GUILLERMO FEIERHERD | PATRICIA PESADO | OSVALDO SPOSITTO
(editors)

Computer Science & Technology Series

XX Argentine Congress of Computer Science Selected Papers

Computer Science & Technology Series

XX Argentine Congress of Computer Science
Selected Papers

RedUNCI

RED DE UNIVERSIDADES CON CARRERAS DE INFORMÁTICA

Computer Science & Technology Series

XX Argentine Congress of Computer Science
Selected Papers

Guillermo Eugenio Feierherd
Patricia Mabel Pesado
Osvaldo Mario Spositto
(eds)

Editorial
de la Universidad
de La Plata

Sposito, Ovaldo
Computer Science & Technology Series 2015: XX Argentine Congress of Computer Science Selected Papers / Ovaldo Sposito; Patricia Mabel Pesado; compilado por Ovaldo Sposito; Patricia Mabel Pesado. - 1a ed. adaptada. - La Plata: EDULP, 2015.
310 p.; 24 x 15 cm.

ISBN 978-987-1985-71-5

1. Actas de Congresos. I. Sposito, Ovaldo, comp. II. Pesado, Patricia Mabel, comp. III. Título.
CDD 005.3

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

Diagramación: Andrea López Osornio

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, 2015
ISBN 978-987-1985-71-5
Queda hecho el depósito que marca la Ley 11.723

Impreso en Argentina

Topics

XV Intelligent Agents and Systems Workshop
Chairs Edgardo Ferretti (UNSL) Ricardo Rodríguez (UBA) Sergio A. Gómez (UNS)

XIV Distributed and Parallel Processing Workshop
Chairs Armando De Giusti (UNLP) Marcela Printista (UNSL) Javier Balladini (UNCOMA)

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

XII Graphic Computation, Images and Visualization Workshop
Chairs Silvia Castro (UNS) Roberto Guerrero (UNSL) Javier Giacomantone (UNLP)

XI Software Engineering Workshop
Chairs Patricia Pesado (UNLP) Elsa Estévez (UNU) Alejandra Cechich (UNCOMA) Horacio Kuna (UNaM)

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

IX Architecture, Nets and Operating Systems Workshop
Chairs Jorge Ardenghi (UNS) Nelson Acosta (UNCPBA) Hugo Padovani (UMorón)

VI Innovation in Software Systems Workshop
Chairs Marcelo Estayno (UNLZ) Osvaldo Spositto (UNLaM) Pablo Fillottrani (UNS) Andrés Bursztyn (UTN)

V Signal Processing and Real-Time Systems Workshop
Chairs Oscar Bría (INVAP) Horacio Villagarcia Wanza (UNLP) Daniel Pandolfi (UNPA)

III Computer Security Workshop
Chairs Javier Diaz (UNLP) Antonio Castro Lechtaller (IESE) Javier Echaiz (UNS)

III 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)
Aguirre Jorge Ramió (España)
Alfonso, Hugo (Argentina)
Ardenghi, Jorge (Argentina)
Baldasarri Sandra (España)
Balladini, Javier (Argentina)
Bertone, Rodolfo (Argentina)
Bría, Oscar (Argentina)
Brisaboa, Nieves (España)
Buckle, Carlos (Argentina)
Bursztyn,Andrés (Argentina)
Cañas, Alberto (EEUU)
Casali, Ana (Argentina)
Castro Lechtaller, Antonio (Argentina)
Castro, Silvia (Argentina)
Cechich, Alejandra (Argentina)
Coello Coello, Carlos (México)
Constantini, Roberto (Argentina)
Cukierman, Uriel (Argentina)
De Giusti, Armando (Argentina)
De Vincenzi, Marcelo (Argentina)
Deco, Claudia (Argentina)
Depetris, Beatriz (Argentina)
Diaz, Javier (Argentina)
Dix, Juerguen (Alemania)
Doallo, Ramón (España)
Docampo, Domingo (España)
Echaiz, Javier (Argentina)
Esquivel, Susana (Argentina)
Estayno, Marcelo (Argentina)
Estevez, Elsa (Naciones Unidas)
Falappa, Marcelo (Argentina)
Feierherd, Guillermo (Argentina)
Fernandez Slezak, Diego (Argentina)
Ferreti, Edgardo (Argentina)
Fillottrani, Pablo (Argentina)
Fleischman, William (EEUU)
GarcíaGarino, Carlos (Argentina)
GarcíaVillalba, Javier (España)
Género, Marcela (España)
Giacomantone, Javier (Argentina)
Gómez, Sergio (Argentina)
Gröller, Eduard (Austria)
Guerrero, Roberto (Argentina)
Janowski, Tomasz (Naciones Unidas)
Kantor, Raul (Argentina)
Kuna, Horacio (Argentina)
Lanzarini, Laura (Argentina)
Leguizamón, Guillermo (Argentina)
Loui, Ronald Prescott (EEUU)
Luque, Emilio (España)
Madoz, Cristina (Argentina)
Malbrán, María (Argentina)
Malberti, Alejandra (Argentina)
Manresa Yee, Cristina (España)
Marín, Mauricio (Chile)
Mas Sansó, Ramón (España)
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)
Rexachs, Dolores (España)
Reyes, Nora (Argentina)
Riesco, Daniel (Argentina)
Rodríguez, Ricardo (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)
Vendrell, Eduardo (España)
Velho, Luiz (Brasil)
Vénere, Marcelo (Argentina)
Villagarcia Wanza, Horacio (Argentina)

Organizing Commitee

Universidad Nacional de La Matanza
Buenos Aires – Argentina
Departamento de Ingeniería e
Investigaciones Tecnológicas (UNLaM)

Dean
Ing. Osvaldo Mario Spositto

President
Pontoriero, Daniel

Coordinator
Gigante, Nora

Members
Stratta, Romina
Tesolín, Yamila
Casalini, Danilo
Salerno, Leslie
Ojeda, Juan M.

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 2014 in La Matanza

CACIC'14 was the twentieth Congress in the CACIC series. It was organized by the Department of Engineering and Technological Research at the La Matanza National University (http://www.unlam.edu.ar/) in La Matanza, Buenos Aires.

The Congress included 13 Workshops with 135 accepted papers, 3 Conferences, 3 technical panels, 2 invited tutorials, different meetings

related with Computer Science Education (Professors, PhD students, Curricula) and an International School with 6 courses. (http://cacic2014.ing.unlam.edu.ar/cacic2014/es/escuela/cursos-becas.jsp).

CACIC 2014 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 230 submissions. An average of 2.5 review reports were collected for each paper, for a grand total of 594 review reports that involved about 206 different reviewers.

A total of 135 full papers, involving 445 authors and 78 Universities, were accepted and 24 of them were selected for this book.

Acknowledgments

CACIC 2014 was made possible due to the support of many individuals and organizations. The Department of Engineering and Technological Research at the La Matanza National University, RedUNCI, the Secretary of University Policies, the National Ministry of Science and Technology, CIC and CONICET were the main institutional sponsors.

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

Ing. Armando De Giusti
RedUNCI

Table of Contents

XV

Intelligent Agents and Systems Workshop

Swarm Intelligence Algorithms on GPUs: One Exhaustive Revision

SILVIA M. MOLINA, FABIANA PICCOLI, GUILLERMO LEGUIZAMÓN

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis, Ejército de los Andes 950, San Luis, Argentina
{smolina,mpiccoli,legui}@unsl.edu.ar

Abstract. *Swarm Intelligence* involves actions of groups of decentralized and self-organized individuals. Those actions can usually be performed in parallel, so that GPU architectures are suitable candidates to provide massive parallelism when necessary. This paper provides an exhaustive review of *Swarm Intelligence* metaheuristics using GPUs. The works analyzed here have been proposed since the year 2012 and they are related mainly with two of the most studied metaheuristics regarding the use of GPUs: *Ant Colony Optimization Algorithm* and *Bee Colony Algorithm*. Also other more recent and less studied metaheuristics are considered, they are *Cuckoo Search Algorithm* and *Fireworks Algorithm*.

Keywords: Swarm Intelligence, Graphics Processing Unit, Ant Colony Optimization, Bees Colony Algorithm, Cuckoo Search Algorithm, Fireworks Algorithm.

1. Introduction

Swarm Intelligence (*SI*) studies the collective behavior of systems composed by many individuals (*swarm*) interacting locally and with their environment. The *swarms* use inherently decentralized forms of control and self-organization to achieve their objectives (Martens et al. [1]). Examples of *SI* metaheuristics are *Ant Colony Optimization (*ACO) (Dorigo and Blum [2]), *Cuckoo Search (CS)* (Yang et al. [3]), *Fireworks Algorithm (FWA)* (Tan and Zhu [4]) and *Bee Colony Algorithm (BCA)* (Pham et al. [5]).

Due to different reasons, *SI*-based algorithms are suitable for applying parallel techniques in their computational solution. One of them is regarding its behavior as the swarms involve individuals with independent actions. Another characteristic is that *SI* works better when the swarm size is big and the process implies many repetitions. Furthermore, in large scale problems (e.g., *Big Data*) *SI* techniques must manipulate large search spaces including high dimensional solutions. To manipulate solutions of these large search

spaces and evaluate the quality of those solutions, a huge time consuming processes is involved.
The purpose of this work is to make a careful analysis of the state of the art *SI* algorithms for GPUs. To do that, we consider two the most studied metaheuristics regarding use of GPUs: *ACO* and *BCA*, besides those more recently proposed: *CS* and *FWA*. Our work considers some proposed algorithms since 2012, making special emphasis about how the hardware recourses of GPUs are assigned to the different algorithmic parts so as to achieve a massive data parallelism and, in consequence, a good performance.
This paper is organized as follows, in Section 2 is presented a brief introduction to the GPU architecture and the CUDA programming model, describing its main features. Section 3 describes the *SI* metaheuristics and Section 4 analyses the algorithms and strategies proposed for *SI* on GPUs. Finally, in Section 5 the conclusions are presented.

2. GPU Architecture and CUDA Programming Model

GPUs (*Graphics Processing Units*) have become a valuable alternative to parallel supercomputers due to its computational power, low cost, and steady progress associated with their hardware and software.
The parallel programming of any application on GPU (*GPGPU*, *General Purpose GPU*) should consider several differences with the parallel programming in typical parallel computers; the most relevant are the number of processing units, the structure of the CPU-GPU memory and the number of parallel threads. Regarding the number of processing units, unlike solutions for massively parallel computers where the number of process units coincides with the number of processing units (processors or cores). In GPGPU, this is not considered (it is possible the execution of hundreds or thousands of processing units in a smaller number of cores). For the structure of memory of CPU-GPU system, there are two different memory spaces: memory of the host (CPU) and memory of the GPU. Thus, transference of the code and data of the application between the host and GPU are necessary because the threads of the GPU running on a separate memory space to host threads. Finally, the programming of the GPU allows initializing a large number of threads with a few overhead. All GPUs has transparent and low cost mechanisms for the creation and management of threads. Also, the GPUs have their own scheduling mechanisms; the threads are scheduled to warps. A warp is a scheduling unit and it consists of 32 threads.
There are different alternatives for GPU processing. The *nVidia* card is the most widely used, with the development environment called *Compute Unified Device Architecture* (*CUDA*) which is designed to simplify the work of synchronization of threads and the communication with the GPU. *CUDA* proposes a *Single Instruction Multiple Data (SIMD*) programming model [6] [7] and it presents the GPU architecture as a set of multiprocessors or

Streaming Multiprocessors (SM) in a *Multiple Instructions-Multiple Data (MIMD*) fashion. Each *SM* has a set of processors *SIMD* called S*treaming Processors (SP*). Regarding the memory hierarchy, each *SP* has a number of records as a local memory (accessible only by it), at the same time each *SM* has a shared memory (is accessible by all the *SP* of the *SM*) and finally the global memory, which is accessed by all the *SMs*. The programmer develops a unique program *CUDA* which contains the CPU code (phase of few parallel tasks or no parallelism at all) and GPU device (phase of data parallelism of fine granularity). The code for the GPU is called *kernel* and it is executed by each thread. Each time that a kernel is activated, a structure called *grid* is generated on the GPU, which consists of blocks of threads. Each block runs on a *SM*. The number of threads per block and resources for each of them is determined by the programmer and he/she can make adjustments to achieve improvements in the performance.

3. Swarm Intelligence

Field related to the *Swarm Intelligence* studies the collective behavior of systems composed of many individuals (*swarm*) that interact locally with their environment. The swarms use forms of inherently decentralized control and self-organization to achieve their objectives (Martens et al. [1]). As mentioned before, in this work we focus on the study of *SI* metaheuristics suitable for efficient implementation on GPUs*,* as *ACO*, *BCA*, *CS,* and *FWA*. The characteristics of each are:

- *ACO* (Dorigo and Blum [2])
  It is a population metaheuristic inspired by the behavior of a species of ants in search of food. The ants initially explore randomly the area around the nest. When an ant finds a food source brings it to the nest. This ant deposits in the environment a chemical substance called *pheromone*, depending on the quantity and quality of food found. *ACO* algorithm can be seen as an interaction of three methods: *Construction of Solutions* (it handles a colony of ants, where the ants visit concurrently and asynchronously the states of the problem considered by the movements across the neighboring nodes of the construction graph representing the search space of the solution); *Modification of Pheromone* (it is responsible of the deposit and evaporation process of pheromone) and *Actions Demons* (it performs the centralized actions which cannot be performed by a single ant). In [8] and [9] it show variants of ACO which propose sequential and parallel versions.

- *BCA* (Pham et al. [5])

It is inspired on the behavior of honey producer bees. When they search for food, they perform a search around the neighborhood combined with a random search.

- *CS* (Yang et al. [3])
  It is based on the behavior parasite of some cuckoos species when they deposit their eggs in alien nests. Cuckoo eggs have evolved in shape and color to mimic to eggs of the bird that is nest owner. If this latter bird discovers that the egg is not yours throws it or it leaves the nest and builds a new nest in elsewhere. When the cuckoo egg hatches the cycle repeats.

- *FWA* (Tan and Zhu [4], Ding et al. [10])
  It is a new *SI* algorithm inspired in the phenomenon of exploding fireworks. The optimization process is guided by the explosion of a firework swarm. The best fitness fire generates more sparks within a small range. To improve the search, it is necessary to assign the computing resources according to the characteristics of the areas of the search space. Thus, more computing resources are assigned to best areas and less on the worst areas. Also, it introduces the possibility of increasing diversity in the swarm by applying Gaussian Mutation.

In the next section we analyze the characteristics of *SI* algorithms proposed for CPU-GPU computing system.

4. *SI* Algorithms and Strategies on GPUs

The proposed work arises from observation that, it have been made several works about the study of state of the art of parallel metaheuristics in general and their trends (Alba et al. [8]). Others works have been directed to more specific *SI* metaheuristics (Krömer et al. [11] and Pedemonte et al. [9]).
In the latter mentioned work is described in detail about how the parts of a given algorithm are parallelized. Also, the authors present a review of *ACO* parallel metaheuristics published from 1993 until 2010, it shows the growing interest in this research field, and it is still more remarkable in the years 2009 and 2010. The GPUs are referenced only in 4 papers. Furthermore, they propose a new taxonomy of strategies for *ACO* parallel implementations, which includes general ideas of taxonomies proposed by Randall and Lewis [12] and Janson et al. [13]. For all these reasons and because the works with features that we suggest so far there have recently published, our work represents a first review, though brief but interesting contribution, which may be useful to researchers of this area and it enable them to deepen in this line of research.

Table 1 shows the *SI* algorithms and strategies for GPUs proposed in the revised publications. These publications are related to metaheuristics that have been most extensively studied in general as *ACO* and *BCA*, but regarding the use of GPUs are still in a maturation stage as in the case of ACO or these are in an exploration stage as is the case of *BCA*. Other publications correspond to recently proposed *SI* and with much to investigate respect to algorithms for GPUs as in the case of *CS* and *FWA*.

Researchers seek to exploit massive parallelism through implementations on GPU trying thus to improve the computation time. This time is decreased due to a good use of hardware and software underlying the GPUs. For example: avoiding high time transfers between the CPU and the GPU, avoiding the use of global memory because it has higher latency, using efficient CUDA library functions, balancing the workload among threads of the same block to maximize the parallelism, avoiding the warp divergences so that in the executions the serializations not occur, etc.

Table 1. *SI* Algorithms and Strategies (e) performed on GPUs. *TSP* is about the *Traveling Salesman Problem* and *VRP* is about the *Vehicle Routing Problem*. Bench. (Benchmark) is about a function set.

SI	**Algorithm – Strategy**	**Problem**	**Year**	**Reference**
ACO	[14]-ACO	VRP	2012	Diego et al. [14]
ACO-MMAS	ANT_{thread} (e) ANT_{block} (e) $COLONY_{block}$ (e) $COLONY_{GPU}$ (e)	TSP	2013	Delévacq et al. [15]
ACO-AS	[16]-AS	TSP	2013	Cecilia et al [16]
ACO-AS	[17]-AS	TSP	2013	Dawson and Stewart [17]
ACO-AS	[18]-AS	TSP	2013	Dawson and Stewart [18]
FWA	GPU-FWA	Bench.	2013	Ding et al. [7] [10]
CS	[19]-CS	Bench.	2013	Jovanovic and Tuba [19]
BCA	*CUBA*	Bench.	2014	Luo et al. [20]

The computation time metrics most commonly used are the *Speedup* and the *Efficiency*. In the reviewed works always superlinear values are reached. Interestingly, some studies have even achieved to improve the quality of solutions.
Regarding the computing platform used in experimentation, in most cases systems with nVidia Fermi GPUs with CUDA are used.
Below we highlight the most relevant characteristics of the algorithms of each of the reviewed publications.

4.1 Review of publications related to *SI* for *GPU*

In this section, we show how the researchers can obtain algorithms with good performance when using GPUs. We focus on describing about how the algorithmic parts are mapped to different processing elements such as warps, threads, blocks, grids, and processors. Also, we mention how the memory hierarchy is managed for storing the main data structures of each type of *SI*. Regarding the characteristics of the architecture of the GPU used and metrics considered in each case, only those that are not included in the introduction to Section 4 will be mentioned.

***ACO* Metaheuristic on GPU**. In most publications reviewed the algorithms for *TSP* problem are proposed (Delévacq et al. [15], Cecilia et al. [16] , and Dawson and Stewart [18] [17]), an exceptional case is presented in Diego et al. [14], where the problem *VRP* is approached.

- Delévacq et al. [5] describe two algorithms *Parallel Ants* and *Multiple Parallel Ant Colony* based at the *Max-Min Ant System* with *3-opt Local Search*. In *Parallel Ants*, ants build a tour of parallel manner. Two strategies are proposed: (i) ANT_{thread} where the ants are distributed among *SP* each in a thread. Each thread computes the state transition rule of each ant in a *SIMD* mode. The data structures are stored in global memory and (ii) ANT_{block} where each ant corresponds to a block, a single thread of block is responsible for constructing a tour, the other threads of block calculate the transition rule and the evaluation of candidate cities for be the next component of the current solution in parallel. Two variants of this strategy are analyzed to evaluate the benefits and limitations of the memory used (shared or global) for the data needed for the computation of the state transition rule: Ant_{Block}^{Shared} and Ant_{Block}^{Global} in this latter is obtained greater speedup.
  For *Multiple Ant Colony*, two strategies are proposed: (i) $COLONY_{Block}$, where the block number and the thread number corresponds to the colony number and ant number respectively and (ii) $COLONY_{GPU}$ where the full ant colonies are assigned to various interconnected GPUs. The Ant_{Block}^{Global} strategy is integrated into each colony.

To prevent transfers between the CPU and GPU, the *Linear Congruential* random number generator is used on the GPU (Yu et al. [21]).

- Cecilia et al. [16] present the first ACO algorithm for *TSP* fully developed on GPU. They propose various designs of *Ant System (AS)* algorithm for *GPU* looking to exploit data parallelism and memory hierarchy. To improve the memory bandwidth, it is used shared memory for data structures. In addition, a *Tiling* technique is proposed where all threads cooperate to load the data from global memory to the shared memory. For the construction of a tour, two types of ants are distinguished: the *Queen* ants and the *Worker* ants; the latter type of ant helps to the first groups of ants in the selection of the next city to visit. Each *Queen* is associated with a block where each thread represents one of the cities that the *Workers* should visit. The warps divergences are avoided with a *Tabu* list (cities that cannot be visited), this list is stored in shared memory or in a register of a file depending if the *Tiling* technique is used or not for to distribute the cities between the threads of a block.
  To improve the *Roulette Wheel* (*RR*) method, the *Independent Roulette* (*I-Roulette*) method is proposed which is more parallelizable than the first method.
  One thread performs the evaporation of pheromone matrix. To the deposit of pheromone is used *Scatter-to-gather* transformations to unify the values that each ant deposits, thus it avoids to perform costly atomic operations. A thread is created in the different blocks for matrix cells.

- Dawson and Stewart [17] propose a parallel implementation of an *AS* algorithm. They extend recent research like Cecilia et al. [16]. On GPUs is made the run the initializing of the structures, building of the solution and the updating of the pheromone matrix. It is proposed the *Double-Spin Roulette* (*DS-Roulette*) method which is alternative to *RR* method, because the selecting of the next component of the solution which involves this latter method. The *RR* is complex to implement on GPU due to its lineal nature, the divergent control structures and the need of random numbers and also, due to the constant synchronization of the threads. *DS-Roulette* method is highly parallel because exploits the parallelism to warp level; it reduces the dependencies of shared memory, the number of total instructions and the work done by the GPU. For the construction of a tour, each ant is mapped to a block of threads. Another proposal is to divide the block threads dedicated to a number of valid cities, thus applying of warps level parallelism, which implicitly guarantees that all threads run without divergences and the execution time decreases significantly.

- Dawson and Stewart [18] is an extension of Dawson and Stewart [17] designed to solve large instances of the TSP problem. Parallel strategies are presented using the set of candidate cities for execution in GPU. This set is essential to limit the search space and to reduce the total execution

time. The best results are achieved when only one ant is associated by a block. In addition, to reduce the speedup are used warps level primitives and branches (or warps divergences) are avoided.

- Diego et al. [14] propose a parallelization strategy to solve the *VRP*. Several kernels are implemented aimed at: (1) the data initialization of the solutions, (2) the construction of the solutions, (3) the choice of the best solution, (4) to record the best solution, (5) the evaporation of the pheromone matrix, and (6) the deposit of pheromone on this matrix. Each kernel corresponds to a single block. In (4) and (6) it is used one thread for kernels, for (2) it is used a number of threads equal to the number of ants; for (3) a thread count equal to half the number of ants is used to the kernel and for (1) a number of threads equal to the solutions number multiplied by the number of ants it is used to the kernel. However, the article does not give any explanation about the amounts of threads selected.

***FWA* Metaheuristic on GPU.**

- Ding et al. [10] propose adapting the *FWA* original algorithm to the GPU architecture, the resulting algorithm is the *GPU-FWA*. This algorithm shows a very good performance, it shows a superlinear speedup respect to *FWA* sequential versions and PSO. Also, it is simple and scalable algorithm. In this algorithm, one warp is assigned to each fire, within every warp the sparks of the fire are generated. In this way, the sparks are implicitly synchronized and it is reduced the communication overhead between them.

  The explosion is implemented in a block; this allows using the shared memory. Only the global memory is accessed to know the position of the fire and the value of fitness, in this way the access latencies decrease significantly.

  The Attract-repulse mechanism mutation is introduced to speed up the search process and to achieve a balance between exploitation and exploration. The CURAND library of GPU is used for the generation of high-quality random number.

***CS* Metaheuristic on GPU.**

- Jovanovic and Tuba [19] propose a parallel algorithm on GPU which shows significant changes respect to the sequential version because the latter has several parts that are not naturally parallel. Parallelization is used on three levels: the parallel reduction in calculating the fitness function of each individual; the calculating the swarm nests which are made in a block; finally several colonies are executed in parallel in separate blocks.

  It is assuming that the algorithm is used to optimize functions with several summations as the functions: Sphere, Rosenbrocks Valley, Schwefels, etc.

The results show that the *CS* parallel algorithm outperforms the sequential version in terms of execution time and the quality of the solutions to some evaluations.

***BCA* metaheuristic on GPU.**

- *Luo et al.*[20] presented the *CUBA* algorithm which it would be the *BCA* first algorithm on GPUs according to its authors. They designed a Bees Multi-colony algorithm where several *Bee* algorithms are executed in parallel. Each bee is assigned to a thread and it is responsible to find a solution for their colony. A block is divided into different colonies which run independently. When an iteration has been completed, the algorithm *CUBA* exchanges information between the colonies of the same block across of the shared memory. The number of bees and the number of colonies depends of the number of threads per block and the number of blocks per grid respectively.
  The parallel algorithm groups the bees in the different areas depending on the workload of the threads unlike the sequential algorithm that includes to the most bees in the most promising areas and the rest of these in other areas. In this algorithm, various colonies make parallel searches in different areas.
  At the end of each iteration, the *Odd-Even* mechanism is used in parallel by blocks for the ordering the solutions, this provides the best solution. The convergence time decreases by the use of a communication mechanism of two phases in the colonies. Inside a block, the best solution of the whole population will replace the worst one.
  The proposed algorithm shows a very good performance. An extensive experimentation is performed for which it is considered various parametric combinations. The Speedup and the Robustness of this algorithm is calculated.

5. Conclusions

In this work an useful reading material is achieved for those interested in the advances of the field of research related to the *Swarm Intelligence* because it is presented an incipient review of algorithms *SI* for GPUs, which are proposed in published work since 2012.
We analyzed the ACO metaheuristic which is one of the most studied metaheuristics in this field and we also reviewed other newer and less explored metaheuristics as *CS, FWA* and *BCA*.
We observed that *Swarm Intelligence* can exploit the massive parallelism by at least applying strategies to minimize the computation time and/or the memory access time using efficient *CUDA* functions.
In general, it is necessary a comprehensive study of the underlying CPU-GPU architecture in terms of the problem at hand and the characteristics of

the metaheuristic itself to manage efficiently the resources than this architecture provides.
Finally, the minimization of the memory latency is interesting line of research when designing *SI* models for GPUs as the memory management is one of the central problems in the design of efficient and scalable algorithms. This problem is exacerbated when the amounts of memory change during the execution, depending on the non-deterministic behavior or the dynamic behavior of algorithm. For example, in *ACO* is necessary pay attention to the pheromone matrix since it is one of the central structures to manage, its size depends on the size of the problem and all the ants have access to the matrix at each iteration; however, in *FWA* is necessary to consider the dynamic and random generation of sparks from the fire explosions.

