easily. As it is aforementioned, an approximate similarity searching can obtain an inexact answer. That is, if a 1-NN query of an element $q \in U$ is posed to the index, it answers with the closest element from U between only the elements that are actually compared with q. However, as we want to save as many distance calculations as we can, q will not be compared against many potentially relevant elements. If the exact answer of 1-NN(q) = $\{x_1\}$, it determines the radius $r_1 = d(x_1, q)$ needed to enclose x_1 from q. An approximate answer of 1-NN(q) could obtain an element z whose $d(q, z) > r_1$.

Recall is a measure commonly used to evaluate the retrieval effectiveness of a method. It is defined as the ratio of the number of relevant elements retrieved for a given query over the number of relevant elements for that query in the database. This measure take on values between 0 and 1. So, for each query element q the exact 1-NN(q) = $Rel(q)$ is determined with traditional LC. The approximate-1-NN(q) = $Retr(q)$ is answered with DLCP index, let be the set $Retr(q) = \{y_1\}$. It can be noticed that the approximate search will also return one element in this case, so $|Retr(q)| = |Rel(q)| = 1$. Thus, we determine the number of elements obtained which are relevant by verifying if $d(q, y_1) = r_1$. We use recall in order to analyze the retrieval effectiveness of our proposal in 1-NN queries.

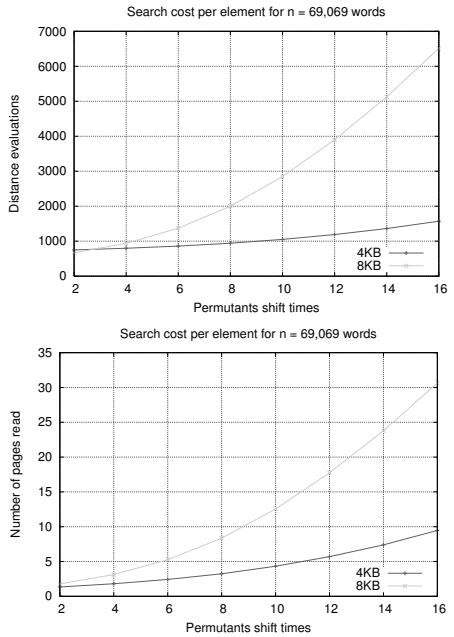


Fig. 4. DLCP *search costs, considering page sizes of 4KB and 8KB.*

Fig. 5 illustrates the recall obtained, as the shifting criterion is relaxed. We evaluate the effect of page sizes. For this matter, better results are obtained with 8KB than with 4KB.

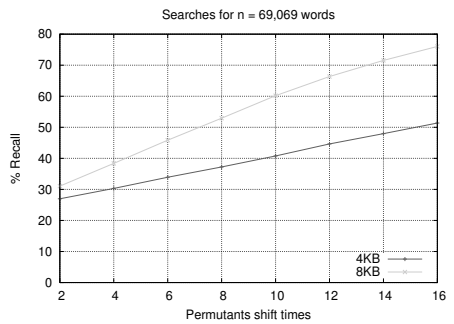


Fig. 5. *Recall of* DLCP*, considering page sizes of 4KB and 8KB.*

6. Conclusions

We have presented a new dynamic index for approximate similarity search in secondary memory. The *DLCP* structure extends an in-memory approximate data structure LCP [9], that offers a good balance between construction and search time. The secondary-memory version also supports approximate

searches by calculating few distances and reading very few disk pages. So, we have obtained a more practical index, because it maintain the good characteristics of LCP, but it can be applied on massive datasets that require secondary memory storage, and it can support insertions of new elements.
As future works we plan to analyze if there is a best disk page size for each space and to validate our results over larger databases. We also have to check how performance is affected when the number of disk pages read and distance calculations are limited with a pair of parameters f and m. As in [9], we also want to explore the use of short permutations for objects into the clusters, because the beginning of the permutation is the most important data portion to process, by trading space to improve the recall results.

References

1. E. Chávez and G. Navarro, "Probabilistic proximity search: Fighting the curse of dimensionality in metric spaces", *Inf. Process. Lett.*, vol. 85, no. 1, pp. 39–46, 2003.
2. E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín, "Searching in Metric Spaces", *ACM Comput. Surv.*, vol. 33, no. 3, pp. 273–321, 2001.
3. P. Zezula, G. Amato, V. Dohnal, and M. Batko, *Similarity Search The Metric Space Approach*, vol. 32, no. XVIII. Springer, 2006.
4. H. Samet, "Foundations of Multidimensional and Metric Data Structures", Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005.
5. G. Navarro and N. Reyes, "Dynamic List of Clusters in secondary memory", *Lect. Notes Comput. Sci.*, vol. 8821, pp. 94–105, 2014.
6. E. Vidal Ruiz, "An algorithm for finding nearest neighbours in (approximately) constant average time," *Pattern Recognit. Lett.*, vol. 4, no. 3, pp. 145–157, 1986.
7. E. Chávez, K. Figueroa, and G. Navarro, "Proximity Searching in High Dimensional Spaces with a Proximity Preserving Order", in *MICAI '05: Advances in Artificial Intelligence*, 2005, pp. 405–414.
8. E. Chavez Gonzalez, K. Figueroa, and G. Navarro, "Effective proximity retrieval by ordering permutations", *IEEE TPAMI.*, vol. 30, no. 9, pp. 1647–1658, 2008.
9. K. Figueroa and R. Paredes, "List of clustered permutations for proximity searching", in *Lecture Notes in Computer Science, SISAP*, 2013, pp. 50--58.
10. A. Esuli, "PP-index: Using permutation prefixes for efficient and scalable approximate similarity search", in *CEUR Workshop Procs.*, 2009, vol. 480, pp. 17–24.
11. K. Figueroa, R. Paredes, and R. Rangel, "Efficient group of permutants for proximity searching", in *MCPR*, 2011, pp. 42–49.
12. R. Fagin, R. Kumar, and D. Sivakumar, "Comparing Top k Lists", *SIAM J. Discret. Math.*, vol. 17, no. 1, pp. 134–160, 2003.
13. E. Chávez and G. Navarro, "A compact space decomposition for effective metric indexing", *Pattern Recognit. Lett.*, vol. 26, no. 9, pp. 1363–1376, 2005.
14. K. Figueroa, G. Navarro, and E. Chávez, "Metric Spaces Library," 2007.

X

Architecture, Nets and Operating Systems Workshop

Structural Locality, division criterion for the execution of non-Autonomous Petri Net on IP-Core

ORLANDO MICOLINI[1], MARCELO CEBOLLADA Y VERDAGUER[1] AND LUIS ORLANDO VENTRE [1]

[1]Laboratorio de Arquitectura de Computadoras - FCEFyN - Universidad Nacional de Córdoba
omicolini@compuar.com, {mcebollada, lventre}@gmail.com

Abstract. *This paper introduces a new concept, which we call "structural locality". Structural locality allows us to represent and divide non autonomous Petri Nets, with the objective of significantly reducing the hardware resources needed to run Petri Nets in an IP-Core. This has the advantage of enabling us to address larger problems.*
Petri Nets represented in this way raise an algorithm of execution that preserves the original model and facilitates parallelism.
Finally, a real case of application is exposed, showing the advantages of applying structural locality to a Petri Net with different temporal semantics and types or arcs, achieving an important reduction in the resources used on the FPGA that implements the IP-Core.

Keywords: *Petri processor, Hierarchical Petri Net, structural locality, IP-Core.*

1. Introduction

Nowadays, to improve the throughput and processing power in multicore computing systems, applications implement threads[1]; which are able to cooperate and execute concurrently. The complexity of multithreading applications is much higher than in sequential applications. This complexity is present in its design, error detection, testing, validation and maintenance [2]. It is also necessary to include a control mechanism, as semaphores, which penalizes execution time. For all these reasons, it is important to build the system's solution as a formal model, in order to make its implementation easier.
Recent researches show that models obtained with Petri Nets (PN) can facilitate the implementation of systems directly, using processors or IP-Cores which execute PN[3-6]. The main problems for these solutions are: the size of the matrix that represents the models, the different timed semantics, types of arcs and events types in transitions. These issues limit the size of the problems that can be addressed due to the hardware resource availability.
The PN processors (PP) have been implemented as IP-Cores in Spartan 6 FPGAs, as shown in [7, 8]. Due to the limitations of the existing hardware, PP

that are able to be synthetized can only include up to 50 places and 50 transitions. To solve larger systems, models with larger numbers of components or elements are required; such as: places, transitions, types of arcs and different timed semantics. Note that the resources demanded to the FPGA to implement the IP-Core, increase in proportion to the product of the maximum capacity by the number of transitions and the number of different types of arms.
For each type of arc there is a matrix which dimension is "place by transition", and there is also an input event queue as well as an output event queue for each transition. Also, depending on the type of time semantics, registers and counters, both of 32 bit, might be needed.
The present paper extends the use of the PN splitting mechanism, introduced in [8], using the concept of structural locality to obtain Hierarchical PN (HPN) that leverage resources in a more effective way, maintaining all the properties of the non-autonomous PN state equation.

2. Hierarchical Petri Nets

In HPN [9], each subnet that composes the system has a particular state at a given time, this results in different transitions sensitized on each subnet of the system at a given time. These transitions can be arbitrarily fired if there is no conflict between them. In case of conflict a priority scheme is applied to keep the system determinism.
There are two types of transitions in subnets: the inner transitions and border transitions [9]. For a border transition to be fired is necessary that all border transitions that represent that unique transition in the original system, are sensitized, this preserves the original semantic of the transitions in PN.
The border transitions of each subnet, are transitions that have been divided and belong to different subnets. This results in distributed transitions.

$$[T_i] \xrightarrow{Ri^{|Ti|x|Tb|}} [BorderTransitionsofSubnet[b]]$$

Where T_i are transitions of subnet i.

All border transitions that do not have internal transitions in the subnet i must be permanently sensitized in the border transition of the subnet i. For this purpose, a subnet i mask is created. The mask has a value of "one" for each border transition that is not in the subnet.

$$border\_sensitized\_of\_\_subnet[i] = BorderTransitionsofSubnet\ [i]\ \ OR\ mask\_subnet[i]$$

$$distributer\_sensitized = \bigwedge_{j=0}^{r-1} border\_sensitized\_of\_subnet[i]$$

Where r is the number of subnets in which the system has been divided.

Figure 1 shows: the divided transitions, distributed transitions, and border transitions of a PN that models a producer-consumer.

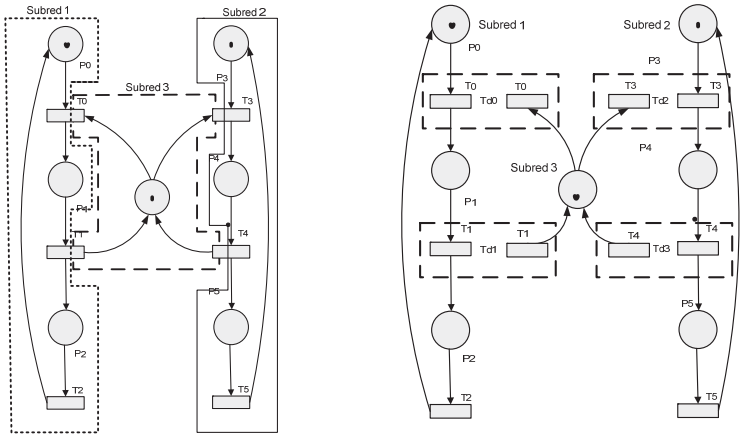


Figure 1. *Producer-consumer net divided in three subnets*

Distributed transitions in figure 1 are: Td0, Td1, Td2 y Td3. While internal transitions of subnet 1 are T0, T1 and T2; in the subnet 2 are T3, T4 and T5; and subnet 3 are T0, T1, T3 and T4.

The HPN firing algorithm maintains the original PN execution, and it is implemented in hardware by a combinational circuit, which executes the formerly described equation logic. This has been implemented with simple logic gates that keep the temporary benefits achieved by the PP [8].

Incidence Matrix range interpretation.

From the firing semantic of a PN we can interpret the incidence matrix considering the columns (transitions) as the conjunctive assessment of restrictions imposed by the rows (places). That is, in a mxn dimensión incidence matrix, n combinations of m logical variables are evaluated, as is expressed in the following equation:

$$if_{h=0}^{n-1}(\bigwedge_{k=0}^{m-1} p_i \geq w_{i,h})$$

Where $w_{i,h}$ are matrix I elements [3] that result from matrix subtraction between I_i^+e I_i^-, and they represent the weight and directions of the arcs of the PN; p_i is the mark on the place i.

3. Petri Net Division

The bibliographic research about division analysis of PN has been realized in [6]. PNs are formed by places, transitions and arcs, which we call elements. These elements participate in the incidence matrix as follows: transitions are column numbers, places are row numbers, and arcs are the values that relate a place with a transition, according a weight and a direction.
In PN we see that these elements are grouped according to subsets, characterized by being strongly interrelated. That means that, there are arcs linking places with transitions and vice versa. In turn, these subsets are weakly related with other subsets, i.e. there are a few arcs that relate to both subsets. Thus we see the incidence matrix, as a case of "sparse matrix" [10] [11] [12].
As we can see in bibliography, the algorithms used to solve these sparse matrixes do not apply to PP implementation with FPGA. The analyzed algorithms use compaction techniques and/or pointers to address these elements, they do not use simple logic operations; all this demands resources and machine cycles, which results in excessive overhead.
In order to obtain a reduction in resources, it is proposed to divide the PN into subnets. To explain this we now assume that a network, with M places and N transitions, can be divided into two subnets, and each one of them has half of the places (M/2) and half of the transitions (N/2), resulting in:

- The original system $\boldsymbol{M} * \boldsymbol{N}$ elements in the matrix
- The divided system $\mathbf{2}((\boldsymbol{M}/\mathbf{2}) * (\boldsymbol{N}/\mathbf{2})) = (\mathbf{1}/\mathbf{2})\boldsymbol{M} * \boldsymbol{N}$, Added to this, we need to consider an overhead that specifies the interrelation between subnets.

We note that the total amount of places and transitions of the systems is maintained, so in principle it's possible to keep its representation as long as we are able to establish how to interrelate the subnets. Moreover, the size of the overhead that specifies the relationships between subnets must be less than$(\mathbf{1}/\mathbf{2}) * \boldsymbol{N}$. This difference will result in reduced resources. For this case, in which the network has been divided in two subnets, the maximum gain is by a factor of two (excluding the overhead).
We can assume that, if the network is divided in J parts, the theoretical maximum gain will be a function of J minus the interrelation cost. The determination of the theoretical maximum gain without the structural locality concept has been treated in [9]. Given the above, we propose to further reduce the amount of resources in the hierarchical PP (HPNP), where the division will be made assuring that every PP that integrates the HPNP has only the needed resources to execute its subnet.

a. Consideration for PN division.

To realize the resource reduction that we have proposed, which is dividing a PN in the proposed form and manner, it is necessary to answer the following questions:

- How much, and which is the gain obtained from dividing the PN? This has been answered, excluding the structural locality concept, in [9].
- Are the resulting PN simpler? The subnets are simpler, because it is possible to divide the transitions as many times as it is desired, and this allows us to choose the simplest subnets.
- In terms of the number of elements: Is there a base for the division of the original PN with which an improvement is obtained with respect to a sub netting balanced division? We seek to answer this question in this work. A divided PN has to express and preserve the original behavior of the non-autonomous Petri Net (NaPN). This has two aspects: the transitions firing rule and the relation between network and the events. The first aspect has been shown in [9] . As for the second one, it is important to maintain the relationship between the transitions and the events that fire the transitions. Also, the communication generated by triggering a transition must be able to be informed. This means that the division does not introduce ambiguity in the events generated by firing a transition, this has been considered and solved in [6]. Finally, the priorities scheme must provide a way to implement and facilitate the parallel execution. For this, the transitions that are in conflict in each subnet are solved using a local priority scheme. Furthermore, the border transitions have a higher priority than the subnet's local transitions, and they have their own priority scheme.
- Figure 1 shows a transition divided PN. Divided transitions will be present in each resultant subnet; the subnets communication is executed by border transitions, which are the subnets boundaries. We can say then that there is a relationship between all parts of each divided transition that originally constituted a single transition. These parts are reported, evaluated, and fired simultaneously. It should be noted that by dividing the networks in this manner, an explicit network hierarchy is generated, with a wider concept that we call "Hierarchical PN divided by transitions". This division results in a resource reduction with respect to the original network. Thus we consider the resultant system as a HPN, which facilitates the implementation and interpretation on different devices interconnected as performed in [13].

b. Considerations for HPN divided by transitions.

The first thing we perceive is that there is only one network. There is no global network and subnets, but all parts of the network or subnets have the same hierarchy and relate to each other through border transitions. However, for each obtained part of the divided network, we will call it subnet.

1. The distributed transitions are constituted by the set of all border transitions, each of which results from dividing a specific transition by the total amount of networks.
2. To obtain the global state of the system we consider all the places together, this means that you need to know the marking of each separated subnet.
3. When a distributed transition is fired, all border transitions from all of the subnets related to it must be fired as well in order to ensure the fire semantic. That is, when all border transitions that belong to the same distributed transition are sensitized, this distributed transition is able to be fired.
4. In order to ensure that firing a distributed transition does not generate conflict (with any border transitions) in a subnet, only one distributed transition is allowed to be fired in each firing cycle.
5. The network is divided by splitting transitions; therefore the transitions chosen as a border between two or more subnets stay divided, resulting in a border transition on each subnet. Border transitions in each subnet are interrelated, forming a unique distributed transition.
6. As each subnet keeps its operation separately, the level of parallelism in the execution of the system is increased; being possible, as a maximum, that all subnets fire at the same time (in transitions that are not borders) in the same cycle.
7. The binary matrix, that represents the relationship between the distributed transitions, shows how distributed transitions regroup into the original transitions on each subnet.
8. To avoid excessive communication between subnets, in case of conflict, distributed transitions have higher firing priority than any internal transitions in the subnet.
9. If it is necessary for an internal transition to have a higher priority than a border transition in conflict, it can be defined as a distributed transition that is only related to the subnet to which it belongs.

Now we introduce the structural locality concept, to guide the network division, that is: *"divide the network by transitions, grouping together the elements with common specifications on each subnet"*. These elements are: arcs with weight of one, arcs with weight higher than one, timed transitions, transitions with time, and other kinds of arcs.

4. An example: Division of network with N readers and a writer in three subnets.

Figure 1 shows the PN that models the system with N readers and a writer, which has been divided in three subnets.
The original network is formed by 7 places (m) and 6 transitions(n), including the priority matrix (nxn), for this PN the PP requires 372 bits.

$$PN = (mxn)8 + (nxn) = 372$$

Dividing this PN, as shown in figure 1, the resources needed are 290 bits; as shown in Table 1.

Table 1: PN resources N readers a writer, divided in three subnets

	Places	Transitions	P. Matrix	Resources
Red0 y Red1	3 ea.	3 ea.	3x3	81 ea. = 162
Red2	1	4	4x4	48
Relationship Matrix	Border transitions are 8. With 4 subnets 4x4x3			48
Priority Matrix border transitions.	4*4			16
Total resources divided network – Reduction resources 79%				290/79%

As we can observe, the division has a resource gain of 21%.
Since subnets one and two only need arcs with weights of one and minus one in the matrix, only 2 bits are needed for those arcs representation. On the other hand, subnet three requires arcs with a weight of N, which needs 8 bits; the resources needed are 55.9%. This additional gain is the result of applying the structural locality concept.

5. Case of application

Now we apply the concept of structural locality into a case of modeling with PN, to analyze performance in web searching applications. The case is taken from [14] (A Modeling Technique for the Performance Analysis of Web Searching Applications). This paper investigates the behavior of a client/server system by using a non- Markovian model made with NaPN to obtain performance indexes. It was also made using an experimental setting, in order to obtain real measurements that are used to validate the analytic model.

a. PN model to calculate client-server system response time.

Figure 2 shows the PN presented by the author; it has been divided into five subnets. The division has been done with the structural locality criterion presented in this paper, gathering the transitions of the same type and also keeping the processing blocks of the original network.

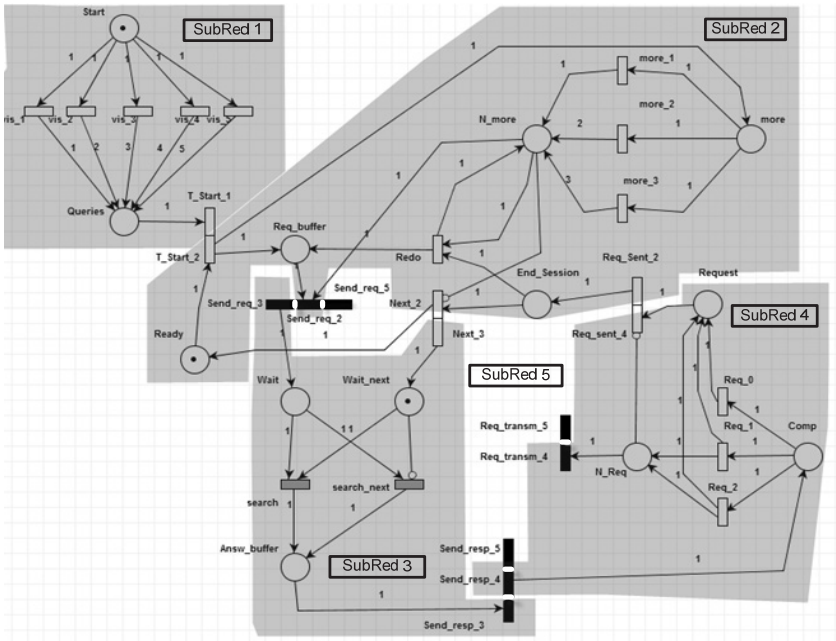


Figure 2: *Divided PN model for web searching analysis*

Figure 2 shows three different types of transitions: immediate transitions represented by an empty rectangle, timed transitions represented by gray rectangle and delayed transitions represented by a black rectangle.

It should be noted that the original PP with time [5, 15] is not able to support the two types of transitions with time semantic simultaneously , while the HPNP supports the two types of transitions with time on different subnets. In accordance with the concept of structural locality, developed in this work, the network has been divided grouping the different types of temporal transitions into the same subnet.

The resources needed to implement the temporal Petri processor (TPP) without divided network are: 3220 bits.
Places = 13, transitions = 20, time comparator 2 x 20 and time counters = 1 x 20. Each comparator and counter resource needs 32 bits. That's the number of bits to support the maximum weight of the 4 bits arcs (including sign).
To simplify the resource calculation, one FF (flip flop) per bit is taken as the unit, so it requires 3220 bits to implement the PP without any network division, while to implement the HPNP requires 1384 bits, so that a 232.6 % gain is obtained.

Table 2: Resources for PN of figure 2 with and without division

	PPT Without Division	Resource needed to implement 5 subnets in HPNP Transitions of network divided by 6				
		Subnet 1	Subnet 2	Subnet 3	Subnet 4	Subnet 5
Places	13	2	5	3	3	0
Transitions	20	11	10	8	9	6
PxT (weight 4bit)	13x20x4	2x11x4	5x10x4	3x8x1	3x9x1	1x8x1
Inhibitors Arcs	13x20x1	no	5x10x1	3x8x1	3x9x1	no
Divided Transitions subnet	NC	1	3	3	2	6
Non divided Transitions	NC	5	4	2	3	0
Time Counters, 32bit	20x32	0	0	8x32	0	6x32
Time comparators 32 bit	2x20x32	0	0	8x32x2	0	0
Resources needed	3220	88	250	792	54	200
Subtotal	3220		1384			
Gain	232,6%					

6. Conclusions and future work

We can highlight that the application of the structural locality concept and network division impacts directly in the hardware resource reduction needed to instantiate the IP-Core. While it is not possible to generalize, for the case of analysis, which is not optimal, the obtained saving in resources is around 57%. This is compared with a processor that instantiates all the possibilities for all the transitions.

The division in subnets, considering that each subnet has common time semantics, allows to instantiate heterogeneous PN using less hardware resources because each subnet only has the strictly necessary resources.

We can infer that the more heterogeneous the transitions semantics are, the more rectangular subnets matrix become, and the lower amount of divided transitions the bigger is the gain.

According to the obtained results in the tests that were done using the IP-Core HPNP for parallel and concurrent systems with time, the improvement in response time regarding the use of semaphores is around 40% and 60%. These are the same results obtained in the PP.

Finally, we mention, that the use of structural locality improves the versatility of PNs to model concurrent and parallel systems, which makes the resulting system flexible to changes in implementation.

As future work, an algorithm for automatic network division is being developed.

References

1. R. A. B. David R. Martinez, M. Michael Vai, *High Performance Embedded Computing Handbook A Systems Perspective*. Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, Massachusetts, U.S.A.: CRC Press, 2008.
2. M. Domeika, *Software Development for Embedded Multi-core Systems*. 30 Corporate Drive, Suite 400, Burlington, MA 01803, USA Linacre House, Jordan Hill, Oxford OX2 8DP, UK, 2008.
3. M. Diaz, *Petri Nets Fundamental Models, Verification and Applications*. NJ USA: John Wiley & Sons, Inc, 2009.
4. M. Pereyra, M. A. N. Gallia, and O. Micolini, "Heterogeneous Multi-Core System, synchronized by a Petri Processor on FPGA," in *IEEE LATIN AMERICA TRANSACTIONS*, 2013, pp. 218-223.
5. J. N. y. C. R. P. Ornaldo Micolini, "IP Core Para Redes de Petri con Tiempo," *CASIC 2013,* pp. 1097-110, 2013.
6. O. Micolini, "ARQUITECTURA ASIMÉTRICA MULTI CORE CON PROCESADOR DE PETRI," Doctor, Informatica, UNLaP, La Plata, Argentina, 2015.
7. O. Micolini, J. Nonino, and C. R. Pisetta, "IP Core Para Redes de Petri con Tiempo," in *CASIC 2013*, 2013, pp. 1097-110.
8. N. G. M. Pereyra, M. Alasia and O. Micolini, "Heterogeneous Multi-Core System, synchronized by a Petri Processor on FPGA," *IEEE LATIN AMERICA TRANSACTIONS,* vol. 11, pp. 218-223, 2013.
9. O. Micolini, E. Arlettaz, S. H. B. Baudino, and M. Cebollada, "Reducción de recursos para implementar procesadores de redes de Petri," in *en Jaiio 2014*, 2014.
10. S. Pissanetzky, *Sparse Matrix Technology*. Bariloche, Argentina: Centro Atómico Bariloche, 1984.
11. S. Kestury, J. D. Davisz, and E. S. Chungz, "Towards a Universal FPGA Matrix-Vector Multiplication Architecture," *Field-Programmable Custom Computing Machines (FCCM), IEEE 20th Annual International Symposium on,* 2012.
12. G. H. Golub and C. F. V. Loan, *Matrix Computations* Johns Hopkins, 2012.
13. R. Pais, Barros, J.P. ; Gomes, L., "From Petri net models to C implementation of digital controllers " *Emerging Technologies and Factory Automation. ETFA 2005. 10th IEEE Conference on,* 2010.
14. M. Scarpa, A. Puliafito, M. Villari, and A. Zaia, "A modeling technique for the performance analysis of Web searching applications," *Knowledge and Data Engineering, IEEE Transactions on,* vol. 16, pp. 1339-1356, 2004.
15. M. Orlando, J. Nonino, and C. R. Pisetta, "IP Core for Timed Petri Nets," in *CASECongreso Argentino de Sistemas Embebidos*, 2013, pp. 3-8.

Topology Control Strategy for Reduce Interference on Multihop Networks

NELSON R. RODRÍGUEZ[1], MARÍA A. MURAZZO[1] AND EDILMA O. GAGLIARDI[2]

[1] Departamento e Instituto de Informática, Universidad Nacional de San Juan
Complejo Universitario Islas Malvinas, 5400, Rivadavia, San Juan, Argentina
[2] Departamento de Informática, Universidad Nacional de San Luis,
San Luis, 5700, Argentina
nelson@iinfo.unsj.edu.ar[1], maritemurazzo@gmail.com[2], oli@unsl.edu.ar[3]

Abstract. *The wireless networks show distinctive characteristics in terms of interference, time battery life and loss of connectivity in mobile networks. Various strategies have been proposed, some to level of MAC layer and other through reducing the traffic. Topology control is a technique used in distributed computing based on graph theory and computational geometry, that allows get a subgraph connected and also reduce interference for each type of network. This work presents strategies applied for reducing interference in multihop networks through topology control and they are compared with model of minimal distance than is used habitually in routing algorithms.*

Keywords*: Topology Control, Interference, MANET, Wireless Networks*

1. Introduction

The advance of wireless networks in recent years has been surprising. This is due to many reasons such as: the growth of cell telephony, reduction of costs of all devices communication (handsets, routers, switchers, antennas, etc.), the increasing sales of end users equipments (smartphones, cell phones, netbooks, tablets, notebooks), the expansion of hotspots, the dissemination of technologies by Linux users, who build wireless communities and the gadgets culture by young people [20].
The rapid development who have experienced wireless networks in recent years it has allowed offering users different solutions. They are presenting fundamental changes in technology, services, and paradigms of business. These changes are essential for the emergence of communication paradigm known as 4A (*"anytime, anywhere, anyhow, anyone"*), also known as ubiquitous computing and basis for pervasive computing [25].
The wireless networks, as an access networking paradigm, are considered to be an inevitable part of the future systems enabling broadband access.

The area of application of such networks is extremely wide as: search and rescue missions, environmental monitoring, surveillance, military communications in battlefield, explorations of space, disaster recovery, and operations under the sea, smart transport systems and sensor systems for agriculture, between others. They can also be used as the last mile technology to provide Internet access in highly populated cities [10].
WMNs (Wireless Mesh Networks) can efficiently satisfy the needs of multiple applications between them: broadband Internet access, indoor WLAN coverage and deliver higher bandwidth than the best 3G technology for mobile user access [21].
Wireless distributed sensor systems will enable fault tolerant monitoring and control of a variety of applications. have been widely used in a variety of long-term and critical applications, including event detection, target tracking, environment and habitat monitoring, localization, safety navigation, seismic activity detection, detection of chemical/biological agents and so on [16] [6].
In the particular case of Ad-hoc networks they are very suitable for situations where infrastructure or is unavailable or unreliable, while mesh networks must have an infrastructure to provide the services for which they were designed.

2. Wireless Networks

The wireless networks multihop, can be classified according to their general characteristics into four types: Ad-hoc Networks, Sensor Networks, Mesh Networks and Hybrid Networks.
The Ad-hoc networks can be fixed or mobiles, in the latter case they are called MANETs (Mobile Ad-Hoc Networks), and mobility adds more difficulty to network connectivity. The most striking feature of MANETs is that unlike WLANs do not require a fixed infrastructure for establish communication, unlike, wireless nodes cooperate to communicate any pair of nodes in the network. MANETs are known as wireless networks without infrastructure [5].
Unlike the MANETs, the WSN nodes (Wireless Sensor Networks) are stationary and the dynamic topology changes are less frequent. Also in WSM, the wireless nodes are access points and Internet Gateway, and therefore have no energy restrictions.
On the other hand, the WSN usually have a high density of nodes, because: many sensors can deplete your battery, including those with solar energy and is difficult to replace.
WSNs usually reach lower speeds than the other networks, due to the reduction of energy consumption and the use of protocols such as Zigbee (of order of Kbps). Unlike, Multi-radio WMNs architecture, in which each access router is equipped with multiple 802.11 radios, is commonly seen as a practical way for efficient utilization of available spectrum and thus reaches higher speeds.
The primary advantages of WMNs consist in its inherent fault tolerance against network failures, simplicity of setting up a network, and the

broadband capability. Unlike cellular networks where the failure of a single base station leading to unavailability of communication services over a large geographical area, WMNs provide high fault tolerance even when a number of nodes fail. However, in WSNs the only way to provide fault tolerance is a higher density of sensors.
Hybrid wireless networks are networks in which any mobile node in a wireless network may have connectivity, either directly or via a gateway node, to an infrastructure network. This latter network may be an IP network as the Internet, a 3G wide area wireless network, or an 802.11 local area wireless network. Actually, any other network technology may be considered. These hybrid networks may use multiple technologies for both WMN backbone and back haul.
The following illustrations (fig. 1) graphically reflect the similarities and differences of the various multi-hop wireless networks [22].

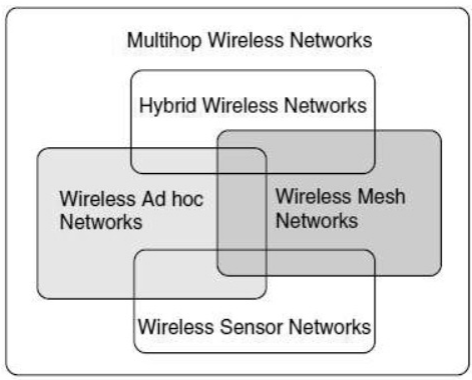


Fig. 1. *Multihop wireless networks types.*

3. Energy Consumption

The energy consumption is perhaps the most important problem of wireless networks, because a good use of it extends the lifetime of the network. In sensor networks and a-hoc networks, energy is consumed mainly for three purposes: data transmission, signal processing and hardware operations.
Power consumption is larger when the signal is transmitted with greater power for greater scope. For example, a sensor can consume 35mA transmitting to 8dBm, while consume 27 mA transmitting to 4 dB. These values are approximate because they depend on various factors such as type of sensor and communication protocol, but in all cases, increases transmit power to reach more nodes, greatly increases consumption [4] [18].
Considering the above, if the it fits transmission power of each node so as to maintain the network related and connect any node of the network to another through intermediate hops or directly, power consumption is reduced.

Moreover, reduces the interference caused by the connection of all with all and the retransmissions by noise are also reduced.
Energy efficiency of a node is defined as the amount of data delivered by a node and the total of energy expended. High energy efficiency means that a large number of packets can be transmitted by a node with a certain amount of energy reserve. The main reasons for power management in wireless ad hoc networks and WSN are: limited reserves of energy, difficulties in replacing batteries, lack of central coordination, restrictions in choosing energy sources and use of shared channels.
On the other hand, must consider that an energy consumption model should include the four states in which can be a node: acquisition, transmission, reception and waiting. By applying the right strategies, consumption of energy to packaged, coding, framing and shipping for transmission, decoding, error detection and checking of the reception direction also it decreases.

4. Interference

Interference occurs when a message from a node gets corrupted due to other concurrent transmission of another node on the same transmission range.
For a network, each node generates a transmission that is modeled with a disk or circle. The interference of a node is defined as the number of disks that they include.
The interference causes loss of messages which results in a high consumption of energy due to retransmissions. This situation can occur in dense networks with high traffic.
Some interference models as published by Meyer [15], they are based on current network traffic. However, the amount and nature of traffic is highly dependent on the type of application chosen. Because usually there is no a priori information about network traffic, a static interference model is desirable, since it depends only on the set of nodes.
Suggestions physical layer to reduce interference, as spectrum analyzers or switchers that administer interference problems, may be useful in static networks. But in MANETs and mobile WSN it is impossible to perform.
Another alternative is to oversize the network capacity, adding high density of access points, but this will cause interference of co-channel and does not apply in MANETs and WSN.
There are several studies that aim to reduce interference in wireless networks and particularly in MANETs and sensor networks, but not all publications are based on topology control using computational geometry. Some of the published papers make interference statistics estimates based on characteristics of the MAC sublayer (and variants), for example, it referred to in Chapter 8 of the book "Ad - hoc Networks " of Ramin Hekmat [11]. Other work presents a model to analyze the performance of the strategies of transmission in networks of packet radio multihop, where each station has radio of adjustable transmission [23]. In both cases present analysis of

throughput but is not proposed changes of strategies to improve the efficiency of the networks.
Often it is disputed that the dispersed topologies with little or limited degrees are fitted to reduce the interference. However, low degree alone does not guarantee low interference. In addition, it can be shown that the majority of topology control algorithms obtains values that are not near to the optimal interference, and therefore should find a graph that minimizes interference first [24].

5. Topology Control

In wireless networks each network node has the possibility to change the topology of the network by adjusting the transmission power in relation to other neighboring nodes. In contrast to wired networks that have a pre-configured fixed infrastructure. This adjustment could go so far as to have zero range (without power consumption but resulting in not connected network) to power maxim for reach all nodes, but resulting maximum interference.
The fundamental reason for the topology control scheme in
MANET and WSN are to provide a control mechanism that maintains the network connectivity and performance optimization by prolonging network lifetime and maximizing network throughput [1].
On the other hand, establishing connectivity in a wireless network can be a complex task for which various (sometimes conflicting) objectives must be optimized. To permit a packet to be routed from any origin node to any destination node in the network, the corresponding communication graph must be connected. In addition to requiring connectivity, various properties can be imposed on the network, including low power consumption, bounded average traffic load, small average hop distance between sender-receiver pairs, low dilation (t-spanner), and minimal interference; this latter objective, minimizing interference (and, consequently, minimizing the required bandwidth), is the focus of much recent research [12].
A wireless network topology can depend on uncontrollable factors such as node mobility, weather, interference, noise as well as controllable factors such as transmission power, directional antennas and multi-channel communications.
Depending on the network, the target of topology control is different. In WMN it comes to getting low latency routes or high transfer speeds that are efficient in energy consumption
This consumption is extremely important in WSN and ad hoc networks, so in such networks, it deals to reduce interference for to obtaining fewer retransmissions.
The techniques used to implement the topology control are based on computational geometry. This discipline provides a theoretical and formal framework for structural design and analysis of algorithms required to provide solutions to problems in various areas of computing, resolving them applying geometric of mode constructively [8]. There are a variety of published papers on the construction of graph algorithms using these techniques and even solving routing problems in networks [2].

Wireless networks are modeled as Disk Unit Graph (UDG) [7] [17]. In this type of graph, two nodes can communicate only if the distance between them is at most the unit; meaning of unit is within range of each node. Usually also it is used with other intersecting planar graphs to get planarity.
Due to limited power and memory, a wireless node prefers to only maintain the information of a subset of neighbors it can communicate, which is called topology control.
The algorithms of topology control are designed for different objectives: minimizing the maximum link length (or node power) while maintaining the network connectivity [19], bounding the node degree, bounding the spanning radio or constructing planar spanner locally (in graph theory, the concept of t -spanner, is to find a graph G ' of a graph G such that G approaches distances with a precision factor t) [6].
Some researchers have tested reduce interference by reducing the power consumption of each node or providing controls to provide low grade topologies for nodes but none of these strategies ensures reduce interference in all cases.
Burkhart [3] proposed several methods to build topologies whose maximum link interference is minimized while the topology keeps connected nodes.
In most cases in which the ad- hoc and WSN networks are used, nodes are not aware of the location of other nodes. Whereby the objective is to obtain a method for constructing the network locally. That is the radio communication of a sensor node or not depends on the location of a sensor or node that is far [13].
The development of this work was based on the following definitions. They are published in various papers graph theory and computational geometry based topology control.

Interference from Topology Control

uv link interference is defined, as the set of nodes whose distance from the node *u* or *v* is less than | *uv* |. That is, the number of nodes covered by two disks centered at *u* and *v* with radius | *uv* |.
Specifically, they define the coverage of a link *uv* as:
cov (*uv*) = {w | w is covered by D(u, |uv|) or D(v, |uv|)}. It is graphed in Figure 2.

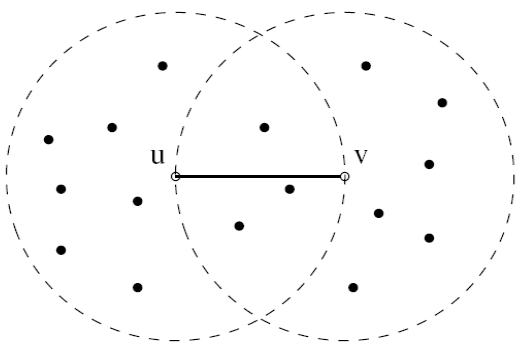


Fig. 2. *Graphical definition of interference on the link uv.*

How, cov (*uv*) represents the set of all nodes that could be affected by nodes u by node v when they communicate with each other using exactly the minimum power need to reach each other.
Then, the *maximum interference* of a graph G (V, E) was defined as highest coverage edge of G.
Total interference of a graph G (V, E) is the sum of all interference.
Another important definition is *average interference*. The same is defined as dividing the sum of all edge coverage in G to edge count.
The network is then represented by a geometric undirected weighted graph G = (V, E) with vertices representing wireless nodes, and edges representing communication links. The weight of each link *uv* is its interference number.