References

1. Martens, D., Fawcett, T., Baesens, B.: Survey: Swarm Intelligence for Data Mining. Machine Learning 82(1), pp. 1-42 (January 2011).
2. Dorigo, M., Blum, C.: Ant colony Optimization Theory: A Survey. Theoretical Computer Science 344(2-3), pp. 243-278 (2005).
3. Yang, X., Deb, S.: Cuckoo Search Via Lévy Flights. In: Nature Biologically Inspired Computing. NaBIC, pp.210-214 (Dec. 2009).
4. Tan, Y., Zhu, Y.: Fireworks Algorithm for Optimization. In Tan, Y., Shi, Y., Tan, K., (eds.): Advances in Swarm Intelligence. LNCS. vol 6145, pp. 355-364. Springer Berlin Heidelberg (2010).
5. Pham, D., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M.: The Bees Algorithm - A Novel Tool for Complex Optimisation Problems. In: Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems (IPROMS 2006), pp.454-459. Elsevier (2006).
6. Kirk, D., Hwu, W.: Programming Massively Parallel Processors: An Approach 1st edn. Morgan Kaufmann Publishers Inc., San Francisco, CA. USA (2010).
7. Sanders, J., Kandrot, E.: CUDA by Example: An Introduction to General-Purpose GPU Programming 1st edn. Addison-Wesley Professional (2010).
8. Alba, E., Luque, G., Nesmachnow, S.: Parallel Metaheuristics: Recent Advances and New Trends. Wiley Online Library, pp. 1-48 (2013).
9. Pedemonte, M., Nesmachnow, S., Cancela, H.: A survey on Parallel Ant Colony Optimization. Applied Soft Computing 11(8), 5181-197 (2011).
10. Ding, K., Zheng, S., Tan, Y.: A GPU-based Parallel Fireworks Algorithm for Optimization. In: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, GECCO '13, pp.9-16. ACM, New York, NY, USA (2013).
11. Krömer, P., Platoš, J., Snášel, V.: Nature-Inspired Meta-Heuristics on Modern GPUs: State of the Art and Brief Survey of Selected Algorithms. International Journal of Parallel Programming 42(5), 681-09 (2014).
12. Randall, M., Lewis, A.: A Parallel Implementation of Ant Colony Optimization. J. Parallel Distrib. Comput. 62(9), 1421-1432 (2002)

13. Janson, S., Merkle, D., Middendorf, M.: Parallel Ant Colony Algorithms. pp. 171-201. John Wiley & Sons. Inc. (2005).
14. Diego, F., Gómez, E., Ortega-Mier, M., García-Sánchez, A.: Parallel CUDA Architecture for Solving de VRP with ACO. In Sethi, S., Bogataj, M., Ros-McDonnell, L., (eds.): Industrial Engineering: Innovative Networks. pp. 385-393. Springer, London (2012).
15. Delévacq, A., Delisle, P., Gravel, M., Krajeck, M.: Parallel Ant Colony Optimization on Graphics Processing Units. Journal of Parallel and Distributed Computing 73(1), 52-61 (2013).
16. Cecilia, A., Nisbet, M., Amos, M., García, J., Ujaldón, M.: Enhancing GPU Parallelism in Nature-inspired Algorithms. J. Supercomput. 3(63), 773-789 (March 2013).
17. Dawson, L., Stewart, I.: Improving Ant Colony Optimization Performance on the GPU Using CUDA. In: Evolutionary Computation (CEC), 2013 IEEE Congress on, pp.1901-1908 (June 2013).
18. Dawson, L., Stewart, I.: Candidate Set Parallelization Strategies for Ant Colony Optimization on the GPU. In Kolodziej, J., Di Martino, B., Talia, D., Xiong, K., (eds.): Algorithms and Architectures for Parallel Processing. LNCS, vol. 8285, pp. 216-226. Springer International Publishing (2013).
19. Jovanovic, R., Tuba, M.: Parallelization of the Cuckoo Search Using CUDA Architecture. In: 7th International Conference on Applied Mathematics, Simulation, Modelling (ASM '13) (2013).
20. Luo, G., Huang, S., Chang, Y., Yuan, S.: A Parallel Bees Algorithm Implementation on GPU. J. Syst. Archit. 60(3), 271-279 (March 2014).
21. Yu, Q., Chen, C., Pan, Z.: Parallel Genetic Algorithms on Programmable Graphics Hardware. LNCS, vol. 3612, pp. 1051. Springer (2005).

Applying CHC Algorithms on Radio Network Design for Wireless Communication

DANIEL MOLINA[1], DANIEL PANDOLFI[1], ANDREA VILLAGRA[1], GUILLERMO LEGUIZAMÓN[2]

[1] Universidad Nacional de la Patagonia Austral, Argentina
{dmolina, dpandolfi,avillgra}/@uaco.unpa.edu.ar

[2] Universidad Nacional de San Luis, Argentina
legui@unsl.edu.ar

Abstract. The diffusion of wireless communication services (telephone, internet, etc.) is continuously growing these days. Unfortunately, the cost of the equipment to provide the service with the appropriate quality is high. Thus, selecting a set of geographical points allowing optimum coverage of a radio frequency signal by minimizing the use of resources is essential. The above task is called the Radio Network Design (RND) and is a NP-hard problem, i.e., this can be approached by using metaheuristics techniques. Metaheuristics are methods comprising local improvement procedures and high-level strategies for a robust search in the problem space. In this work, different versions of the CHC algorithm with a fitness function based on the efficiency of resource use are proposed. The achieved results are encouraging in terms of efficiency and quality in all the analysed scenarios.

1. Introduction

Nowadays wireless media are perhaps the largest generators of the growth in communications. A message is encoding using electromagnetic waves between the transmitter and the receiver in order to let the messages exchange. Waves are subject to noise through the free space (attenuation, reflection, refraction, and diffraction) caused by atmospheric phenomena and obstacles [15]. Consequently there are losses in the signal level and the receiver may not be able to decode the message. The focus of this work is based on radio frequency, allowing to extend the application of metaheuristics to any service (radio, television, Internet, cell phone, etc.) [9]. Hence is a paramount challenge to achieve positioning the antennas where the most of the receivers can have an unobstructed line of sight and thus, ensuring as much as possible a high quality signal level during the whole transmission time. The task for solving this problem in the traditional method is called RND [8]. In this work we propose to analyze and evaluate population-based metaheuristics, more precisely a variant of

Genetic Algorithms (GAs) [6] to find an acceptable solution for the RND problem. Former versions of GAs are implemented with one cut point, two cut point and uniform recombination operators. The CHC Algorithm (Crossover elitism population, Half uniform crossover combination, Cataclysm mutation) is a variant ofa GA proposed in [5]. Our proposed CHC Algorithm consists of the implementation of three alternative methods for the "Cataclysm mutation" which is used in CHC for breaking a local optimum convergence: (1) the classical mutation method, (2) retaining a proportion of bits of the best individual method, and (3) a local search iterated hybridization [7]. Besides we introduce a variation mechanism of the Hamming distance for incest prevention and also a fitness function based on the efficiency of surface coverage with respect to the radiated signal. In this way, each installed resources (e.g., antennas) will be used to its full potential, and consequently minimize the amount of them necessary to fulfill the objective. This paper is organized as follows: Section 2 contains the problem definition and explains the proposed fitness function. Section 3 describes the proposed algorithms and develops alternative mechanism to detect the population convergence. Section 4 presents the experimental study and the achieved results, and finally in Section 5, the conclusions of this work are given.

2. Problem Definition

According to [4] the problem of positioning antennas can be described as: given a set of candidate sites, a discretized geographic area, and a set of points that need to intercommunicate with each other; the objective is to select a subset of sites from the set of candidate sites that maximize the coverage by using a minimum number of resources and satisfying the traffic estimation and threshold signal reception between points. The signal level between the transmitter and receiver [17] determines the available bandwidth, being the indicator to satisfy the estimated traffic and threshold reception. This is regardless of the equipment capacity, technology or service to provide (e.g., radio, television, mobile phone, Internet, etc.). In order to discretize the ground surface, a matrix M of $f \times c$ is used, relating the latitude and longitude with the subscripts i, j of the cells in the matrix. Then, the height of each point at sea level is stored in each cell in M. This form of discretization responds to the raster model used in geographic information systems (GIS) and has been used in [10] and [12] to solve the RND problem. The candidates for the installation of antennas sites are represented as a subset of cells belonging to the matrix M. For this problem, omnidirectional antennas (radiating in all directions) are used and a ground plane is considered. Figure 1 shows the achieved

coverage by a possible radio network. The value 1 in a cell indicates that the site is covered by a single antenna brand, the value -1 indicates that the site is covered by more than one cell; therefore, an interference is produced. Finally, a value 0 means that the site is not covered by any antenna. Each possible radio network is modeled by the activation or not of each of the candidate sites. The problem search space is determined by 2^n where n is the number of candidate sites.

The fitness function uses the values in the ground matrix M and assigns a unique numerical value to each solution. This value is a measure of the quality of the radio network evaluated and leads to different search algorithms on the radio network to maximize coverage, and minimize interference and use of resources.

1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	-1	1	1	0	1	1	1
0	0	0	1	1	-1	1	1	0	1	1	1
0	0	0	1	1	-1	1	1	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Fig. 1. Discretized ground by a matrix with possible covering values for a candidate network radio

The amount of $1s$, $0s$, and $-1s$ indicate respectively the total number of covered sites, uncovered sites, and illuminated sites by more than one antenna. Equation 1 displays the fitness function using the values obtained, the first factor determines the degree of covering ground and the second factor determines the quality of the covering ground. When the factors tend to one, the maximization of the covered area and the minimization of the use of resource is satisfied. Finally, the closer to 1 is the fitness value; the better is the solution (i.e., the maximum coverage with minimum use of resources).

$$f(x) = (1 - \frac{\#UncoveredPoints}{\#GroundPoints}) \times (1 - \frac{\#InterferedPoints}{\#CoveredPoints}) \quad (1)$$

3. CHC Algorithm

The CHC Algorithm combines a conservative selection strategy that preserves the best individuals found [5], [16]. The recombination operator generates offspring that maximizes their genetic differences from their parents. Reproduction occurs only if the Hamming distance (genetic difference between the parents) is greater than the threshold of incest (1/4 the size of the chromosome). The half uniform crossover scheme (HUX) is used to maximize the genetic distance between individuals. The new population is generated with a selection of the best individuals between parents and children. For each not improving generation, the genetic distance is decreased if the number of bits different from the selected parents is less than thresholds, i.e., the individuals are too similar. This determines that the population has converged, so that the population is restarted by triggering a cataclysm method [3]. Algorithm 1 shows the pseudo-code of CHC.

3.1 Proposed CHC Algorithms

For the RND problem with large chains bits (e.g., 200 or more bits) incest threshold for restarting the population demands a large computational effort

Algorithm 1 CHC pseudo-code

```
1: t = 0 // time generation
2: Initialize (Pa, incestThreshold)
3: while not (endCondition(t, Pa)) do
4:         parents=parentsSelection(Pa,incestThreshold)
5:         offspring=HUX(P(t))
6:         evaluate(offspring, parents)
7:         Pn=elitismSelection(offspring, parents)
8:         if not (improvement(Pa,Pn))
 9: incestThreshold=incestThreshold-1
 10:        if incestThreshold <= k
11:     Pn =cataclysm(Pa)
12:  end if
13:        endif
14:        t = t+1
15:        Pa = Pn
16: end while
17: return best solution
```

and number of iterations. The proposed method differs from [1][11] in the use of the genetic variability of the population to measure the level of diversity of the individuals. It is based on the probability of acceptance of the Simulated Annealing (SA)[2], but in reverse way as the temperature increases with the number of iterations and therefore increasing the chance of a cataclysm. For each unimproved iteration, the variability of the generated population is compared against a variability pattern. The difference between these two variability patterns must always be lower than the environment centered on the pattern variability and the radius must always be equal to the probability of acceptance. As iterations progress, the environment is reduced and if the variability of the generated population is outside of the environment, then the cataclysm is triggered. The variability *v* is defined by:

$$v = \frac{\sum_{i=1}^{n} \left(\sum_{j=1}^{c} x_{ij} \right)}{n \times c}$$

where $x \in (0, 1)$, v is the population variability, c the chromosome size and n the number of individuals in the population. The acceptance environment *ea* is defined by:

$$ea = 1 - e^{\left(\frac{(f(x') - 1)}{T} \right)}$$

where $f(x^r)$ is the local optimum value and T the number of iterations without improvements. The proposed based CHC algorithm is obtained by replacing the lines 9 to 12 of Algorithm 1 by the pseudo-code displayed in Algorithm 2. The parameter "method" in line 5 varies from values 1 to 3 (see explanation below) giving rise to the three CHC versions proposed here.

Algorithm 2 Pseudo-code for detecting population convergence

```
1: varPattern=0.5// variability pattern
2:          standingPopulationVariability(Pa)
3: acceptationRadius(T, fitnessUnimprovedValue)
4:                  if variability is out of range
5:   Cataclysm(Pa, method)
6: else
7:          T = T+1 // T increases and decreases acceptance range 8:
  end if
```

The restart operators proposed for this work are: (1) Random Method (RM), the new population keeps a copy of the best individual and the rest individuals are obtained by mutating the best individual [13]; (2) Bits Conservation Method (BCM), the new population is generated retaining the best individual and varying only a percentage of bits of the original string [14]; (3) hybridization with Iterated Local Search (ILS)[7]. The proposed CHC versions differ from the canonical CHC in the mechanism to escape from suboptimal regions, so they are called QCHC (Quasi CHC). To identify the method used in each version we call QCHC-RM for method (1), QCHC-BCM for method (2), and QCHC-ILS for method (3).

4. Experiments and Analysis of Results

To analyze the performance of the proposed CHC Algorithms (QCHC-RM, QCHC-BCM, and QCHC-ILS) we additionally considered a set of GAs with different crossover operators: one-point crossover (GA-OPX), two-point crossover (GA-TPX), and uniform crossover (GA-UX). For each problem instance (i.e., matrix) the number of candidate sites was varied. The configurations of the seven instances used are shown in Table 1. Column Instance displays the name of each instance, column *Matrix* shows the matrix size used to discretize the respective ground scenery. Column *size* indicates the number of cells of each ground matrix as a result of product of the #rows by #columns. Column *Sites* indicates the number of candidates for each instance.

The behavior of the studied algorithms is analyzed using various degrees of complexity related to the matrix dimensions and number of sites conforming the optimal solution. The antennas use

Table 2. Success rate obtained by GA-UX, GA-OPX, GA-TPX, QCHC-RM, QCHC- BCM and QCHC-ILS Algorithms

Instance	GA-UX	GA-OPX	GA-TPX	QCHC-RM	QCHC-BCM	QCHC-ILS
Inst 01	**100%**	**100%**	**100%**	**100%**	**100%**	**100%**
Inst 02	93.33%	70%	**100%**	**100%**	**100%**	**100%**
Inst 03	66.7%	66.7%	**100%**	**100%**	90%	**100%**
Inst 04	**100%**	**100%**	86.67%	**100%**	90%	**100%**
Inst 05	90.00%	80.00%	70.00%	**100%**	93.33%	**100%**
Inst 06	63.33%	66.67%	86.67%	90%	93.33%	**100%**
Inst 07	26.67%	33.33%	30%	43.33%	86.67%	**100%**
Average	77.41%	73.81%	81.90%	90.48%	93.33%	**100%**

A significance level of $\alpha = 0.05$ was used for all the applied statical tests. Due to the nature of the experiments, the values obtained by the algorithms satisfy the condition of independence, thus Kolgomorov-Smirnov test is applied to determine whether the data fit a normal distribution and the Levene test is applied to verify whether the data have homoscedasticity. As the results obtained by the algorithms do not meet the conditions of normality and homoscedasticity, non-parametric test (Kruskal-Wallis) was used for the algorithms on instances Inst 01 to Inst 04. The Wilcoxon test is applied to the algorithms on the Inst 05 since only two algorithms are compared: **QCHC-RM** and **QCHC-ILS**.

Tables 3 and 4 show the results obtained by the statistical tests for performance variables of time and number of function evaluations to achieve the best solution. For both tables, column Instance indicates the scenario configuration, columns GA-UX, GA-OPX, GA-TPX, QCHC-RM, QCHC-BCM, and QCHC-ILS show respectively the average values obtained for each evaluated algorithm on the instances that achieved 100% of success. N/A (Not Available) indicates that the algorithm does not meet this condition. Column KW/W represents the initials of the Kruskal-Wallis and Wilcoxon test respectively. The symbols $(+)$ and $(-)$ stand respectively for there are and there are not statistical difference. The symbol $*$ indicates that the Wilcoxon test was applied. Tukey's test applies in all cases where statistically significant differences are detected to identify the best performing algorithm.

Table 3. Average values for the performance variable time and the statistically significant differences obtained in each test

Instance	GA-UX	GA-OPX	GA-TPX	QCHC-RM	QCHC-BCM	QCHC-ILS	KW/W
Inst 01	1.710	2.808	1.953	0.424	1.020	**0.405**	(+)
Inst 02	N/A	N/A	3.317	18.219	7.639	**0.754**	(+)
Inst 03	N/A	N/A	5.674	3.634	N/A	**1.127**	(+)
Inst 04	9.878	17.528	N/A	6.664	N/A	**1.806**	(+)
Inst 05	N/A	N/A	N/A	12.562	N/A	**2.527**	*(+)

Table 4. Average values for the performance variable number of evaluations and the statistically significant differences obtained in each test

Instance	GA-UX	GA-OPX	GA-TPX	QCHC-RM	QCHC-BCM	QCHC-ILS	KW/W
Inst 01	17023	28217	19663	6550	16167	**6417**	(+)
Inst 02	N/A	N/A	32307	298033	107733	**10550**	(+)
Inst 03	N/A	N/A	52793	40650	N/A	**13183**	(-)
Inst 04	86247	153673	N/A	67000	N/A	**18333**	(+)
Inst 05	N/A	N/A	N/A	112133	N/A	**22100**	(+)

We can see in Table 3 that the results obtained by **QCHC-ILS** in terms of performance time variable are lower than the remaining studied algorithms. The statistical test applied to the results obtained by the algorithms in terms of this performance variable indicate that there are statistically significant differences between the results. In all analyzed cases the differences arise between **QCHC-ILS** with respect to the other algorithms. In Table 4 we can observe that in the number of evaluations required to achieve the best solution, again the **QCHC-ILS** algorithm is one that requires the fewest number of evaluations in all instances tested. In this case, the algorithms present differences statistically significant in four of five instances. Here again statistically significant differences occur between **QCHC-ILS** with respect to the other algorithms. Figure 2 represents the number of instances with 100% of the hits obtained by each algorithm. We can observe that the proposed **QCHC-ILS** algorithm is the only one in all instances obtaining 100% success reaching the optimum value.

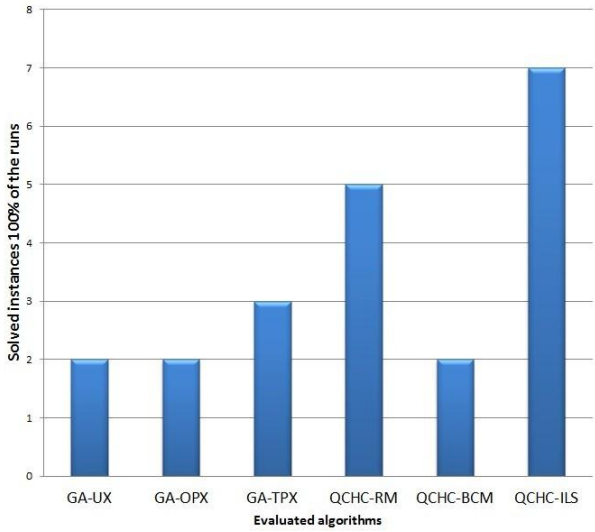


Fig. 2. Number of instances solved by the evaluated Algorithms

5. Conclusions

Three GAs called respectively **GA-UX**, **GA-OPX**, and **GA-TPX** were ana- lyzed, and three versions of the **CHC** algorithms called respectively **QCHC-RM**, **QCHC-BCM**, and **QCHC-ILS** are proposed and analyzed in this work. New incest prevention mechanism based on population variability and a fitness function considering coverage efficiency of signal was used in the scenarios. The proposed mechanism can detect incest prevention to avoid premature convergence at a low computational cost. All the **CHC** versions used the **HUX** operator and the elitist selection as usually applied in the former CHC algorithm. The QCHC- **RM**, **QCHC-CBM** and **QCHC-ILS** algorithms use the best individual to generate the new population. The **QCHC-ILS** algorithm, in addition, applies an iterated local search procedure that intensifies the search around each newest generated individual during the evolutionary process.

The first instance was solved by all competitors. However, as the complexity/size of the instances increase several algorithms were not able to reach the optimum in the 30 runs (i.e., a hit ratio less than 100%). From a statistical point of view, the hybrid algorithm (**QCHC-ILS**) is the best performing algorithm with respect to the complete set of studied algorithms. This is the only algorithm that achieves the optimal for all the instances considered. In addition to the performance variables of time and number of evaluations, the

hybrid approach has the best performance with 95% confidence level for the statistical test applied. This means that this version of the **CHC** Algorithm is an effective improvement over the competitors and the other QCHC versions. Future studies will address different size of problems in order to study scalability properties for large dimensional search spaces. Also, other aspects of the CHC improvements will be analyzed such as the selection mechanism and the use of parallel architectures.

Acknowledgments We acknowledge the constant support by the Universidad Nacional de la Patagonia Austral. The last author also acknowledges the constant support afforded by the Universidad Nacional de San Luis that finances his current researches.

References

1. E Alba, G Molina, and F Chicano. Optimal placement of antennae using meta- heuristics. In Numerical Methods and Applications, pages 214-222. Springer, 2007.
2. V.Cˇerny`. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of optimization theory and applications, 45(1):41-51, 1985.
3. P.K. Chawdhry, R. Roy, and R.K. Pant. Soft Computing in Engineering Design and Manufacturing. Springer London, 1998.
4. D. Corne, M. J. Oates, and G.D. Smith. Telecommunications optimization: heuristics and adaptive techniques. John Wiley, 2000.
5. L.J. Eshelman. The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination. In Foundations of Genetic Algorithms, pages 265-283. Morgan Kaufmann, 1991.
6. D.E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Artificial Intelligence. Addison-Wesley, 1989.
7. O. Martin H. R. Lourenco and T. Stutzle. Iterated local search. In Handbook of Metaheuristics. In Vol. 57 of Operations Research and Management Science, pages 321-353. Kluwer Academic, 2002.
8. N. Gerlich K. Tutschku and P. Tran-Gia. An integrated Approach to Cellular Network Planning. In Institute of Computer Science. University of Wurzburg, 1995.
9. A.R.C. Lechtaler and R.J. Fusario. Comunicaciones: una introducción a las redes digitales de transmisión de datos y señales isocronas. Área: computación. Subarea: comunicaciones. Alfaomega, 2013.
10. F. Luna, J. J. Durillo, A. J. Nebro, and E. Alba. Evolutionary algorithms for solving the automatic cell planning problem: a survey. Engineering Optimization, 42(7):671-690, December 2010.
11. G Molina, F Chicano, and E Alba. Optimal location of antennas in telecommunication networks. Optimization Techniques for Solving Complex Problems, pages 287-307, 2009.

12. P. Kuonen D. Kobler P. Callegari, F. Guidec. Parallel island-based genetic algorithm for radio network design. In Journal of Parallel and Distributed Computing (47), pages 86-90, 1997.
13. S.K. Pal and P.P. Wang. Genetic Algorithms for Pattern Recognition. Computer science mathematics. Taylor & Francis, 1996.
14. I.C. Parmee. Evolutionary and Adaptive Computing in Engineering Design: With 98 Figures. Springer London, 2001.
15. J.D. Parsons. The Mobile Radio Propagation Channel. Wiley, 2000.
16. S. Priem-Mendes, G. Molina, M. A. Vega-Rodríguez, J. A Gomez Pulido, Y. Saez, G. Miranda, C. Segura, E. Alba, P. Isasi, C. Leon, and J. M. Sanchez-Pérez. Bench- marking a wide spectrum of metaheuristic techniques for the radio network design problem. IEEE Trans. Evolutionary Computation, 13(5):1133-1150, 2009.
17. T.S Rappaport. Wireless Communications: Principles and Practice, 2/E. Pearson Education, 2010.

XIV

Distributed and Parallel Processing Workshop

Methodology to predict the energy consumption of checkpoints in HPC systems

JAVIER BALLADINI[1], MARINA MORÁN[1], DOLORES REXACHS[2], EMILIO LUQUE[2]

[1]Facultad de Informática. Universidad Nacional del Comahue
Buenos Aires 1400. 8300 - Neuquén. Argentina
{javier.balladini, marina}@fi.uncoma.edu.ar

[2]Department of Computer Architecture and Operating Systems.
Universitat Autònoma de Barcelona
Campus UAB. Edici Q. 08193 Bellaterra (Barcelona). España
{dolores.rexachs, emilio.luque}@uab.es

Resumen. While the performance of high performance computing applications keep growing, the machines grow up significantly its processing units. This makes that fault tolerance and energy consumption became issues even more important. The fault tolerance methods have great influence on energy consumption making important knowing the impact that diferent methods and configurations can have on an application before we run it. In this paper we show a methodology to predict energy consumption produced by remote coordinated checkpoint method. The metodology is based on a energy caraterizations of the application and an analitic model that instantiet with the caracterized parameters. The model allow to predict the energy consumption of the checkpoints for any problem size and any cpu frequency. The results of the predictions show a presicion greater than 95%.

Key words: checkpoint, energy consumption, high performance computing.

1. Introduction

While the performance of high performance computing (HPC) applications keep growing, the machines grow up significantly its processing units. This increase in the number of components decreases realiability and increase energy consumption of a computing system. So, a few to arrive to the exascale era (scheduled for 2020), fault tolerance and energy consumption have been identified as the two biggest challenges we face [4, 3]. The magnitude of the problem is significant. On the one hand, in an exascale machine is expected to have a failure every few minutes [3], which added to the larger number of processing units that will use applications (as you increase the performance of HPC systems, also increases the complexity of the applications, either by increasing the size of the problem, the accuracy of

their calculations, or because new problems are viable to solve), produce an unavoidable need for fault tolerance mechanisms.