6. Simulations

Regular topologies (triangle, square, hexagon, or other geometric figures) and random uniform distributions are widely used in analytically tractable models. But the real cases also occur arbitrary or mesh topologies, which case is more appropriate simulation model.
The results presented in this paper were developed under certain circumstances:

- The links are bidirectional and symmetric, meaning that a message sent over a link can be acknowledged by the receiver.
- It is considered to stationary nodes, since there may be differences in traffic models and mobility.
- The transmission range of each node is modeled by a circle.
- Not are modeled directional antennas per sector.
- All nodes are equal in terms of processing capabilities, battery and memory.
- The nodes are randomly generated in 70% of the graphs; the remaining graphs are generated linear topologies, ring and geometric shapes.

The values of the simulations have been compared with the minimum distance model. That is, the subgraph obtained reduces interference and produces a graph of greater length, secondly the modeled graph presents minimum distance shorter but with a greater degree of interference.
We put different numbers of nodes that are randomly placed in a 500 x 500 meters square.
Simulations for networks with 5, 10, 50, and 100 nodes were performed. The number of nodes on the stage dimensions allows that some simulations presented some networks scattered and other dense.
For each number of nodes (5, 10, 50 and 100), 20 independent simulations were performed.
For graphs with nodes 5, in 50% of cases, values it matches interference between the subgraph obtained by applying the interference model and the minimum distance model; and of course the remaining 50 %, the interference value was lower. Regarding the overall length of the graph obtained, also in 50% of cases they coincided and the rest the length was greater interference graph.

For graphs of 10 nodes, the results were similar, but an important result is that in the topologies with nodes randomly generated, interference is important and therefore result inadequate the model based on distance minimal. However, geometric figures topologies, both models show few differences and thus may be used interchangeably.
With 50 nodes and 100 nodes the relation between the two models remain, but obviously when increase the number of nodes, interference values increase between 5 and 10 on average in both simulations, resulting most convenient interference model.
Then in figure 3 and 4, the results obtained for graphs with nodes 100 are shown, comparing the interference model with the minimum distance model. In the figures, values are shown: total interference graph and the graph full length.
It may be noted in the graph comparing the interference (Figure 3) that the interference model is better, and the graph comparing the distances (Figure 4) the distance model has advantages. Yet when relayed that interference occurred, it produces a greater delay than most meters on the link and consumes more battery power.
Finally , figure 5 shows the average interference for nodes 100.

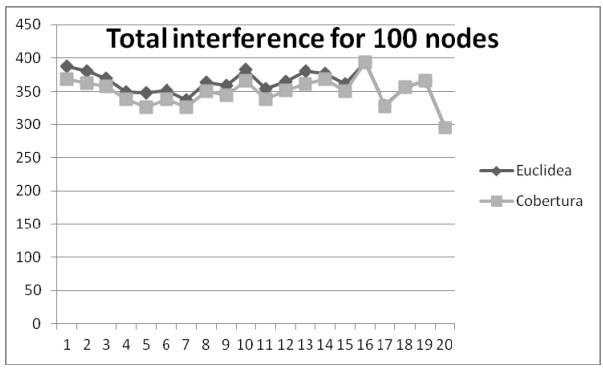


Fig. 3. *Comparing coverage model and length, regarding interference*

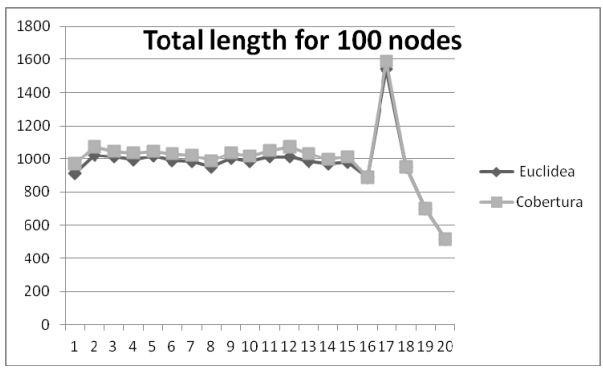


Fig. 4. *Comparing interference model and length, taking into account the length*

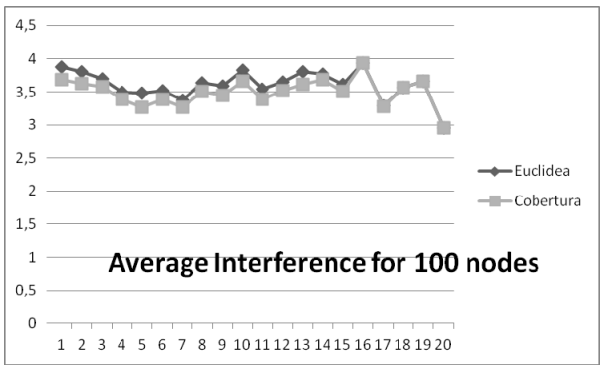


Fig. 5. *Average interference*

7. Conclusions and Future Work

The topology control allows obtaining a subnet of a given network to improve certain parameters thereof. Because wireless networks are very different, the objectives of this control also changes.
In WSN and MANET reducing interference directly impacts energy consumption in this type of network determines the "life" of the network. However, in WMN the performance is more important since nodes are usually connected to a power line, although such networks if interference is reduced also impacts in the performance.
Wired networks use metrics such as lower bond length or number of hops, which are efficient for such networks. However, wireless networks have characteristics that difference. This work shows that it is appropriate to replace the usual metrics by interference or coverage.
It is expected to work in the future with mobility scenarios, even without implementing traffic models, primarily for networks with little change as pedestrian networks. Also it foresees to him carry out in analysis of loss of packages in function of the density of the nodes.
It is also planned to perform the analysis of QoS for application class (in this case without applying mobility models). For example, applications for latency sensitive side and ability as voice over wireless LAN (VoWLAN) and on the other side applications less sensitive to delay as Web browsing.

References

1. T. Asha, N. Muniraj, "Network Connectivity based Topology Control for Mobile Ad Hoc Networks", International Journal of Computer Applications, pp. 975-987, Vol. 56, No. 2, 2012.

2. M. Berón, O. Gagliardi, G. Hernández Peñalver, Estrategias de ruteo alternativas para redes móviles. In: XI Congreso Argentino de Ciencias de la Computación, 2005.
3. M. Burkhart, P. Von Rickenbach, R. Wattenhofer, A. Zollinger, "Does Topology Control Reduce Interference?", Proceedings of ACM Mobi-Hoc, 04 pp. 9–19, 2004.
4. M. Calle Torres, "Energy consumption in wireless sensor networks using GSP", Master of Science in Telecommunications. University of Pittsburgh, 2006, http://d-scholarship.pitt.edu/7682/1/callemariag072606.pdf
5. P. Chandra, "Bulletproof Wireless Security - GSM, UMTS, 802.11, and Ad Hoc Security", Elsiever Communications engineering series pp. 121-127, 2005.
6. Z. Chen, P. Xu, X. Deng, "A Distributed Planar t-Spanner Topology Control Algorithm in Wireless Sensor Networks", Journal of Computer Research and Development. Vol. 49, No. 3, 2012.
7. B. Clark, C. Colbourn, D. Johnson, "Unit Disk Graphs", Discrete Mathematics 86, pp. 165-177, 1990.
8. O. Gagliardi, M. Taramilla, M., Berón, G. Hernández Peñalver, "La geometría computacional a nuestro alrededor", IV Workshop de Investigadores en Ciencias de la Computación, 2002.
9. P. Garg, N. Pawar, "A Review on the topology control approaches for utilizing MRMC over WMNs", SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE), EFES 2015. ISSN: 2348 – 8387, www.internationaljournalssrg.org.
10. L. Gaurilovska., R. Prasad, "Ad hoc netwoking towards seamless communications", Ed. Springer, Holanda ,2000.
11. R. Hekmat, "Ad-hoc Networks: Fundamental properties and Network topologies, Chap. 8, Interference in Ad-hoc Networks", pp. 77 a 94, 2006.
12. M. Khabbazian, S. Durochez, A. Haghnegahdaz, "A Bounding interference in wireless ad hoc networks with nodes in random position", Networking, IEEE/ACM Transactions. Issue 99, 2014.
13. M. Korman, "Minimizing interference in ad hoc networks with bounded communication radius", Information processing Letters,112, pp 748-752, 2012
14. M. Li, L. Zhenjiang, and A. Vasilakos, "Survey on Topology Control in Wireless Sensor Networks: Taxonomy, Comparative Study, and Open Issues", Proceedings of the IEEE, Vol. 101, No. 12, 2013.
15. Meyer auf der Heide F., Schindelhauer C., Volvert K.: Congestion, Energy and Delay in Radio Networks. Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures. pp..230-237, 2002
16. R. Min et al., "Low Power Wireless Sensor Networks", In the Proceedings of International Conference on VLSI Design. Bangalore, India, 2001.
17. G. Peñalver Hernandez, http://www.dma.fi.upm.es/gregorio/grafos/prorout/grafos.htm
18. D. Puccinelli, M. Haenggi, "Wireless Sensor Networks-Applications and Challenges of Ubiquitous Sensing", IEEE Circuits and Systems Magazine, pp.19-29, 2005.

19. R. Ramanathan, R. Rosales, R. Hain, “Topology control of multihop wireless networks using transmit power adjustment”, INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2000.
20. N. Rodríguez, “Nuevas configuraciones de redes inalámbricas y consideraciones en la elección”, II Congreso de Informática del nuevo cuyo - Jornadas de Informática – San Juan, 2007
21. M. Sichitiu, “Wireless Mesh Networks: Opportunities and Challenges”, Wireless World Congress, 2005, http://www4.ncsu.edu/~mlsichit/Research/Publications/wwcChallenges.pdf
22. C. Tchepnda, H. Moustafa, H. Labiod, France Telecom R&D, “Hybrid Wireless Networks: Applications, Architectures and New Perspectives “, Sensor and Ad Hoc Communications and Networks”, SECON '06, 3rd Annual IEEE Communications Society on, 2006.
23. H. Ting-Chao, “Transmission Range Control in Multihop Packet Radio Networks”, IEEE Transactions on communications, Vol. Com 34, Nª 1, 1986.
24. P. von Rickenbach, R.Wattenhofer, and A. Zollinger, “Algorithmic Models of Interference in Wireless Ad Hoc and Sensor Networks”, IEEE/ACM Transactions on Networking, vol. 17, No. 1, 2009.
25. P. Wirth., AT&T Labs, “The Role of teletrafic modeling in the new communications paradigms”, IEEE Communications Magazine, 1997.

VII

Innovation in Software Systems Workshop

3D Mobile Prototype for Basic Algorithms Learning

FEDERICO CRISTINA[1], SEBASTIÁN DAPOTO[1], PABLO THOMAS[1] AND PATRICIA PESADO[1,2]

[1] Instituto de Investigación en Informática LIDI
Universidad Nacional de La Plata – Argentina
[2] Comisión de Investigaciones Científicas de la Provincia de Buenos Aires - Argentina
{fcristina, sdapoto, pthomas, ppesado}@lidi.info.unlp.edu.ar

Abstract. *The educative environment must adapt itself to changes and new ways of learning. M-learning proposes modern support methods for the learning process through the use of mobile devices. This way, it is possible to count on e-learning features at any place and time. There is a particular interest in the development of software tools that provide learning support at initial levels of computer science careers. This, in conjunction with current mobile devices processing power, allows the development of a visual 3D application prototype for learning basic algorithms, which is presented in the current paper.*

Palabras clave: *M-learning, Unity, basic algorithms*

1. Introduction

Currently, people use mobile devices such as cell phones, smart phones or tablets just like any other every day accesory. By means of a wide variety of applications, relevant information is retrieved from several Internet electronic services, entertainment applications are enjoyed, commercial transactions are performed, and it is even possible to control health-related issues. Besides, many of these applications allow users to be interconnected [1].

In a similar way, this new technology has the potential to be used in teaching and learning processes [2][3]. M-learning is an evolution from e-learning through the use of mobile devices. The learning process is then transformed into an interactive, cooperative, portable and personalized activity.

One of the most important characteristics that M-learning proposes is to provide complete flexibility for students; that is, it allows content selection according to the user needs, at the required time and place. Besides, contents must not depend on any particular device. Finally, this technology independency must be augmented with content adaptation considering navegability, processing power and connection speed for a wide range of mobile devices.

However, most current mobile devices have evolved considerably. Their technological evolution allows the execution of complex, hardware demanding applications. This has a particular importance when developing

tools with 3D graphics that require this computational power in order to achieve a correct execution of the application.
The tools that provide a 3D environment are visually more pleasant and users are more attracted to them. Largely this is due the fact that 3D environments are more similar to reality than 2D environments, which allows a better and more active involvement from the user.
The rest of the paper is organized as follows: section 2 describes the motivation behind the building of the prototype; section 3 shows a preliminary analysis of current 3D mobile development tools; section 4 introduces the developed prototype; section 5 presents the conclusions; finally, section 6 poses possible related future work.

2. Motivation

The main concepts to be included in initial levels of computer science careers provide a favorable context for the creation of tools that support the features required for m-learning.
In particular, inaugural classes at the Facultad de Informática - Universidad Nacional de La Plata, a special tool for learning basic algorithm concepts is used, called R-Info [4][5][6][7][8]. By its use, students can create simple to moderate complexity programs and visualize its execution. Thus, the tool allows students to solve problems in an visual and attractive environment.
R-Info is a simple abstract machine, a mobile robot controlled by a reduced set of primitive instructions, which allows the modeling of tasks for the robot within a city composed of streets (horizontal paths) and avenues (vertical paths). Figure 1 shows that the city is a square with 100 streets and 100 avenues.

The robot can perform the following basic tasks:
1. Move forward one block.
2. Turn right (rotate clockwise 90 degrees).
3. Recognize two special types of objects: flowers and papers. These object can be placed at the corners of the city (street and avenue intersections).
4. Carry flowers and papers in a bag that the robot holds. Besides, the robot is able to pick up and deposit these objects in a corner, but only once at a time. The bag has unlimited capacity.
5. Change the position of the robot to any other corner of the city.
6. Perform simple calculations, even using variables if necessary.
7. Make use of control structures, such as *mientras* (while) or *repetir* (repeat).
8. Display results to the user.

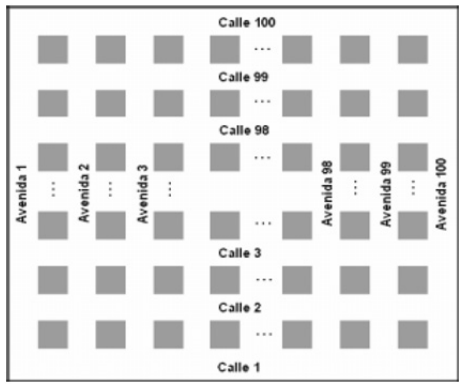


Fig. 1. *The city where the robot moves around*

The main screen of R-Info has a control panel that allows - among other things - writing the source code of the algorithm and then executing it. Moreover, the application has a tool for editing the city, setting flowers and/or papers on each corner, and also displaying portions of the city.
Figure 2 shows the control panel and the algorithm área (top), and the path made by the robot throughout the city according to the execution of the algorithm (bottom). This path moves the robot from corner (1,1) to corner (3,2). As shown in figure 2, it is also possible to see the corners that contain flower and/or papers.
Currently, mostly every computer science students have mobile devices. Thus, they are familiarized with this type of technology, now transformed to a daily basis activity. Besides, students consider the use of desktop devices not as comfortable as mobile devices. With this in mind, a mobile version of R-Info is an interesting approach in order to encourage the use of applications targeted to improve learning.

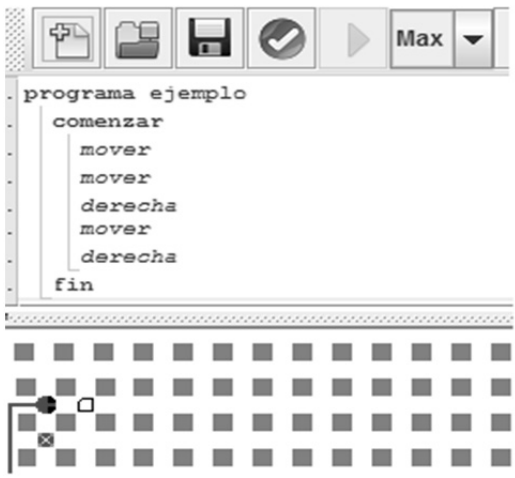


Fig. 2. *R-Info.*

It is then possible to bring R-Info to a new level through the development of a mobile version with a substancial improvement in the visuals of the application, the robot and the activities that perform along the city. This can be achieved by developing a 3D graphical interface that converts the resulting application into a more interesting, attractive, and prone to use visual tool [9][10].

3. 3D development tools

Although a complete analysis of the tools and libraries most widely used in order to achieve the proposed goal is not the main purpose of this paper, a general review of these tools is then featured.
Particularly, there are two main applications in this area that stand out from the rest: Unity[11] and Unreal Engine[12] mostly due to their popularity, functionality and versatility. Both considerably differ in certain aspects such as programming language, user license, technical support, etc. This is why in an early stage of the development process a comprehensive evaluation was made in order to establish the right tool to be used.
Several factors were analyzed, like: learning curve, cost, supported programming languages, execution performance, users community, hardware requirements, development and deployment platforms, etc.
As a result of the preliminary analysis, it was determined that both Unity and Unreal Engine meet the requirements based on the previously mentioned criteria. However, Unity is simpler and more intuitive as regards its use.
Additionally, the use of C# language in Unity vs C++ language in Unreal Engine was considered an advantage, given that the original R-Info project is written in Java language, which is similar to C# in several aspects.
Although with certain differences, both alternatives offer a free end-user license, thus there are no great advantages of one over another in this matter.
The amount of users - hence its community and support - with which Unity account, considerably surpasses the one of Unreal Engine. For instance, the site StackOverflow [13] contains 11147 questions with the *Unity* tag, while only 140 with the *Unreal Engine* tag at the time of writing this paper. Figure 3 shows the popularity according to Google Trends [14], which also presents a considerably advantage of Unity over Unreal Engine, and the trend seems to remain at least in the medium term.

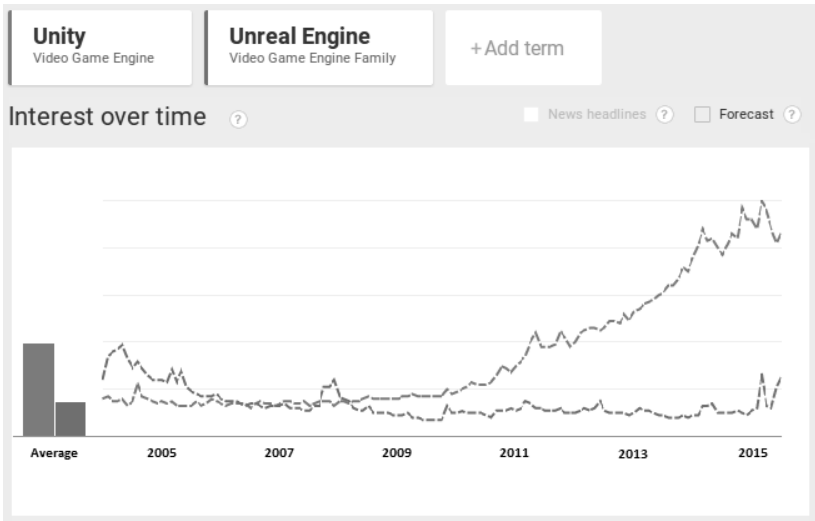


Fig. 3. *Google Trends: Unity vs. Unreal Engine.*

Finally, the execution performance of applications on mobile devices was considered better on Unity in the performed tests. In this tests, an exact scenario was tried to reproduce in both platforms, using the same number of poligons for the objects, identical textures, shadows and illumination configurations.
For all the above, Unity was the chosen tool for developing the prototype presented in this paper.