On the other hand, energy consumption is today a big problem, and it will be bigger with the exascale machines. To give an idea of the problem magnitude, we use as an example the most powerfull machine of these days, the chinese Tianhe-2 machine. According to the TOP500[1], this machine demand 17.8 MW of power, which is required to supply homes in a city with 200,000 inhabitants (calculation based on consumption of a home in Argentina). Whereas a relatively cheap energy contract of 1 million dollars per MW, per year should pay a 17.8 million dollars to the energy supplier, causing a high economic impact. But no only that, such energy generation has a high social impact (for example, the largest source of global energy is derived from coal, with a highly dangerous extraction) and environmental (eg, hydroelectric dams that modified the ecosystem).

The fault tolerance methods have an impact on energy consumption and the power required by the system to run applications. This is because any method we use adds significant processing, and possibly disk storage, additional to the application. Presently, the fault tolerance method more widely used in large computer systems is the coordinated checkpoint / restart. This strategy consists mainly in restarting the application after replacing the component that failed, but to avoid a complete reexecution of the application, we periodically save the execution state in secondary storage, local or remote. When a software or hardware failure occurs, the application is restarted from the last stored execution state. The widespread use of this method is because It has a simple implementation, since synchronization points required to store the state of the processes are naturally present in most parallel applications.

It is very importance to know in advance the energy impact that can produce different configurations of the method utilized for checkpointing on each application. Thus, a system administrator could decide which configuration to use to maintain a certain balance between performance, fault tolerance and energy consumption. In this paper we study the energy behavior of remote coordinated checkpoints in cluster systems, considering different cpu frequencies, different time intervals of checkpoints, and different application data sizes. The main contribution of this paper is a methodology that allow to build a model to predict energy consumption which involves the use of the fault tolerance mechanism we mentioned in a given application, under new problem sizes and CPU frequencies in execution of checkpoints.

The rest of this article is organized as follows. Section 2 gives a brief review of the work in this area. Section 3 presents the methodology of prediction of energy consumption of remote coordinated checkpointing. In Section 4,

1 http://www.top500.org/

experimental results are presented used to validate the methodology. Finally, Section 5 presents the conclusions and future work.

2. Related Work

Evaluation of energy consumption for various methods of fault tolerance has been treated in a few items. In [5], it presents a energy assessment of coordinated and uncoordinated checkpoint protocols, differentiating the three main tasks associated with the checkpoint method (writing of checkpoint, recovery, and log of messages). In [6] is performed a comparison of three methods fault tolerance methods based on checkpoint: checkpoint / restart, message logging and parallel recovery. Propose a analytical model to predict the behavior of these protocols in exscales. In [7] power consumption is examined at the component level during checkpoint operations, and explore the reduction of energy consumption, using dynamic frequency and voltage scaling during checkpoint operation with intensive input/output. In [8], use an analytical model to study the performance of techniques based on coordinated checkpoint and replication. They also propose an alternative model for replication to improve energy efficiency and response time compared to checkpoint / restart and traditional replication techniques.

Our work tries to predict the energy consumption of the remote checkpoint technique, based on the characterization of the computer system and application. After making the characterizations, the analytical model allows to predict the energy consumption of checkpoints for any size problem. Only one real energy measurement is performed on the characterization of the system, so once made, we can work with any new application without measuring the energy again. We believe that there is no previous studies of this nature.

3. Energy consumption prediction methodology

The checkpoints are limited by input and output operations, and specifically a remote checkpoint is intensive in the use of the network. In this case, when you want to save the state of a process that is running on a given node, the checkpoint data are stored in a diferent node. Depending on the software and hardware of the computer system that you are using, the CPU may have a high incidence in time and energy consumption of the checkpoint. This can occur on systems that require significant CPU resources for network transmissions. Therefore, it is necessary to study the impact of the change in clock frequency of the CPU involved in the transmission of data over the network. Another important factor to analyze is the checkpoint size, which clearly impacts the transmission time of checkpoint data to the remote node.

Our methodology does not consider energy consumption of the node performing the checkpoint writing in secondary storage (hard disk), but

focuses on cluster nodes running processes that we want to preserve their state. Furthermore, power measurements are performed to full node, including the energy source.

To predict how much energy the different applications and their checkpoints consume, we have defined a methodology that consists of three phases. The first one, called *system characterization phase*, is necesary to have a representative equation of the energy behavior of the system running checkpoints. The second, called *characterization of the application phase*, seeks an equation that represents the temporal behavior of the checkpoints for that application. In the third phase, called *instantiation of the energy model phase*, the analytical model is constructed to predict energy consumption of the application and of the fault tolerance mechanism.

3.1. System caracterization phase

The objective of this phase is to obtain representative energetic attributes of the computer system, considering both hardware and software, relative to the average power demanded by it to implement the remote checkpoint. It is necessary to know the average power to make it possible to calculate the energy consumed by the system to run the checkpoint. Fig. 1 shows the power demand over time to make checkpoints of the same application at two different frequencies of CPU, the first interval between 97.63 and 150.35 is at low frequency, while between 201.822 and 253.42 is at high frequency. You can see at a glance how the power is increased when the CPU frequency is increased. In turn, the average power of checkpoints does not depend on a particular application or size of the checkpoints. Table 1 shows some measurements of average power of checkpoints execution of different applications (NAS parallel benchmarks) and data sizes at the same CPU frequency. The results shows that the average power is just 1.9% larger than the lower. Thus, as the average power required for the execution of checkpoint only depends on the CPU frequency, we will focus on finding an equation to determine the average power as a function of the frequency of CPU.

The methodology for the average power equation comprises two stages. The first one is to repeat a series of checkpoints (for some application) for different frequencies of CPU, and register potential samples. Finally power samples are averaged for each frequency. Then a regression analysis is performed in order to find the equation representing the average power as a function of the frequency of CPU. Figure[fig:Regresion] shows average power samples for three frequencies of CPU, and a polynomial regression of the second degree (representing the average power), $PowerCheck(f)=76.806-11.5988*f+5.28026*f^2$, which fits exactly the three samples. This equation is then used to instantiate the analytical model and to predict the energy consumption of the application with its mechanism of fault tolerance.

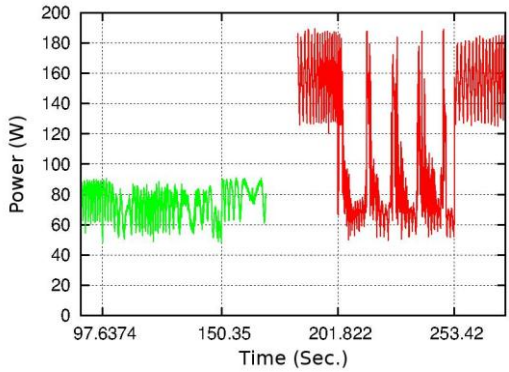


Fig. 1. Power consumption of a checkpoint at diferent CPU frequencies.

Table 1. Power consumption of a checkpoint at dierent data size.

Application	Checkpoint Size (MB)	Average Power per node (W)
LU clase B	106.81	84.84
CG clase C	367.48	83.52
IS clase C	540.09	85.11

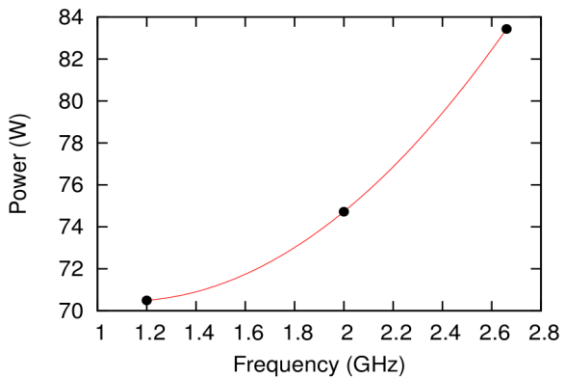


Fig. 2. Power regression.

3.2. Application caraterizations phase

Besides knowing the average power demanded by the node when running a checkpoint, to make it possible to calculate the energy consumed by the system, also you need to know the time it takes to execute the checkpoint.

The execution time of an application checkpoint depends on two variables, the CPU frequency and size of the problem. Table 2 (a) shows the execution time of a checkpoint corresponding to the same application and of problem size (CG class C of NAS), to different CPU frequency. It can be seen how the runtime decreases as the CPU frequency is increased. As runtime varies then confirm that it depends on the frequency of the CPU. In the table 2 (b) the execution time of a checkpoint for an application is displayed (heat transfer simulation), keeping fixed the CPU frequency and varying the size of the problem (size of a grid). In this case we can appreciate the high incidence of problem size. Then, we will focus to find an equation to determine the checkpoints runtimes depending on the CPU frequency and the problem size.

Tabla 2. Influences of variables on execution time of checkpoints.

(a) CPU Frequency.

Frequency (GHz)	Execution Time (Sec)
1.2	36.7
2	35.8
2.67	35.4

(b) Problem Size.

Size	Execution Time (Sec)
1000x1000	2.34
5000x5000	17.56
10000x10000	61.37

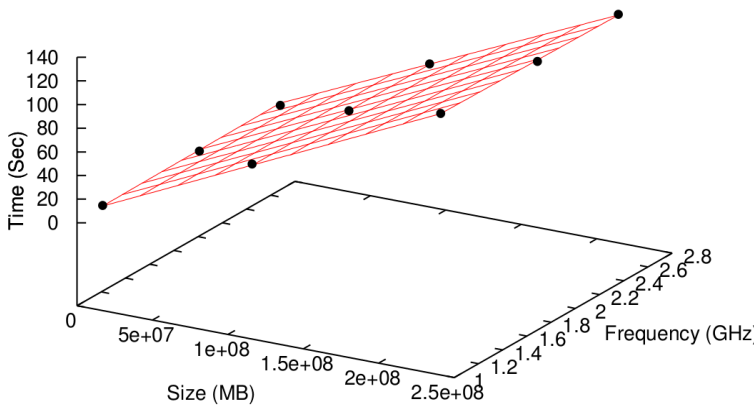


Fig. 3. Execution time regression of checkpoints.

Similarly to the power analysis, the methodology for the runtime equation of a checkpoint comprises two stages. The first one consists in, for diferent CPU frequencies and problem sizes of an application, repeat a series of checkpoints and record the times it takes to execute them. Averaging times, the execution time of each checkpoint is obtained to every combination of CPU frequency and size of problem. You must select at least three frequencies of CPU (high, medium and low) and three sizes of problems (large, medium and small), obtaining in this case a total of nine samples; however, the number of samples required may vary depending on the application. Next, a regression analysis is performed in order to find the equation for the checkpoint execution time depending on the CPU frequency and size of the problem. Fig. 3 shows observed checkpoint runtime samples for different frequencies of CPU and data sizes, and a first-degree polynomial regression (representing the runtime) with a residual variance of 1.6, and whose equation is:

$$TimeCheck(tam, f) = 1 + 6{,}05086 * 10^{-7} * tam + f - 1{,}16005 * 10^{-8} * tam * f$$

This equation is then used to instantiate the analytical model and predict power consumption of the application with its method of fault tolerance.

3.3 Instantiation of the energy model phase

Having characterized the system and the application, the next step is to instanciate the model with the equations obtained through characterization. Once instantiated the model it can predict the energy consumed by the application with its checkpoints, at any CPU frequency and problem size. Next, the analytical model is described in detail. The analytical model defines Equation 1 which seeks to determine the total energy that consume the application execution with its fault tolerance mechanism. The equation considers the energy consumed by the application computation itself E_{app} and energy consumption caused by the execution of the checkpoints E_{checks}.

$$E_{total} = E_{app} + E_{checks} \quad (1)$$

E_{app} is an input parameter of the model; in particular, for type SPMD (Single Program Multiple Data), E_{app} can be obtained applying the methodology proposed in [2]. Equation 2 defines E_{checks}, which depends on the number of nodes *nNodes*, the runtime of the application T_{app}, the time interval between checkpoints *CheckInt* and the energy consumption of each checkpoint E_{check}.

$$E_{checks} = \sum_{j=1}^{nNodes} \sum_{i=1}^{T_{app}/CheckInt} E_{check_{i,j}} \quad (2)$$

CheckInt and *Tapp* are input parameters of the model. In particular, for SPMD type of application, T_{app} can be obtained by applying the methodology proposed in [2]. $E\text{check}_{i,j}$ is the energy consumed by the i-th checkpoint at node j, which is defined in Equation 3, and depends on the time it takes the checkpoint execution T_{check} and the average power *PCheck(f)* demanded by the computer system for executing a checkpoint at CPU frequency *f*.

$$E_{check} = T_{check} * PowerCheck(f) \quad (3)$$

The equation defining *PowerCheck(f)* is obtained from the system power characterization, made in the characterization phase of the system. The equation defining T_{check} is obtained from the characterization of checkpoint execution time made in the application characterization stage (ie, the equation *TimeCheck(tam,f))*. Note that *PowerCheck(f)* is system dependent but independent of the application, while T_{check} is dependent on both the system and the application.

4. Experimental results

This section describes the hardware and software platform used, power measurement methodology, and finally the experimental validation of the energy prediction methodology is presented.

4.1 Experimental computing platform (hardware and software)

The experiments were performed on a cluster of PCs, with a 100 Mbps Ethernet network, and nodes consist of a main memory of 4GB, a hard disk SATA 500GB 7200 rpm, and a processor Intel Core i5-750, with a frequency range of 1.2 GHz to 2.66 GHz (without using the Intel Turbo Boost mechanism), four cores (without multithreading), and 8MB of cache. The nodes use the Debian GNU / Linux wheezy / sid operating system, OpenMPI version 1.4.5 as the MPI message passing library, and the checkpoint tool DMTCP [1] (Distributed MultiThreaded Checkpointing) version 1.2.5. The selected applications for system characterization are part of the series of parallel NAS benchmarks while the parallel application of interest for prediction of energy consumption is a heat transfer simulator usign MPI. The network file system, used for remote checkpointdata writing is NFS (Network File System).

4.2 Power measurement methodology

To realize the power measurements we use the PicoScope 2203, TA041 active differential probe, and the PP264 current clamp 60 A AC / DC, all Pico Technology products. The electrical signals captured by the two channels oscilloscope are transmitted in real time to a notebook through a USB connectio. The tension is measured using the TA041 probe that connects to an oscilloscope input channel. The electricity of the phase conductor, that supply the energy of the complete node (including power supply), is measured using the PP264 current clamp that connects to another input channel of the oscilloscope. Then, the power is calculated as the

product of voltage and electricity measurements. The sampling rate used for both channels of the oscilloscope was set at 1000 Hz.

4.3 Experimental validation of the methodology

To validate our method, we define a two compute nodes scenario (cluster) where the heat transfer simulatior is executed, and two dedicated nodes to input and output, where checkpoints data are written. Power measurements are performed on one of the two nodes running the parallelapplication. For simplicity, we present a case where a single checkpoint to the application is performed.

First, the system characterization phase is realized, comprising the power analysis while executing the checkpoints. In this case, we take the equation presented in section 3.1, since it corresponds exactly to this experimental computing platform. The NAS parallel benchmarks used for this phase were CG, IS and LU. Then, the application characterization phase is performed, comprising the analysis of checkpoints runtimes for the heat transfer simulation application. Finally, we performs the energy model instantiation phase, where the equations obtained in the two previous phases are added to the model.

In the table 3 it is shown, for one of the nodes, actual and predicted results of time and power, for two particular sizes of the problem and a specific CPU frequency (other than those used for system characterization and application). In the table 4 it is shown for a single node, the actual, predicted and error commited of energy consumption, for checkpoint alone and the application with the checkpoint (which not including the checkpoint are around 54 seconds). The precision error for a checkpoint is only 4.8%, which naturally decreases when considering the complete application.

Table 3. Prediction of execution time and power.

Tamaño	Frequency (GHz)	Actual Time (Sec)	Estimated Time (Sec)	Actual Power (W)	Estimated Power (W)
6.000x6.000	2.399	23.47	24.18	77.94	79.36
12.000x12.000	1.599	88.26	87.06	72.98	71.76

Table 4. Accuracy of energy consumption prediction.

Size	Checkpoint			Application with checkpoint		
	Actual E. (Joules)	Est. E. (Joules)	Error (%)	Actual E. (Joules)	Est. E. (Joules)	Error (%)
6.000x6.000	1829.77	1918.92	4.8	7.795.39	7.710.816	1.08
12.000x12.000	6441.19	6247.42	3	11.682.11	11.477.56	1.75

5. Future work and conclusions

In this paper we present a methodology to predict the energy consumption of a system of high performance computing, in a cluster, running an application with remote coordinated checkpoints. The methodology starts with an energy system characterization. Whenever you have a new application, a temporal (non-energy) characterization of checkpoints is made. With these data the analytical model to predict the energy consumption is instantiated, for any problem size and CPU frequency. The methodology is relatively simple to apply, and experimental results with a heat transfer application shows a prediction accuracy greater than 95%. We believe this work can be useful to the high performance computing community in various ways. In particular, to determine the energy impact of the checkpoint when an application scales up and the problem size at each node is reduced. It could also be used for an energy forecasting at different CPU frequencies and then determine which one is more convenient in terms of time, power and energy consumption. Future work will focus on expanding the experimental evidence, supplement the model with the application restart after a failure, cover more methods of fault tolerance, consider the power consumption of the input and output nodes that store the checkpoints, and integrate the model presented here to a model for SPMD applications prediction energy.

References

1. Ansel, J., Arya, K., Cooperman, G.: DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop. In: Proceedings of the 2009 IEEE International Symposium on Parallel&Distributed Processing. pp. 1-12. IPDPS '09, IEEE Computer Society, Washington, DC, USA (2009)
2. Balladini, J., Muresano, R., Suppi, R., Rexachs, D., Luque, E.: Methodology for predicting the energy consumption of SPMD application on virtualized environments. Computer Science & Technology 13(1), 130-136 (Dec 2013).
3. Bergman, K., et al.: ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems (2008).
4. Dongarra, J., et al.: The international exascale software project roadmap. Int. J. High Perform. Comput. Appl. 25(1), 3-60 (Feb 2011).
5. El Mehdi Diouri, M., Gluck, O., Lefevre, L., Cappello, F.: Energy considerations in checkpointing and fault tolerance protocols. In: Dependable Systems and Networks Workshops (DSN-W), 2012 IEEE/IFIP 42nd International Conference on. pp. 1-6 (June 2012).
6. Meneses, E., Sarood, O., Kale, L.V.: Assessing energy e-ciency of fault tolerance protocols for hpc systems. In: Proceedings of the 2012 IEEE 24th International Symposium on Computer Architecture and High Performance Computing. pp. 35-42. SBAC-PAD '12, IEEE Computer Society, Washington, DC, USA (2012).
7. Mills, B., Grant, R.E., Ferreira, K.B., Riesen, R.: Evaluating energy savings for checkpoint/restart. In: Proceedings of the 1st International Workshop on Energy E-cient Supercomputing. pp. 6:1-6:8. E2SC '13, ACM, New York, NY, USA (2013).
8. Mills, B., Znati, T., Melhem, R., Ferreira, K.B., Grant, R.E.: Energy consumption of resilience mechanisms in large scale systems. In: Proceedings of the 2014 22Nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing. pp. 528-535. PDP '14, IEEE Computer Society, Washington, DC, USA (2014).

Performance tuning of the HDA* algorithm for multicore machines

VICTORIA SANZ[1,2], ARMANDO DE GIUSTI[1,2], MARCELO NAIOUF[1]

[1]III-LIDI, Facultad de Informática, UNLP, La Plata, Buenos Aires, Argentina. [2]CONICET.
{vsanz, degiusti, mnaiouf}@lidi.info.unlp.edu.ar

Abstract. This paper analyzes the performance achieved by an own version of HDA* (Hash Distributed A*) for shared-memory architectures, a parallel algorithm which allows finding solutions to combinatorial optimization problems. This algorithm was implemented using *Pthreads*, the *Jemalloc* dynamic memory allocator, and the *15-Puzzle* as study case. Experimental work is focused on assessing the deviation of execution times when the algorithm is run on a multicore machine and varying the values of its parameters, for different input instances and number of cores/threads. Therefore, algorithm parameters are tuned in order to obtain a higher performance. Finally, algorithm scalability is studied using the parameters values that improved experimental performance for each instance and number of threads/cores.

Keywords: HDA*; Multicore; Performance tuning; Scalability; Combinatorial optimization problems.

1. Introduction

In the area of Artificial Intelligence, heuristic search algorithms are used as the basis to solve combinatorial optimization problems that require finding a sequence of actions that minimize a goal function and allow transforming an initial configuration (which represents the problem to be solved) into a final configuration (which represents the solution).
One of the most widely used search algorithms for that purpose is known as Best-First Search (*BFS*) [1], which browses the graph that represents the state space of the problem using a cost function $\hat{f}$ to value the nodes, which is partly composed by heuristic information that will guide the search faster towards the solution and will reduce the number of nodes to be considered. The algorithm is different from conventional methods because the search tree is implicit and dynamically generated, i.e. nodes are created as the search progresses. During the process, it keeps two data structures: one for the unexplored nodes sorted by function $\hat{f}$ (*open list*), and another for the already explored nodes (*closed list*) used to avoid processing the same state multiple times. In each iteration, the most promising node available on the open list is removed (according to function $\hat{f}$), it is added to the closed list, and legal actions are applied to it to generate successor nodes that will be added to the open list under certain conditions (verification step known as *duplicate*

detection). The search process continues until the node that represents the solution is removed from the open list.
The A* algorithm [2] is one of the most commonly used *BFS* variants because it guarantees finding optimal cost solutions. To that end, function $\hat{f}$ contains known cost information of the path from the initial node to the current node, as well as heuristic information to estimate the unknown cost of the path that goes from the current node to the solution node, which can never overestimate the actual cost. Thus, the search is guided to process the most promising paths first.
On the other hand, over the last years the development of parallel heuristic search algorithms has been promoted because the high requirement of computing power and memory, as a consequence of the exponential or factorial growth of the state space of the problem, makes its resolution on a single-core processor difficult. Moreover, it is common to find multi-core machines today, so sequential applications should be adapted to take advantage of the computing power that this architecture provides.
So far, different authors have presented various techniques to parallelize *BFS* algorithms, which vary in the method used to manage the open and closed lists and the strategy used to balance load among processors during the execution. The technique to be chosen will depend on the architecture and the problem to solve [3].
The simplest way to implement a parallel *BFS* algorithm on a shared-memory architecture is to keep only one open-list and only one closed-list shared by all the threads (*centralized strategy*). Therefore, thread synchronization is needed to ensure data structure consistency, and thus will limit performance [3][4]. Although the open and closed lists can be implemented through data structures that allow concurrent access to different portions in order to reduce resource contention, several authors have shown that this technique yields improvements only for problems with high heuristic computation time [3], and especially current studies have shown that it does not get a competitive performance on multi-core machines [5].
In order to solve the previous problem, each process/thread is equipped with its own local open and closed lists (*decentralized strategy*) and performs a quasi-independent search. This strategy is suitable both for shared-memory and distributed-memory architectures. However, communication among the processes/threads (processors) is needed due to the following reasons:

- Since at the beginning only one processor has the initial node on its open list and the search tree is generated at run time, the workload should be distributed dynamically.
- The nodes located on the processor's open list might not be the global best ones, so it will be necessary to equalize the nodes between processors.
- Duplicate nodes (nodes representing the same state) can be generated by different processors. If the *duplicate detection procedure* is only performed by the processor that has generated the node and/or by that which has received the node owing to load balancing, the detection and pruning of duplicate nodes will be *partial* because another processor may have a node representing the same state on its open or closed lists. Instead, if *absolute* detection and pruning are required, strategies that assign each state to a particular processor will be needed.

- The termination criterion should be modified because, if the search is ended when a solution is found, there would be no guarantee that such solution is the best one. This is because there are multiple inconsistent open lists and, as a consequence of dynamic load balancing, there may be some graph nodes that are being communicated among processors.
- The costs of the partial solutions found should be communicated in order to use them to prune the paths that lead to suboptimal cost solutions.

In this sense, the Hash Distributed A* (*HDA**) algorithm [6] parallelizes A* by applying a *decentralized strategy* and using *Zobrist's hash function* [7] to assign each state of the problem to a single processor. Thus, when a processor generates a node that does not belong to it, it determines who the owner is and transfers the node to that owner. This mechanism allows balancing the workload, leveling node quality, and pruning duplicates in an *absolute* way, as the nodes representing the same state are always sent to the same processor. The implementation of HDA* proposed by these authors uses only the MPI message passing library and asynchronous communication, so the algorithm is suitable for execution both on distributed-memory and shared-memory architectures.
On the other hand, the research carried out by [5] presents an implementation of HDA* for multicore machines, developed using the *Pthreads* library; in this way, it is possible to eliminate some inefficiencies that arise when the original HDA* algorithm is run on a shared-memory machine. Additionally, the *Jemalloc* library [8] was used to avoid performance degradation due to contention in the access to the data structures managed by the dynamic memory allocator, caused by the frequent alloc/free operations. Also, the authors present the *PBNF* (Parallel Best-*N* Block-First) algorithm that allows threads to work during synchronization free periods. Although a better performance is obtained with *PBNF*, the algorithm is complex and does not use the same implementation approach as A*, so *high superlinear speedup* is achieved in some cases.
Furthermore, in article [9] we presented the implementation of our own version of HDA* for shared-memory architectures. The algorithm is based on the one proposed in [5], with the following differences:

- The termination detection procedure is carried out in a decentralized way. To that end, an adaptation of the algorithm proposed by *Dijkstra and Safra* [10] is used.
- The algorithm incorporates the LNPI (Limit of Nodes per Iteration) parameter that indicates the maximum number of nodes to be processed per iteration of the algorithm. In this sense, our version is different from the original algorithm, which expands a single node per iteration.
- Threads accumulate a customizable quantity of nodes (*LNPT* or Limit of Nodes per Transfer) addressed to another thread before attempting their transfer, i.e. there are no transfers after each node generation as in the original algorithm.
- The *Memory Pool* technique is used to avoid performance degradation caused by alloc-free operations in a producer-consumer relation among different threads.

The effectiveness of the *Memory Pool* technique was confirmed and the algorithm scalability was analyzed on a multicore machine, using a fixed value of LNPT and the value of LNPI that improved experimental performance.

2. Contribution

This article presents an analysis of the performance achieved by the HDA* algorithm suitable for shared-memory architectures, which was presented in [9], when it is run on a machine with multicore processors. The experimental work was carried out using *15-Puzzle* [11] instances of various complexity levels, varying the number of threads/cores used and the values of the LNPI and LNPT parameters. Thus, the benefit obtained when incorporating the LNPI and LNPT parameters is documented.
Also, an analysis of the speedup and efficiency achieved as the workload and the number of threads/cores used increase is presented, using the values of LNPI and LNPT that improved experimental performance for each input instance.