4. 3D mobile prototype

In order to have a new, flexible software tool, which can be used at any time and place, and also being visually atractive, a basic 3D mobile prototype was developed. This Unity prototype implements only a subset of the complete instruction set of R-Info.
The visual interface is completely three-dimensional, thus the city and the rest of the objects can be seen from different perspectives. With this advantage, the prototype allows selecting various cameras, each one of them showing a specific point of view. Besides, the views have a zoom control in order to display the scene near or far away.
Figure 4 shows an example of the prototype, in which the code editor can be seen, and on the background, the city where the simulation takes place when the *Execute* button is pressed. In this case, the first person is the chosen view.
Figure 5 shows the execution of the algorithm using a third person view with a medium zoom level. At the top of the image, information about the robot can be seen, such as position, orientation, number of flowers and papers in the bag and also at the corner where the robot is positioned. At the bottom, the current instruction being executed is shown. Zoom level and execution speed can be configured with the controls at both sides of the screen.

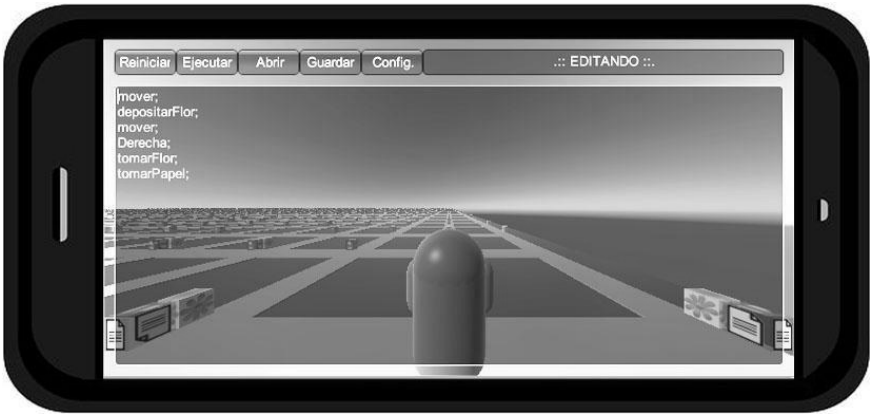


Fig. 4. *Code editor panel and first person view of the city.*

In addition to the first and third person views previously mentioned, there is a third option that allows to visualize the city from a top view, which is similar to the R-Info rendering mode. This last option can be seen in figure 6.
Papers and flowers located along the city corners are also rendered, as shown in the mentioned figures.

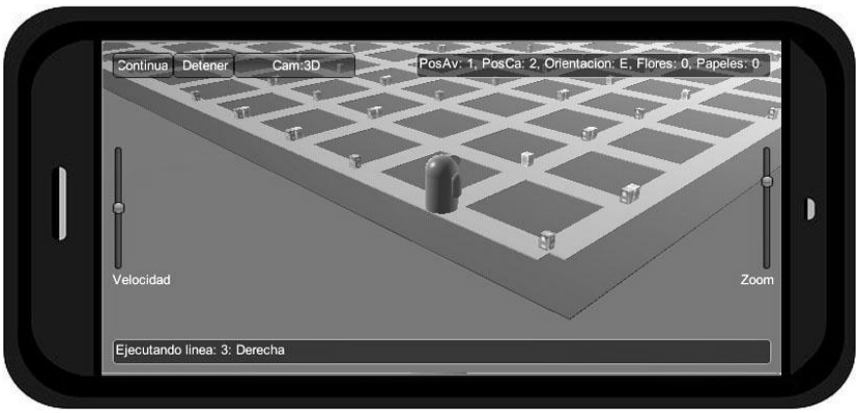


Fig. 5. *Execution panel. Third person view.*

The prototype allows additional adjustments such as animation execution speed control, loading source code from external files, and language selection. Besides, the program can also be executed step by step, or even restart it at any time.

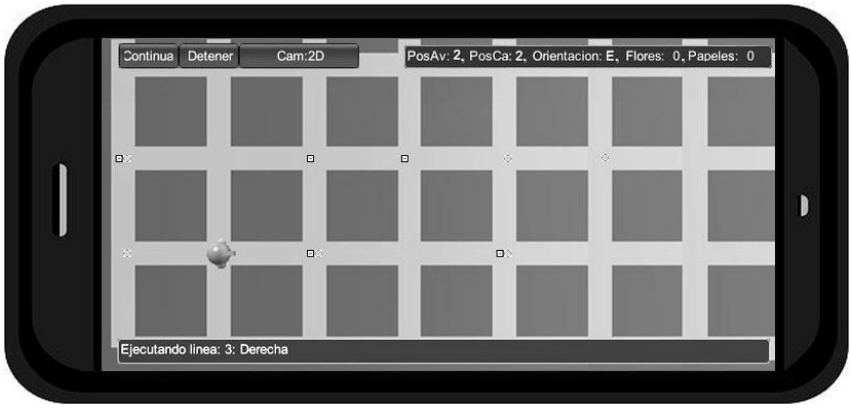


Fig. 6. *Top view.*

The three-dimensional environment allows the students to choose their point of view of preference, thus enhancing the user experience when executing the algorithms. Additionally, being a mobile prototype, students can use it at any moment and place, which is a considerable advantage over the desktop version.

5. Conclusions

A mobile 3D prototype of a tool that provides the support for learning basic algorithms at the initial levels of computer science careers has been developed.
Being a mobile prototype, it motivates the student and simplifies the task of learning basic concepts of algorithmics, given that execution of programs can be performed wherever the student is located. This way, one of the main characteristics of M-learning is accomplished, offering total flexibility regarding time and place of use.
The 3D environment is an additional appeal for students in order to feel interest in using the prototype. Besides, with several points of view available, the prototype allows a better visualization of the tasks performed by the robot along the city.
In summary, the 3D mobile prototype tries to enhance the experience for the student in its learning task, becoming a very useful tool for any course containing basic algorithmic topics.

6. Future work

The main future work is to provide full support of the functionalities provided by R-Info, adapting them to the 3D mobile solution.
In order to avoid a complete migration of the base source code of R-Info, a refactor of R-Info will be applied with the goal of providing two modes of

execution, an “application” or traditional mode, and a “library” or support mode for external uses.
This last mode will be the one used by the 3D mobile solution. This way, a source code migration is avoided and code compatibility is guaranteed in case of a potential evolution of R-Info.
Finally, visual enhancements are expected to be implemented for the 3D mobile solution, such as video quality configuration, robot character selection, visual and sound effects for the animations, among other things.

References

1. Cristina, F.; Dapoto, S.; Thomas, P.; Pesado, P. “A simplified multiplatform communication framework for mobile applications”. IEEE 9th International Conference on Computer Engineering & Systems (ICCES); El Cairo, Egipto. Dec. 2014. ISBN 978-1-4799-6593-9.
2. Kantel E., Tovar G., Serrano A.”Diseño de un Entorno Colaborativo Móvil para Apoyo al Aprendizaje a través de Dispositivos Móviles de Tercera Generación.” IEEE-RITA 5, no. 4. Nov. 2010. ISSN 1932-8540.
3. Yadegaridehkordi, E.; Iahad, N.A.; Mirabolghasemi, M. “Users’ Perceptions towards M-learning Adoption: An Initial Study”. IEEE International Conference on Research and Innovation in Information Systems (ICRIIS); Kuala Lumpur, Malaysia. Nov. 2011. ISBN: 978-1-61284-295-0.
4. De Giusti A.; Frati E.; Leibovich F.; Sanchez M.; De Giusti L.; Madoz M. “LMRE: Un entorno multiprocesador para la enseñanza de conceptos de concurrencia en un curso CS1”. XVII Congreso Argentino de Ciencias de la Computación (CACIC). Oct. 2011. ISBN: 978-950-34-0756-1
5. De Giusti A.; Frati E.; Leibovich F.; Sanchez M.; De Giusti L. "LIDI Multi Robot Environment: Support software for concurrency learning in CS1". International Conference on Collaboration Technologies and Systems (CTS); Denver, USA. May 2012. ISBN: 978-1-4673-1380-3
6. De Giusti L.; Leibovich F.; Sánchez M.; Chichizola F.; Naiouf M.; De Giusti A. "Desafíos y herramientas para la enseñanza temprana de Concurrencia y Paralelismo". XIX Congreso Argentino de Ciencias de la Computación (CACIC). Oct. 2013. ISBN: 978-987-23963-1-2.
7. De Giusti L.; Leibovich F.; Sánchez M.; Rodriguez Eguren S.; Chichizola F;, Naiouf M.; De Giusti A. "Herramienta interactiva para la enseñanza temprana de Concurrencia y Paralelismo: un caso de estudio”. XX Congreso Argentino de Ciencias de la Computación (CACIC). Oct. 2014. ISBN: 978-987-3806-05-6.
8. De Giusti A.; De Giusti L.; Leibovich F.; Sanchez M.; Rodriguez Eguren S. "Entorno interactivo multirrobot para el aprendizaje de conceptos de Concurrencia y Paralelismo". Congreso de Tecnología en Educación y Educación en Tecnología (TE&ET). 2014.

9. Paredes R.; Sánchez J.A.; Rojas L.; Strazzulla D.; Martínez-Teutle R. "Interacting with 3D Learning Objects". IEEE Latin American Web Congress; Merida, Mexico. Nov. 2009. ISBN: 978-0-7695-3856-3.
10. Hesse S.; Gumhold S. "Web based Interactive 3D Learning Objects for Learning Management Systems". International Conference on Education, Training, and Informatics (ICETI); Orlando, USA. Mar. 2011. ISBN: 978-161-8394-87-3.
11. Unity 3D Homepage: https://unity3d.com/
12. Unreal Engine Homepage: https://www.unrealengine.com/
13. Comparison between Unity and Unreal Engine in terms of number of questions for each platform: http://stackoverflow.com/questions/tagged/unity3d vs http://stackoverflow.com/questions/tagged/unreal-engine40
14. Interest comparison between Unity and Unreal Engine according to Google Trends: https://www.google.com/trends/explore#q=%2Fm%2F0dmyvh%2C%20%2Fm%2F025wnp&cmpt=q&tz=Etc%2FGMT%2B3

VI

Signal Processing and Real-Time Systems Workshop

Real Time Operating Systems evaluation over Microcontrollers

SANTIAGO MEDINA[1], MARTÍN PI PUIG[1], JUAN MANUEL PANIEGO[1], MATÍAS DELL'OSO[1], FERNANDO ROMERO[1] AND FERNANDO G. TINETTI[1,2]

[1]Instituto de Investigación en Informática LIDI (III-LIDI), Facultad de Informática, Universidad Nacional de La Plata, 50 y 120 2do piso, La Plata, Argentina.
[2]CIC – Comisión de Investigaciones Científicas de la Pcia. de Buenos Aires
{smedina, mpipuig, jmpaniego, mdelloso, fromero, fernando}@lidi.info.unlp.edu.ar

Abstract. *This work presents some measures of different operating systems that support real time characteristics, installed in a microcontroller system.*
These assessments characterize the response time, that is, a limit that determines the type of the application in which they could be applied, within the development of Real Time Operating System application. A major metric is latency time, which represents the time elapsed between the moment when an input becomes effective and the output is issued. Apart from evaluating this time, its variability is also analyzed, since in real time systems determinism is essential. Although this is not the only condition that an operating system must meet to be considered real time, it is a necessary requirement. Latency times and their variability determine the dimensions of temporary requirements provided by the system.

Keywords: *Real Time, Microcontrollers, Operating Systems, Embedded Systems, Latency.*

1. Introduction

Real Time Systems (RTS) [1] [2] [3] [4] are those which warrant a service within a limited response time interval. This restriction requires a careful design in hardware and software.
RTS are comprised by a set of electronic devices and a Real Time Operating System (RTOS). In a bottom-up approach, these systems can be implemented by:

- Electronic system, that is, only hardware specially designed to satisfy control requirements.
- Programmable electronic system. In general, hardware is designed for a generalized set of requirements and it is customized through software. Only one program is used with instructions to achieve the desired functionality over hardware with no operating system. They are also known as Bare-metal environment. They are systems in

which a virtual machine is installed directly in the hardware instead of in a host operating system [6].

- Electronic system with a Real Time Operating System (RTOS) and a program to achieve the desired functionality. The complexity of these systems is varied: from a microcontroller with an elemental RTOS (FreeRTOS, MQX) to complex computers running RTOS derived from UNIX (QNX) or Linux (RTLinux, RTAI, Linux RT-Preempt).

The advantages in complex systems with a RTOS are:

- Minimal development time: support provided by the RTOS.
- Flexibility: software can easily be changed.
- As a result of the above, these systems are easy to maintain.
- Meet strict time commitments: when an event occurs, it provides the programmer with tools to answer within a bounded time span. This implies that a poorly designed application may fail in serving events even when you use a RTOS.
- Time management: it provides management of timers and waiting time.
- Idle Task: when none of the tasks requires the processor, the system executes a task called idle, which allows to measure the occupancy level in the CPU, to put it in energy-saving mode or to run any task that could be useful for the system when it should not serve any of its events.
- Multitask: it simplifies multitask system programming.
- Scalability: while running multiple tasks at the same time, other tasks can be easily added providing that we have cautiously inserted them in the system execution scheme.
- Greater code reuse: with an optimal task design, with null or minimal dependence, it is easier to incorporate them into other applications.

Some disadvantages are:

- High power consumption: in real time systems applications, we found mobile systems as well as systems disconnected from mains that use batteries for power supply. Their autonomy depends on consumption.
- High failure rate: as it contains larger quantities of hardware and software components, a real time system requires continuous operation. Furthermore, simpler systems are preferred. High latency: any surplus software component involves an overhead resulting from its execution (for example, task scheduler).

In this work, latency measurement is analyzed, that is, the time elapsed between an event occurrence and the execution of the Interrupt Service Routine.
As the system becomes more complex, latency tends to increase by the large number of elements involved. Latency is analyzed over a TWR-K70 development board, part of Freescale Kinetis family. The different samples obtained without RTOS are compared against the same system with FreeRTOS and MQX RTOS.

A real time system may require bounded response times, in the order of seconds to milliseconds or microseconds. The correction of a real time system, as opposed to conventional systems, involves both the generated outputs and the time involved in producing them. Latency is a fundamental measurement of the hardware and RTOS platforms that will be used to develop a certain solution and it should be measured. Otherwise temporary restrictions of real time systems might not be met. Furthermore, the whole system might be at failure risk.

2. Objectives and Methodology

As these are embedded systems interacting with the physical environment, events occurring in this area must generate a system's response. These responses must be correct and issued within the time limits. The delay in producing them is called latency and should be measured. The objective of this work is to obtain a measure of that latency using both a RTOS as well as without using it (Bare-metal). The methodology is to generate an event, to take the time when it is produced, and measure the time again when the interrupt is generated. The difference between both determines latency time. The process that is executed as a response of the interrupt event is called Interrupt Service Routine (ISR).
As the interrupt events can still occur periodically ─a typical situation in RTOS─ it is desirable that the ISR executes in the shortest possible time. Therefore, in general, the interrupt response is implemented in two parts: one as small as possible inside the ISR and the rest through deferred routine. In this work, the methodology that implements this technique is analyzed.

2.1 Experimentation

To carry out tests, we used a Freescale development board and we worked with Kinetis Design Studio IDE (KDS IDE). Experiments were performed with RTOS and without it. The two RTOS system used were a free code one and an owner one.
The measurement method consisted in taking the start instant before the interruption of the system and the finishing time while detecting the event and serving the interruption. For that purpose, a pulse was generated by a determiner port of the system, which one was configured to trigger an interrupt on that pulse. This situation was repeated several times, to achieve better approximations of the finishing time.
Similarly, the times obtained were sent by the serial port, through the attached board TWR-SER, to a local host for further processing of data.

On the other hand, different tests were carried out over cases in which part of the task of elaborating the response are made on delayed mode, which makes the RTOS scheduler to act in a direct way. These tests were carried out over the RTOS FreeRTOS.

Three situations referred:

- Scenario A: main task generates an output signal. Then, the system captures it and calls the ISR. Later, it unlocks a mutex and then the main task stops the timer.
- Scenario B: Main task generates an output signal. Then, the system captures it and calls the ISR which wakes up a second task. This one stops the appropriate timer. Main and second tasks have the same priority.
- Scenario C: Main task generates an output signal. Then, the system captures it and calls the ISR which wakes up a second task. This one stops the appropriate timer. Main and second tasks have the same priority. Main task priority is lower than the second task priority.

2.2 Hardware

The TWR-K70 was used, a development kit belonging to the Freescale Kinetis family [12]. It is composed of an ARM-CortexM4 32bits microcontroller, with a 120MHz clock. It has several components: a 1GB SRAM, 2GB Flash memory, an accelerometer, 4 general purpose LEDs, 4 touch pads sensors, a potentiometer, a MicroSD slot and a USB port for standalone or debug.
While the board has a lot of hardware modules, it allows a large set of applications. Moreover, these modules can be added to form a tower structure. Freescale provides several modules to use in different applications. It is important to mention that this project uses a communication module (TWR-SER) that enables Ethernet and Serial interfaces.

2.3 Operating Systems Evaluated

Two RTOS were used: FreeRTOS and MQX. Both are called embedded operating systems, which are a combination of the operating system and the application.

2.3.1 FreeRTOS

FreeRTOS [7] [8] [9] [10] [11] is a real time operating system kernel, very popular among embedded systems. It has been ported to more than 35 microcontroller architectures and hardware platforms, even PCs. It is distributed under a standard GLP open source license, but it also allows

running property code over the open source core. In addition, property application can benefit from using FreeRTOS.
FreeRTOS is designed to be simple: the kernel consists of a set of C program files. Also, in some architectures, it includes assembler code lines, mainly in the scheduler routines.

FreeRTOS provides:

- Low memory usage
- Low overload
- Fast execution
- Multi-threading
- Synchronization
- Software timers
- Clock tick reduction for low power mode
- Task priority
- Four memory allocation scheme
- Open source
- Large community of users and developers.

The scheduler can change task depending on its priority and round-robin scheme. It can also be configured as preemptive or non-preemptive. Minimal task priority is equal to 0 and it is usually referred to the idle task priority (tskIDLE_PRIORITY) .

2.3.2 MQX

It is a real time operating system property from Freescale, but freely distributed.
It has two parts:

- PSP: Platform Support Package, MQX subsystem that is directly related to the core over which it will run. e.g.: ARM, ColdFire, PowerPC,etc.
- BSP: Board Support Package, MQX subsystem that contains all the code that supports the main board with a determined processor. This code has drivers for all modules.

This system provides:

- Real-time behavior
- Tasks
- Multithreading
- Device drivers
- Communication protocol stacks
- Multitasking with preventive programming and fast interrupt and response kernel
- Synchronization
- File Systems

- 6KB Configurable ROM, including kernel, interrupts, mutexes, queues and memory management
- MS-DOS File System (MFS).
- USB stacks
- Multiple platform support (Kinetis, ColdFire, Vybrid, i.MX, Power Architecture, etc.)

2.4 Measuring system

Two different tools were used to obtain latency times:

- System clock: ARM CortexM4 core clock called Systick. It is a 24bits down counter at 24MHz which has 4 configurable registers.
- Processor Expert's (PE) component: TimerUnit_LDD is the component used that provides an interface to configure FlexTimer board module. This timer is a 16bits counter that allows the programmer to configure frequency and count direction. The module was configured at 20.9MHz frequency.

The advantage of using the Systick core clock is that it allows to unify all the proposed scenarios (BareMetal, FreeRTOS, MQX).
On the other hand, PE method only applies over BareMetal and FreeRTOS being that Freescale MQX does not provide support for PE projects creation over TWR-K70 board.
Anyway, both methods return the same results.

3. Results

In table 1 the results obtained for each scenario can be seen.

Table 1. Latency Times

Bare-metal	6.9 µs
FreeRTOS	6.9 µs
MQX	9.0 µs

As it can be seen, the latency time corresponding to the development kit in a Bare-metal environment is really small. This value is the main reference for comparison against the different RTOS. Additionally, the time obtained sets a limit generated by the board for the development of applications requiring responses in shorter periods than those obtained.
In the case of FreeRTOS, no additional time due to its use is observed. This is because the RTOS scheduler is not executed, since in the interrupt routine of the event no call for any FreeRTOS API is made. In conclusion, the handler of the interrupt generated in response to pulse does not introduce any

overhead in the final latency time, thus being identical to the one obtained in the scenario without RTOS.
However, when evaluating the microcontroller system with the MQX, a minimum increase in the latency time is observed. When using the standard interrupts provided by the MQX, an additional time of about 2 μs [5] is generated. Similarly, as in the previous case, the code of the interrupt handler does not make any call to a RTOS API. . In case of requiring latency times shorter than those generated by the MQX, it is necessary to make kernel mode interrupts which are completely supported by the MQX. In that case, installing the required event interrupt with kernel-exclusive functions provided by the MQX will be enough. In any case, the application will not have the benefits provided by the classic MQX interrupts such as using flags and RTOS events as and API for other MQX processes.
As it was previously analyzed, the interrupt handler of both RTOS does not execute any call to an API. As a consequence, the latency time varies minimally or does not vary at all, with regard to the use of a bare metal scenario.
Afterwards, several tests were carried out in which there is an explicit call to certain functions of the RTOS to observe a possible overhead.
Therefore, in Table 2 the results obtained for the scenarios in which the TWR-K70 board incorporates the FreeRTOS are observed, with the difference that it uses a delayed section in the input event management.

Table 2: Results obtained using the delayed sections with FreeRTOS

Scenario	Min, T.	Max. T. .
A	22 μs	32 μs
B	55 μs	75 μs
C	40 μs	50 μs

As a consequence of the explicit call to a certain API of the RTOS, the response times are substantially increased.
In the first scenario, the event handler unlocks the mutex waiting for the main task, thus generating an overhead typical of the FreeRTOS API mutex call. As it can be seen, this delayed process adds an extra time of between 15 and 25 μs to the time obtained in Table 1.
Then in scenario B, the interrupt locks the mutex waiting for a secondary task, different from the main task generating the input pulse. Nevertheless, both tasks present the same priority. In this situation, the time obtained as well as its variability is increased to a greater extent. This is due to the fact that the RTOS generates different situations when deciding which of the two tasks is executed, after the interrupt handling is finished.
Finally, scenario C is identical to the previous one with the difference that the secondary task has a greater priority tan the main task. Therefore, the time measured decreases in comparison with the previous situation but it shows an increase in comparison to scenario A. . This is because when the execution of the interrupt routine is finished, FreeRTOS evaluates if the task unlocked by the mutex has greater priority tan the task previously interrupted. If this is the

case, the task scheduler is executed introducing an overhead generated by the use of a RTOS API. On the other hand, the variability is identical to the one in the first scenario since there are no tasks with the same priority.

4 . Conclusion

The main purpose of this work consisted in verifying, through the latency time, the performance of a RTS using two different RTOS and comparing the results obtained in those systems with the ones obtained executing applications directly over the hardware under the Bare-metal mode. In the different tests it was verified that the latency times over the Bare-metal and FreeRTOS are identical if the response process is made within the ISR, where the RTOS does not introduce any overhead. In case of the execution of a delayed routine with RTOS API calls, additional times with greater variability appear, due to the tasks carried out by the RTOS, mainly generated by its scheduler.
In another scenario, the RTOS MQX owner introduces a minimal overhead in the latency time of the system provided that classic interrupts are carried out.

References

1. N.C. Audsley , A. Burns , M. F. Richardson , A. J. Wellings: Hard Real-Time Scheduling: The Deadline-Monotonic Approach. Proc. IEEE Workshop on Real-Time Operating Systems and Software (1991).
2. L. Buhr. “An Introduction to Real Time Systems”. Prentice Hall (1999).
3. A. Burns, A. J. Wellings: Designing hard real-time systems. Proceedings of the 11th Ada-Europe international conference on Ada. ISBN:0-387-55585-4 (1992).
4. A. Burns & A. Wellings: Real-Time Systems and Programming Languages. Addison Wesley, ISBN 90-201-40365-x.
5. L. Prokop, Freescale Semiconductor Application Note. Motor Control Under the Freescale MQX Operating System.
6. A. Silberschatz, Peter Galvin, Greg GAGNE. Operating System Concepts.
7. R. Barry. Using the FreeRTOS Real Time Kernel.
8. C. Becker. RTOS en sistemas embebidos, available athttp://ieee.eie.fceia.unr.edu.ar/PDF_RTOS.pdf
9. A. Celery. Sistemas Operativos de Tiempo Real, available at http://www.sase.com.ar/2011/files/2010/11/SASE2011Introduccion_RTOS.pdf
10. FreeRTOS, available athttp://en.wikipedia.org/wiki/FreeRTOS
11. FreeRTOS, available athttp://www.freertos.org/
12. TWR-K70F120M: Kinetis Tower System Module, available athttp://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=TWR-K70F120M

Data acquisition system for measuring hydrogen absorption or desorption thermally activated

JORGE RUNCO AND MARCOS MEYER

IFLP - Depto.de Física – Facultad de Cs.Exactas – UNLP
CONICET
{ runco, meyer }@fisica.unlp.edu.ar

Abstract. *This work is part of the activities carried jointly by the Laboratory of Electronics and Experimental Physics group whose line of research aims to study nanostructured materials suitable for hydrogen storage.*
Besides the usual characterization techniques (X-ray diffraction , electron microscopy, differential scanning calorimetry and volumetric measurements) [1] [7], in samples containing Fe , it is possible to use Spectroscopy Mössbauer [2] sensitive atomic environment of Fe atoms. In this sense, it would be very important to observe the evolution of the Mössbauer signal (constant speed mode) simultaneously with the detection of hydrogen flow associated with thermally activated desorption or absorption.
For this purpose, a furnace has been constructed based on an annular heater [3] placed in a water cooled stainless steel chamber coupled to a flow meter. Experience involves heating the sample in a controlled manner by recording the time, temperature and hydrogen flow while simultaneously and independently a Mossbauer spectrum is measured.
This paper describes the development and implementation of a system that control and automates the acquisition of the aforementioned parameters
The experiment is controlled by a computer system (or Notebook PC), measuring, controlling, recording and automatically plotting the evolution of the variables of the experiment (temperature and gas flow) versus time. The software was developed in a programming language of high level (Matlab , Delphi) offering the user a typical graphic user interphase (GUI) visual languages .

Keywords: *Mössbauer Spectroscopy, absorption and desorption process, data acquisition*

1. Introduction

One of the most important challenges for the development and use of hydrogen as an energy vector is the ability to store it safely and efficiently.
Currently there are several ways to do this, each has advantages and disadvantages. Storage of gaseous hydrogen is the simplest, but it is very bulky and requires high pressures.
Liquid storage tanks needs very low temperatures, using large amounts of energy to maintain such cryogenic temperatures.

Because hydrogen is highly reactive there are a lot of elements capable of reacting with it to form hydrides. If the conditions of pressure and temperature are adequate, some hydrides could be used as a very efficient alternative to store hydrogen.
Among other advantages, the absorption of metals [1], forming a hydride phase over current systems (compression and liquefaction), it does not require work to compress or liquefy nor cryogenic temperatures.
An important issue in experimental research is the analysis of the properties absorption -desorption of hydrogen of new materials, as well as the study of the kinetics of absorption and desorption

2. Measuring system

Continuing development of equipment that allows for new experiences to study the absorption - desorption kinetics of hydrogen at different temperatures, in this work the built oven, its control and the system that automates the experience is showed.
The system allows to study the kinetics absorption -desorption of hydrogen, at different temperatures, maintaining constant pressure in the reaction chamber, in a wide temperature range (300K to 1000K) and pressure (1 mbar to 50 bar).

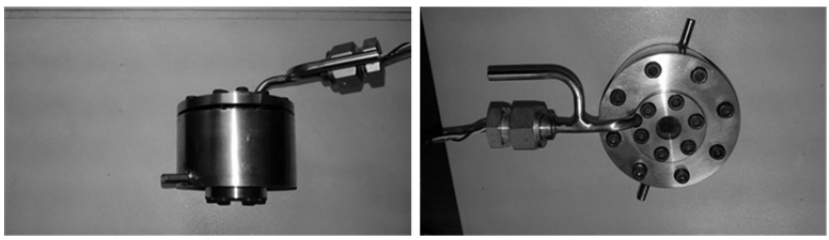


Figure1. *Used reactor*

The image shows the oven where the sample is tested. The same is cooled with water and its response was studied in order to model it and design the control of it.
A step power was applied and the response thereof (evolution of temperature) was measured and based on this most suitable strategy control was determined. As a result a model with different control techniques (proportional, proportional+ integral) according to the specifications of the experience.
Two types of experiences were made a) take the sample at a certain temperature and keep at that value or b) performing heating at a linear temperature rise (ramp). All this was taken into account when selecting the control algorithm.
Figure 2 shows the schematic diagram of experience where the interconnection of the various modules and components is depicted.

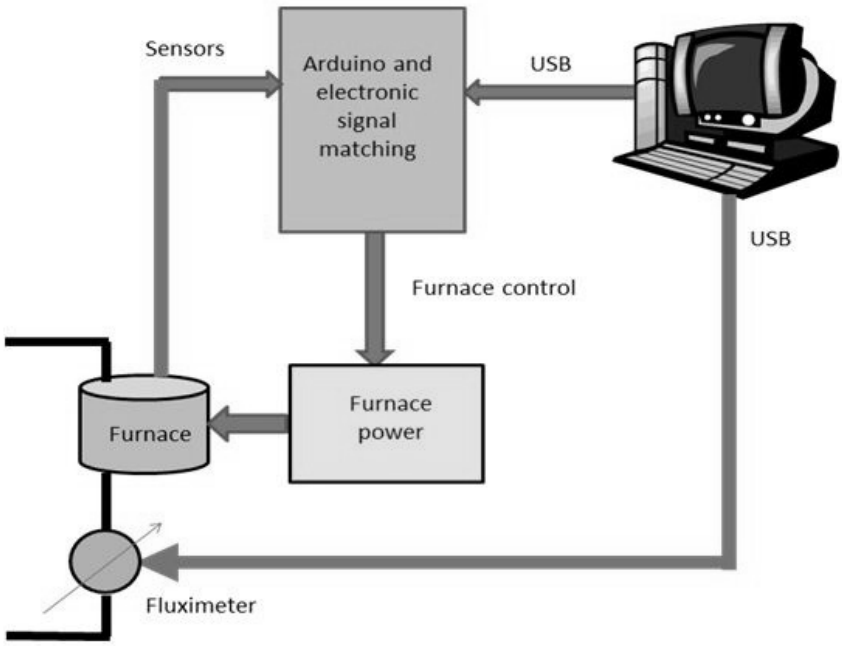


Figure 2: *schematic diagram of the measuring system*

The measuring instrument was developed on the computer system, notebook or PC type, usb ports through arduino controls a [4] microcontroller and flow meter.
The A/D converter that has onboard the arduino and electronic added to match the signals, are responsible for measuring temperature and pressure [5]. Arduino digital outputs are used to control the power stage feeding the furnace. While the flow meter is controlled directly from a USB port, as the instrument delivers information in digital form (MODBUS Protocol [6]) .
During experience the aforementioned parameters, temperature, pressure and gas flow, are acquired and stored. The system has the possibility of experience at constant temperature and linearly increasing temperature (temperature ramp).
This experiment was designed to measure simultaneously, with additional equipment, the Mössbauer effect and the variation of this signal with hydrogen desorption and absorption temperature dependent. Figure 3 shows the arrangement of additional equipment that was not shown in Figure 2.

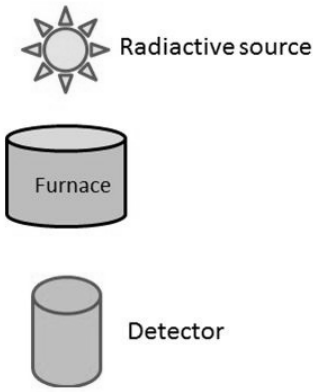


Figure 3: *Schematic diagram of Mössbauer spectrometer*

3. Conclusions

This equipment was developed at the request of a Research group of IFLP (Institute of Physics of La Plata) - Physics Department (Fac Sciences - . UNLP) that investigate the hydrogen sorption properties of several hydrides. It allows to follow the evolution of Mössbauer signal (additional equipment) depending on the hydrogen desorption and absorption thermally activated (equipment shown in this paper).
It was developed and implemented a "low cost" equipment made with commercial components that allows to study the hydrogen absorption and desorption in metals. The development was made "tailored" to meet the requirements of experiences regarding the different parameters measured. It was developed and implemented the signals adaptation module and the software that controls and automates the experience as well as the chamber containing the sample and the heating furnace. Also the power stage feeding the furnace was developed.

References

1. Optimización de un hidruro complejo para almacenamiento de hidrógeno. Tesis doctoral Junio 2009 – Cardozo, César Luis - Centro Atómico Bariloche.
2. Blatt, Frank J, Modern Physics, McGraw Hill, 1992. Efecto Mossbauer, Ch 15.
3. Watlow ULTRAMIC Advanced Ceramic Heaters. https://www.watlow.com
4. http://dfists.ua.es/~jpomares/arduino/page_03.htm
5. Data Acquisition and Control Handbook. Keithley
6. MODBUS Protocol Specification - http://www.modbus.org
7. Sievert-type measurement and acquisition system for the study of hydrogen storage in solids. Runco J. Meyer M. CACIC 2013 – Mar del Plata - Argentina

IV

Computer Security Workshop

Automated Analysis of Source Code Patches using Machine Learning Algorithms

ANTONIO CASTRO LECHTALER[1,2], JULIO CÉSAR LIPORACE[1], MARCELO CIPRIANO[1], EDITH GARCÍA[1], ARIEL MAIORANO[1], EDUARDO MALVACIO[1] AND NÉSTOR TAPIA[1]

[1] Grupo de Investigación en Criptografía y Seguridad Informática (GICSI), Instituto Universitario del Ejército (IUE), [2] Universidad Nacional de Chilecito (UNdeC), Argentina
{antonio.castrolechtaler, edithxgarcia, jcliporace,maiorano, cipriano1.618, edumalvacio, tapianestor87}@gmail.com

Abstract. *An updated version of a tool for automated analysis of source code patches and branch differences is presented. The upgrade involves the use of machine learning techniques on source code, comments, and messages. It aims to help analysts, code reviewers, or auditors perform repetitive tasks continuously. The environment designed encourages collaborative work. It systematizes certain tasks pertaining to reviewing or auditing processes. Currently, the scope of the automated test is limited. Current work aims to increase the volume of source code analyzed per time unit, letting users focus on alerts automatically generated. The tool is distributed as open source software. This work also aims to provide arguments in support of the use of this type of tool. A brief overview of security problems in open source software is presented. It is argued that these problems were or may have been discovered reviewing patches and branch differences, released before the vulnerability was disclosed.*

Keywords: *automated, source code review, source code analysis, patch analysis, machine learning, text mining, software quality.*

1. Introduction

This work presents a software tool for automated source code patches and branch differences analysis. It aims to systematize updates and source code reviews to alert on potential bugs, implying some degree of system security compromise, or vulnerabilities. The tool is distributed as an open source project and available at http://github.com/gicsi/aap. Currently, we have not found other open source tools available for the proposed functionality presented here. Other projects have been published but they largely deal with software engineering. For instance: trackable recoveries between source code and corrected bugs through patch analysis [5], or the use of patches for bug reporting. In the first case, a data processing tool was presented for Bugzilla systems, identifying the information with CVS tags.

In the case of open source analysis with cryptographic functionalities, Android case studies show that, for instance, only 17% of 269 vulnerabilities - reported in the period between January 2011 and May 2014 - were attributed to glitches in the cryptographic libraries. The remaining 83% were attributed to flaws in the application´s use of these libraries [14].
Furthermore, Android applications were analyzed in [8]: 10.327 out of 11.748 applications (88%) reported at least one implementation error. This last work presented an automatic detection tool, but was not distributed freely [15].
Other tools are available for general use to assist in source code management. However, they do not focus on security problems or patch reviews . They operate over the entire development cycle. Among the most common available open source alternatives, [12] and [23] are worth mentioning.
In the next section, references are surveyed on open and commercial software alternatives, dealing with security issues [4, 19, and 22].
It should be noted that control systems aiming at specific revisions have their own difficulties and limitations.
A study analyzing 210 samples showed that over 40% were not reported in the evaluation of C/C++ source code with five analytical tools; while only 7% of those samples were appropriately reported by the five tools [30]. Analogous results were found in the same study, using six tools based on Java.
The Open project OWASP [22] has identified the weaknesses of automated tests. It has also pointed out their strengths: ability to perform fast and repetitive analysis over a high volume of source code, the capacity to detect high probability errors, and the specificity provided in the reports.

2. Source Code Analysis

2.1 The Cost of Software Quality

Source code analysis and inspections are a continuous best-practice, aiming to improve software quality. It should be noted that although the quantity of bugs in a software project is not the only quality indicator, it constitutes a valuable measure for control and potential enhancements to development processes. Strictly, the goal of these reviews is to reduce the cost of software quality by identifying and correcting bugs at early stages of development [25].
It has been shown empirically [9] that software quality depends on control mechanisms used as an integral part of processes. Furthermore, the rate of personal inspections (measured in quantity of source code lines per time unit) has been shown to affect the effectiveness in detection (and correction) of bugs. Data shows, for instance, that review quality declines as this rate exceeds its recommended maximum of 200 code lines per hour. The authors cite previous articles in which a rate of 125 lines per hour is considered the optimum [3]. They also note: *"it is almost one of the laws of nature about inspections, i.e., the faster an inspection, the fewer defects removed"* [25].

2.2 Analysis Systematization

When dealing with software quality and the effectiveness of source code review, analysis systematization may yield efficient quality and development control processes when accounting for the time demanded from an analyst. Quality assurance tools are an essential resource in the improvement of software applications.
In a NIST publication, the minimum required functionality specifications in a source code analyzer for searching vulnerabilities are laid out [2]. The article suggests that it should identify security weaknesses, and report them tagging their type and location.
According to SEI/CERT, relying exclusively on policies, standards, and good practices for software development have proven inadequate [30]. It points out that these factors are not being applied consistently and that manual source code audits can be complemented and enhanced with systematized and automated tests.

2.2.1 Current Available Tests

Different organizations maintain lists from which a great range of tools for testing source code are referenced. Among them: the NIST webpage, Source Code Security Analyzers [19], and the Secure Coding Tools from CERT [4].
Finally, the OWASP project is an obliged reference, maintaining a thorough list of source code analysis tools [22].

3. Post Patch Vulnerabilities

3.1 Errors in the Categorization of Vulnerabilities

In his recent thesis [34] and revising previous work [35, 36, 33], Jason Wright - best known for his work in the cryptographic framework of the OpenBSD operating system [13] – emphasizes the importance of patch reviews. In the third section of his last publication, he introduces the concept of *hidden impact bugs*. The concept is also found in [35] and [33]. In the latter the term used is *hidden impact vulnerabilities*, referring to previous work [1]. However, they all deal with the same concept; i.e., vulnerabilities which were reported as bugs and whose real impact – their vulnerability category – was recognized after issuing the initial report which included a patch to solve what was believed to be a software flaw or defect with no security implications.
Impact delay is defined as the time elapsed since the report of the flaw – as a patch – until the time when the bug was tagged with a CVE. In 2012, the author, contributing with others at a second stage of analysis, published a review of hidden impact bugs in the source code (patches) of the Linux

Kernel and MySQL database [33]. Later, in 2013, they published an extended analysis, focusing on MySQL exclusively [35].
The tests consisted of detailed reviews of a subset of the reported bugs. The source code associated to the bugs is analyzed to determine the percentage of unclassified vulnerabilities. The authors extrapolate the results to yield an estimate of the total percentage on reported flaws. From the available data, the results are summed up as follows:

- In the second stage analysis of the Linux kernel - from January 2009 to April 2011 - results show that from a total of 185 hidden impact bugs, 73 (39%) had an impact delay of less than 2 weeks; 55 bugs (30%) had a delay of less than 4 weeks; and 29 (16%) had a delay of less than 8 weeks.
- In the MySQL study, total hidden impact bugs sum up to 29. From them, 19 (65%) had an impact delay of less than 2 weeks, also 19 had a delay of less than 4 weeks, and 16 bugs (55%) had a delay of less than 8 weeks.