3. Sequential A* algorithm

A* [2] is a variation of the *Best-First Search* technique, where each node *n* is valuated in accordance to the cost of reaching it from the root of the search tree $\hat{g}(n)$ and a heuristic that estimates the cost to go from *n* to a *solution node* $\hat{h}(n)$. Thus, the cost function will be $\hat{f}(n)= \hat{g}(n)+ \hat{h}(n)$. If the heuristic is *admissible* (i.e., it never overestimates the real cost), A* will always find an optimal solution.
The algorithm keeps a data structure with unexplored nodes (*open list*), sorted by function $\hat{f}$, and another data structure with already explored nodes (*closed list*), used to avoid processing the same state several times. Initially, the open list contains only one element, the initial node, and the closed list is empty.
In each step of the algorithm, the node with the lowest $\hat{f}$-value (the most promising node) is removed from the open list and examined. If the node is the solution, the algorithm ends. Otherwise, the node is expanded (generating successor nodes by applying legal moves) and added to the closed list. Each successor node is added to the open list if it is not found on either list, or if it is but its cost value improves that of the previous node (this verification is known as *duplicate detection*).
Once the node that represents the final state has been found, the sequence of actions applied to get to it can be obtained by tracing backwards the optimal path found.

4. HDA* algorithm for shared memory architectures

The HDA* algorithm proposed in [5] was developed using *Pthreads* and *Jemalloc*, and is based on the following:

- Each thread has: its own open and closed lists; an input queue known globally where the rest of the threads will deposit nodes that must be processed by this thread, and which must be protected by a lock to keep its consistency; a local output queue for each peer thread, which does not need to be protected since it will be for thread's own use to prevent it from waiting.
- When a thread t_i generates a node that belongs to another thread t_j, it must be communicated by adding it to t_j's input queue at some point. In order to do this, the thread tries to take the lock associated to t_j's input queue. When the lock is obtained immediately, node transfer is done by copying the pointer, and then the lock is released (this enables subsequent accesses to the queue by other threads). Otherwise, the pointer is added to the local output queue for t_j (there is no waiting time associated with this operation).
- After thread t_i carries out a certain number of node expansions from its open list:

- For each non-empty local *output queue*, the thread tries to communicate the stored nodes to its owner thread. In order to do this, the thread tries to take the lock associated to the input queue of the corresponding thread. If the lock is obtained, all the pointers to node are transferred, leaving the local output queue empty. Otherwise, it is not forced to wait.
- The thread tries to consume the nodes left by other threads on its own input queue. To do this, the thread must take the lock but it is only forced to wait if its open list is empty (in this case, it does not have any nodes to keep on working).

The allocation of states to threads is done through the *Zobrist Function*. The input and output queues are implemented as dynamic arrays that contain pointers to node.

5. Implementations

5.1 Sequential A*

The data structure selected to implement the open list is a *MinHeap* [12] whose content is indexed by an *Extensible Hash Table* [13]. This data structure allows nodes to be sorted according to the value of function $\hat{f}$, so that the operations of inserting a node, removing the node with the lowest $\hat{f}$-value and decreasing the priority of a node can be carried out in logarithmic order; at the same time, it allows carrying out searches in constant order to determine the existence of a node representing a particular state. Additionally, an *Extensible Hash Table* [13] was used to implement the closed list, which allows the operations of insertion/removal of nodes and searching to determine the existence of a node representing a particular state to be carried out in constant order.
The keys associated to the elements (nodes) stored in the *Hash Tables* are obtained by calculating the *Zobrist Function* [7] over the representation of the state.

5.2 HDA* for shared-memory architectures

An own version of the HDA* algorithm for shared-memory architectures, similar to the one studied in Section 4, was implemented. To that end, the *Pthread* library was used and the *Zobrist Function* was used to assign states to threads.

Each *thread* performs an A* search locally, keeping its local open and closed lists. Node communication strategy is based on the use of input and output queues.

To avoid making a single thread responsible for detecting termination by checking the state of the other threads and the state of their input queues, an adaptation of the *Dijkstra and Safra's* termination detection algorithm was carried out, allowing all threads to cooperate with such purpose. To that end, each thread keeps a *state* and a *counter* of sent and received *nodes* - instead of the number of "communications" that were done[1]. The *termination token* is represented by a shared variable with the following information: a *counter* of nodes in transit, a *state*, and the *identifier* of the thread that *owns* the token at the moment; these data are not protected since only one thread will be able to modify them at a given point in time. The end of the computation is communicated through a shared variable *end*.

With the aim of allowing the pruning of nodes that would lead to suboptimal solutions, threads share a pointer to the best global solution found so far (*best_solution*) and its cost (*best_solution_cost*). Both variables must be protected, since two threads can find two different solutions and try to update these values at the same time.

Even though the code is the same for all threads, thread 0 will be in charge of: generating the initial node and adding it to the input queue of its owner thread; initializing the common structures; verifying the termination state; and recovering from the shared memory the steps sequence that represents the solution to the problem once computation is finished.

Each thread carries out a series of iterations until it detects the end of the computation (through a change in the value of the variable *end*). In each iteration, the following stages are performed:

- *Node consumption stage*: the thread checks whether its own input queue is not empty. In that case, it tries to take the lock associated to the queue. When it obtains the access immediately, it takes all the pointers to nodes that were deposited on the queue, releases the lock, and then for each node whose cost is lower than *best_solution_cost* it performs the *duplicate detection process* adding the node to the open list as appropriate.
- *Processing stage*: the thread expands at most *LNPI* (*Limit of Nodes per Iteration*) nodes from its open list. For each extracted node, it verifies if its cost is at least *best_solution_cost*. If so, the thread empties the open list since the nodes stored in it would lead to suboptimal solutions. Otherwise, it checks if the node represents the solution and in that case it empties the open list and updates *best_solution* and *best_solution_cost,* after having taken the lock that protects them and having consulted again

[1] The input queues count the total amount of stored nodes (logical dimension). Consequently, the amount of nodes in transit is calculated instead of the amount of "deposits" that have not been received yet. This is a modification of the *Dijskstra and Safra* algorithm.

if the node cost is less than *best_solution_cost*[2]. When the extracted node is not the solution, it is added to the closed list, it is expanded (generating successors), and then, for each successor, the *Zobrist Function* is calculated to determine which thread has to process it. When the node belongs to this very thread, it carries out the duplicate detection and adds the node to its open list as appropriate. Otherwise, the thread stores the node in the local output queue for the destination thread; when the amount of stored nodes is higher than the limit *LNPT* (*Limit of Nodes per Transference*), it tries to take the lock associated to the destination thread's input queue and, if it obtains the lock immediately, it transfers the stored nodes leaving the output queue empty[3]. A node transfer involves a pointer copy. If this thread is the first to deposit on that input queue (i.e., the input queue was empty), it must notify the destination thread, just in case the latter was idle waiting for work.

- *Idle stage*: the thread goes into this stage when its open list is empty. Firstly, it transfers the nodes stored on each non-empty output queue. Then, it remains waiting for any of the following events until having received work or having detected the end of computation:

  - *End of computation:* thread 0 detected the *termination state* and changed the value of the *end* variable.
  - *Termination token arrival*: the thread must update the *termination token*, based on the termination detection algorithm, and pass it to the following thread, which involves changing the *owner* field in order to notify the successor thread that it is the *new owner*. Instead, thread 0 performs a different task which consists in verifying if the termination conditions are given; if so, it changes the value of the *end* variable; otherwise, it starts a new round to detect termination.
  - *Work deposit notification*: the thread must obtain the lock associated to its input queue, it must take all the pointers to nodes that were deposited on the queue and release the lock. For each node whose cost is lower than *best_solution_cost* the *duplicate detection* is carried out, adding the node to the open list as appropriate.

The termination detection algorithm involves updating the thread-local variables *state* and *counter* every time it adds nodes to the input queue of another thread or every time it removes nodes from its own input queue, either increasing or decreasing the local *counter* by the amount of nodes that were deposited or received, respectively.

In order to prevent performance degradation due to an *alloc-free* relation between threads, a *pool* of pointers to node (*Memory Pool*) was incorporated to each thread, where the pointers to node that the thread wishes to "set free" for a further use are stored. This technique prevents thread A from accessing the structures assigned to another thread B by the dynamic memory allocator, when the former wants to "free" a pointer allocated by the latter, situation that would produce contention.

---

2 This is necessary because two threads could find two solutions with different cost and attempt to update the shared data at the same time. If the condition about the cost is not verified again, a suboptimal solution could be stored.

3 Our version is different from the one proposed by Burns, in which each time a thread t_i generates a node that belongs to another thread t_j, the former tries to take the lock associated to the latter's input queue in order to transfer the node individually.

6. Experimental results

Experimental tests were carried out on a machine with two Intel ® Xeon ® E5620 processors and 32GB RAM (DDR3 1066Mhz). Each processor has *four* 2.4Ghz physical *cores*; each *core* has two 64KB L1 caches for data and instructions, respectively, and one 256KB L2 cache; all processor cores share a 12MB L3 cache. Each processor has a memory controller, so the machine has NUMA architecture and uses a *QuickPath Interconnect (QPI)* interconnection of 5.86 GT/s.

The tests considered sixteen 15-Puzzle instances used in [14] - numbered 3, 15, 17, 21, 26, 32, 33, 49, 53, 56, 59, 60, 66, 82, 88, 100; six of the 10 configurations that are part of the test suite proposed by [15], numbered 101-106 in this article; and the final instance proposed by [14]. Thus, the 22 configurations used present different levels of complexity, which is measured in terms of the number of nodes processed by Sequential A*, and varies between 1 and 103 millions of processed nodes.

The sequential A* and HDA* algorithms were configured to use the *Jemalloc* dynamic memory allocator (with 256 arenas) and the heuristic H4. The HDA* algorithm was compiled to use the *Memory Pool* technique and *active waiting*. We showed in [9] that these techniques improve the performance of both algorithms.

The sequential A* is deterministic. For each initial configuration, 10 runs were performed and the *average runtime* in seconds was calculated.

On the other hand, HDA* is non-deterministic. For each initial configuration and set of parameters, 100 tests were carried out. The parameters and values used were: number of cores/threads (4, 8); LNPI (1, 5, 50, 500); and LNPT (26, 210, 1680) [4]. The data obtained with these 100 runs were then averaged for each configuration and set of parameters, which will be referred to as *mean sample*. Each thread was bound to an exclusive core using the function *sched_setaffinity();* in those tests with 4 threads, 1 pair of threads was mapped to each machine processor, and in those tests with 8 threads, 1 thread was mapped to each physical core of the machine.

6.1 Impact of the LNPI parameter on performance

The LNPI parameter determines how many nodes a thread must expand per algorithm iteration, i.e., it establishes the amount of nodes to be processed by the thread before it tries to consume nodes deposited on its own input queue again.

To analyze the impact of this parameter on execution time, all *mean samples* were taken and those with the same initial configuration, number of cores and LNPT were grouped. That is, each group contained the *mean samples* whose only difference among them was the LNPI parameter value. For each group, the Coefficient of Variation (CV) was calculated from the Standard Deviation (SD) of the execution time, since it measures how much the execution time of a *mean sample* in the group tends to deviate from the *group mean time*.

From the data obtained for LNPT=26, it was observed that the CV for the groups is between 0.001 and 0.06; this indicates that when varying the LNPI

[4] These values correspond to 1KB (26 nodes), 8KB (210 nodes) and 64KB (1680 nodes) of data transferred, respectively.

parameter between the values defined, *mean sample* execution time tends to deviate between 0.1% and 6% from its *group mean;* it should be noted that 95.5% of the groups has a CV value below 0.03. From the results obtained for LNPT=210, it was observed that the CV for the groups ranges from 0.0009 to 0.036; 95.5% of the groups has a CV value below 0.03. Finally, tests that used LNPT=1680 showed a CV value between 0.001 and 0.062, 97.7% of the groups has a CV value below 0.03.
Based on the low CV values, it is concluded that there is a non-significant variation in execution times when changing the value of the LNPI parameter among those defined. This parameter determines the *frequency* at which a thread tries to consume nodes deposited on its input queue:

- As accesses to the input queue are *more frequent*, the quality of nodes stored on the thread's open list will be more updated. When a thread fails in its attempt to access its input queue, performance is not affected since the thread is not forced to wait.
- As accesses to the input queue are *less frequent* (higher LNPI), the thread will be forced to expand a higher amount of nodes stored on its open list *speculatively,* i.e. newly received nodes which might be among the global best ones are not added to the open list and thus the thread will only follow its local heuristic information. Consequently, execution time increases due to a growth of Search Overhead[5]. This hypothesis was confirmed by running tests for the 22 initial instances, using Memory Pool, active waiting, limiting LNPT to 26 and LNPI to 10000.

6.2 Impact of the LNPT parameter on performance.

The LNPT parameter indicates the number of nodes that the thread must accumulate on the respective output queue before attempting their transfer to the recipient thread.
In order to evaluate the benefits of using the LNPT parameter, the parallel algorithm was run limiting LNPT to 1. Then, the execution times of *mean samples* that used LNPT=1 were compared with those that used LNPT=26, taking into account the same initial configuration, LNPI value and number of cores. The results obtained show that 97.7% of the *mean samples* that used LNPT=26 present improvements in execution times, which range from 0.24% to 10.61%; the remaining samples that used LNPT=26 (2.3%) present a deterioration ranging from 0.39% to 6.13%. Previous results indicate the benefits of using the LNPT parameter.
For the purpose of analyzing the effect of the LNPT parameter on execution time, all *mean samples* obtained from runs limiting LNPT to 26, 210 and 1680 were taken. Then, all *mean samples* with identical configuration, number of cores and LNPI values were grouped; i.e., each group contained *mean samples* whose only difference was the value for their LNPT parameter. For each group, the Coefficient of Variation (CV) was calculated from the Standard Deviation (SD) of the execution time, since it measures

[5] The Search Overhead represents the percentage of nodes that the parallel algorithm expands in excess as compared to the sequential algorithm. It is calculated using the formula 100x(NP/NS - 1), where NP is the number of nodes processed by the parallel algorithm and NS is the number of nodes processed by the sequential algorithm.

how much the execution time of a *mean sample* in the group tends to deviate from the *group mean time*.
In general, the results obtained show that the CV for the groups is between 0.0035 and 0.26; this indicates that when varying the LNPT parameter between the values defined, *mean sample* execution time tends to deviate between 0.35% and 26% from its *group mean*; it should be noted that 70.45% of the groups has a CV value below 0.1, 27.8% has a CV value below 0.05 and only 10.2% has a CV value below 0.03.
Based on the moderately high CV values, it is concluded that the LNPT parameter affects execution time. This is because this parameter impacts the *frequency* at which threads try to acquire the locks associated to the input queues in order to transfer nodes. The following can be inferred:

- A *high frequency* (low LNPT) causes a higher contention in the access to locks associated to input queues. This, in turn, increases the number of failed attempts to access the thread's own input queue when it is performing the *node consumption stage.* Consequently, a significant increase in Search Overhead was noted because each thread performed a greater amount of *speculative expansions* on its local nodes.
- A *low frequency* (high LNPT) delays nodes transfer among threads, which in turn increases idle time, unbalances the quality of open nodes stored by threads and increases the amount of *speculative work* on local nodes and thus the Search Overhead.

In general, the value of LNPT that improves performance for each configuration and number of cores is 210 (there is an exception with configuration 102 and 4 cores, for which performance improved using LNPT=1680). Figure 1 shows the speedup obtained for each initial configuration, using 4 and 8 threads/cores and LNPT=210, while Figure 2 shows the efficiency achieved; the *mean sample* that minimized resolution time was selected for each configuration and number of cores, that is, the sample whose LNPI value resulted in the best performance. The various selected *mean samples* were sorted based on configuration complexity (that is, based on the number of nodes processed by A*, which is related to sequential execution time).
Tests with 4 cores obtained a speedup that ranges from 3.39 to 4.39, and efficiency between 0.85 and 1.099. Tests with 8 cores show a speedup between 5.17 and 8.42, and efficiency between 0.65 and 1.05.
The superlinear speedup observed in some cases is due to the fact that the parallel algorithm processed a lower number of nodes than the sequential algorithm (i.e., a negative *Search Overhead* was obtained). The remaining cases are due to the decrease in the number of elements in the multiple open-list and closed-list structures, which causes an acceleration of the operations carried out on them.

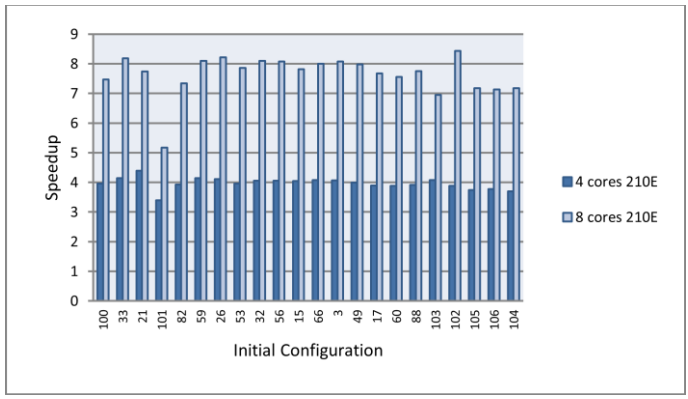


Figure 1. Speedup achieved by HDA*, LNPT =210.

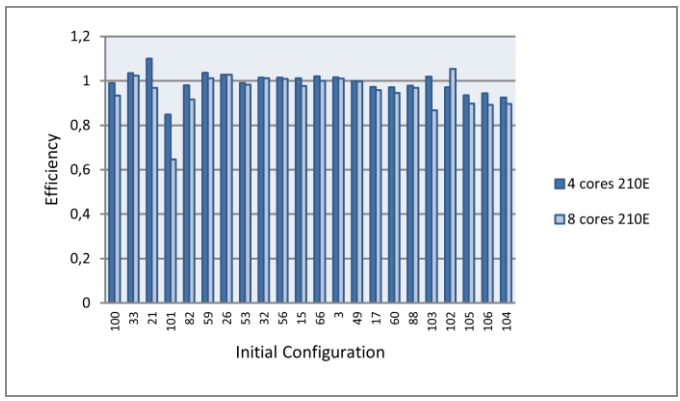


Figure 2. Efficiency achieved by HDA*, LNPT =210.

7. Conclusions and future lines of work

An own version of the HDA* algorithm suitable for shared-memory architectures was presented, and the effect of the LNPI and LNPT parameters on performance was analyzed. Additionally, algorithm scalability was assessed as the number of cores and workload increase, using the LNPT value that improved experimental performance. Future lines of work focus on contrasting the algorithm presented in this paper against an own version of HDA* implemented exclusively with MPI, comparing the performance achieved and the amount of memory used when they are run on a multicore machine.

References

1. S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd edn. New Jersey, USA. Prentice Hall, 2003.
2. P. Hart, N. Nilsson and Raphael B., "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 1, pp. 100-107, 1968.
3. V. Kumar, K. Ramesh and V. N. Rao, "Parallel Best-First Search of State-Space Graphs: A Summary of Results", *Proceedings of the 10th Nat. Conf. AI, AAAI*, 1988, pp. 122-127.
4. V. Cung and B. Le Cun, "An efficient implementation of parallel A*", *Proceedings of the First Canada-France Conference on Parallel and Distributed Computing*, Montréal, 1994, pp. 153-168.
5. E. Burns, S. Lemons, W. Ruml and R. Zhou, "Best-First Heuristic Search for Multicore Machines", *Journal of AI Research*, vol. 39, pp. 689-743, 2010.
6. A. Kishimoto, A. Fukunaga and A. Botea, "Evaluation of a simple, scalable, parallel best-first search strategy", *Artificial Intelligence*, pp. 222-248, 2013.
7. A. Zobrist, "A New Hashing Method with Application for Game Playing", Computer Sciences Department, University of Wisconsin, TR 88, 1968.
8. J. Evans, "A Scalable Concurrent malloc(3) Implementation for FreeBSD", in *Proceedings of the 3rd annual Technical BSD Conference*, Ottawa, 2006.
9. V. Sanz, A. De Giusti and M. Naiouf, "On the Optimization of HDA* for Multicore Machines. Performance Analysis", *Proceedings of the PDPTA*'14, Las Vegas, USA. pp. 625-631.
10. E. Dijkstra, "Shmuel Safra's version of termination detection EWD-Note 998", 1987.
11. D. Ratner and M. Warmuth, "The (n2-1)-puzzle and related relocation problems", *Journal of Symbolic Computation*, vol. 10, no. 2, pp. 111-137, 1990.
12. A. Aho, J. Ullman, and J. Hopcroft, *Data Structures and Algorithms*, 1st edn. Boston, MA, USA. Addison-Wesley Longman Publishing Co, 1983.
13. R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 2nd edn. McGraw-Hill Education, 1999.
14. R. Korf, "Depth-first Iterative-Deepening: An Optimal Admissible Tree Search", *Artificial Intelligence*, pp. 97-109, 1985.
15. A. Brüngger, *Solving Hard Combinatorial Optimization Problems in Parallel: Two Cases Studies*. Zurich, Switzerland, 1998.

XII

Information Technology Applied to Education Workshop

Determining the Profiles of Young People from Buenos Aires with a Tendency to Pursue Computer Science Studies

MARÍA EMILIA CHARNELLI[1], LAURA LANZARINI[2], GUILLERMO BALDINO[3], JAVIER DÍAZ[1]

[1] LINTI - Laboratorio de Investigación en Nuevas Tecnologías Informáticas
[2] III LIDI - Instituto de Investigación en Informática LIDI
Facultad de Informática, Universidad Nacional de La Plata
[3] LINSI - Laboratorio de Innovaciones en Sistemas de Información
Dpto de Sistemas - UTN Facultad Regional La Plata
mcharnelli@linti.unlp.edu.ar, laural@lidi.info.unlp.edu.ar, gbaldino@linsi.edu.ar, jdiaz@unlp.edu.ar

Abstract. Data Mining is a discipline related to the development and application of techniques for the extraction of new and useful information from large amounts of available data. The goal of this paper is to use these techniques on data extracted from a poll in order to identify the profiles of the young people showing interest in pursuing undergraduate studies in the field of computer science. The paper describes the process of identification of the most salient features for high school students in a wide age range. It also includes the data preprocessing stage, fundamental in the process, as it strongly influences the development of the model obtained. Finally, results and conclusions are presented, as well as future lines of work.

Keywords: Data Mining, Feature Selection, Classification and Prediction Techniques, Computer Science.

1. Introduction

Nowadays, owing to the advancement of technology, most processes have digital historical information large enough to make manual processing difficult.
Data Mining, one of the most important stages of the Knowledge Discovery in Databases process, gathers a set of techniques capable of modeling and summarizing these historical data, facilitating their understanding and aiding in future decision making processes.

This paper sets out to identify and select the most relevant features for establishing and defining the profile of young people showing interest in pursuing studies in the field of computer science. This study will allow every Academic Unit to recognize the aspects that attract the attention of their prospective students, which can be used to target the right audience, thus solving the two problems implied by wrong choice of studies: the frustration of discovering the course of studies does not fulfill expectations and the financial losses caused by wrongly allocating resources incurred in by students, institutions and the state.

2. Knowledge Discovery in Databases

This paper is framed in what is known as Knowledge Discovery in Databases or KDD, which consists of a series of phases that define the methodology used. The order of these phases is not strict and there may be variation, depending on the result of each phase, which may result in a cyclic process.

- **Phase 1. Domain understanding:** Before starting to work, it is necessary to understand the situation, determine the goals and design a work plan that will help us solve the problem.
- **Phase 2. Data collection and integration:** This phase starts with the obtention of the data, which are later studied and their origins identified. This stage features data collection, description, exploration and quality verification.
- **Phase 3. Data preparation**: It is necessary to select and prepare the subset of data to be used. This phase covers all the activities to build the final data set that will be used by the modeling techniques.
- **Phase 4. Modeling:** Also called Data Mining, because it is the most characteristic of KDD, it is the phase in which multiple modeling techniques are selected and applied, configuring their parameters for result obtention. Here is where new knowledge is produced, building models from the collected data.
- **Phase 5. Interpretation and evaluation:** The models obtained in the previous phase are interpreted and evaluated in order to check whether they fulfill the goals set in preliminary phases. Here, it is critical to determine whether important parts of reality have been considered sufficiently and to decide on the reuse of the DM process results
- **Phase 6. Result dissemination, use and measurement:** Knowledge acquired through model creation must be organized and presented in such a way as to be understood and used by the end user. The applicability of the model depends on this phase.

3. Description of the Problem

The information to be used comes from a survey done by the Sadosky Foundation in multiple high schools of the Province of Buenos Aires. The questions seek to collect the impressions students have in relation to computer science and to reveal why there are few female participants in software production processes; more specifically in the Software and IT sector.

The information gathered involves 627 young people between 13 and 22 years of age. The answers obtained have generated 236 different attributes or variables. Therefore, although the number of surveys was small, the amount of attributes composing each of them makes it difficult to identify patterns or relations existing in the opinions of different subjects. In relation to this, it is a good option to use objective techniques that allow for the identification of the most relevant attributes.

However, before applying techniques specific to Data Mining, it is necessary to verify information in order to avoid discrepancies. Modifications and transformations operated on the original data are described following:

- Attributes with inconsistent data

Attributes with an excessive amount of missing data were eliminated. Anomalous values resulting from loading errors and constant values such as language (all were done in Spanish) were cleaned. Redundant attributes were eliminated because they were the same value as another one or because they had the same value in all subjects.

- Attributes with non generalizable data

Non generalizable attributes such as student names were eliminated. The cardinality of some attributes was reduced by using more generic categories. For example, school names were replaced by the corresponding geographic area: zona Norte, Sur, Oeste y Matanza.

- Transformations

Some attributes were numerized and their range normalized, according to the requirements of the Data Mining techniques used. This type of transformations was applied to ordinal attributes such as ``hability to perform different activities'' or ``academic level of the head of the family''. In the case of questions with tabulated answers, a binary representation was used, composed by as many attributes as possible answers to the question. For example, this transformation was applied to questions such as ``Use you give to the PC'', ``Activity you would like to work in'', ``What do you do in your free time?'' and ``How do you learn to program?''.

- Data Mining applied to open questions

Open questions in the survey required special treatment as they cannot be processed directly using any Data Mining technique. Some examples of these questions were ``What is the first word that comes to mind to define a computer?'' or ``What is the first word that comes to mind to define a computer program?'', where answers included lists of very different words.