3.2 Fixed Issues in NTP-devel

A reference implementation of the protocol used to synchronize clocks over the network, known as *Network Time Protcol* (NTP), is distributed as part of the Linux operating systems, among others. In December 2014, a series of security issues – some of them categorized as critical – were identified. The information went public along with patches to fix the problems. However, the bug database used by the development team showed that some of these bugs had already been fixed in the *devel* version, years before.
Additional information is available in their security webpage [16]. A brief outline of the particular details is given below to illustrate the way in which a comparison between two versions can point out critical fixes not implemented in the stable version.
Although Bugzilla´s historical data show that the information was made accessible to the public on 12/12/2014, messages on bug 2665 [17] - regarding a weak cryptographic default key - point out that the vulnerability had already been "corrected" four years before, in the NTP-devel (4.2.7), particularly "(4.2.7p11) 2010/01/28."
An additional example involves bug 2666 [18] dealing with a random number generator – cryptographically insecure with a weak initialization seed. Although it was later fixed once again before the release of version 4.2.8, messages point out that the development version n 4.2.7 p30 (of November 1st, 2011) had already corrected the problem, referring an enhancement of the seed management.

3.3 Multiple Patches for Shellshok Vulnerabilities

Recently discovered in the UNIX bash shell system, the Shellshok vulnerabilities represent an example of critical security problems which are not corrected completely or adequately in the first published patches. The

importance and criticality of the vulnerability was highlighted in The Register [28], in which they recommended immediate installation of the patch to avoid wide open access to Linux and OS X operating systems. The news, along with a public notice from Red Hat [26], and the updated packages are dated September 24th, 2014. Nonetheless, later, related problems were identified and for which Red Hat did not offer updates until September 26th, 2014 [27].

3.4 Simultaneous Discovery of the Heartbleed Vulnerability

During 2014, a critical vulnerability, known as Heartbleed, was disclosed. It consisted of an exploitable bug in the OpenSSL library. According to Bruce Schneier, half a million sites were infected, turning the problem catastrophic [29]. A Red Hat Security representative [6] considered that the coincidence of two simultaneous findings of a single problem would increase the risk of maintaining the vulnerability secret. Consequently, it can be inferred that this rationale could have encouraged OpenSSL to release updated packets hastily. The rush in the releases may be responsible for the lack of coordination in issuing the patches. Thus, this illustrates how, under certain circumstances, several organizations can publish patches in an non-coordinated manner.

3.5 Vulnerability in OpenBSD after a Misclassified Bug

In 2007, an IPv6 vulnerability was found in the OpenBSD kernel [21]. A specially crafted ICMP packet could generate a remote buffer overflow, when handled by an internal function which miscalculated the buffer´s length. According to the author, the vulnerability was identified while analyzing – and trying to reproduce – a problem already fixed by a patch rated as reliability fix instead of vulnerability or security fix [20].

4. Patch Analysis Tool

4.1 Automated Analysis of Patches, or AAP, version 0.2b

4.1.1 Idea and Rationale

Beside obvious differences in source code and binary analysis, the project is based on the criteria used by automated malware analysis tools, such as the Cuckoo Sandbox.
In [11], two problems are identified when performing this type of analysis manually: the increasing volume of malware in the wild and the time required for a thorough inspection.

The goal in this work also involves an efficacy assessment in the use of IA techniques, especially text mining and machine learning, applied to source code patches and "texts" in general comments.
The rationale relies on published work regarding the discovery of vulnerabilities in the patches distribution of open source code projects. The references in the third section of this article constitute examples of this problem.

4.1.2 Machine Learning

Although beyond the scope of this article, a brief overview is presented here on the techniques and applications of machine learning implemented in the tool presented in this article:

Natural Language Processing (NLP). Natural language is understood as the language used by human beings to communicate: English, Spanish, ... Generally, NLP involves the manipulation of a natural language through the use of computer systems. For instance, text analysis enables the detection of positive or negative sentiments in tweets – in favor or against.
Supervised Classification. Classifying comprises assigning tags of the appropriate category to particular entries or inputs. The set of tags is defined beforehand. Some classification examples are SPAM identification or news categorization (sports, finance, and others). A classifier is supervised when it is trained based on data (corpora) containing the correct tag for each entry or input.

AAP Implementation. For the generation of tagged data, bug reports, with attached patches, were downloaded from Red Hat and OpenBSD public bug database system and errata pages, respectively. The labels "vulnerability" and "bug" were defined according to the criticality established in the original reports. With this set of tagged entries or inputs, classifiers - of the naïve Bayes and decision tree types - were trained. These classifiers can then be used from a plugin to estimate the appropriate tag for each automatically analyzed patch.

4.1.3 Functionality Currently Implemented

Through its web interface, the tool maintains and organize information of software projects, reference to its repositories, branches, updates, and analysts users assigned to them.
In addition, the web interface handles the general configuration for the automatic reviews of patches and different versions. These rules are implemented in the form of plugins, programmed in Python and editable on the interface itself.
The verifications of these rules, automatically executed over patches or over branch differences of a software project update, activates alerts whenever necessary. Currently, these rules include, among others, searching for configured patterns in source code and comments, detecting source code

implementing cryptographic functionality, and checking for changes in specific files or done by particular authors or committers.
Since version 0.2b, the tool contemplates the application of machine learning techniques through trained classifiers with patches and comments from public bug information repositories. Classification analysis applies a per "vulnerability" or "bug" criteria.
Configured rule inspections are performed over patches recovered from GIT repositories [10].
The periodic batch process updates the repositories and automatically performs the analysis configured for each of its branches. The outcome includes the generation of system alerts -and optionally sending notification emails with analysis results- to the assigned analysts.

4.1.4 Technologies

The tool consists of a web-based application developed in the Python language [24] and using Web Django framework [7]. It makes use of GIT versatility for updating and handling source codes [10]. Other external tools may be used optionally.
Currently, plugins to classify patches and comments are implemented using an open source library or toolkit, also developed in Python: NLTK (Natural Language Toolkit). It includes software, data and documentation, and can be freely downloaded from http://nltk.org/.
NLTK Trainer constitutes another resource – a set of Python scripts for NLTK model training.
Particular plugins for static analysis were implemented for different languages (C/C++, Python, Perl, PHP) based on the use of open source tools like FlawFinder, RATS, Splint and CppCheck.

4.1.5 Open Source Software

The tool's source code is publicly published using a free service for open source projects management: http://github.com/gicsi/aap. It is licensed under the terms of the GPL, or GNU General Public License, from the Free Software Foundation.

5 Further Work

Functionality enhancements are under design. The expectation is to capture the interest of other developers and encourage them to get involved with the project. Among other improvements, the following features are under consideration: to achieve a higher degree of sophistication in rule and algorithm design, to extend the use of machine learning in the tool, to incorporate information retrieved from developers´ discussion forums, and to integrate the tool with other databases dealing with vulnerability reports and security advisories.

References

1. Arnold, J., Abbott, T., Daher, W., Price, G., Elhage, N., Thomas, G.A. Kaseorg Security Impact Ratings Considered Harmful. Proceedings 12th Conference on Hot Topics in Operating Systems. USENIX. May 2009. [Online – accessed 29/12/2014: http://www.inl.gov/technicalpublications/Documents/5588153.pdf].
2. Black, P., Kass, M., Koo, M., Fong, M. Source Code Security Analysis Tool Functional Specification Version 1.1. NIST Special Publication 500-268 v1.1. February 2011. [Online – accessed 29/12/2014: http://samate.nist.gov/docs/source_code_security_analysis_spec_SP500-268_v1.1.pdf].
3. Buck, F. Indicators of Quality Inspections. IBM Technical Report TR21.802, Systems Comm. December 1981.
4. CERT Division - Secure Coding Secure Coding Tools. CERT, Software Engineering Institute (SEI), Carnegie Mellon University. [Online - accessed 29/12/2014: http://www.cert.org/secure-coding/tools/index.cfm].
5. Corley, C., Etzkorn, L., Kraft, N., Lukins, S. Recovering Traceability Links between Source Code and Fixed Bugs via Patch Analysis. University of Alabama. 2008. [online - accessed 29/12/2014: http://www.cs.wm.edu/semeru/tefse2011/papers/p31-corley.pdf].
6. Cox, M. Heartbleed. Mark Cox, J. Google+. [Online - accessed 29/12/2014: https://plus.google.com/+MarkJCox/posts/TmCbp3BhJma].
7. Django Framework. Django overview. Django Software Foundation. [Online - accessed 29/12/2014: https://www.djangoproject.com/start/overview/].
8. Egele, M., Brumley, D., Fratantonio, Y., Kruegel, C. An empirical study of cryptographic misuse in android applications. CCS '13 Proceedings of the 2013 ACM SIGSAC conference on computer & communications security. Pages 73-84. 2013. U.S.A. [Online – accessed 29/12/2014: http://www.cs.ucsb.edu/~chris/research/doc/ccs13_cryptolint.pdf].
9. Kemerer, C., Paulk, M. The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. 35, NO. XX April 2009. [Online - accessed el 29/12/2014: http://www.pitt.edu/~ckemerer/PSP_Data.pdf].
10. Git SCM. About Git. Git - Software Freedom Conservancy. [Online: http://git-scm.com/about - accessed 2912/2014].
11. Guarnieri, C. One Flew Over the Cuckoo's Nest. Hack in the Box 2012. May 2012. Netherlands. [Online – accessed 29/12/2014: http://sebug.net/paper/Meeting-Documents/hitbsecconf2012ams/D1T1%20-%20Claudio%20Guarnieri%20-%20One%20Flew%20Over%20the%20Cuckoos%20Nest.pdf].
12. Gerrit Website Gerrit Code Review. Google Inc. [online: https://code.google.com/p/gerrit/ - accessed 29/12/2014].

13. Keromytis, A., Wright, J., de, T. Raadt. The Design of the Open BSD Cryptographic Framework. International Conference on Human System Interactions (HSI). June 2012. Australia. [Online - accessed 29/12/2014: http://www.thought.net/papers/ocf.pdf].
14. Lazar, D., Chen, H., Wang, X., Zeldovich, N. Why does cryptographic software fail? a case study and open problems. Proceedings of 5th Asia-Pacific Workshop on Systems Article No. 7. 2014. U.S.A. [online - accessed 29/12/2014: http://pdos.csail.mit.edu/papers/cryptobugs:apsys14.pdf].
15. Mujic, A. Reimplementation of CryptoLint tool. Blog for and by my students. December 2013. [Online - accessed 29/12/2014: http://sgros-students.blogspot.com.ar/2013/12/reimplementation-of-cryptolint-tool.html].
16. Network Time Protocol project. NTP Security Notice. NTP support website. Network Time Foundation. [Online - accessed 29/12/2014: http://support.ntp.org/bin/view/Main/WebHome].
17. Network Time Protocol project. Bug 2665 - Weak default key. NTP Bugzilla. Network Time Foundation. [Online - accessed 29/12/2014: http://bugs.ntp.org/show_bug.cgi?id=2665].
18. Network Time Protocol project. Bug 2666 - non-cryptographic random number generator with weak seed. NTP Bugzilla. Network Time Foundation. [Online - accessed 29/12/2014: http://bugs.ntp.org/show_bug.cgi?id=2666].
19. NIST Source Code Security Analyzers. SAMATE - NIST. [Online - accessed 29/12/2014: http://samate.nist.gov/index.php/Source_Code_Security_Analyzers.html].
20. Ortega, A. OpenBSD Remote Exploit. Core Security. Julio de 2007. [Online – accessed 29/12/2014: https://www.blackhat.com/presentations/bh-usa-07/Ortega/Whitepaper/bh-usa-07-ortega-WP.pdf].
21. Ortega, A., Richarte, G. OpenBSD Remote Exploit. Core Security. April 2007. [Online - accessed 29/12/2014: https://www.blackhat.com/presentations/bh-usa-07/Ortega/Whitepaper/bh-usa-07-ortega-WP.pdf].
22. OWASP Wiki. Source Code Analysis Tools. The Open Web Application Security Project (OWASP). Ult. mod. 29/10/2014. [Online: https://www.owasp.org/index.php/Source_Code_Analysis_Tools - accessed 29/12/2014].
23. Phabricator Website. Phabricator, an open source, software engineering platform. Phacility, Inc. [online: http://phabricator.org/ - accessed 29/12/2014].
24. Python Website. About Python. Python Software Foundation. [Online: https://www.python.org/about/ - accessed 29/12/2014].
25. Radice, R. High Quality Low Cost Software Inspections. Paradoxicon Publishing. 2002.
26. Red Hat. Security. CVE Databases. CVE-2014-6271. Red Hat Customer portal. September 24th, 2014. [Online: https://access.redhat.com/security/cve/CVE-2014-6271 - accessed 29/12/2014].

27. Red Hat. Security. CVE Databases. CVE-2014-7169. Red Hat Customer portal. September 24th, 2014. [Online: https://access.redhat.com/security/cve/CVE-2014-7169 - accessed 29/12/2014].
28. The Register. Leyden, J. Patch Bash NOW: 'Shellshock' bug blasts OSX, Linux systems wide open. The Register online tech publication. September 24th, 2014. [Online: http://www.theregister.co.uk/2014/09/24/bash_shell_vuln/ - accessed 29/12/2014].
29. Schneier, B. Heartbleed. Schneier on Security, Blog. April 2014. [Online: https://www.schneier.com/blog/archives/2014/04/heartbleed.html – accessed 29/12/2014].
30. Seacord, R., Dormann, W., McCurley, J., Miller, P., Stoddard, R., D.Svoboda, Welch, J. Source Code Analysis Laboratory (SCALe). CERT, Software Engineering Institute (SEI), Carnegie Mellon University. April 2012. [Online: https://resources.sei.cmu.edu/asset_files/TechnicalNote/2012_004_001_15440.pdf - accessed 29/12/2014].
31. W. Weimer. Patches as Better Bug Reports. University of Virginia. 2006. [Online: https://www.cs.virginia.edu/~weimer/p/p181-weimer.pdf - accessed 29/12/2014].
32. Wheeler, D. Flawfinder. David Wheeler, A.'s Personal Home Page-Flawfinder Home Page. [Online: http://www.dwheeler.com/flawfinder/ - accessed 29/12/2014].
33. Wijayasekara, D., Manic, M., Wright, J., McQueen, M. Mining Bug Databases for Unidentified Software Vulnerabilities. Proceedings International Conference on Human System Interactions (HSI). June 2012, Perth, Australia. [Online: http://www.inl.gov/technicalpublications/Documents/5588153.pdf – accessed 29/12/2014].
34. Wright, J. Software Vulnerabilities: Lifespans, Metrics, and Case Study. Master of Science Thesis. University of Idaho. May 2014. [Online: http://www.thought.net/papers/thesis.pdf – accessed 29/12/2014].
35. Wright, J., Larsen, J., McQueen, M. Estimating Software Vulnerabilities: A Case Study Based on the Misclassification of Bugs in MySQL Server. Proceedings International Conference of Availability, Reliability, and Security (ARES). September 2013. pp. 72-81. Regensburg, Germany. [Online – accessed 29/12/2014: http://www.inl.gov/technicalpublications/Documents/5842499.pdf].
36. Wright, J., McQueen, M., Wellman, L. Analyses of Two End-User Software Vulnerability Exposure Metrics (Extended Version). Information Security Technical Report, 17(4), Elsevier. April 2013. pp. 44-55. [Online: http://www.thought.net/papers/INL-JOU-12-27465-preprint.pdf – accessed 29/12/2014].

IV

Innovation in Computer Science Education Workshop

Information Systems: Professional Competencies 2020

MARISA CECILIA TUMINO, JUAN MANUEL BOURNISSEN AND KAREN BARRIOS

Universidad Adventista del Plata, Libertador San Martín, Entre Ríos, Argentina
{marisatumino, juanbournissen}@doc.uap.edu.ar

Abstract. *A search for current professional competencies in Information Systems-related degree courses of renowned institutions was proposed for this work, gathering those that best represented professional performance in this discipline. The selected competencies were analyzed in light of the degree course objectives with the aim of subjecting them to being assessed by employers, teachers, students, and graduates. Afterwards, those professional competencies that met the requirements identified in the study were defined and the subsequent conclusions were drawn.*

Key words: *professional competencies, competency based training, professional profile, information systems.*

1. Introduction

At present, most curricula are not flexible enough to delve deeply into certain issues or include others that enrich students' vocation by providing them with study options, research and professional practice opportunities. According to Duk and Loren [1],

> *"El currículum oficial de un país proyecta la visión de futuro de la sociedad y sus aspiraciones para con las nuevas generaciones; asimismo concreta las finalidades de la educación a través de la selección de las competencias que permitan a las personas desarrollarse y participar en las distintas esferas de la vida."*

As stated in Tuning America Latina project [2], reaching a graduation profile in the Informatics field poses particular difficulties, since the recent origin of Informatics and the rapid growth of associated technologies lead to a constant evolution of the supporting knowledge and techniques. On the other hand, the use of Informatics is constantly increasing in almost every area of human tasks, generating several and various scenarios for the professional practice.
As pointed out by the Systems Engineering Department of University of Antioquia [3], the world is increasingly demanding advanced means to create, spread, adjust, and use knowledge as the foundation for generating richness. As a consequence, societies must be educated to enable a diversified education of their members so as to timely and efficiently satisfy the demands brought about by the growing volume of knowledge.

The aforementioned document poses the importance of the functioning of the information systems program on the basis of a research-type pedagogical model that is centered on the three main conceptual ideas of being, knowing, and doing, and oriented to problem solving and competencies achievement design. This pedagogical model conceives education as an essential subsystem of the social system, since it comprehensively educates individuals to interact in that system. Comprehensive education entails three dimensions: education as a person, intellectual development, and capacity for intervening in social realities in a competent, responsible, and supportive way.

Education in these dimensions is supported by a problem solving-oriented curriculum that is focused on competency development to intervene in diverse realities. Some competencies are developed by professional areas that use specialized knowledge to solve specific problems. Others are generated by basic areas that support competencies of professional areas and contribute to the development of skills and strategies to understand real situations and formulate and solve problems.

In this sense, Garcia Almiñana, Sánchez Carracedo, and Gavaldà Mestre [4] classify the professional competencies of a degree into three categories: (a) technical competencies, which are related to the discipline itself, (b) transversal competencies, which are necessary for this academic level but are not related to the degree field knowledge, and (c) deontological competencies, which are related to social personal education. For practical purposes, only technical competencies have been considered in this work, though taking into account those aspects of transversal and ontological competencies that add natural value to technical competencies.

Fuente, De Andrés, Nieto, Suárez, Pérez, Cernuda, Luengo, Riesco, Martínez, Lanvín and Fondón [5] point out that one of the objectives proposed in the framework of the European Higher Education Area (EHEA) consists in actually bringing that training provided by the university environment closer to the professional needs demanded by society. For that reason, they propose to thoroughly consider professional competencies when designing new degree courses. Nowadays, however, many degree courses offered by European universities do not meet this requirement, keeping a significant distance between graduate profiles and the professional profiles demanded by society.

Aiming at an advance in university-company integration, the 'white books' of the National Agency for Quality Assessment and Accreditation of Spain (ANECA, for its initials in Spanish) have been developed, where an expert board defined the set of competencies that constitute professional profiles. As a result of the critical study on the proposals made in the white book (ANECA) [6], Fuente *et al.* developed the 'blue book' as an alternative arising from a redefinition of disciplinary competencies and profiles.

Openlibra [7] prepared a manual for training on Informatics and information competencies, which was also a valuable source for drawing up the proposal and included TIC competencies on Information provided by Romeu Fontanillas [8].

These experiences have been undertaken considering both the academic and professional fields so as to facilitate this integration. Since these books have resulted from the emerging conditions in the EHEA, it is worth considering

the contextualization of that proposal, adjusting it to the current Argentine situation with a projection for 2020.

2. Problem Definition

Since it is necessary to specify the competencies of Informatics professionals by adjusting them to the historical and geographical context of Argentina, it is intended to define those competencies that emerge from the current professional profiles so that they can be subjected to being assessed by the collectives that are really incumbent upon this definition.

3. Scientific, Academic, Institutional, and Social Justification

In recent years, this competency approach to professional education in Latin America was consolidated as a result of this issue rise in European countries since the Bologna Process, which is evidenced by the creation of the European Higher Education Area (EHEA). This trend was not noticeable firmly enough in Informatics degrees in Latin American countries, keeping the observed distance between the graduate profile meant by the university and the professional profile imposed by social demand. Currently, Argentina is showing an express will to define competencies in order to introduce the concept in the Informatics field.

In Universidad Adventista del Plata (UAP), the path is being cleared in this respect; and it is expected to start designing curricula and syllabi using a competency-based approach to professional education.