When it comes to operating with textual information, it is necessary to use Text Mining techniques in order to identify the terms most frequently used by the subjects. This was done using a process composed of multiple stages. One stage comprised the application of a stopword filter, which filters the words that match any stopword included in a file provided for this purpose. Then each word in the text was reduced to its root using a stemming algorithm 1. The importance of this process lies in that it eliminates the syntactic variations related to gender, number and tense. Once the root of each word was obtained, the frequency of incidence was calculated and the three most frequent words were chosen.
This resulted in the three most representative terms for each of the open questions, which summarized the answers of each subject as a sequence of words in a specific and representative category.

4. Feature Selection

Data Mining techniques applied to structured information composed by a large amount of features results in complex models. Depending on the technique used, data with a high dimension produce either very large trees or sets of rules with high cardinality and backgrounds formed by a great amount of conjunctions [2] or discriminating functions that are difficult to interpret.
In order to solve this problem, it is necessary to identify the most representative attributes of the information available before the model is constructed. Thus, the technique used will be simplified in its task and result in a simpler, easier to interpret model [3].
In the particular case of the problem in this article, selecting the features is fundamental, as the goal is to identify the most important matters to young people with a tendency to pursue computer science related studies. The answer sought is, no doubt, the result of a feature selection process [4].
The two main feature selection techniques are: filter methods and wrapper methods [5]. Filter methods are based on general features of the training set to select some without using any learning algorithm. Wrapper methods require a predefined learning algorithm to select, and use its performance to evaluate and determine which features will be selected [6].
Due to the high dimensionality of the data set provided by the Sadosky Foundation, a filtering method was used to select the features, called Chi2. This method, proposed by Liu et al. in [7], is one of the most used feature selection methods and is based on a statistical method for comparing proportions. The χ^2 metric is used to measure attribute performance, as it determines a value proportional to the relation existing between a class C and a feature f which can take r possible values.
Given a set of data D with n examples, the χ^2 metric is calculated using the following formula:

$$\chi^2(D,c,f) = \sum_{i=1}^{r} \frac{(n_{i_{pos}} - \mu_{i_{pos}})^2}{\mu_{i_{pos}}} + \frac{(n_{i_{neg}} - \mu_{i_{neg}})^2}{\mu_{i_{neg}}}$$

where $n_{i_{pos}}$ and $n_{i_{neg}}$ represent the amount of positive and negative examples for value i of feature f, respectively, and $\mu_{i_{pos}}$ and $\mu_{i_{neg}}$ are the expected values if the data had a uniform distribution.

The score obtained when evaluating $\chi^2(D,c,f)$ follows the distribution χ^2 and the goal of the selection algorithm is to simply choose a subset of features among those with the highest scores, as they will be the most relevant when discriminating the classes.

The following criteria were used in order to determine how many attributes were to be selected:

- **Criterion 1**: Selecting the attributes whose score was higher than the value of the mean plus one and a half standard deviations.
- **Criterion 2**: Selecting attributes evaluating the performance of a classifier as the features are incorporated, one at a time, in order of decreasing score.

A classifier is built for this purpose from the first feature and its performance is measured. This process is repeated for the remaining two characteristics with the highest score. The process continues this way, incorporating features one by one and evaluating the performance reached until no changes occur for a certain number of iterations [8].

For the classifier performance criterion, the amount of correct answers as to who chooses computer science was the focus, i.e., the performance on true positives. This is due to the marked imbalance among the classes, as only 10% of the subjects showed interest in computer science; therefore, the general accuracy of the classifier, considering true predictions in both classes, may be high even if it does not perform well on the class of interest.

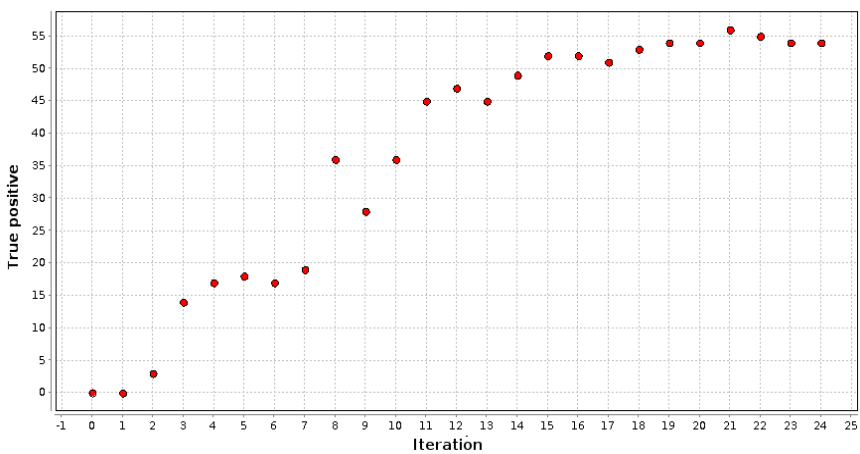


Fig. 1. Amount of true predictions in the True class of each iteration of Criterion 2

5. Results

Table1 shows the list of attributes selected by each of the criteria described in the previous section. Regarding Criterion 2, figure 1 shows a slow improvement in the classifier until it reaches a maximum with 22 features. The classifier used in Criterion 2 was a tree generated with the popular method C4.5 [9].
The aforementioned is summarized in the two columns of table 1, where Criterion 2 can be seen to include more attributes than Criterion 1. The difference between the two criteria is less than 3% of the original amount of attributes. The 22 attributes selected constitute 9.3% of the original attributes.
It is worth mentioning that the fact that an attribute appears in table 1 does not mean that the answer to the corresponding question was positive, but rather that the answer given by the subject helps determine whether they will choose to pursue computer science related studies.
Table 2 shows the confusion matrix of the tree obtained with the C4.5 method using the 22 features in table 1. Observing the structure, it is proved that the tree correctly predicts the 80% of the cases in which subjects have answered that they would choose to study computer science. This result shows that feature selection has been successful, since the tree built from the 236 attributes shows an equivalent true prediction rate.
Finally, with the goal of identifying relevant attributes on different groups of subjects, the sample set was divided using two different criteria: by gender and by whether the subjects were in their junior years or in their senior years. Each subset was applied the Chi2 feature selection method with criterion 2. Results are represented in table 3 where only features with at least 3 matches among the different data sets were included.
For example, the first row of table 3 shows women in their senior high school year do not show a strong tendency when answering the question related to the gender of a domestic worker.
Moreover, the attributes selected show the relevance of computer use patterns in discriminating the classes. Using the computer outside of social networks, as well as for games (soccer games or other, more complex ones) seems to be an indicator of a preference for computer science related studies. Also salient are an aptitude for assembling and disassembling things and installing and configuring programs. Some less evident yet relevant attributes for the classification are an aptitude to show and receive affection.

6. Conclusions and future lines of work

This paper identifies the 22 most relevant attributes of a survey by the Sadosky Foundation that help determine whether a person will choose to pursue computer science related studies. This reduces the total of questions by 90% as the original number of attributes was 236.
The preliminary results evidence the importance of showing young people the multiple potential functions of a computer, i.e., high schools should

encourage the creation of spaces and activities that take students closer to computers in unconventional ways in order to broaden their range of application, e.g, through music, image processing, robotics, etc.
Using games and configuring and managing software applications seem to be strong indicators of a tendency in students to choose computer science careers. In this sense, proposing workshops on these topics could increase an interest in the field.

Table 1. Characteristics selected by criteria 1 and 2 as relevant to determine whether a person will choose computer science related studies using the full data set.

Attribute	*Criterion 1*	*Criterion 2*
01. Most adequate gender for a domestic worker	X	X
02. Aptitude for showing affection	X	X
03. Aptitude for assembling and disassembling	X	X
04. Would you like to do informational work	X	X
05. Using the PC for social networks	X	X
06. Using the PC for soccer games and other complex games.	X	X
07. Using the PC to configure, investigate or update programs.	X	X
08. Use of the PC to play online games	X	X
09. ¿Do you know what a computer program is?	X	X
10. ¿Is programming about creating or inventing?	X	X
11. Computer Science courses by gender	X	X
12. ¿Does a new car need a program to operate?	X	X
13. ¿Are computer scientists like me?	X	X
14. Would you like to work assisting people	X	X
15. ¿Do you like assembling and disassembling things?	X	X
16. ¿Do you think computer scientists make money?	X	X
17. ¿Do you like to google?		X
18. ¿Do you like to show affection?		X
19. Would you like to work in the arts or show business		X
20. Would you like to work in a professional activity		X
21. ¿Is the salary of a scientist high?		X
22. ¿Does a lamp need a program to operate?		X

Table 2. Confusion matrix of the tree obtained from the C4.5 method using the 22 features in the table 1

	Chooses computer science	Does not choose computer science	Class precision
Predicts chooses computer science	56	6	90.32%
Predicts does not choose computer science	14	552	97.53%
Class recall	80.00%	98.92%	

Table 3. Common features selected by at least three data subsets

Attribute Selected	Full	Only women	Only men	First age	Last age
Domestic worker gender	X		X	X	
Aptitude for showing affection	X		X	X	X
Aptitude for assembling and disassembling	X		X	X	X
Activity you would like to work in: Informational work (not software)	X		X	X	X
Free time to use the computer not for social networks	X	X	X	X	X
Using the computer for soccer games and other complex games	X	X		X	X
Using the computer to configure investigate or update programs	X	X	X		X
¿Is programming about creating or inventing?	X		X		X
¿Does a new car need a program to operate?	X		X	X	
¿Like me?	X	X		X	
¿Does a lamp need a program to operate?	X	X		X	

This paper represents a first step in defining the features of computer science students. The obtained results will allow for a definition of the direction of future surveys that are similar to the one performed by the Sadosky Foundation. As a future line of work, other techniques will be used to generate models from the set of attributes selected and similar surveys will take place within the National University of La Plata.

References

1. Gupta, V., Lehal, G.S.: A survey of common stemming techniques and existing stemmers for indian languages. Journal of Emerging Technologies in Web Intelligence 5 (2013) 157-161.
2. Sebban, M., Nock, R., Chauchat, J.H., Rakotomalala, R.: Impact of learning set quality and size on decision tree performances. Int. Journal of Computers, Systems and Signals 1 (2000) 85-105.
3. Thrun, S.B., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., Jong, K.D. Dzeroski, S., Fahlman, S.E., Fisher, D., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R., Mitchell, T., Pachowicz, P., Reich, Y.,Vafaie, H., Welde, W.V.D., Wenzel, W., Wnek, J., Zhang, J.: The monk's problems a performance comparison of diferent learning algorithms. Technical report (1991).
4. Alelyani, S., Tang, J., Liu, H.: Feature selection for clustering: A review. (2013)
5. Das, S.: Filters, wrappers and a boosting-based hybrid for feature selection. In: Proceedings of the Eighteenth International Conference on Machine Learning. ICML 2001, Williamstown, MA, USA, Williams College (2001).
6. Song, Q., Ni, J.,Wang, G.: A fast clustering-based feature subset selection algorithm for high-dimensional data. Knowledge and Data Engineering, IEEE Transactions on 25 (2013) 1-14.
7. Liu, H., Setiono, R.: Chi2: Feature selection and discretization of numeric attributes. In: Proceedings of the Seventh International Conference on Tools with Artificial Intelligence. TAI '95, Washington, DC, USA, IEEE Computer Society (1995) p. 388.
8. O'Mahony, M.P., Cunningham, P., Smyth, B.: An assessment of machine learning techniques for review recommendation. In: Artificial Intelligence and Cognitive Science. Springer (2010) 241-250.
9. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993).

A Novel Authoring Tool for Augmented Books

NICOLÁS FERNANDO GAZCÓN[1, 2] AND SILVIA MABEL CASTRO[1]

[1] VyGLab, Department of Computer Science and Engineering,
Universidad Nacional del Sur, Av. Alem 1253,
8000 Bahía Blanca, Buenos Aires, Argentina
[2] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
Av. Rivadavia 1906, 1033 Buenos Aires, Argentina
{nfg,smc}@cs.uns.edu.ar
http://vyglab.cs.uns.edu.ar

Abstract. Augmented Reality (AR) has been studied from different application fields since the early 60s. Different research works in the educational field have been recently conducted, particularly based on a versatile AR application called Augmented Book. This paper focuses on a novel authoring tool for this kind of books. Our approach allows the possibility of adding virtual contents to any existing printed book. The application was implemented following a framework-based approach, making easy to extend with new features. Informal evaluations and observations by academics of different fields were conducted obtaining promising results.

Keywords: Augmented Book, Augmented Reality, Authoring Tool, Computer Graphics

1. Introduction

It has been more than fifty years since researchers addressed Augmented Reality (AR). Sutherland research work [18] is considered as one of the first contributions on this field. He introduced what he called the "*ultimate display*", however, due to technological limitations these features was mostly not adopted. In fact, both widely accepted definitions of AR were introduced in 1994 and 1997, by Milgram & Kishino [16] and Azuma [5] respectively.
The first definition introduces the *Reality-Virtuality Continuum*. As it is shown in Fig.1, it consists of a combination of real and virtual elements. It has at one end of the spectrum the Real Environment, representing our real world. At the other end of the spectrum is situated the Virtual Reality, representing a pure virtual world. The middle region of this continuum is considered the Mixed Reality, which combines both worlds. The AR is closer to the Real Environment (i.e. real world elements are predominant over the

virtual elements) in contrast to the Augmented Virtuality which is closer to the pure virtual world. On the other hand, Azuma's definition has been adopted by the research community as the basic principle for an AR application. This definition states that AR applications have to meet three basic requirements, namely: (i) a combination of real and virtual worlds; (ii) real-time interaction; and (iii) three-dimensional objects visualization superimposed onto the real environment.

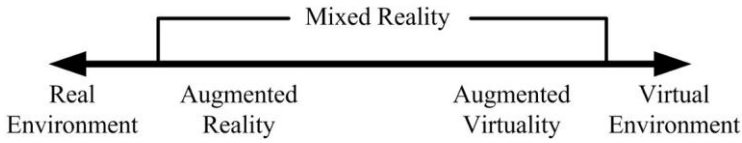


Fig. 1. Reality-Virtuality Continuum.

Based on these definitions, researchers have introduced contributions to different fields, such as medical sciences [7], training [20], games [17] and education [14] among many others. In particular, in the area of education there has been growing interest in AR technology in recent years due to its great potential [12]. AR environments allow students to interact with both the real and the virtual world, exploring objects, learning concepts, developing skills and carrying out collaborative activities. These aspects motivated the incorporation of different types of AR learning environments for regular classroom practices [8].

An AR application with promising capabilities for educational purposes has been expanded lately, named as *MagicBook* by Billinghurst et al [6]. More recently, this type of application is also known as *Augmented Books*. It uses normal printed books with AR fiducials (i.e. special markers that can be recognized by a computer software) as the main interface objects. People can turn pages of the book, look at the pictures and read the text like they used to. However, if they look at the pages through an AR display (e.g. computer screen) they would be able to appreciate virtual contents appearing over the pages. Thereby, augmented books give new alternatives to enhance traditional books with inter- active visualizations, animations, 3D models and/or audio, and open up a whole new range of possibilities to merge real and virtual experience.

Many researchers and companies [3, 21] explored AR book experiences, adding new ways of interactions [9, 12] or introducing collaborative tasks [15]. However, in general these approaches consist on books conceived as augmented books from scratch. We propose a new approach for adding AR capabilities to traditional books without having to generate them from scratch. Thus we can enrich pre- existing books embedding different kinds of

virtual content. In addition to this approach, we have implemented our authoring tool using a framework-based approach in order to facilitate the incorporation of new features to it (i.e. new interactions and/or new types of augmented contents).
The rest of this paper is structured as follows. Section 2 provides the related work concerning to *augmented books*. In Section 3 details of the approach are presented. In Section 4, the application architecture and details of the implementation are provided. Section 5 presents the discussion of the proposed approach based on the results from informal evaluation and observations and also we outline research lines for future work. Finally, the conclusion is presented in Section 6.

2. Related Work

Several research works explored various alternatives for augmented books in different areas such as chemistry [4], science [22] or cultural heritage [19] among others. More recently, this simple approach has been proved to be effective, for example for spatial skills development [14] or learning geometric shapes [12].
Despite this relatively large amount of research, many of them were developed for a specific pre-defined domain aiming to a specific target people using augmented books created from scratch. Other researchers proposed authoring tools for the creation of augmented books [10, 11]. Besides these approaches allow people with less technical knowledge to create these kind of books, the created books are conceived as augmented books from the beginning. Other authoring tools have been developed [1, 2, 13], however these approaches are too general and lack of specific features for books (e.g. interactions or contents designed for enhancing reading experience).
As we described, despite the great potential of augmented books, there has been a great limitation with authoring tools that do not consider pre-existent books. Our work tries to overcome this constraint, making the contribution of a novel approach to add the capabilities of augmented books to any traditional book.

3. Authoring Tool for Augmented Books

We present a novel tool for enhance pre-existent books with AR content. The main idea of our work is the traditional books' enrichment by means of the AR, not only showing augmented information on them but also giving the facilities to users (or readers, used indistinctly throughout this manuscript) to incorporate augmented contents to traditional books.

As in common AR systems, this tool is aimed to desktop computers with a camera device. The camera is the input sensor of the real world, and the computer display acts as a window to the augmented world, showing augmentations superimposed on the frames captured by the camera. We considered the use of at least one marker in the surrounding of the book in order to incorporate virtual contents over it. The next subsections describe the design of an augmented book and the usage concept of the proposed approach.

3.1 Book Model

We designed the augmented book as a composition of two concepts: a digital representation of the book itself and its digital contents. The digital representation of the physical book consists of descriptive attributes, such as the title, authors, cover page, a brief description of the book and its ISBN code. In addition to these attributes, the augmented book can be populated with several augmented contents in a structured fashion.
The augmented book is conformed by containers which are associated to a specific page of the physical book and can be used to separate content thematically. These containers can be attached with different virtual contents. Thus, these elements represent a logical container for the digital contents of the book. Several virtual contents can be added to the augmented book, such as text, web-links, images, 3D-models and sounds. These containers as well as the digital contents are created by the readers. Figure 2 shows an example of the structure of an augmented book.

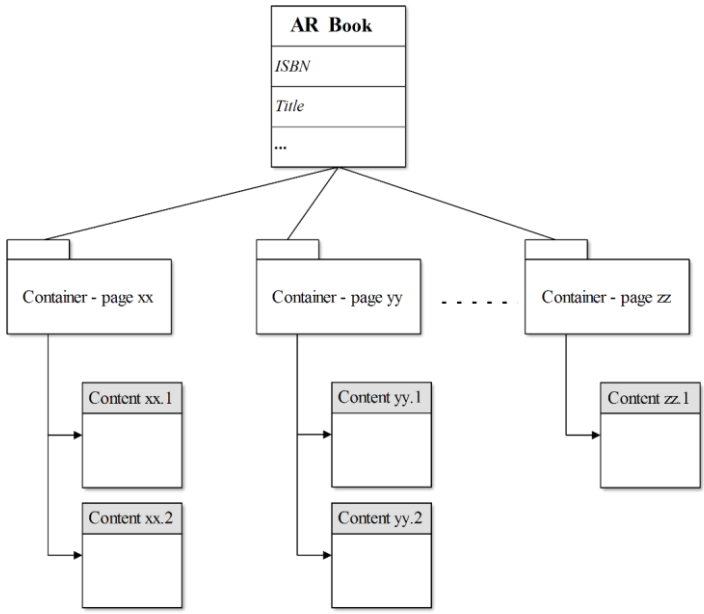
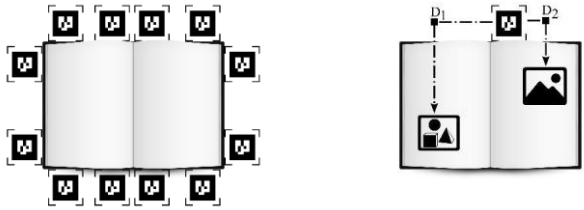


Fig. 2. An example of an augmented book structure.

The main attraction of digital contents is the possibility of being associated with two important features: *renderable objects* and *trackable objects*. The *renderable objects* allow the digital content to be displayed by the Render Engine of the authoring tool viewer, which in turn can be textured images or 3D-models.

The *trackable objects* are the main elements which enable AR. They consist of markers, that are interpreted by the AR Module to obtain its pose (i.e. position and orientation). With this information it is possible to assign a 3D transformation to a *renderable object* obtaining a correct registration in the rendering of the AR content. We opted for an adaptable way of adding virtual content for the physical books as it is shown in Fig. 3(a). The requirement of extra devices for the application is minimum (i.e. only one marker) as well as easily adapt- able to every book. From the chosen marker position, the reader can add the augmented contents displacing the content with 3D transformations as Fig. 3(b) shows. Despite this approach, our authoring tool supports several markers for the AR content input as well as *free markers* (i.e. markers that are not constrained to a fixed position, and the reader can freely move them throughout the book). Users can choose to add more markers for the content creation, tailoring the augmented book design to their particular needs.



(a) (b)

Fig. 3. Considered marker fixed positions (a) and example of the concept of displacements (D1 and D2) from a single marker to situate two augmented elements over the book (b).

3.2 Usage Concept

The usage idea of the proposed approach is basically divided in two phases: a stage among books and another stage within a book.

Among books. On one hand the user can create a new augmented book for his/her desired physical book. On the other hand, the reader can open previously created books. In this way, several books can be managed concurrently.

Within a book. Once the augmented book is open, the reader can navigate through its containers and contents. The reader can select each content type and perform the corresponding interactions (e.g. view images, play sounds, etc.). The selected augmented contents will be displayed over the book on the viewport of the application.
In order to add a new content to the issued augmented book, the user must select a container. When there is no container associated to the desired page or the user may want to separate contents thematically, the reader ought to create a new container. Then, he/she can select the content type to be created and choose specific characteristics about this content.
Finally, the reader can add augmented features for the issued content. In such case, the reader must select a marker (trackable object) and its position in the surrounding of the physical book. Also, besides the predefined positions for markers, a floating marker is considered. This feature allows to use this marker freely (i.e. user can move it without restrictions).
The basic 3D transformations (rotation, translation and scaling) can be performed to set the desired spatial position of the content over the book, that is translating the content from the marker location. All this process is totally interactive and is shown in real-time in a seamlessly manner.

4. Authoring Tool Implementation

4.1 Architecture

The proposed authoring tool was designed with several modules, maintaining them independent, in a loosely coupled manner. In this way, future modifications will not affect the overall application. Moreover, several object oriented patterns were followed and a framework-based approach was employed, allowing to easily extend the system with new features. The overall architecture of the system is shown in Fig. 4.1.

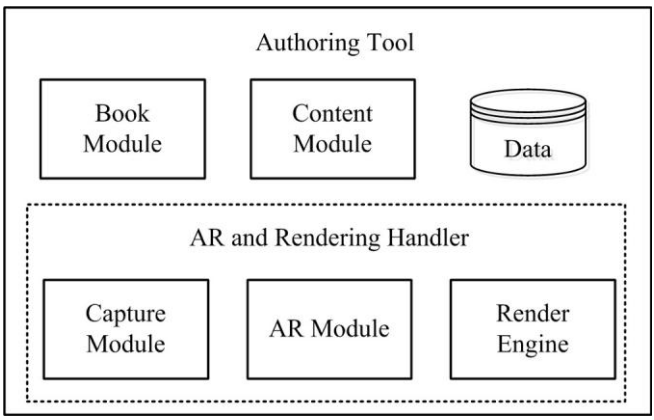


Fig. 4. System Architecture overview.

Each module aims to the different operations of the application. Hence, the *Book Module* handles all the actions related to search and creation of the augmented books. The creation of contents is achieved by the *Content Module*.

In addition to the described modules, the application consists of a group of modules to handle the rendering and the AR detection. The camera is accessed via the *Capture Module*. Frames captured are displayed by the *Render Engine* which in turn will display virtual objects over the book images. These AR features are handled by the *AR Module*, detecting the markers on the captured screen and obtaining the pose information of them. The functionalities from these modules are managed by a special handler, which performs the communication among them.

4.2 Implementation Details

The proposed authoring tool has been implemented in C++ using the Qt Framework[3]. The *Render Engine* was developed using OpenGL and the *AR Module* was designed using the ALVAR[4] library.
The application was developed for the Windows platform and deployed on a machine with AMD Phenom II X4 840 CPU, 4GB DDR3 memory and an ATI Radeon HD 5750 video card running Microsoft Windows 7 64-bit operating system. The application showing AR contents using the mentioned hardware was able to run smoothly with frame rates of about 30fps.

5. Discussion and Future Work

5.1 Informal User Observations

The application was tested by academics from different areas in order to obtain informal feedback and observations about the proposed approach. We suggested a simple task that involved the creation of an augmented book for their desired book, adding contents with AR features. Like in every first contact with an AR application, users needed a brief introduction to this technology (e.g. what is AR, the need of a marker, etc.). The feedback obtained was very positive and users were pleasantly surprised by the natural way of adding contents to the book. In general, they did not experienced problems with the interface and the proposed marker approach around the book. Figure 5 shows examples of different augmented contents created by academic readers.
The results of these informal observations are promising. Several suggestions were provided for special uses in different fields of endeavor, such as in a classroom environment as well as for exhibitions or demonstrations purposes. In general the users were not distracted with the marker location, however they had to be careful to not occlude the marker. All the users agreed that this technology would be very useful in their respective areas to motivate students with this enhanced reading experience.

5.2 Drawbacks and Future Work

Although promising feedback was obtained, this application still should be tested using a formal evaluation in order to obtain detailed results. The authoring tool provides several

[3] http://qt-project.org/
[4] http://virtual.vtt.fi/virtual/proj2/multimedia/index.html

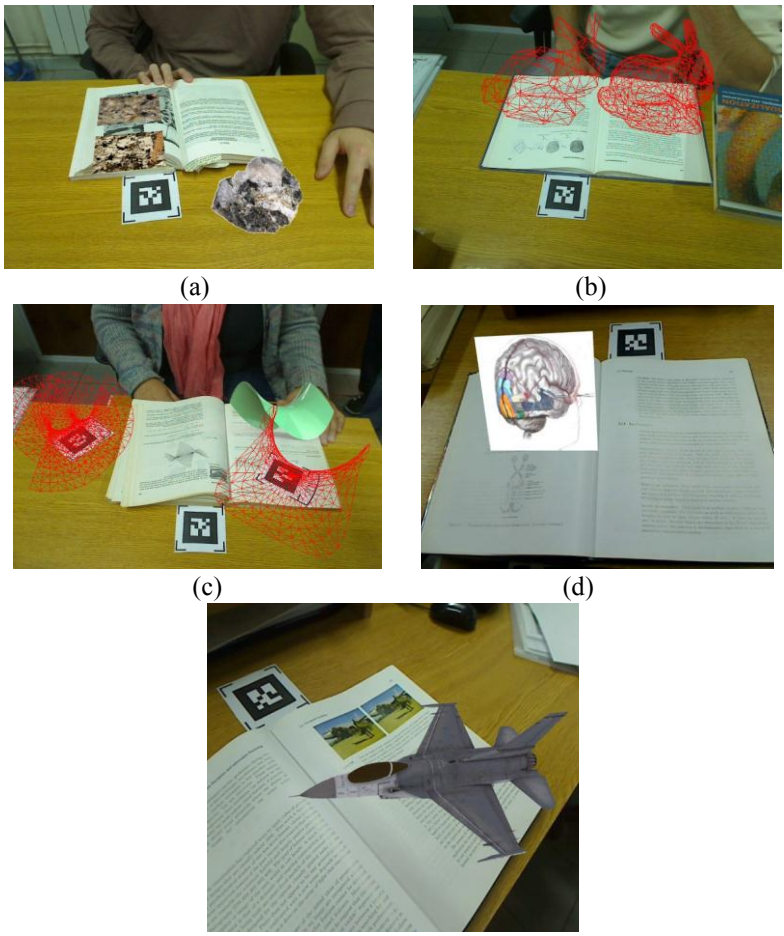


Fig. 5. Examples of four augmented books created using the authoring tool: a book of Geology (a), Computer Graphics (b), Mathematics (c) and Visualization (d,e).

digital contents and interactions for these contents. However, there are still other contents to be considered (e.g. videos) as well as other kind of interactions (e.g. multimodal interactions). From the informal conducted user observation, suggestions to incorporate additional contents like digital notes or some kind of handwrite notes were provided.

Though the single marker or several markers approaches are effective, we plan to improve the existent AR features adding *Markerless Registration and Tracking* in order to obtain a more natural interface with the book. This kind of registration methods were considered in the design of the application but these are not implemented yet.

Finally we also plan to port our authoring tool to mobile devices. A mobile version could be used in different environments where it would be difficult to have a desktop computer to every reader (e.g. a library).

6. Conclusion

In this paper we presented a novel authoring tool for augmented books. The application allows readers to incorporate augmented contents to pre-existing printed books. The application has been used to create Augmented Books by academics of different areas. Informal user observation and interviews were con- ducted and the feedback obtained suggests that this technology is easy to use and opens up great possibilities to enrich the traditional reading. Users were able to create new digital contents for their desired books and experienced the facilities offered by the proposed authoring tool to complement the reading experience. We summarized future directions for this approach and we expect to use this technology in different environments such as libraries or classrooms.

References

1. http://www.buildar.co.nz/ (BuildAR, commercial authoring tool developed by HIT Lab NZ).
2. http://www.metaio.com/creator/ (Metaio Creator, commercial authoring tool from METAIO).
3. AR-Books: Commercial augmented reality books. http://www.ar-books.com/ (2012).
4. Asai, K., Kobayashi, H., Kondo, T.: Augmented instructions - a fusion of augmented reality and printed learning materials. In: Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on. pp. 213–215 (2005).
5. Azuma, R.: A survey of augmented reality. Presence 6(4), 355-385 (1997).
6. Billinghurst, M., Kato, H., Poupyrev, I.: The magicbook - moving seamlessly between reality and virtuality. Computer Graphics and Applications, IEEE 21(3), 6-8 (may/jun 2001).
7. Buchs, N.C., Volonte, F., Pugin, F., Toso, C., Fusaglia, M., Gavaghan, K., Majno, P.E., Peterhans, M., Weber, S., Morel, P.: Augmented environments for the targeting of hepatic lesions during image-guided robotic liver surgery. Journal of Surgical Research 184(2), 825- 831 (2013).
8. Cuendet, S., Bonnard, Q., Do-Lenh, S., Dillenbourg, P.: Designing augmented reality for the classroom. Computers & Education 68(0), 557- 569 (2013).
9. Grasset, R., Dunser, A., Billinghurst, M.: The design of a mixed-reality book: Is it still a real book? In: Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality. pp. 99-102. ISMAR '08, IEEE Computer Society, Washington, DC, USA (2008).

10. Ha, T., Woo, W., Lee, Y., Lee, J., Ryu, J., Choi, H., Lee, K.: Artalet: Tangible user interface based immersive augmented reality authoring tool for digilog book. In: Ubiquitous Virtual Reality (ISUVR), 2010 International Symposium on. pp. 40-43 (july 2010).
11. Jee, H.K., Lim, S., Youn, J., Lee, J.: An immersive authoring tool for augmented reality-based e-learning applications. In: Information Science and Applications (ICISA), 2011 International Conference on. pp. 1-5 (April 2011).
12. Kirner, T., Reis, F., Kirner, C.: Development of an interactive book with augmented reality for teaching and learning geometric shapes. In: Information Systems and Technologies (CISTI), 2012 7th Iberian Conference on. pp. 1-6 (2012).
13. Layar: http://www.layar.com/
14. Martín-Gutiérrez, J., Saorín, J.L., Contero, M., niz, M.A., Pérez-López, D.C., Ortega, M.: Design and validation of an augmented book for spatial abilities development in engineering students. Computers & Graphics 34(1), 77- 91 (2010).
15. Matcha, W., Awang Rambli, D.: Design consideration for augmented reality book- based application for collaborative learning environment. In: Computer Information Science (ICCIS), 2012 International Conference on. vol. 2, pp. 1123-1126 (2012).
16. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. In: IEICE Trans. Information Systems. vol. E77-D, pp. 1321-1329 (1994).
17. Von der Ptten, A.M., Klatt, J., Broeke, S.T., McCall, R., Krmer, N.C., Wetzel, R., Blum, L., Oppermann, L., Klatt, J.: Subjective and behavioral presence measurement and interactivity in the collaborative augmented reality game timewarp. Interacting with Computers 24(4), 317- 325 (2012), special Issue on Presence and Interaction.
18. Sutherland, I.E.: The ultimate display. In: Proceedings of the IFIP Congress. pp. 506-508 (1965).
19. Walczak, K., Wojciechowski, R.: Dynamic creation of interactive mixed reality presentations. In: Proceedings of the ACM symposium on Virtual reality software and technology. pp. 167-176. VRST '05, ACM, New York, NY, USA (2005).
20. Webel, S., Bockholt, U., Engelke, T., Gavish, N., Olbrich, M., Preusche, C.: An augmented reality training platform for assembly and maintenance skills. Robotics and Autonomous Systems 61(4), 398- 403 (2013), models and Technologies for Multi-modal Skill Training.
21. Wonderbook: an augmented reality peripheral for the playstation 3 console. http://wonderbook.eu.playstation.com/ (2012).
22. Woods, E., Billinghurst, M., Looser, J., Aldridge, G., Brown, D., Garrie, B., Nelles, C.: Augmenting the science centre and museum experience. In: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia. pp. 230-236. GRAPHITE '04, ACM, New York, NY, USA (2004).

Participation Metrics within Virtual Collaborative Workspaces Oriented to Generation of Didactic Interventions

NORBERTO CHARCZUK, DARÍO RODRÍGUEZ, RAMÓN GARCÍA-MARTINEZ

Laboratory of Virtual Workspaces
Information Systems Research Group
Department of Productive and Tecnological Development
National University of Lanus
norcknet@gmail.com, {darodriguez,rgarcia}@unla.edu.ar

Abstract. The incorporation of technology as tool in the learning process in undergraduate and post-graduate level has generated a multitude of factors to be considered, one of them, students´ interaction in Virtual Collaborative Workspaces (EVTC), where observing the student during the group learning process must also be mediated by technology. In this paper, we propose an adaptation of certain metrics used in SNA field (Social Network Analysis) as a professor's tool to decide proper didactic intervention in order to achieve the course objectives.

Keywords. Technology as a tool, learning process, virtual collaborative workspaces, adaptation of sociometric metrics, didactic intervention.

1. Introduction

The technological contributions in educational area always have been present. They have evolved from slate to computer. Nowadays what has changed is that there is a new "environment" in which educational process may take place: the web [Litwin, 2000]. There are several educative models: presential, semi-presential or distance. This models coexist and each one of them has its particularity. In all of them we find professors and students, but, as work environment for students has changed because of using web, this implies that there is a change in ways of student´s observing by professor, during the group learning process in virtual workspaces. Hence the need of new concepts which allow evaluate in virtual workspaces, the learning process of student who interacts with his group, focusing on levels of participation and membership. This paper proposes a metric based on the participants´ interaction in a virtual collaborative workspace (EVTC), having as reference framework the SNA (social network analysis).
Within this context, the theoretical framework for observing groups is presented in detail, characteristics of the different roles and adaptation of sociometric techniques as a tool for observing interactions within group

(section 2); the problem is defined (section 3); a solution based on the adaptation of definitions of sociometric field and of social dynamics in virtual groups is proposed, and a method of calculation of the indicators defined in adaptation (section 4) ; the application of the proposed definitions and techniques are shown in a proof of concept (Section 5); and preliminary conclusions on this research line are summarized (section 6) .

2. Theoretical Framework

In this section the theoretical framework that considers the interaction among participants in working groups is presented (section 2.1), the concept and characteristics of role in working groups (section 2.2) and sociometry as a tool of observing the roles (section 2. 3).

2.1. Interaction in Virtual Workspaces (EVTC)

An interpretation of the factors that make the interaction of participants in a EVTC, are the asynchronous processes among them, where the new web-based learning environments and training web platforms allow greater exchange in time [Williams et al., 1999]. Hence one can distinguish the different skills and roles of the participants. Facing these new elements shifts the focus of teaching and how to get information from the need in an didactic intervention.
The "visible" things,, where which is visible may have implications related to knowledge and the way to access to it. In word of Querrien [1979] "... this established order but not altered by the individual, by his sudden movements, his unreasoned natural needs ...if what is seen, is readed, is observed; is the representation of what is done or what should be done; so that what should be done has been previously set, and can not therefore be modified in the future ... .this identification of imposition order and visible order, at same time that thoughts and visible things also have enormous consequences in the innovation domain ... "
In an EVTC, exchanges can be synchronous and asynchronous, but the degree of visibility of the participants is zero, hence the importance of observation and its characteristics. De Ketele [1980] defines that "...observing is a process that includes voluntary-attention and intelligence, guided by a target or an organizing goal, and is directed onto an object for obtaining information from him". Due to the particular circumstances of the environment, its degree of visibility is determined by the interaction / participation of members and the need to generate appropriate metrics.

2.2. Concept "Role" in Working Groups

Unlike traditional / formal education, which prescribes fixed roles to extreme of stereotyping them, this methodology proposes changeable roles (but not

symmetrical) for participants. The "informal" expression refers to a group of persons in which roles have not been previously agreed. The system of relations is dynamic; is developed and modified over time, but it may be taken in mind that certain behaviors can be repeated, such as aggression, apathy, rejection among participants. Hence, the importance of being able to identify bonds among members, in order to review and make didactic interventions. This seeks for to avoid competition that generates aggression and impoverishes the quality of the achievements of the group. It should aim to roles that enable collaboration between participants and achieve different visions of common project [Pasel & Asborno, 1993].

2.3. Sociometry as Observing Tool for Roles

Sociometry is the technique that investigates centralism index, communication channels and experiences within a human group, hence the degree of belonging to it and participation with their respective functions in the group. It was created by the psychiatrist Jacob Levy Moreno (American psychiatrist born in Bucharest, Romania, 1889-1974). Among the techniques developed in this discipline are the sociometric test and sociometric perception test, aimed at the systematic study of the psychological properties of people, putting into action an experimental technique based on quantitative methods. Among the applications in the educational field we can cite the work in early childhood education by Bastin [1966], at the primary level by Leibovich [1980], and at the secondary level by by Perez-Alvarez [1991]. The related process may be adapted to be applicable to obtain different measures of virtual interaction of a group within an asynchronous collaborative environment. [Charczuk et al.,, 2013]

3. Identified Problem

Sociometric techniques are used to find out who are the actors of the dynamics of a working group: leaders, marginalized, rejected, normal, and others. Is an open problem to determine these type of roles in groups that interact mediated by internet based technology, either students in a virtual classroom, or professionals in a virtual office.
In virtual environments the socializing link occurs through interactions [Rodriguez and Garcia-Martinez, 2014]. Then, relationships all-parts and parts-all are determined by the flow of interactions, this is consistent with the fact that systems are constituted by actions among complex units formed turn by interactions [Morin, 1980].
In [Charczuk et al., 2013] has been proposed the goal of developing sociometric techniques for project development groups mediated by EVTC based on sociometric test techniques such as Bastin [1966] and Leibovich [1980].

4. Proposed Solution

This section presents definitions to be used for reformulation of sociometric techniques to diagnose of group dynamics in virtual collaborative workspaces (section 4.1), and the process for calculating the measures of diagnostic/membership in group dynamics (Section 4.2)..

4.1. Definitions

In this section we define and redefine sociometric concepts to enable their application in virtual collaborative workspaces:

4.1.1. Communication Density (DC)

Is the number of messages sent per time unit by the member S_i the member S_j

4.1.2. Interaction Matrix (M)

It can be built the matrix M of working group interactions, where M (i,j) is the amount of messages sent / answered of the member S_i to member S_j.

4.1.3. Communication Density Matrix Member-Group (MDCSG)

It is built using the matrix M and contents in the position MDCSG(i,j) the density that member S_i has of each member of the group.

4.1.4 Segmentation Step (ES)

It is the value obtained dividing by 5 the subtraction of the higher value of Communication Density minus lower value of Communication Density (this generates five quintiles)

4.1.5 Conversion of DC values to Indicators of Metrics

It is the value obtained by assigning each DC value within the ranges generated by distribution in quintiles (see Table II).

4.1.6. Communication Density Reference Matrix Interactions (MRDCI)

It is built using the matrix MDC and contents in the position MRDCI(i,j) the resulting value from the conversion of communication density parameter related into the segmentation rank of each group member (see Table II).

4.1.7. Auto- collaboration Index (IAC_i)

It is the value representing the degree of collaboration that member S_i has about himself in relation to the group. This index is calculated by the expression (where X represents the weighting value, n represents the relative frequency of value x, N represents the absolute frequency):

$$IAC = \frac{\sum_{i=1}^{n} xi.ni}{N}$$

4.1.8. Hetero-collaboration Index (IHCi)

t is an approximate value of the degree of collaboration that has the group with respect to member S_i. This index is calculated by the expression (x, n, N; iden IAC_i):

$$IHC = \frac{\sum_{i=1}^{n} xi.ni}{N}$$

4.1.9. Collaborative Reality Index (IRC_i)

It is a value that represents the difference between the above indices and to determine the degree of collaboration member-group / group-member. This index is calculated by the expression: IRC_i=10 x (IAC_i – IHC_i). This value has the coefficient 10 which multiplies the difference between indices, because of that differences between values may be a value by rounding not significant enough. It is proposed that this index is interpreted according to the following decision rule:

IRCi > 0 The member S_i is over-estimated in relation to estimated by the group in the collaborative process.

IRCi = 0 The valuation of the member S_i is similar with the which one group has of him.

IRCi < 0 The member S_i is under-estimated in relation to estimated by the group in the collaborative process.

4.1.10. Absolute Reality Index (IRCA)

It is the absolute value of the Collaborative Reality Index of a member and a group obtained by accumulation of IRCA of members. The average value of the index of absolute reality (MIRCA) indicates the overall assessment of the group, if it is close to zero, it means that group is mature in collaborative issues. Otherwise, the more it moves away from zero, it means the non-maturity of the grpup in collaborative tasks. The Absolute Reality Index is calculated by the following expression (where L is the number of subjects in the group):

$$V_{MIRCA} = \frac{\sum_{i=1}^{n} IRCAi}{L}$$

4.1.11. Degree of Colaboration Member-Group (GCSG)

The GCSG of member W allow observing which is the interaction behavior among member W and group. GCSG(Si,P) represents the amount of interaction with value P that has member W. The assessment of GCSG of the member W is given by the vector:

($Acceptor_i$, $Normal_i$, $Rejectorr_i$)

Where:

$Acceptor_i$ = GCSG(Si,5) + GCSG(Si,4)

$Normal_i$ = GCSG(S_i,3) + GCSG(S_i,2)

$Rejector_i$ = GCSG(S_i,1) + GCSG(S_i,0)

The decision rules proposed in this paper are:

S_i is $Acceptor_i$ of Group IF

$Acceptor_i >= Normal_i + Rejectorr_i$

S_i is $Normal_i$ of Group IF

$Normal_i >= Acceptor_i + Rejectorr_i$

S_i es $Rejector_i$ of Group IF

$Rejector_i >= Normal_i + Acceptor_i$

If the weighting find with the same value in two of the three components (Acceptor, Normal, Rejector), then the attributes of weighting will be shared.

4.1.12. Degree of Colaboration Group-Member (GCGS)

The GCGS of Group related to a member W allows observing by pooling the amounts of interactions between other members of the group and S_i which is group behaviour in interaction with the member S_i. GCGS (S_i,Q) denotes the amount of weights with Q value of the group with respect to member S_i. The assessment of GCGS of the member S_i is given by the vector:

($Important_i$, $Normal_i$, $Outcast_i$)

Donde:

$Important_i$ = GCGS (S_i,5) + GCGS (S_i,4)

$Normal_i$ = GCGS (S_i,3) + GCGS (S_i,2)

$Outcast_i$ = GCGS (S_i,1) + GCGS (S_i,0)

The decision rules proposed in this paper are:

S_i is Important of Group IF

$Important_i >= Normal_i + Outcast_i$

S_i is Normal of Group IF

$Normal_i >= Important_i + Outcast_i$

S_i is Outcast of Group IF

$Outcast_i >= Normal_i + Important_i$

If the weighting find with the same value in two of the three components (Important, Normal, Outcast), then the attributes of weighting will be shared. The "Important" role may correspond to a potential leader.

4.1.13. Communication Channels

They are the various ways in which all members of Group can communicate. It will be denoted with the letter K. Since the maximum number of possible communication channels is determined by the following expression (where L represents the number of subjects in the group):

$$K = \frac{L}{2} * (L - 1)$$

4.1.14. Percentage of Effective Channels of Communications

This Value represents the percentage of cases relating to effective communication; it is proposed to interpret as effective communication to

cases in which the analyzed weight corresponds to the upper half of the matrix MRDCI. $Ft_i(5,4,3)$ is the sum of the relative frequencies using weights with value 5, 4 and 3 in the lower triangular submatrix of interactions matrix, and $Ft_s(5,4,3)$ is similar one using upper triangular submatrix of interactions matrix. From both submatrices a value is obtained, the lowest of them is selected by interpreting the worst case in the communication process, which divided by the maximum number of channels allows to obtain:

$$Percentage\ of\ Effective\ Channels\ of\ Comunications = \frac{F(5,4,3)}{K} * 100$$

4.1.15. Percentage of Communication Lack

The lack of communication is determined by the amount values of zeros (using weighting matrix MRDCI there is no communication) divided by the possible communications in the interactions matrix.

$$Percentage\ of\ Communication\ Lack = \frac{Amount\ of\ Zeros}{(L-1) * L} * 100$$

4.2 Proposed Method

In this section we present the calculus method for predefined indicators as follows:

Step 1:	Identification of amount of sent / received messages and time-lapse records.
Step 2:	Construction of Interactions. Matrix M(i, j).
Step 3:	Construction of Communication Density Matrix MDC(i,j).
Step 4:	Calculus of Segmentation Step (ES).
Step 5:	Assigning DC values related to corresponding ranges of Segmentation Steps.
Step 6:	Reinterpretation of table MDC(i, j), assigning parameters to Table I and Table II
Step 7:	Construction of Matrix MRDCI(i,j)
Step 8:	Calculus of $IAC_{n+1,\,j}$ and del $IHC_{i,n+1}$
Step 9:	Calculus of Collaborative Reality Index (IRC_i) for member S_i.
Step 10:	Interpretation of resulting Collaborative Reality Index (IRC_i) for each member S_i.
Step 11:	Extended of Communication Density Reference Matrix Interactions Member-Group (SG) with the row Degree of Collaboration Member-Group (GCSG)
Step 12:	Interpretación of matrix MRDCI of Member-Group (SG) interactions.
Step 13:	Extended of Communication Density Reference Matrix Interactions Group-Member (GS) with the row Degree of Collaboration Group-Member (GCGS).
Step 14:	Calculus of: Communication Channels, Percentage of Effective Channels of Communications, and Percentage of Communication Lack
Step 15:	Interpretation of results of Channels Calculus
Step 16:	Calculus of DC value bias with respect to its mean value
Step 17:	Graphing DC bias and its mean value with respect to the communication links.

5. Case Study

In this section we present a case study (section 5.1), the application of the proposed method (section 5.2), and the interpretation of obtained results (section 5.3).

5.1. Case Study Description

The case study is based on the e-mails among the four members of a student-group. The group has to use the virtual workspace to manage the group work related to solve a problem of software development assigned by their professor. Proposed metrics were evaluated using e-mails among students of the group.

5.2. Applying the Proposed Method

In this section we present the results of each step of the proposed method:

Step 1: Identification of amount of sent / received messages and time-lapse records

Step 2: Construction of Interactions. Matrix M(i, j)

	S_1	S_2	S_3	S_4
S_1		5	5	4
S_2	12		12	10
S_3	55	55		67
S_4	48	48	60	

Results Step 1: Amount of time: 22 days

Results Step 2

	S_1	S_2	S_3	S_4
S_1		0,23	0,23	0,18
S_2	0,55		0,55	0,45
S_3	2,50	2,50		3,05
S_4	2,18	2,18	2,73	

Step 3: Construction of Communication Density Matrix MDC(i,j).

ES= (3,045-0,182) / 5

ES= 0,57272

Step 4 and 5: Calculus of Segmentation Step (ES); Assigning DC values related to corresponding ranges of Segmentation Steps.

a. 2,473 – 3,045
b. 1,900 – 2,473
c. 1,327 – 1,900
d. 0,755 – 1,327
e. 0,182 – 0,755

Results Step 3

Member i	Member j	DC	Order Nber
S_3	S_4	3,045	1
S_4	S_3	2,727	2
S_3	S_2	2,500	3
S_3	S_1	2,500	4
S_4	S_1	2,182	5
S_4	S_2	2,182	6
S_2	S_1	0,545	7
S_2	S_3	0,545	8
S_2	S_4	0,455	9
S_1	S_2	0,227	10
S_1	S_3	0,227	11
S_1	S_4	0,182	12

Results Steps 4-5

Step 6: Reinterpretation of table MDC(i, j)

Step 7: Construction of Matrix MRDCI(i,j)

	S_1	S_2	S_3	S_4	Amount 5	Amount 4	Amount 3	Amount 2	Amount 1	Amount 0	Acceptor	Normal	Rejector	IAC
S_1		1	1	1					3		0	0	3	1,00
S_2	1		1	1					3		0	0	3	1,00
S_3	5	5		5	3						3	0	0	5,00
S_4	4	4	5		1	2					3	0	0	4,33
Amount 5	1	1	1	1										
Amount 4	1	1												
Amount 3														
Amount 2														
Amount 1	1	1	2	2										
Amount 0														
Important.	2	2	1	1										
Normal														
Outcast	1	1	2	2										
IHC	3,33	3,33	2,33	2,33										
Amount of 5,4,3:	6													

Amount of 5,4,3 = 6

Results Steps 6-7

Step 8, 9 and 10: Calculus of IACn+1, j and del IHCi,n+1; Calculus of Collaborative Reality Index (IRCi) for member Si; Interpretation of resulting Collaborative Reality Index (IRCi) for each member Si.

	IRC	IAC	IHC	IRCA	Int. IRC
S_1	-2,33	1,00	3,33	2,33	-
S_2	-2,33	1,00	3,33	2,33	-
S_3	2,67	5,00	2,33	2,67	+
S_4	2,00	4,33	2,33	2,00	+

Results Steps 8-9-10

Step 11, 12 y 13: Extended of Communication Density Reference Matrix Interactions Member-Group (SG) with the row Degree of Collaboration Member-Group (GCSG); Interpretation of matrix MRDCI of Member-Group (SG) interactions; Extended of Communication Density Reference Matrix Interactions Group-Member (GS) with the row Degree of Collaboration Group-Member (GCGS).

Results of Steps 11,12 y 13 are included in matrix MRDCI of steps 6 and 7.

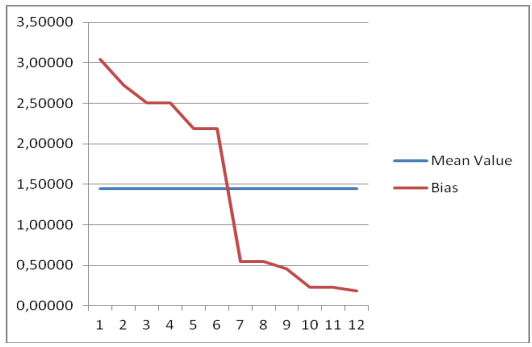
Step 14 and 15: Calculus of: Communication Channels, Percentage of Effective Channels of Communications, and Percentage of Communication Lack; Interpretation of results of Channels Calculus

Amount of Communication Channels	6
Percentage of Effective Channels of Communications	100
Percentage of Communication Lack	0

Step 16 and 17: Calculus of DC value bias with respect to its mean value; Graphing DC bias and its mean value with respect to the communication links.

DC	Mean Value	Bias
3,045	1,44318	1,602
2,727	1,44318	1,284
2,500	1,44318	1,057
2,500	1,44318	1,057
2,182	1,44318	0,739
2,182	1,44318	0,739
0,545	1,44318	-0,898
0,545	1,44318	-0,898
0,455	1,44318	-0,989
0,227	1,44318	-1,216
0,227	1,44318	-1,216
0,182	1,44318	-1,261

Results of Step 16



Results of Step 17

5.3. Interpretation of Results

The interpretation of resulting Collaborative Reality Index (IRC_i) for ech member S_i, is denoted by "+" when member S_i overestimates himself in relation with the group estimation. It is denoted by "-" when member S_i underestimates himself in relation with the group estimation. It is denoted by "0" when member Si estimates himself in the same way as do group. This value represents the lack of communication among participants, and is used to calculate the percentage of Communication Lack (section 4.1.15). The interpretation of the results Member-Group (Matrix MRDCI) is based on the conventions specified in definitions 4.1.11 and 4.1.12.
From the results it follows that in the polarized group behavior are noticeable: S1 and S2 = Rejecters; S3 and S4 = acceptors; S1 and S2 = Important; and S3 and S4 = Outcast.