The introduction of the concept of competency, which demands evidence of the holistic development of students from knowledge assimilation –reflected in skills and attitudes–, is consistent with the educational institution intention of training professionals that are capable of facing the challenges imposed by science and technology in the organizational community in the 21st century. For that reason, it is important to reach a consensus over the definition of competencies consistent with these trends.

4. Methodology

The followed steps are summarized below:

1. Data collection. At the beginning of the study, competencies were proposed on the basis of the survey and a collection of trends and syllabi from other national and Latin American universities, analyzing informatics professional profiles demanded by organizations and the informatics graduate profiles provided by those profession-related degree courses. At the same time, the study considered the reserved activities of informatics professionals that are included in Resolution Nr. 786/09 issued by the Ministry of Education of Argentina on the Accreditation

Standards for Bachelor's degrees in Computer Science, Computer Systems/Information Systems/System Analysis, Informatics, and Engineering degrees in Computer Science and Informatics/Information Systems [9]. A data analysis was carried out and there was a preliminary proposal of competencies.
2. Assessment of the whole proposal by the collectives. Participants of the study called 'Competencies of the Informatics Professional for 2020", including teachers, employers, and graduates of informatics degree courses from different schools of the country analyzed if those competencies included in the list were sufficient; they also added those that were considered to be wrongly omitted, removed those they considered as repeated or unnecessary, and modified those that needed a change.
3. Adjustment and definition of the informatics professional competencies according to suggestions made by participants. After receiving suggestions, analyzing them in light of professional profiles, and introducing the corresponding changes, the new list of competencies of the informatics professional for 2020 was ready. Later, the online survey was composed, using a response scale that represents a continuous valuation from null relevance (0) to optimal relevance (7) for each competency –organized by areas–, each of them containing a set of indicators or competencies whose quantity varies according to each area.
4. Once the tool was created, the various collectives (teachers, employers, graduates, and students) were asked to assess the relevance level of the proposed competencies for training information systems professionals. A valuation on the relevance degree was obtained for each defined competency.

5. Data Processing and Analysis

The result analysis was adjusted to Maldonado Rojas (2007: 235-237) [10] methodology. The response scale was codified according to the scheme shown in Table 1. For that purpose, the frequencies observed in each value of the scale were used. The criterion to analyze the obtained results was supported by interpretative categories designed from the quartile scale. Only percentages obtained in the High level were considered, after carrying out the codification according to the response scale shown in Table 1. Final categories are shown in Table 2.

Table 1: Response Scale and its codification

Scale	Grade
0	Low
1	
2	
3	Medium
4	
5	High
6	
7	

Competencies included in the final proposal were those that were positioned in interpretative categories equal to or higher than the 'Satisfactory' level. The study describes the percentage of subjects that agree with each option of the scale.
The expected result of this initiative is to lead any participating institution to distinguish the proposed competencies for all systems-related degree courses, including bachelor's degrees in Systems, Computer Science, and Informatics, and Engineering degrees in Systems and Computer Science.

Table 2: Interpretative categories used as analysis criteria

% of subjects that consider competency has a High level of relevance	Interpretative category
Lower than or equal to 25%	Deficient
Higher than 25% and lower than or equal to 50%	Moderately satisfactory
Higher than 50% and lower than or equal to 75%	Satisfactory
Higher than 75%	Optimal

6. Results

It is worth highlighting that the characteristics of the experts that participated in the study fulfilled those conditions described in Tables 3 and 4, as regards the role they played and their experience in professional practice. As shown, 59% of the involved participants had more than 10-year experience, which allows classifying the obtained results as reliable information.
As regards played roles, it is noted that the total percentage exceeds 100% since this field allowed for multiple responses and some survey respondents played more than one role. This is evidenced by the 54% corresponding to informatics graduates, which is an expected value for the accomplishment of functions that are inherent to that profession.

Table 3: Role distribution among participants

Role	Number	%
Employer	7	17%
Informatics graduate	22	54%
Informatics teacher	14	34%
Informatics student	7	17%

Table 4: Distribution for participants' experience time

Experience time	Number	%
Less than 6 years	9	22%
6 to 10 years	7	17%
11 to 15 years	9	22%
16 to 20 years	4	10%
21 to 25 years	3	7%
more than 25 years	8	20%

Once informatics professional competencies were gathered, which were collected from the survey performed in national and Latin American universities whose academic offer includes related degrees, they were subjected to being evaluated by the study participants. According to their expertise, these participants evaluated each competency, its modification or omission, and suggested changes or added competencies they considered should be included.

The resulting list was forwarded to the various collectives, who assessed the relevance of each competency included in the list. The received data were processed with the aim of obtaining the total percentages of individuals that scored the relevance of each competency within the 5-7 range of the scale, which represents a High degree of relevance (Table 1). Those percentages are shown in a decreasing order in Table 5.

Table 5: Summative percentages of the High degree in the relevance scale

	Competencies	%
1	Analysis of the characteristics, functionalities, and structure of Operating Systems for implementing applications.	92.68%
2	System and data assurance according to the needs of usage and security conditions settled to prevent failures and external attacks.	92.68%
3	Integration of communication equipment in network infrastructures, setting the configuration to assure connectivity.	90.24%
4	Planning, design, organization, development, maintenance, and direction of system, service, and application projects in the field of Informatics, leading their startup and continuous improvement, and assessing their economic and social impact.	90.24%
5	Analysis, design, implementation, maintenance, and efficient and secure use of data bases adjusted to the functionalities of information applications.	90.24%
6	Knowledge and application of the necessary tools for software development and data base management.	90.24%
7	Peer-to-peer service and training for leadership in identifying the community needs and team work with the aim of finding altruist and supportive solutions with social responsibility.	90.24%
8	Pro-activity that encourages full development while being grounded in universal values, human rights, cultural and democratic values, environmental responsibility, and ethical commitment.	90.24%
9	User management according to specifications settled to guarantee the availability of system resources.	87.80%
10	Counseling and design of data security strategies in data transmission networks.	87.80%
11	Evaluation and selection of hardware and software platforms for developing and executing information systems, services, and applications.	87.80%
12	Design and efficient use of data types and structures that best suit the resolution of a problem.	87.80%
13	Autonomy for investigating new languages or solutions from different reliable sources so as to provide responses for the new professional demands.	87.80%

14	Management of network services (web, electronic messenger, and file transfer, among others) and software installation and configuration under quality conditions to meet organizations' needs.	85.37%
15	Appropriate management of information resources (storage, space, energy, money, time), according to the commitments made with customers.	85.37%
16	Development of applications for mobile devices, according to current technologies and paradigms.	85.37%
17	Design, development, evaluation, and assurance of accessibility, ergonomy, usability, and security of information systems, services, and applications as well as of the managed data, in compliance with ethical principles and current legislations and regulations.	85.37%
18	Analysis, selection, and application of methodologies and life cycles that suit the needs of the application to be built.	85.37%
19	Development, maintenance, and evaluation of reliable and efficient software systems and services that meet user requirements, in compliance with quality norms and applying the good practices of Software Engineering.	85.37%
20	Responsible, collaborative, and respectful integration in software development teams.	85.37%
21	More appropriate selection of programming paradigms and languages in the analysis, design, construction, and maintenance of robust, secure, and efficient applications.	85.37%
22	Clear expression of ideas, grounded in reflection and interpretation of relevant social, scientific, or ethical issues, adjusting them to real problems and to the solutions demanded by the diversity of opinions and situations.	85.37%
23	Use of the English language to access relevant information for constant research and training.	85.37%
24	Analysis of information systems structure, organization, functioning, and interconnection, programming foundations, and application for the effective resolution of engineering problems.	82.93%
25	Selection, design, deployment, integration, and management of communication networks and infrastructures in an organization.	82.93%
26	Supervision of physical security according to specifications and the security plan so as to avoid interruptions to the provision of system services.	82.93%
27	Design and development of centralized or distributed information systems or architectures, integrating hardware, software, and networks according to the appropriate knowledge.	82.93%
28	Application of security, confidentiality, integrity, and privacy standards that are inherent to information systems within the framework of current legislation, regulations of the corresponding professional association, and corporate politics of the employing company.	82.93%
29	Selection, design, construction, integration, management, and maintenance of information systems that satisfy the organization's needs according to cost, quality, and technological innovation criteria throughout the software life cycle.	82.93%
30	Use of programming environment tools to create and develop applications.	82.93%
31	Analysis and assessment of the social and environmental impact of informatics solution development, considering specifications, in compliance with current legislation and taking ethical and professional responsibility for the activity.	82.93%

32	Leadership in human relationships, in negotiation and solution of conflicts, and in effective working habits, resorting to communication and motivation skills in all areas of work task performance.	82.93%
33	Design and implementation of applications based on the analysis of characteristics, functionalities, and structure of Distributed Systems, Computer Networks, and the Internet.	80.49%
34	Design and evaluation of human-computer interfaces (HCI) that guarantee accessibility and usability of information systems, services, and applications.	80.49%
35	Design of informatics solutions using software engineering methods that integrate ethical, social, legal, and economic aspects.	80.49%
36	Appropriate knowledge of security, privacy, and intellectual property aspects inherent to the information systems of an organization.	80.49%
37	Use of arguments and evidence as means to retain or reject conjectures and advance towards conclusions.	80.49%
38	Analysis of the organization and structure of computers and their basic components.	78.05%
39	Knowledge of the functioning of real and virtual machines so as to allow improving the efficiency of computational algorithms of information systems.	78.05%
40	Communication-oriented software development, according to security and reliability regulations for data transmission networks.	78.05%
41	Appropriate management of customer requirements and constraints, reaching a compromise of conflicting objectives by searching acceptable solutions in relation to the restrictions derived from costs, time, technologies, the existence of already developed systems, and organizations themselves (technical and economic feasibility).	78.05%
42	Innovating, autonomous, proactive, and creative endeavoring in support of problem solving and capitalization of new opportunities.	78.05%
43	Active participation in economic, social, and cultural life with a critical and responsible attitude.	78.05%
44	Evaluation of hardware-software and software-software compatibility criteria so as to assure the correct functioning of components.	75.61%
45	Development of new methods and technologies as versatile contributions that adjust to the new situations that arise in the professional field.	75.61%
46	Management of Information and Communication Technologies in business processes that contribute with effective solutions to information needs of companies, providing them with competitive advantages.	75.61%
47	Application of basic algorithmic proceedings of information technologies to solutions design, analyzing their aptitude and complexity.	75.61%
48	Development of algorithmic solutions that consider the characteristics of the equipment in which they are to be implemented.	75.61%
49	Writing of documents containing technical conditions for installing information devices in compliance with current standards and regulations.	73.17%
50	Application of the fundamental principles and techniques of intelligent systems.	73.17%
51	Planning and development of professional updating and training courses on information systems for colleagues, system users, and for any audience.	73.17%

52	Acquisition, formalization, and representation of human knowledge related to perception and action aspects in a computable way for problem solving by means of an information system.	73.17%
53	Appropriate use of organization and user-oriented development strategies for the evaluation and management of information technology-based applications and systems that ensure accessibility, ergonomy, and usability of systems.	70.73%
54	Planning, organization, writing, development, and signature of projects in the software engineering field with the aim of taking advantage of information systems, services, and applications, managing available tools and techniques, their approaches and paradigms.	70.73%
55	Mediation among technical and management communities of an organization, applying the principles and good practices of organizational communities.	70.73%
56	Analysis and evaluation of computer architectures, including parallel and distributed platforms in software projects that meet particular requirements.	68.29%
57	Constant evaluation and optimization of the system performance, configuring hardware devices according to functioning requirements.	68.29%
58	Responsible diagnosis and efficient solution of technical problems in data transmission networks.	68.29%
59	Application of principles related to organization, economy, human resources, risk management, legislation, and regulations in projects of the information systems field.	68.29%
60	Knowledge on and application of the fundamental principles and basic techniques of parallel, concurrent, distributed, and real time programming.	68.29%
61	Command of mathematical algorithms and proceedings and knowledge about how, when, and why using them in a proper way.	68.29%
62	Adequate modeling of specific problems, under restrictions imposed by the software-hardware relation.	68.29%
63	Objective analysis of the artificial intelligence approach in the search for solutions based on control systems and robotics.	68.29%
64	Scientific research on subjects related to information systems.	68.29%
65	Design, implementation, and documentation of encapsulated program components	68.29%
66	Comprehension and evaluation of components that constitute high performance computation.	65.85%
67	Modeling, design, and evaluation of data transmission networks behavior, considering the logical functioning of their active components.	65.85%
68	Management of own resources in the field according to work load and maintenance plan.	63.41%
69	Efficient planning and execution of auditing tasks on information systems.	63.41%
70	Objective analysis of the artificial intelligence approach in the search for solutions based on classical problems and intelligent games, expert systems, neuronal networks, and fuzzy logic.	63.41%
71	Coordination and collaboration in research aimed at strengthening scientific and technological development, communicating methods and results in an effective way.	63.41%

72	Efficient comprehension, application, and evaluation of the means for physical storage.	60.98%
73	Optimization of computation algorithms of information systems, grounded in the management of resources used by finite-state machines.	60.98%
74	Valuations, expert appraisals, and reports of informatics tasks or works according to particular needs.	60.98%
75	Proper comprehension and application of the principles related to risk management in the drawing and execution of action programs.	60.98%
76	Application of simulation tools and methods in modeling various types of information systems.	60.98%
77	Knowledge of the basic aspects of the data processing machine theory that allow properly operating software components of information systems.	53.66%
78	Knowledge of the fundamentals of hardware description languages and the functionality of hardware components that provide higher security when counseling potential customers.	53.66%
79	Interpretation and modeling of phenomena by means of mathematical models.	53.66%
80	Comprehension of basic concepts of electricity, magnetism, electromagnetism, electromagnetic waves, electric, magnetic, and electromagnetic fields; concept of electronics (analogic and digital), circuits and properties, semiconductors, electronic devices, and their appropriate application to solve engineering-related problems.	51.22%
81	Construction and representation of mathematical formulations, applying computational methods adjusted to specific requirements.	39.02%
82	Creation and refinement of simple programs, using an assembling code.	29.27%

The indicator preceding each competency shows the disciplinary area it belongs to, as detailed below:
Architecture, Operating Systems, and Networks (ARSORE, for its initials in Spanish): 80, 24, 38, 66, 1, 33, 25, 3, 14, 26, 9, 68, 53, 56, 72, 77, 39, 78, 73, 44, 15, 57, 81, 67, 40, 10, 58, and 82.
Software Engineering, Databases, and Information Systems (ISBDSI, for its initials in Spanish): 4, 16, 17, 11, 27, 45, 74, 59, 54, 49, 28, 2, 29, 5, 6, 50, 18, 34, 19, 41, 46, 55, 69, 75, 76, and 20.
Algorithms and Languages (AyL, for its initials in Spanish): 47, 12, 21, 60, 48, 30, 61, and 13.
Professional and Social Aspects (APyS, for its initials in Spanish): 22, 31, 35, 42, 36, and 51.
Control Systems / Artificial Intelligence: 52, 62, 70, and 63.
Various – Generic: 7, 32, 8, 23, 64, 65, 71, 79, 43, and 37.

7. Discussion and Conclusion

The relevance of the present project lies on the valuable contribution of the highly experienced information systems professionals that devoted time and effort throughout this two-stage research process with the aim of building a curricular proposal grounded in competency training. As shown in Table 5, forty eight (48) competencies came out to fall into the Optimal interpretative category, since the sum of frequencies corresponding to high scores, between 5 and 7, represents a percentage higher than 75%. Moreover, thirty two (32) competencies fell into the Satisfactory interpretative category, since they became associated to a percentage in the 50%-75% range. Only two (2) competencies were found to be related to a Moderately satisfactory level, since their corresponding percentages ranged from 25% to 49%, and thus only these two competencies are considered to be removed from the original list.

According to the research-oriented approach adopted by this study, it is estimated that the list created throughout the proposed process, in general terms, resulted to be appropriate for the informatics professional profile in 2020, since 97,6% of the proposed competencies –eighty (80)– were positioned at Satisfactory and Optimal interpretative relevance levels. The final list composed of 80 professional competencies is intended to be a contribution for designing the curricula of the five Information terminal degrees in the Argentine education system as well as the curricula of related professions in other countries.

References

1. C. Duk H. y C. Loren G., "Flexibilización del Currículum para Atender la Diversidad", Revista Latinoamericana de Educación Inclusiva, 4, 1 (2010): 187.
2. Tuning América Latina. Documento elaborado en la tercera reunión general del 2 al 4 de mayo de 2012 en Santiago de Chile; disponible en; http://www.tuningal.org/es/publicaciones/doc_download/71-documento-de-trabajo-de-la-reunion
3. Departamento de Ingeniería de Sistemas Universidad de Antioquia. "Transformación curricular programa Ingeniería de Sistemas". Medellín: Publicaciones Universidad de Antioquia (2006); disponible en; http://es.scribd.com/doc /54299048/Ingeniera-de-Sistemas
4. J. Garcia Almiñana, F. Sánchez Carracedo, y R. Gavaldà Mestre, "Recomendaciones para el diseño de una titulación de Grado en Informática", IEEE-RITA, 2, nº 2, (2007): 99
5. A. J. Fuente, J. De Andrés, C. Nieto, M. Suárez, J. R. Pérez, A. Cernuda, M. C. Luengo, M. Riesco, B. Martínez, D. F. Lanvín y M. D. Fondón, "El Libro Azul de La Ingeniería en Informática: una alternativa al Libro Blanco"; disponible en; http://di002.edv.uniovi.es/~cernuda/pubs/jide2005-c.pdf

6. Agencia Nacional de Evaluación de la Calidad y Acreditación. “Libro Blanco, Título de grado en Ingeniería Informática”, 2004; disponible en; http://www.fic.udc.es/files/266 /libroblanco_informatica_0305.pdf; Internet (consultada el 07 de marzo de 2014).
7. Manual para la formación en competencias informáticas e informacionales (Madrid: Cardiff University, 2011), 160, disponible en; http://ci2.es/sites/default /files/documentacion/ manual_ci2_completo.pdf
8. Teresa Romeu Fontanillas, “Competencias TIC en Información y Documentación” (Catalunya, España: UOC, 2009).
9. Res ME Nº 786/09 – “Estándares de acreditación de los títulos de Licenciado en Ciencias de la Computación, Licenciado en Sistemas/Sistemas de Información/Análisis de Sistemas, Licenciado en Informática, Ingeniero en Computación e Ingeniero en Sistemas de Información/Informática”; disponible en: http://www.coneau.edu.ar/archivos /Res786_09.pdf
10. Mónica Maldonado Rojas, “Valoración de la formación recibida usando un perfil de referencia basado en competencias profesionales”, Educación Médica, 10, nº 4, (2007): 233-243.

Production of Learning Objects for University Teaching. Call for Educators of the School of Computer Science of the UNLP

ALEJANDRA ZANGARA[1], CECILIA SANZ[1], LUCRECIA MORALEJO[1], FERNANDA BARRANQUERO[2] AND MARCELO NAIOUF [1]

[1] Institute of Research in Computer Science LIDI (III LIDI). [2] Pedagogical Direction, School of Computer Science, National University of La Plata
La Plata, Argentina

Summary: *The production of Learning Objects (LOs) for University-level teaching is a topic that is both interesting and controversial. There is disagreement in relation to the definition itself of what a Learning Object is, as well as the methodologies that should b applied to its design and development and the issue of reutilization, including the value of the educator as reutilization and resignification agent for the contents and format of the original LO. The truth is that, in a field with so many conflicting issues, both the institutional decision of going for this kind of projects as well as educator training become key factors. The starting point should then be a clean and thorough definition of the concept of Learning Object, both from a didactic as well as a technological point of view. It is essential also that the competencies that educators need in order to be able to design, develop, use and reuse LOs are defined. The definition of these issues necessarily starts with the institutional perspective. Institutions should lead these innovation projects, providing educators the necessary conditions and encouragement to undertake this training path. This idea resulted in the "Call for the Production of Learning Objects to Innovate in University Teaching" that was organized by the School of Computer Science of the UNLP in August 2014, and which received the joint support of the Direction of Distance Education and Technologies Applied to Education, the Direction of Pedagogy, and the Master in Information Technology Applied to Education of the aforementioned School. This article describes the steps of the Call, related training activities, and the results obtained after the first year.*

Keywords: *Learning Objects, Learning Object Production, Innovation, Learning Object Reutilization.*

1. Introduction

It has been quite a few years now since teaching and learning virtual environments (TLVEs) became part of numerous educational scenarios,

especially at universities. Educators and students already know how to use them, and have been involved in working with digital educational materials, which became a core pillar when considering proposals mediated by Information and Communication Technologies (ICTs). Thus, educators also got involved in the production of their own digital educational materials or resources (presentations, websites, videos, images), which opened the discussion in relation to their reutilization in different educational and technological contexts.