6. Conclusions

Our work aims to develop analytical tools that bring out the underlying structure of the group interaction mediated by a virtual workspace, allowing hypotheses with quantitative basis on likes and dislikes of individuals to each other and in relation to collaborative task [Cirigliano & Villaverde, 1966]. Preliminary results obtained in the experimental virtual workspace, show binding elements with distance education, where participants have synchronous and asynchronous communications are observed. In the future, it is expected: [a] do a systematic validation of indicators in a large sample

group in virtual collaborative work spaces; and [b] explore ways of tailoring Sociometric other indicators.

The human component involved in the interaction mediated by virtual workspaces, generates effects on individual learning, raising the need for tools to assist the diagnosis for the necessary intervention of teachers in the educational processes in use of virtual workspaces dedicated to education.

References

1. Bastín, G. (1966). *Los test sociométricos*. Kapeluz.
2. Charczuk, N., Rodríguez, D., García-Martínez, R. 2013. *Propuesta de Técnicas de Diagnóstico Sociométrico de Dinámicas Grupales para Utilizar en Ambientes de Trabajo Colaborativo*. Proceedings VIII TEYET. ISBN 978-987-1676-04-0. UNSE
3. Cirigliano, G. & Villaverde, A. (1966). *Dinámica de Grupos y Educación*. HVMANITAS.
4. Cohen, S. & Bailey, D. 1997. *What Makes Team Work: Group Effectiveness Research from the Shop Floor to the Executive Suite*. Journal of Management, 23(3): 239-290.
5. Cols, A. (1977). *Conducción del aprendizaje*. Marymar.
6. De Ketele, J. & Roegiers, X. (1995). *Metodología para la Recogida de Información*. Editorial La Muralla.
7. Etzkowitz, H. 2003. *Research Groups as 'Quasi-firms': The Invention of the Entrepreneurial University*. Research Policy, 32: 109-121.
8. Leibovich, É. (1980). *Medida de la relación social en el aula*. CINAE.
9. Li, C. & Bernoff J. (2008). *Groundswell: Winning in a world transformed by social technologies*. Harvard Business Press.
10. Litwin, E. (2000). *La educación a distancia* (compiler) Colección Agenda Educativa.
11. Morin, E. (1980). *Ciencia con Conciencia*. Anthopos.
12. Pasel, S., & Asborno, S. (1993). *Aula-Taller*. Aique Grupo Editor S.A.
13. Perez-Alvares, S. (1991). *El Diagnóstico de la Situación Educativa*. Editorial Braga 3° Edición.
14. Pérez-Alvarez, S. (1991). L*as investigaciones exploratorias y descriptivas en las Ciencias de la Educación*. Editorial Braga.
15. Querrien, A. (1979). *Trabajos elementales sobre la escuela primaria*. Ed. La Piqueta. Madrid.
16. Rodriguez, D., Garcia-Martinez, R. 2014. *A Proposal of Interaction Modelling Formalisms in Virtual Collaborative Work Spaces*. Lecture Notes on Software Engineering, 2(1): 76-80. ISSN-2301-3559.
17. Williams, M. & Paprock, K. & Covington, B. (1999). *Distance Learning The Essential Guide*. Sage Publications.

18. **Acknowledgments.**The research reported in this paper has been partially funded by Research Subsidies applied to Project 33A166 within Laboratory of Virtual Workspaces of Department of Production and Technological Development of the National University of Lanus, Argentina.

XII

Graphic Computation, Images and Visualization Workshop

Time of Flight Image Segmentation through Co-Regularized Spectral Clustering

LUCIANO LORENTI AND JAVIER GIACOMANTONE

Instituto de Investigación en Informática (III-LIDI),
Facultad de Informática - Universidad Nacional de La Plata - Argentina.
La Plata, Buenos Aires, Argentina.
{llorenti, jog, degiusti}@lidi.info.unlp.edu.ar

Abstract. Time of Flight (TOF) cameras generate two simultaneous images, one for intensity and one for range. This allows tackling segmentation problems where the information pertaining to intensity or range alone is not enough to extract objects of interest from a 3D scene. In this paper, we present a spectral segmentation method that combines information from both images. By modifying the affinity matrix of each of the images based on the other, the segmentation of objects in the scene is improved. The proposed method exploits two mechanisms, one for reducing the computational demand when calculating the eigenvectors for each matrix, and another for improving segmentation performance. The experimental results obtained with two sets of real images are presented and used to assess the proposed method.

Keywords: Segmentation, Range Images, Time of Flight Cameras, Spectral Clustering

1. Introduction

Segmentation is generally the first stage in an image analysis system, and it is one of its most critical tasks because it affects subsequent stages [5][21]. Computer vision algorithms, in particular segmentation ones, that have been successfully used in industrial environments, with controlled colors and lighting, do not obtain similar results in different contexts. An alternative to tackle problems in which outline conditions are not suitable for adequate segmentation would be to add depth information, i.e., the distance at which the objects forming the scene are located in relation to the capture device [13][12]. In this context, image segmentation consists in using algorithms that use both sources of information rather than just intensity levels [3][9]. With this perspective, the segmentation problem can be presented as the search for effective shapes that will allow correctly partitioning a set of samples with intensity and distance information. In particular, for this work we used a TOF camera, the MESA SR 4000 [6], that allows obtaining range and intensity images simultaneously. The SR 4000 is an active camera that

uses its own lighting source through a matrix of infrared, amplitude-modulated light emitting diodes. Camera sensors detect the light that is reflected on the illuminated objects and the camera generates two images. The intensity image is proportional to the amplitude of the reflected wave, and the range or distance image is generated from the phase difference between the emitted wave and the wave reflected on each element in the image [2]. The main advantages versus other 3D measurement techniques is the possibility of obtaining images at speeds that are compatible with real-time applications, as well as the possibility of obtaining clouds of 3D points from a single point of view [10][7]. Due to the computational complexity required for spectral clustering algorithms, methods facilitating the calculation of the eigenvectors of the affinity matrix have been proposed recently [11][1][8]. The clustering techniques that have been used allow improving, from multiple data representations, the clustering process [15][14][4]. The method proposed exploits two mechanisms, one for reducing the computational demand when calculating the eigenvectors for each matrix, and another for improving segmentation performance. The improvement in the computational demand is achieved by approximating the eigenvectors of the affinity matrix obtained from each of the images. Segmentation is improved, with respect to the use of a single image, by means of an iteration mechanism that allows obtaining the optimal eigenvector space to carry out the segmentation. The method proposed is assessed by means of two datasets of real images, one obtained with a TOF camera, and another one provided by the Laboratory of Multimedia Technology and Telecommunications of the University of Padua [20].
This article is organized as follows: in Section 2, a review of the main concepts used for the method proposed is presented. The method itself is described in Section 3. In Section 4, experimental results are detailed. Finally, the conclusions are discussed in Section 5.

2. Spectral clustering

Given a set of patterns $X = \{x_1, x_2, ...x_n\} \in \mathbb{R}^m$, and a similarity function $d : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$, an affinity matrix W can be built so that $W(i,j) = d(x_i, x_j)$. Spectral clustering algorithms obtain a representation of the data in a lower dimension space by solving the following optimization problem:

$$\begin{array}{ll} \underset{U \in \mathbb{R}^{n \times k}}{\text{máx}} & Tr\left(U^T L U\right) \\ \text{s.t.} & U^T U = I \end{array} \tag{1}$$

where $L = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ is the Laplacian matrix of W in accordance to [17] and D is a diagonal matrix with the sum of the rows in W on its main diagonal. After U is obtained, its rows are considered as the new coordinates

for the patterns. With this new representation, traditional clustering algorithms are easier to apply [19].
Image segmentation spectral methods are based on the eigenvectors and eigenvalues of a N x N matrix derived from the affinities among pixels. It should be noted that one of the main limitations of this type of algorithms is the amount of memory required due to the fact that the dimensions of W exhibit quadratic growth with respect to the number of elements in the image. A possible approach to deal with this problem is using a sparse matrix that codes the local information from each pixel. In this representation, each element is connected only to some of its closest neighbors, and all other connections are assumed to be zero [19] [16]. Alternatively, the affinities of a small set of pixels can be calculated and the remaining affinities, approximated [11] [1].

2.1 Approximate Calculation of Eigenvectors

One of the initial proposals to define spectral clustering relates the weights matrix W to a graph incidence matrix and to the clustering problem as a graph partitioning problem [19]. From this perspective, each of the $\mathbf{X}_i$ patterns is considered as the vertex of a weighted, non-directed graph $G = (V, E)$ and the element $W(i, j)$ is the weight of the edge that connect vertex i with vertex j. Be $G = (V, E)$ the similarity graph derived from a set of patterns $X = \{x_1, x_2, ..., x_n\}$, $A \subset V$ a subset of sampled vertexes and $B = V - A$, the remaining, non-sampled vertexes. G_A is the graph that results from connecting the vertexes of A to each other, and G_B is the graph that results from connecting the vertexes of A to the vertexes of B. Be W_A the adjacency matrix of G_A and L_A the Laplacian matrix of G_A. Be W_B and L_B the corresponding matrixes of G_B. We can then formulate the adjacency matrix of G, which we will call W, and the Laplacian matrix of G, which we will call L, as follows:

$$W = \begin{bmatrix} W_A & W_B \\ W_B^T & W_C \end{bmatrix} \qquad L = \begin{bmatrix} L_A & L_B \\ L_B^T & L_C \end{bmatrix}$$

Considering the diagonalization of $A = U\Lambda U^T$, using Nystrom's extension [11]: $\overline{U} = \begin{bmatrix} U \\ B^T U \Lambda^{-1} \end{bmatrix}$ as approximate eigenvectors of W, an approximation of W, called $\hat{W}$, can be obtained by calculating only A and B:

$$\hat{W} = \bar{U}\Lambda\bar{U}^T = \begin{bmatrix} A & B \\ B^T & B^T A^{-1} B \end{bmatrix}$$

To obtain the eigenvectors for $\hat{L} = \hat{D}^{\frac{1}{2}}\hat{W}\hat{D}^{\frac{1}{2}}$, i.e. the approximate Laplacian matrix generated from $\hat{W}$ the proposed technique can be used, which only requires calculating $\hat{L_A}$ and $\hat{L_B}$:

$$\hat{L_{Aij}} = \frac{W_{A_{ij}}}{\sqrt{\hat{d}_i \hat{d}_j}} \qquad \hat{L_{Bij}} = \frac{W_{B_{ij}}}{\sqrt{\hat{d}_i \hat{d}_{j+|A|}}} \tag{2}$$

where $\hat{d} = \hat{W}\mathbf{1} = \begin{bmatrix} a_r + b_r \\ b_c + B^T A^{-1} b_r \end{bmatrix}$ and a_r represent the sum of the rows in A, b_c represents the sum of the columns in B y b_r represents the sum of the rows in B. If $\hat{L_A}$ is defined positive, the approximate orthogonal eigenvectors can be found in a single step. Be $\mathrm{S}=\hat{L_A} + \hat{L_A}^{-\frac{1}{2}} \hat{L_B} \hat{L_B}^T \hat{L_A}^{-\frac{1}{2}}$ and its diagonalization $\mathrm{S}=\mathrm{U}_S \Lambda_S U_S^T$, Fowkles et al [11] showed that if a matrix V if defined as:

$$V = \begin{bmatrix} \hat{L_A} \\ \hat{L_B}^T \end{bmatrix} \hat{L_A}^{-\frac{1}{2}} U_S \Lambda_S^{-\frac{1}{2}} \tag{3}$$

$\hat{LN}$ is diagonalized by V and by Λ_S and $V^T V = I$

2.2 Co-regularization

When the dataset has more than one representation, these representations are referred to as "views". In the context of spectral clustering, co-regularization techniques help support similarities in the examples in the new representation generated from the eigenvectors of each of the views. Let $\mathrm{X}^{(v)} = \{x_1^{(v)}, x_2^{(v)}, ..., x_n^{(v)}\}$ be the examples for view V y $\mathrm{L}^{(v)}$ the Laplacian matrix created from X for view V. We define $\mathrm{U}^{(v)}$ as the matrix formed by the first k eigenvectors corresponding to matrix $\mathrm{L}^{(v)}$ accordance with (1).

In [14] a criterion to measure the dissimilarity degree between two representations was proposed:

$$\mathrm{D}(\mathrm{U}^{(v)}, U^{(w)}) = \left|\left| \frac{K_{U^{(v)}}}{\left|\left|K_{U^{(v)}}\right|\right|_F} - \frac{K_{U^{(w)}}}{\left|\left|K_{U^{(w)}}\right|\right|_F} \right|\right|_F^2$$

Where $K_{U^{(v)}}$ is the similarity matrix generated from the patterns in the new representation $\mathrm{U}^{(v)}$ and $||\cdot||_F$ is Frobenius norm. If the inner product between the vectors is used as a similarity measure, we obtain

$\mathrm{K}_{U^{(v)}} = U^{(v)}U^{(v)^T}$. Ignoring additive and escalation constants, the previous equation can be formulated as follows:

$$\mathrm{D}(\mathrm{U}^{(v)}, U^{(w)}) = -Tr\left(U^{(v)}U^{(v)^T}U^{(w)}U^{(w)^T}\right)$$

The goal is minimizing the level of disagreement among the representations obtained from each view. Therefore, the following optimization problem is obtained that combines the individual objectives of spectral clustering and the objective that determines the disagreement among representations:

$$\begin{array}{ll} \underset{\substack{U^{(v)} \in R^{n\times k} \\ U^{(w)} \in R^{n\times k}}}{\text{máx}} & Tr\left(U^{(v)^T}L^{(v)}U^{(v)}\right) + Tr\left(U^{(w)^T}L^{(w)}U^{(w)}\right) + \\ & \lambda Tr\left(U^{(v)}U^{(v)^T}U^{(w)}U^{(w)^T}\right) \\ \text{s.t.} & U^{(v)^T}U^{(v)} = I \\ & U^{(w)^T}U^{(w)} = I \end{array} \quad (4)$$

The parameter λ balances the objective of spectral clustering and the disagreement among representations. The joint optimization problem can be solved by using alternating maximization with respect $\mathrm{U}^{(v)}$ and $\mathrm{U}^{(w)}$. For a given $\mathrm{U}^{(w)}$ the following optimization problem is obtained for $\mathrm{U}^{(v)}$:

$$\begin{array}{ll} \underset{U^{(v)}\in R^{n\times k}}{\text{máx}} & Tr\left(U^{(v)^T}\left(L^{(v)} + \lambda U^{(w)}U^{(w)^T}\right)U^{(v)}\right) \\ \text{s.t.} & U^{(w)^T}U^{(w)} = I \end{array} \quad (5)$$

Which results in a traditional clustering algorithm with the modified Laplacian matrix $\mathrm{L}^{(v)} + \lambda U^{(w)}U^{(w)^T}$

3. Method Proposed

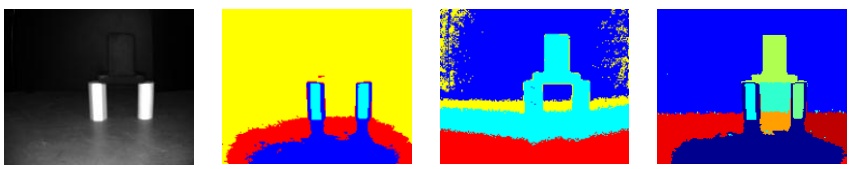
Let I be an amplitude image and be R a dimension range image $\mathrm{n}\times m$, both for the same scene.

1. From I and R, the approximate Laplacian matrixes $\hat{L_I}$ and $\hat{L_R}$ are obtained, as described in (2).
2. Let $\hat{V_I}$ be the approximate eigenvectors for $\hat{L_I}$ calculated in accordance with (3).
3. $\hat{V_R}$, the eigenvectors of the modified Laplacian matrix $\hat{L_R} + \lambda\hat{V_I}\hat{V_I}^T$ (5) are obtained using method (2).

4. V_I, the eigenvectors of the modified Laplacian matrix $\hat{L}_I + \lambda \hat{V}_R \hat{V}_R^T$ (5) are obtained using method (2).
5. $V = [V_I \quad V_R]$
6. A clustering algorithm is applied to V.
7. The criterion proposed in [8] is used to assess segmentation algorithm performance. If performance improves, then go to 3; if not, end.

4. Experimental Results

The performance of the segmentation algorithm proposed was assessed over 50 captured images using the MESA SwissRanger SR4000 time-of-flight camera [6] and the entire dataset provided by the Laboratory of Multimedia Technology and Telecommunications of the University of Padua [20].



(a) (b) (c) (d)

Figure 1. Segmentation applied to an image captured with the SwissRanger SR4000 time-of-flight camera. (a) Amplitude image of a real scene. (b) [11] method applied to the amplitude image. H=0.24 (c) [11] method applied to the range image. H=0.18 (d) Proposed method $\lambda = 3$. H=0.29.

The MESA SwissRanger SR4000 time-of-flight camera provides two images: an amplitude image and a range image, both with 144 x 176 pixels. The dataset [20] contains images captured with a time-of-flight camera and a traditional RGB camera. The performance of the segmentation algorithm was assessed using the criterion proposed in [8] and [18], which we called H.
The similarity function used in every case takes into account the spatial distribution of pixels in the image and the difference between their values:

$$W(i,j) = e^{-\frac{\|X_{(i)} - X_{(j)}\|_1^2}{sX}} * e^{-\frac{\|F_{(i)} - F_{(j)}\|_1^2}{sY}}$$

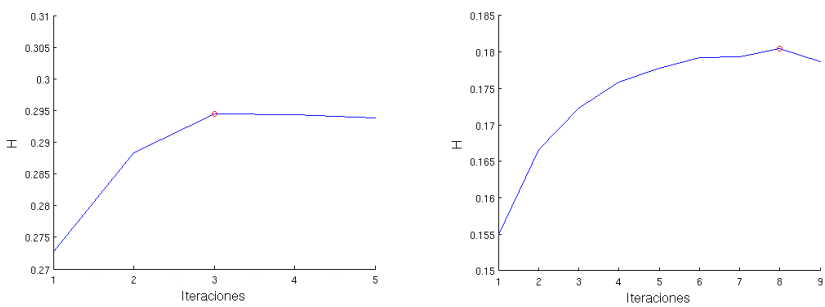
where $X(i)$ is the spatial location of pixel i, $F(i)$ is the value of the i-th pixel in the image, $sX = E(\|X_{(i)} - X_{(j)}\|_1^2) + \frac{3}{4}\sigma(\|X_{(i)} - X_{(j)}\|_1^2)$ of the pixels in set A and $sY = E(\|F_{(i)} - F_{(j)}\|_1^2)$ of the pixels in set A. Figure 1 shows the experimental results of the method proposed applied to an image obtained with the MESA SwissRanger SR4000 time-of-flight camera. The amplitude image of the captured scene 1(a) presents 3 objects over a black background, all at the same distance. One of the objects has an intensity level that is similar to that of

the background, which makes segmentation difficult. Since all objects at the front of the image are at the same distance, their range values will be similar. Figures 1(b) and 1(c) show the result of applying method [11] to the amplitude image and the range image, respectively. Figure 1(d) shows the result of applying the proposed method at the optimal operation point. The method correctly combines the information from both noisy images to segment the objects found in the scene. Figure 3(a) shows the performance assessed for each iteration of the algorithm. Figure 2 shows the result of applying the proposed algorithm to a scene in the dataset provided by the University of Padua. Figure 2(a) shows the amplitude image for the scene. Figures 2(b) and 2(c) show the result of applying algorithm [11] to the amplitude and range images. Separately, both images do not provide the necessary information to extract all objects in the scene.

(a) (b) (c) (d)

Figure 2. Segmentation of an image of the dataset provided by the University of Padua. (a) Amplitude image of a real scene (b) H=0.11 (c) H=0.13 (d) Proposed method using $\lambda = 3$. H=0.18.

The proposed method, through co-regularization, is successful in extracting the useful information from both images, maximizing segmentation performance, as Figure 2(d) shows. The performance assessed in each iteration of the algorithm is shown in Figure 3(b).



(a) (b)

Figure 3. Performance in relation to the number of iterations

5. Conclusions

In this article, we presented a clustering method for segmenting images captured by TOF cameras. The preliminary results obtained both on intensity and range images are satisfactory. The algorithm correctly combines the information provided by both images, even in the presence of noise, by using co-regularization techniques. The performance obtained was better when using semi-supervised learning instead of using concatenated characteristics in all tested cases.
In a future stage, we plan adding color information to the segmentation algorithm. Also, we will consider the convenience of using an alternative disparity measure.

Bibliography

1. Nicholas Arcolano. *Approximation of positive semidefinite matrices using the Nyström method*. PhD thesis, Harvard University, 2011.
2. N. Blanc, T. Oggier, G. Gruener, J. Weingarten, A Codourey, and P. Seitz. Miniaturized smart cameras for 3d-imaging in real-time [mobile robot applications]. In *Sensors, 2004. Proceedings of IEEE*, pages 471-474 vol.1, Oct 2004.
3. Amit Bleiweiss and Michael Werman. Fusing time-of-flight depth and color for real-time segmentation and tracking. In *Dynamic 3D Imaging*, pages 58-69. Springer, 2009.
4. Hanneke Borgdorff, Evgeni Tsivtsivadze, Rita Verhelst, Massimo Marzorati, Suzanne Jurriaans, Gilles F Ndayisaba, Frank H Schuren, and Janneke HHM van de Wijgert. Lactobacillus-dominated cervicovaginal microbiota associated with reduced hiv/sti prevalence and genital hiv viral load in african women. Nature Publishing Group, 2014.
5. John Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679-698, Nov 1986.
6. Miguel Cazorla, Diego Viejo, and Cristina Pomares. Study of the sr 4000 camera. In *XI Workshop de Agentes FÃ□sicos*, 2004.
7. F. Chiabrando, D. Piatti, and F. Rinaudo. R-4000 tof camera: Further experimental tests and first applications to metric surveys. In *Proceedings of V Symposium on Remote Sensing and Spatial Information Sciences*, pages 149-154, 2010.
8. C. Dal Mutto, P. Zanuttigh, and G.M. Cortelazzo. Fusion of geometry and color information for scene segmentation. *Selected Topics in Signal Processing, IEEE Journal of*, 6(5):505-521, Sept 2012.
9. G. Danciu, M. Ivanovici, and V. Buzuloiu. Improved contours for tof cameras based on vicinity logic operations. In *Optimization of Electrical and Electronic Equipment (OPTIM), 2010 12th International Conference on*, pages 989-992, May 2010.
10. Adrian A Dorrington, Cameron DB Kelly, Shane H McClure, Andrew D Payne, and Michael J Cree. Advantages of 3d time-of-flight range imaging cameras in machine vision applications. In *Proceedings of 16th Electronics New Zealand Conference*, pages 95-99, 2009.

11. Charless Fowlkes, Serge Belongie, Fan Chung, and Jitendra Malik. Spectral grouping using the nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214-225, Feburary 2004.
12. Guruprasad M Hegde and Cang Ye. A recursive planar feature extraction method for 3d range data segmentation. In *Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on*, pages 3119-3124. IEEE, 2011.
13. Dirk Holz, Stefan Holzer, Radu Bogdan Rusu, and Sven Behnke. Real-time plane segmentation using rgb-d cameras. In *RoboCup 2011: Robot Soccer World Cup XV*, pages 306-317. Springer, 2012.
14. Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In J. Shawe-Taylor, R.S. Zemel, P.L. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1413-1421. Curran Associates, Inc., 2011.
15. Abhishek Kumar, Avishek Saha, and Hal Daume. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 478-486, 2010.
16. Luciano Lorenti and Javier Giacomantone. Segmentación espectral de imágenes utilizando cámaras de tiempo de vuelo. In *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.
17. Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849-856. MIT Press, 2001.
18. C. Rosenberger and K. Chehdi. Genetic fusion: application to multi-components image segmentation. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 6, pages 2223-2226 vol.4, 2000.
19. J. Shi and J. Malik. Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings, 1997 IEEE Computer Society Conference on*, pages 731-737, Jun 1997.
20. Multimedia Technology and University of Padova Telecommunications Laboratory. Joint color and depth segmentation datasets, July 2014.
21. R. Wu, Z. y Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101-1113, Nov 1993.

Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1413-1421. Curran Associates, Inc., 2011.
15. Abhishek Kumar, Avishek Saha, and Hal Daume. Co-regularization based semi-supervised domain adaptation. In *Advances in Neural Information Processing Systems*, pages 478-486, 2010.
16. Luciano Lorenti and Javier Giacomantone. Segmentación espectral de imágenes utilizando cámaras de tiempo de vuelo. In *XVIII Congreso Argentino de Ciencias de la Computación*, 2013.
17. Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849-856. MIT Press, 2001.
18. C. Rosenberger and K. Chehdi. Genetic fusion: application to multi-components image segmentation. In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 6, pages 2223-2226 vol.4, 2000.
19. J. Shi and J. Malik. Normalized cuts and image segmentation. In *Computer Vision and Pattern Recognition, 1997. Proceedings, 1997 IEEE Computer Society Conference on*, pages 731-737, Jun 1997.
20. Multimedia Technology and University of Padova Telecommunications Laboratory. Joint color and depth segmentation datasets, July 2014.
21. R. Wu, Z. y Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101-1113, Nov 1993.

Biometric Iris Identification in Bovines

JUAN I. LARREGUI[1], JOAQUÍN ESPINOSA[1], M. LUJÁN GANUZA[1], SILVIA M. CASTRO[1]

[1] VygLab, Lab. De Visualización y Computación Gráfica,
Dpto. De Cs. e Ing. De la Computación,
Universidad Nacional del Sur. Bahía Blanca, Buenos Aires, Argentina.
juan.larregui@cs.uns.edu.ar, joaco.espinosa@gmail.com,
{mlg, smc}@cs.uns.edu.ar

Abstract. Animal identification has become a topic of increasing worldwide importance as this method is desirable for genetic and performance monitoring and other applications such as animal health and food safety. In particular, bovine identification aims to avoid the problems caused by the intentional or unintentional contamination of food. This paper proposes a complete biometric iris identification method for bovines, based on images that could be taken with a regular camera in a farm. The iris region of the bovine eye is segmented and analyzed to recognize the animal. Experimental results show the effectiveness of the method.