Learning Objects (LOs) propose a road for designing reusable digital educational materials.

Even though some authors track their origin to Computer Science and link them to object-oriented programming based on their reutilization characteristics, others state that the term was coined by Wayne Hodgins who, in 1992, proposed building digital educational materials from separate, reusable modules that could be coupled together to create increasingly complex modules, the same way Lego blocks are used to create buildings. While he was watching his son play with some Legos, Hodgins realized that the building blocks he was using could be used as a metaphor to explain how teaching materials are built; he was thinking about small teaching blocks that would facilitate learning and which could be connected to one another to create structures or products that are more complex or have a greater scope [1].

By the end of the 90s, James L'Allier defined a LO as "the smallest independent teaching experience that has an objective, learning activities and an assessment" [2].

Polsani (2003) defines a LO as "an autonomous and independent unit of learning content that is suitable for reutilization in multiple teaching contexts" [3].

There are numerous definitions on the concept of Learning Object, and this has resulted in the generation of different educational materials with varying levels of granularity. In fact, learning objects are linked to a design for reutilization, interoperability, access (from different repositories), and durability, to mention some of the features upon which authors most frequently agree. Also, learning objects are undoubtedly within the educational materials category and, therefore, their purpose is educational. Granularity has become one of the most controversial issues around LOs. Consequently, there are materials with the most varied levels of granularity, and some consider them to be LOs while some others do not. In this paper, a specific definition is taken as a starting point, interpreted within the context of the Master in Information Technology Applied to Education of the School of Computer Science, and the main features upon which key authors in the subject agree are considered, while at the same time establishing the difference between LOs and other educational resources and materials. Thus, a LO is "a type of digital educational material that is characterized, from a pedagogical point of view, for its orientation towards a specific learning objective, and having: a series of contents that will allow presenting the topic related to the objective, activities that will allow students to put the contents presented into practice or consider related problems, and a self-evaluation that will allow students determine if they have understood the contents linked to the objective. From a technological standpoint, it is characterized for containing a set

of standardized metadata used fir search and retrieval operations, as well as for being integrated, using a standard-compliant packing model, which allows the interaction with different technology environments" [4].
Thus, the concept of LO is separated from those of other educational materials, in this case, by defining its components: a specific objective that identifies its educational goal, a number of contents, activities, and a self-evaluation step that are interrelated so that the objective proposed by the LO can be achieved. Its granularity is also established by stating that it is oriented towards a specific objective. Specifications include having metadata that follow a standard for storage, searching and retrieval, and packing should follow a model that allows interoperability.
A learning object is then an autonomous unit that allows achieving a certain educational objective and that, when combined with other learning objects, can result in more complex structures, such as lessons, modules or courses.
Based on this definition, the creation of a LO design methodology, called CROA [5], has also been pursued; it is intended as a guidance for the educator (or the development team) in the design and creation of LOs that comply with the definition at hand. This methodology has been developed after a detailed review of several methodologies oriented to the creation of LOs [6]. CROA contributes by providing guidelines from the point of view of both the instructional as well as the technological design of a LO. It combines both outlooks and is aimed at guiding educators, who may not have specific programming knowledge.
In this paper, we present the experience developed in the context of the School of Computer Science of the National University of La Plata, oriented towards the proposal to encourage the production of LOs in the institution and the creation of a specific repository. In Section 2, the project and its various stages are described, Section 3 discusses some preliminary results, and Section 4 details some conclusions and future work.

2 Project for the Production of Learning Objects in the School of Computer Science

In agreement with the project carried out by the School to open the road for the interaction between the educational and the technological worlds — in this case through the design of learning objects, — a project was designed and carried out to focus on this issue. This project was launched simultaneously in the relevant management and training areas. It had the participation of the Pedagogical Direction, the Direction of Distance Education and Information Technologies Applied to Education and the Master in Information Technology Applied to Education. The project included defining a number of stages, which are presented in the figure below. These are: Presentation of the topic (raising awareness); opening the call; training and follow-up of educator needs; submission of deliverable documents, and publication of LOs in an institutional repository. In the following sub-sections, the planning corresponding to each of these stags will be detailed.

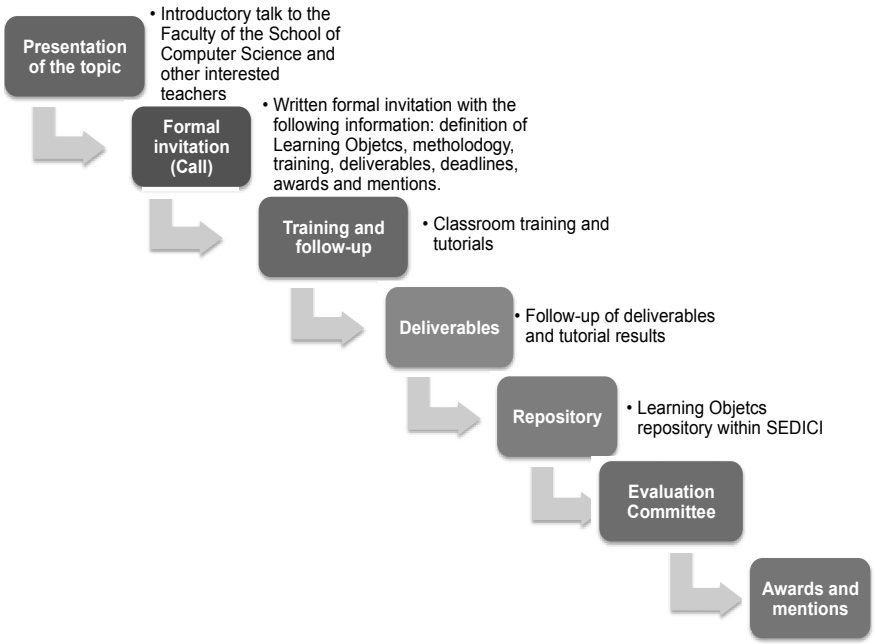


Figure 1: *Stages of the project Production of Learning Objects in the School of Computer Science*

2.1. Presentation of the Topic – Raising Awareness

This presentation was oriented to explaining the foundations of the project. It focused on providing the definition of LOs, possible methodologies for designing them, and the resignification of the role of the educator to facilitate LO use and reutilization. It was organized by the offices of both Directions and the Master mentioned above. The title of the initial presentation, carried out in September, 2014, was "*Concept, Design and Production of Learning Objects*." Its objectives were as follows:

- Presenting the concept of Learning Object (LO) as a controversial one, and then going on to discuss the definition presented in the first section of this article. After that, a discussion followed on the usefulness of LOs for graduate and post-graduate teaching in several content areas.
- Showing different LO repositories.
- Presenting a Call for LOs organized by the School, through the Direction of Distance Education, for educators of both graduate and postgraduate courses to design LOs for their corresponding courses.

It should be noted that it is a common practice of this School to present issues that are central to the management of technology in education as a joint collaboration of both Directions working in these issues. The advantages of this methodology is that it allows discussing several aspects from various specialist perspectives, as well as encouraging the dialogue/discussion among educators,

which often goes on after the end of the face-to-face presentation. Therefore, this was the road chosen to start working on this topic.

2.2. Call

The information in the Call was one of the central aspects for this project. It was called: "*Production of Learning Objects to Innovate in University Teaching*." It gave a clear idea of its objective (the production of educational materials with the format of LO), the idea of innovation in teaching, and it defined the target audience (as a first stage, only educators from the School).

From its very beginning, and as a continuation of the research and conceptualization line that was discussed in the first section, the Call specified the concept of LO. Additionally, educators were informed of the following intention: "*The School wants to take the initiative and compile all materials that respond to a design of learning object in order to innovate in our teaching practices and generate a repository of materials at an institutional level.*" This already anticipated one of the main goals of the project — the creation of a specific institutional LO repository.

The Call also invited educators to take part in a training program based on their specific needs. This will be detailed more thoroughly in the next sub-section.

As regards the methodology for designing LOs, which is as relevant as controversial, educators were given complete freedom to select their preferred methodology. It should be noted that some educators in the School were already working on this area before the Call, using their own methodologies. For uniformity, educators were asked that any LO they submitted followed a specific definition, and a number of guiding questions/deliverables were provided to assist educators with the basic planning of their LOs, regardless of the methodology used.

The Call also mentioned the creation of a board of reviewers for the evaluation of the LO to be created, in order to grant some type of "award" to those that were evaluated with the highest scores by the reviewers and thus offering an additional incentive to educators. It was decided that this board of reviewers would be formed by renowned experts in the area from both national and foreign universities. The following indications were also included in the Call:

- *LOs must refer to contents taught in the courses given at the School, and the author of the LO must be an educator.*
- *It is not required that any submitted LO has already been used to teach the selected topic or that its use has been assessed.*

One of the key aspects of the Call, aimed at providing guidance for educators from the start, was a work schedule that included two "deliverable" documents that were required to organize the production cycle. This will be specifically discussed in the following sections.

2.3. Training

Training is considered to be one of the key aspects of the project, since it represented a significant challenge due to the complexity of the topic in itself and the heterogeneity of the participating educators, all of whom had developed an interest for the topic in previous stages. Thus, separate training modules were used, which were based on the needs of the different educators. The following modules were proposed:

MODULE 1: Concept and methodology for the production of LOs. The CROA methodology designed in the context of a research project of the School of Computer Science was specifically selected. In the context of this methodology, the step required for the analysis, design, implementation, and evaluation of LOs were discussed. Some metadata standards were introduced, and LOM was recommended. Various packing models were also analyzed.

MODULE 2: Instructional design of LOs. Organized by the Pedagogical Direction, specific training sessions aimed at the instructional design of LOs in line with the CROA methodology. Specific documentation was provided.

MODULE 3: Tools for developing LOs and editing metadata. The use of different tools for creating LOs was discussed (Ardora, ExeLearning, both of these combined, Reload), tools for editing metadata were introduced (ExeLearning, Reload), and recommendations were made, aligned with CROA, to fill metadata in.

Additionally, an online course was added to the teaching and learning virtual environments WebUNLP, which is used by several courses of the School, with all materials and to provide answers to queries from participants (using the system of virtual tutoring). This was also used to provide information about the progress of the Call.

The Direction of Distance Education and Information Technologies Applied to Education offered support for the LO design and production stages through online tutoring sessions (using WebUNLP and, if needed, an email address) and face-to-face meetings at the School if required.

2.4. Deliverables and Follow-Up

Once the training period was over, the next stage of this proposal was that of "Deliverables and following up educators work". In this stage, following up the progress of the educators is essential; it is done through:

- tutoring (face-to-face or using the virtual environment) and
- organizing intermediate deliverable documents (which we called "deliverables") that the educators must submit in order to move forward in the design and production of their LOs.

The first deliverable was the submission of general information about the intended LO and a brief of its design. The second deliverable was the implementation of the LO and updating the first deliverable accordingly, based on the progress made.
For the first deliverable, the information required was as follows:

General data:
- Title for the Learning Object
- Author(s) (this includes the educator(s) creating the LO and the educator supporting its production)
- Course for which the LO is created

Planning the LO
- What is it that the target audience needs to learn?
- What is the target educational level?
- What is the topic that will be developed by the LO?
- What prior knowledge should the student have to use the LO?
- What other knowledge related to the LO can be acquired after the LO has been used?
- What is the specific objective proposed for the LO?
- LO navigation ma (it can be a diagram)
- Activities that will be proposed to students in relation to the objective proposed
- Self-evaluation that will be proposed to students in relation to the objective proposed

The staff of both Directions involved in the organization of the project and the Master in ITAE[1] created an Evaluation Committee to evaluate the deliverables submitted by participating educators or groups of educators. Additionally, face-to-face tutoring sessions were offered to explain/expand on the comments.

2.4. Repository

In parallel with the design and implementation of the Call, and in relation to the production of LOs, joint work was carried out with the SEDICI (institutional repository of the National University of La Plata) for the creation of a specific repository for learning objects for the School of Computer Science.
The LO publication stage is aimed at starting the process for making these educational materials available for use by educators and students. This is the ultimate goal of the project, which will then become a permanent call to continuously expand the repository.

[1] *Master in Information Technologies Applied to Education: http://postgrado.info.unlp.edu.ar/Carreras/Magisters/Tecnologia_Informatica_Aplicada_en_Educacion/Tecnologia_Informatica_Aplicada_en_Educacion.html*

3. Preliminary Results

Each stage of the project was reviewed and analyzed once implemented. At the time of writing this, the deliverable evaluation stage is ongoing, after which LOs will be posted to the repository. Below, some preliminary results are presented in relation to the project stages that have already been implemented:

Eighteen educators from graduate and post-graduate courses attended the **initial presentation**. As regards the background of the educators, it was surprising that only 56% came from computer science subjects that were related to software development. There were several "non-computer" educators, which was interesting and in a way indicative that the project was actually reaching beyond its niche, since educators from various areas were showing interest. Figure 2 shows an analysis, based on subject matter taught, of the educators that attended the initial presentation at the School of Computer Science (September 2014).

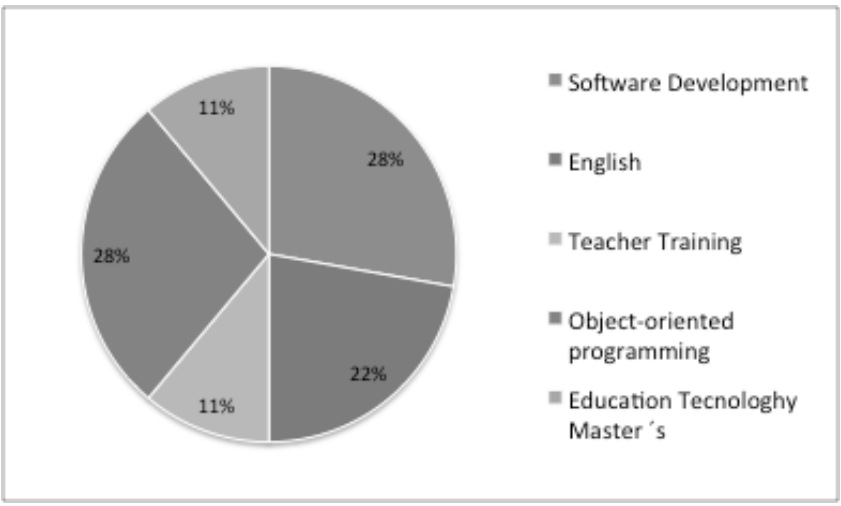


Figure 2: *Attendance to the presentation "Concept, Design and Production of Learning Objects"*

Ten educators or groups or educators with various backgrounds enrolled in the **Call** for developing LOs and have a chance of winning the prize. Figure 3 shows the topics proposed.

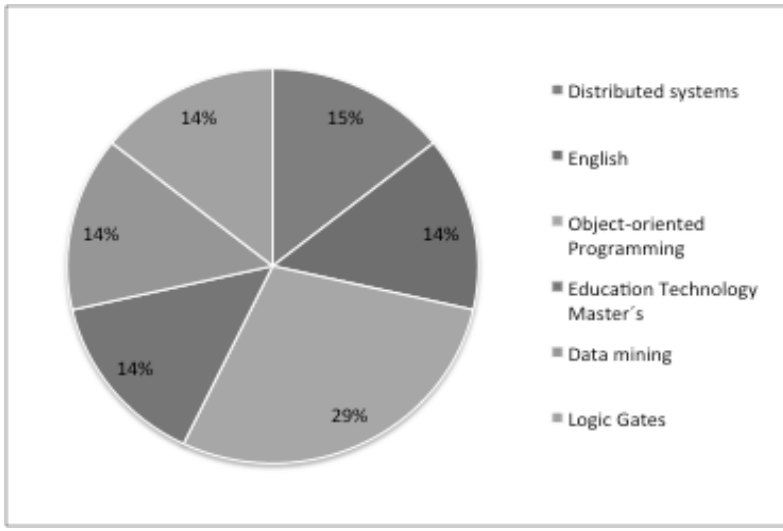


Figure 3: *Enrollment in the Call for creating LOs and proposed topics*

During the **training stage**, several educators took advantage of the various modules offered. Each of the educators who enrolled in the Call participated in those modules they felt they needed the most. Some attended every single module, while some others only took one of the modules. There were even educators who had not enrolled in the Call but decided to take the modules anyway out of interest in the topic.

During the **deliverables stage**, all participating educators submitted their LO planning documentation and most of them also submitted their implemented LOs, with the exception of one educator that asked for an extension.

Another aspect to be highlighted is the importance of the **tutoring sessions offered between deliverables**, since they were extremely helpful for the educators, both to reinforce already imparted knowledge as well as to work in the framework with the objectives and the contents of each planned LO. Those tutoring sessions that were carried out after the feedback for the first deliverable, three important topics were developed, all of them related to central issues discussed in the literature also in relation to LOs: Instructional Design of LOs, Development Tools and Metadata.

As regards the **instructional design**, one of the most common comments was in relation to consistency, when going from objective-contents-practical activities to self-evaluation. In some cases, the feedback after the first deliverable and the subsequent tutoring sessions resulted in the full reorganization of the LO to tend to didactic uniformity and LO granularity issues. For instance, one of the teams of educators ended up designing a chain of 5 LOs, starting with an initial objective that was of a more general nature and was later on split into lower level, specific objectives (and, thus, their corresponding LOs).

The LOs submitted by the educators in response to the Call are currently being evaluated.

4. Conclusions and Future Work

Even though these preliminary results are satisfactory, the road has only begun and there are more challenges ahead. Some of the highlights for the School of Computer Science are the following:
- The topic of learning objects as a specific type of educational material is now know to educators of both graduate and post-graduate courses at the School.
- The Call was well received by educators, considering their background and the diversity of topics used.
- The educators were strong in production, even in interdisciplinary work: graphics, instruction, contents, tools.
- Educators are interested in documenting (in the form of papers or reports) their progress, both in producing the LOs as well as using them with other educators and students.
- The School of Computer Science is on its way to having its own repository in the area of the SEDICI, which will be an innovation at the institutional level.

On the other hand, our future design and production activities in relation to LOs at the School of Computer Science will be as follows:

- Generating a new Call, this time a permanent one, to encourage the continuous production of LOs and offer support to the educators working on them.
- Strengthening the production of LOs that can be assembled for some courses.
- Implementing new functionalities for the repository.
- Moving forward with the evaluation of LOs in use.
- In this sense, generation of an ad hoc design to strengthen research on the topic of LO reutilization, which would be a significant contribution since there are currently no projects being carried out in this area of interest.

Bibliography

1. Hodgins, H. W. (2000). The future of learning objects. The Instructional Use of Learning Objects: Online Version. Retrieved from http:// reusability.org/read/chapters/hodgins.doc
2. L'Allier, J. (1998). NETg's Precision Skilling: The linking of occupational skills descriptors to training interventions. Retrieved from http://www.netg.com/research/pskillpaper.htm
3. Polsani, P. R. (2003). Use and Abuse of Reusable Learning Objects. Journal of Digital Information, 3(4). Retrieved from journals.tdl.org/jodi/article/viewArticle/89
4. Sanz C., Moralejo L., Barranquero F. (2014). Materiales del Curso de Doctorado: "Diseño y Producción de Objetos de Aprendizaje".

5. Sanz C., Moralejo L., Barranquero F. (2014). Metodología CROA. http://croa.info.unlp.edu.ar
6. Maldonado J., Sanz C., Fernandez Pampillón A.M. (2014). Desarrollo de un marco de análisis para la selección de Metodologías de Diseño de Objetos de Aprendizaje (OA) basado en criterios de calidad para contextos educativos específicos". Tesis de la Maestría de Tecnología Informática Aplicada en Educación. March 2015. http://sedici.unlp.edu.ar/handle/10915/45063

Esta edición de 150 ejemplares
se terminó de imprimir en Estudiocentro,
Bolívar, Buenos Aires, Argentina,
en el mes de septiembre de 2016.

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 58 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 978-987-4127-00-6