1. Introduction

Animal Identification has become a very important topic nowadays. This stems from the fact that animals suffering from diseases might result in contaminated food, which in turn, could affect human beings. Performance animals, such as horses and dogs, need also be controlled as they are constantly involved in business negotiations where fraud and deception might take place. In addition, vulnerable, endangered and critically endangered animal species can be protected if they are identified and controlled. Another important field of application for Animal Identification is Zoos. These facilities identify each animal, create detailed histories and record all health, legal and behavioural information.
In recent years Bovine Identification has become a topic of intense worldwide interest in the aftermath of terrorist incidents, outbreaks of bovine spongiform encephalitis (BSE) and, more recently, reports of E. coli contamination in beef [4]. In fact, food could be intentionally contaminated as a terrorist act. The most efficient and effective way of countering all emergencies including food terrorism, is through sensible precautions coupled with strong surveillance. Bovine Identification is a means to implement the mentioned surveillance and there are a number of benefits that come with it such as routine animal health management, animal disease control programmes, enhanced breeding programmes and ensuring market access.

Bovine Identification is based on different methods such as ear, neck and brisket tags, tattoos, hot and cold branding and ear notches. Some of these methods are susceptible to loss, theft and tampering, while others might be subject to illegibility and mutilation concerns. Bovines can also be identified using DNA fingerprinting. However, three major drawbacks are present in this method: it is expensive, slow and can be defeated by cloning. Lastly, electronic devices can be used to achieve Bovine Identification. These can be attached to all the aforementioned tags. In that case, they fall under the category of External Electronic Identification Devices (External EIDs) and the same disadvantages can be attributed. Internal EIDs refer to injectable and ingestible devices. Both methods share two important drawbacks: they are invasive and their recovery during slaughter is very complex and may result in wounding the animal. All mentioned techniques have important disadvantages and therefore, a non-invasive, precise and accurate method is needed for animal identification.

Biometric Iris Recognition in bovines is an alternative to the aforementioned methods which is based on the iris' random patterns and other particular attributes that have been shown capable of generating highly unique identification codes. This method is precise and hard to deceive. In addition, it is fast and inexpensive when compared to the previously mentioned identification techniques. More importantly, it is non-invasive and therefore, the animals' physical integrity is ensured.

Given the presented analysis, a Bovine Biometric Iris Identification and Recognition system based on Image Processing is proposed.

2. Previous Work

A number of contributions related to this field of application have been presented in recent years. In 2009, Zhang *et al.* [12] proposed an iris localization algorithm that is based on geometrical features of cow eyes. In this contribution, the iris is approximated by using a Sobel edge detection operator and the coordinates of boundary points are determined with quadratic B-spline interpolation curves. Next, two circles are created which do not exactly delimit the pupil and iris of the cow but instead isolate part of the eye's iris. This work proposes a localization algorithm but does not solve the segmentation and recognition problems in bovines. In the same year, Zhao *et al.* [13] proposed a recognition method which uses a iris localization scheme almost identical to the one described above. In this contribution, however, a feature extraction algorithm is created using 2D Gabor filters and a matching test is implemented using Boolean XORs. In December 2009, Sun *et al.* [10] presented an iris recognition method which first detects the iris inner and outer borders using the Snake model introduced by Chan and Vese [5] and store the resulting ring belt. Then, 2D Gabor wavelets are employed to describe the isolated iris data and Boolean Exclusive-ORs are used for recognition purposes. In 2012, Sun and Zhao [9] proposed a method for segmenting the iris using a global and local region-based active contour model. Firstly, the iris and pupil contours are found using and active contours model. Secondly, a pair of concentric ellipses is used to approach these contours and then

the iris is regularized into an elliptic ring. Next, part of the elliptic ring is reshaped into a rectangle with polar coordinate transformation. In 2013, a recognition scheme based on Scale Invariant Feature Transform (SIFT) and bag of features was presented by Sun *et al.* [8]. Once the pupil is localized using region-based active contours, SIFT algorithm is employed to detect key-points in the iris image and to generate a descriptor. Next, a feature vocabulary is constructed and a histogram representation is generated for each image. Then, histogram distance is used for the matching test. All the previously mentioned contributions consider the images stored in the SEU database. On these images, the iris covers most of each image and is well defined. Furthermore, access to this database is limited and the circumstances under which these photos were taken are not disclosed. In 2014, Lu *et al.* [6] proposed a new cow identification method. Once the captured images are properly selected, the pupil and iris boundaries are fitted respectively as two ellipses based on the edge images during segmentation. Then, the segmented cow iris is normalized using a geometric method. Finally, 2D complex wavelet transform (2D-CWT) is used to extract local and global characteristics of the cow iris and the phase of the filtered cow iris is encoded as features.
This work proposes a system which will be referred to as Bovine Identification System, or BIS. Given images of bovines' eyes, the BIS offers some wanted features, namely: (i) the iris region on the input images does not need to be centered; (ii) the iris region does not need to cover most of the image; (iii) the input images can be captured using a regular visible-light camera on a farm; (iv) iris and pupil segmentation is achieved on images without previous enhancement.

3. Bovine Eye Description

The eye of the bovine consists of several distinctive components. This paper, only focuses on three: sclera, iris and pupil (Fig. **¡Error! No se encuentra el origen de la referencia.**). Like other ungulates, bovines have pupils that resemble a transverse ellipse. The pupil is defined as a light receptive opening which allows light to enter the eye and fall on the retina. As indicated by Budras *et al.* [3], when the pupil dilates, it becomes rounder, resembling a circle. The black projections (*granula iridica*) on the upper and lower margins of the pupil are vascular appendages covered by pigmented epithelium from the back of the iris. The iris is an asymmetrical structure in the eye that may be irregularly pigmented. The bovine iris is dark because of the heavy pigmentation of the posterior epithelium (*pars iridica retinae*). This structure is characterized by texture information, including pigment and structural related features, as well as several features relating to the function of controlling the size of the pupil that are visible from the exterior of the eye. It has random patterns of striations, ciliary processes, crypts, rings, furrows and other features which had been shown capable of generating highly unique iris images for identification purposes [7]. The last element to be considered is the sclera. This is part of the fibrous tonic which encloses part of the bulb in its dense white connective tissue and the transparent cornea.

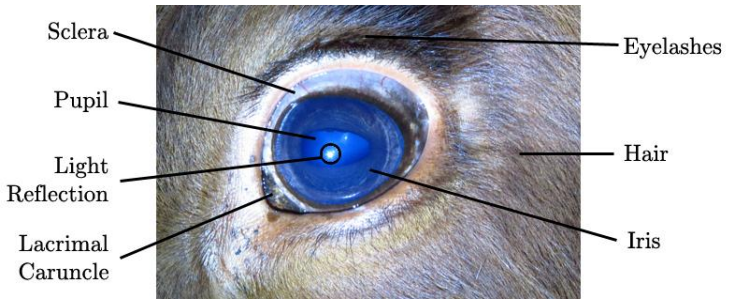


Fig. 1. Bovine Eye Description.

4. Bovine Identification System (BIS)

This paper proposes a biometric identification and recognition system for bovines based on iris features obtained from images. It works in two different modes: Training Mode and Test Mode. In Training Mode, the system needs to be filled with images from every animal to be identified. These images constitute the Training Set, and will be stored in a database. For each bovine, the system will generate and store a unique ID code. Therefore, once these images are uploaded, every animal will be registered with its associated ID. In Test Mode, new images from any animal can be entered into the system. These images constitute the Test Set. Every new image will then be analyzed and processed. Once the processing is completed, the BIS will provide one of two output results: (i) a Unique ID code corresponding to the identified bovine; (ii) an error code indicating that the animal was not found in the system.

4.1 Challenges

It is essential for any biometric recognition method to identify the regions of interest (ROIs) that are going to be analyzed. This implies that the system needs to be capable of determining if a given identified component is useful or not. Several iris segmentation methods in human beings have shown their effectiveness. However, these methods cannot be applied on bovines due to the differences in the anatomy of the eye. Human iris segmentation methods rely on the fact that human iris and pupil borders are round and can be considered as two concentric circles. Conversely, bovine iris and pupil borders are highly asymmetrical and eccentric, and therefore, such assumption cannot be made.

Identifying the ROIs of the bovine eye becomes a complex task due to the presence of hair on the input images as its colour is sometimes similar to the colour of the pupil or the colour of the iris. Besides that, the eyelashes might cover part of the eye ball thus hiding the iris' inner and/or outer borders and also hiding sensitive information useful for description purposes.

In addition to finding the ROIs, the system must be capable of representing and describing the iris data in order to identify each bovine. The description will be based on the information residing in the region of interest. The representation will in turn be based on each description.
At last, a method for comparing bovines must be designed. The system must be able to compare each animal's representation in order to return a result depending on whether the animal was identified and recognized or not.

4.2 BIS' Pipeline

Next, the different stages of the Bovine Identification System's pipeline are detailed (Fig.**¡Error! No se encuentra el origen de la referencia.**).

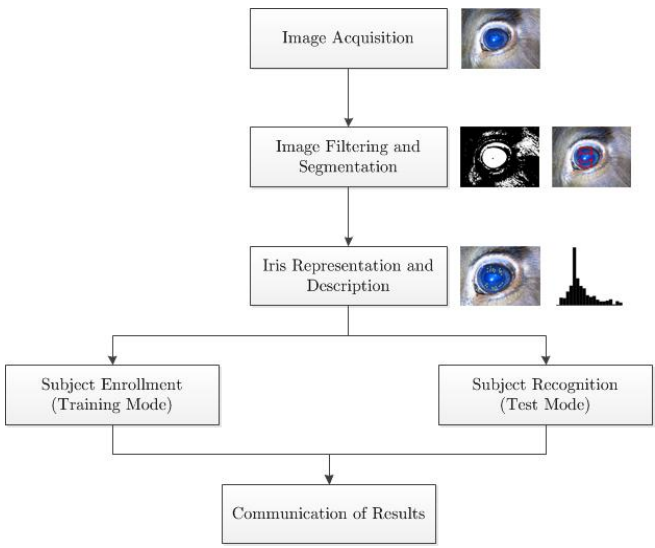


Fig. 2. The BIS' pipeline.

Image Acquisition. For practical purposes, input images for the BIS consist of photos obtained using a regular visible light camera. Because of the non-cooperative behavior of bovines, these images can be taken from a cautious position. In addition, the iris region does not need to cover most of the image, and therefore, the number of non-suitable images in which the iris is not entirely seen is reduced.
Input images for the BIS could either be used in Training Mode, adding new iris data to the database, or in Test Mode, testing against stored data to determine if the iris description is already in the database in order to identify the bovine.

Image Filtering and Segmentation. Image filtering and segmentation of the region of interest (ROI) works in the same way in both system modes (Fig.**¡Error!**

No se encuentra el origen de la referencia.). Initially, three matrices are created based on the input image. These are referred to as the Red Matrix, the Green Matrix and the Blue Matrix and store the information residing in the red, green and blue channels of the image respectively.
The first goal is to find the iris' outer border which separates the iris from the sclera. This border is highlighted by calculating the difference between the Blue matrix and the Green matrix. The resulting image is then filtered using an adaptive threshold with a specific threshold value based on the size of the image. In addition, a morphological opening operation is applied by eroding and dilating the image. Next, a search is conducted to look for all contours with a length bigger than a Minimum Length Value using the algorithm presented by Suzuki and Abe in 1985 [11]. The goal here is to ignore all small contours found due to the presence of hair. Then, for each found contour, an ellipse based on that contour is generated. Every contour is then compared with its associated ellipse and a Match Value is calculated. The goal is to disregard all contours which are not ellipsoidal. This choice is based on the bovine eye anatomy. Consequently, only contours with a Match Value lower than a Maximum Acceptable Mismatch Value are considered. To find the iris, we look for the largest contour. This is the contour for the iris' outer border.
Once the iris has been found, the system starts looking for the iris' inner border which divides the iris from the pupil. First, the difference between the Blue Matrix and the Red Matrix is calculated to darken the iris while barely affecting the pupil. This follows from the fact that the iris and the pupil of a bovine are mostly blue but there is more red information on the iris than on the pupil. As a result, these regions become more distinct and the border that divides them is highlighted. Moving forward, the new image is filtered using an adaptive threshold with a specific threshold value. The image is then eroded and dilated. Since the iris outer border has already been found, a search is conducted to look for all contours inside it. Lastly, we look for the contour closer to the iris' centroid that also encloses the iris' centroid. This is the contour for the iris' inner border.
Once the iris' inner and outer borders are obtained, a mask is generated in such a way that the only information available after applying this mask corresponds to the set of pixel data representing an upper region of interest (uROI) of the iris region above the pupil region, and a lower region of interest (lROI) of the iris region below the pupil region (Fig. **¡Error! No se encuentra el origen de la referencia.**). This selection is based on the bovine eye characteristics given that, from an anatomical point of view, these regions contain the most representative information of the iris [7].

Iris Representation and Description. This stage is based on the SURF (Speeded Up Robust Features) [2] algorithm for feature detection. In both system modes, the original image is converted to gray-scale and its histogram is equalized. Next, both the mask generated in the previous stage and the processed image are provided to the SURF key-point detector. As a result, a set of key-points is obtained which correspond to the iris region of the treated image. Subsequently, a descriptor for each key-point is generated by the SURF descriptor extractor. The generated descriptors are stored in per-image matrices

which represent the characteristic features of each iris. Then, in Training Mode, a set of clusters is created based on all descriptors for all training images. This is accomplished by using the K-Means Clustering method

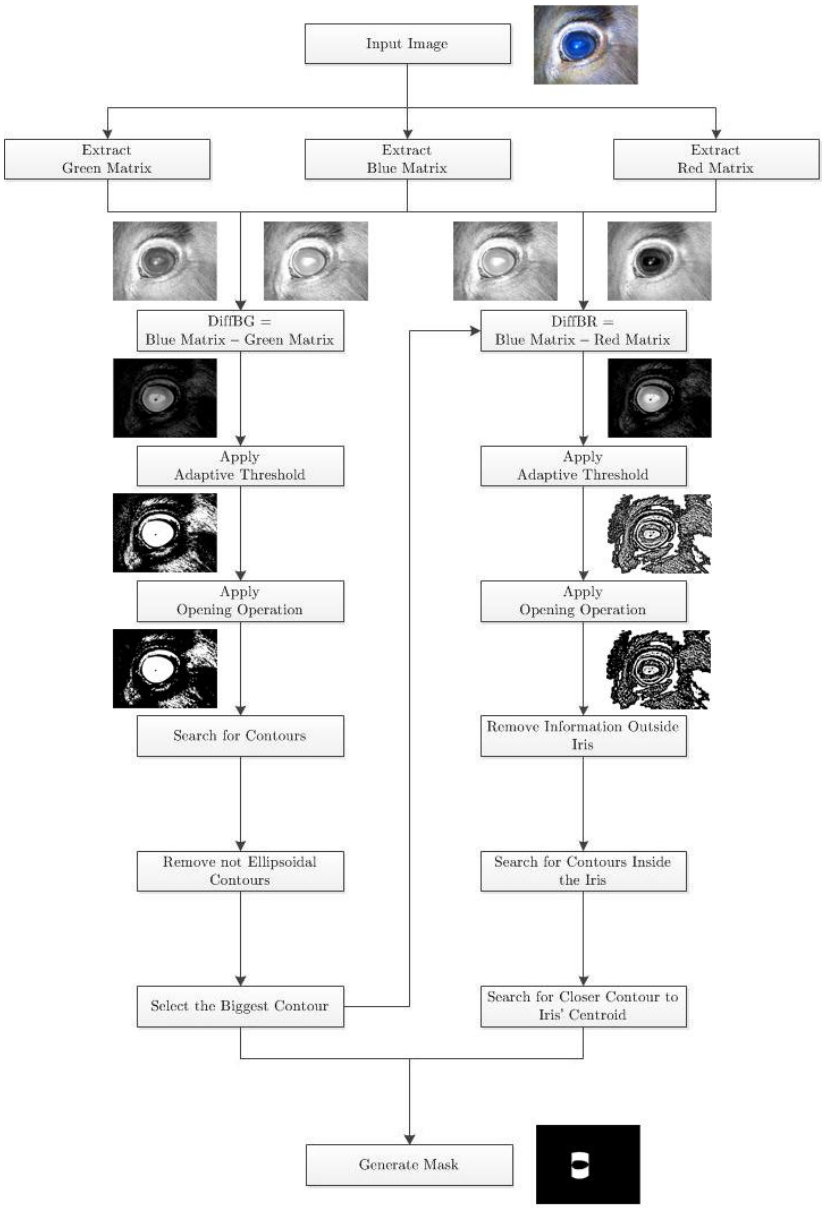


Fig. 3. Image Filtering and Segmentation algorithm.

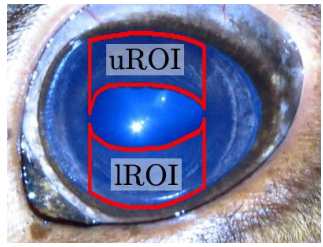


Fig. 4. Upper ROI (uROI) and lower ROI (lROI).

which ultimately calculates the centroid for each generated cluster. The algorithm is initialized using the K-Means++ method [1]. These centroids will be used to classify both training and test images. Moving forward, in both Training and Test modes, the following operations are carried out:

1. All descriptors for the image are classified using the aforementioned centroids and the K-Nearest algorithm.
2. Once all descriptors are classified into clusters, the ratio of features per cluster is obtained for each cluster.
3. These obtained proportions are then used to create a matrix of (1 x No. of Clusters) elements which will be used as the classification histogram for the iris.

Subject Enrolment. In Training Mode, once the classification histogram is calculated, it is stored in the database. A unique ID code is associated with the histogram representing the code that identifies the bovine in the database.

Subject Recognition. In Test Mode, once the classification histogram is calculated, it is compared with each classification histogram for training images that was stored in the Subject Enrolment stage. This is carried out by obtaining the quadratic-chi distance between the stored histogram and the test histogram. If none of the previous calculated distances is smaller than a Maximum Acceptable Distance value, the BIS returns an error code. Otherwise, the stored histogram with the minimum quadratic-chi distance to the test histogram is considered to belong to an iris image from the same animal as the one from the test image.

Communication of Results. In the last stage of the pipeline, the system communicates the obtained results. In Training Mode, it returns the unique ID code generated for the bovine. In Test Mode, it either returns the ID code associated to the histogram selected in the Subject Recognition stage, or a proper error code.

4.3 Implementation

The identification system was implemented in the C++ programming language, using the Visual Studio development environment. For image processing operations, the OpenCV 2.4.6 open source library was used.

5. Experimental Results

The proposed methodology was applied to an image database consisting of 48 eye images from 8 different Aberdeen Angus breed bovines. The Training Set was integrated by 16 images, while the remaining 32 images represented the Test Set. Images were obtained on a farm, using a Canon PowerShot SX40 HS camera. These images were captured in the evening in order to avoid sunlight reflections on the animal's eyes which could hide useful information. The target was illuminated using a visible light LED flashlight source positioned approximately two meters away from the eye and pointed directly into the pupil of the subject so that the facula lies on the animal's pupil, and therefore, no useful information stored in the iris is lost. Examples of obtained images can be observed in Fig. **¡Error! No se encuentra el origen de la referencia.**.

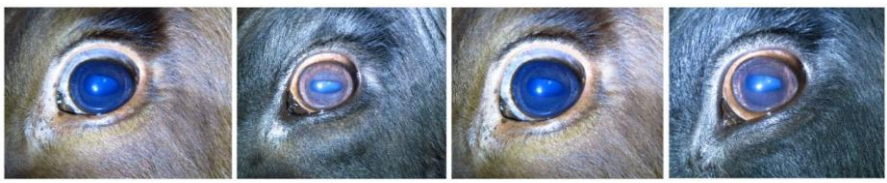


Fig. 5. Samples from BIS database.

Segmentation and Recognition Results. The algorithm properly created the region of interest (ROI) mask of the iris for each image on the entire Training Set. When processing test images, the segmentation was successful in 28 images out of 32, making 44 correctly segmented images from a total of 48. Thus, an accuracy of 91.67% was achieved. Only properly segmented images were considered in the Recognition experiments. Then, the classification histograms for the 44 properly segmented images were calculated. Each histogram for test images was compared with each histogram for training images, successfully recognizing the animal in 24 images from the total of 28, with an accuracy of 85.71%.

6. Conclusions and Future Work

In this paper, a bovine iris recognition system is proposed. The iris inner and outer borders are segmented using different filtering techniques and conducting a search looking for relevant contours. SURF is used for description while K-Means Clustering and K-Nearest Neighbours are used for classification. Finally, a precise comparison method is employed to recognize and identify each animal. Existing recognition methods achieve greater accuracy [6, 8, 9, 12]. However, the proposed approach is proven to be effective, as shown in Section 5, considering that images are taken with a regular camera, farther from the animal's eye, in the field and without any previous enhancement. Furthermore, it deals with the problem of iris segmentation when the iris does not cover the most of the image. Since the results are based on a limited number of photos, these experiments will be conducted on a database containing a larger number of bovine iris images.

Acknowledgments. The authors gratefully acknowledge the help provided by Eduardo and Paula Mainini, Raúl Larregui, Roberto Espinosa, Carlos Rodríguez, Cristian Kleine and Estancia Hogar Funke during the image acquisition process. The authors would like to thank Laura Popp for providing the imaging apparatus used to capture high-quality iris images. We would also like to acknowledge the support from Secretaría General de Ciencia y Tecnología, Universidad Nacional del Sur, Bahía Blanca, Argentina. This research is partially supported by the PGI 24/N028.

References

1. David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
2. Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Computer Vision–ECCV 2006, pages 404-417. Springer, 2006.
3. Klaus Dieter Budras, Robert E Habel, Christoph KW Mülling, and Paul R Greenough. Bovine anatomy: an illustrated text. Manson Publishing, 2011.
4. James L Cambier. System and method for animal identification using iris images, November 20 2012. US Patent 8,315,440.
5. Tony F Chan and Luminita A Vese. Active contours without edges. Image processing, IEEE transactions on, 10(2):266-277, 2001.
6. Yue Lu, Xiaofu He, Ying Wen, and Patrick SP Wang. A new cow identification system based on iris analysis and recognition. International Journal of Biometrics, 6(1):18-32, 2014.
7. Clyde Musgrave and James L Cambier. System and method of animal identification and animal transaction authorization using iris patterns, July 23 2002. US Patent 6,424,727.
8. Shengnan Sun, Shicai Yang, and Lindu Zhao. Noncooperative bovine iris recognition via sift. Neurocomputing, 120:310-317, 2013.

9. Shengnan Sun and Lindu Zhao. Bovine iris segmentation using region-based active contour model. International Journal of Innovative Computing, Information and Control, 8(9):6461-6471, 2012.
10. Shengnan Sun, Lindu Zhao, Qiang Kong, and Xiaoqiang Wang. Quality control of meat supply chain based on iris recognition. In Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on, pages 700-704. IEEE, 2009.
11. Satoshi Suzuki and Keiichi Abe. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32-46, 1985.
12. Menglu Zhang, Lindu Zhao, and Qiang Kong. An iris localization algorithm based on geometrical features of cow eyes. In Sixth International Symposium on Multispectral Image Processing and Pattern Recognition, pages 749517-749517. International Society for Optics and Photonics, 2009.
13. Lindu Zhao, Shengnan Sun, and Xiaoqiang Wang. Tracking and traceability system using livestock iris code in meat supply chain. International Journal of Innovative Computing, Information and Control, 7(5A):2201-2212, 2011.

XI

Software Engineering Workshop

Integrating Software Metrics for Fortran Legacy into an IDE

MARIANO MÉNDEZ AND FERNANDO G. TINETTI*

III-LIDI, Fac. de Informática
Universidad Nacional de La Plata
50 y 120, La Plata
Argentina
*CIC Prov. de Bs. As.

Abstract. Software Metrics have being used since the 70s, their purpose is to measure different software attributes, such as complexity and maintainability, to name a few. Software Metrics help programmers obtain valuable information about programs. That information is essential when working with legacy systems. Scientists have been producing Fortran programs for the last six decades, and some of those programs became legacy years ago. We have implemented a set of well known software metrics for Fortran into a widely used IDE (Integrated Development Environment) by means of AST (Abstract Syntax Tree). This integration allows developers to obtain software metrics from their source code while they are programming.

Keywords: Fortran Legacy Systems, Software Metrics, Integrated Development Environments.

1. Introduction

Software attributes measurement and software process improvement has been a research topic for many years. Improving software processes that can be tracked down [34, 23] while measuring software attributes has been deeply treated since the 70s [22, 17, 24]. In this paper, we will focus in software metrics: "A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possesses a given attribute that affects its quality" [18]. Most of the relevant studies focusing on software metrics initially concentrated on Fortran Programming Language [6, 22, 24], among others. Fortran is one of the most long lived programming languages [2] which has been widely used by scientists to produce large amounts of source code ever since it came into existence, it has become as the "de facto" scientific programming language [26, 3, 21, 28, 29]. Throughout its long lived existence it has experimented a particular evolutionary process since it was the first programming language to be standardized [2]. Fortran has passed through 10 different versions:

FORTRAN i, FORTRAN ii, FORTRAN iii, FORTRAN iv, FORTRAN 66, FORTRAN 77, Fortran 90, Fortran 95, Fortran 2003, and Fortran 2008. Six of those versions resulted in standards [19, 13, 20]. The Fortran standard has passed through several revision processes during the last decades (1966, 1978, 1991, 1997, 2004, 2010). As a matter of fact, Fortran is considered to have passed through a "seven ages" evolutionary process [27].
Fortran evolution has also been described as a pragmatic process, given the fact that updated versions of Fortran are fully backward compatible with older versions of the language. Updated versions have kept full compatibility even with obsolescent features already deleted from the language standard, with minor exceptions: "Unlike Fortran 90, Fortran 95 was not a superset; it deleted a small number of so-called obsolescent features. This incompatibility is more theoretical than real however, as all existing Fortran 95 compilers include the deleted features as extensions" [7]. Annex B of the Fortran 2008 Standard (ISO/IEC 2008) enumerates the obsolescent features of the language that have not been deleted, some of which may be found in the Fortran 66 and Fortran 77 specification. Fortran 2008 was released in 2010, and it is the current revision of the Fortran standard.
Taking into account the Fortran evolution described above, it seems to be paradoxical that there are a few tools implementing software metrics for Fortran [38, 37]. Furthermore, most of them (if not all) have closed source and commercial/paid licenses. In addition, there are no modern IDEs including metrics for the Fortran language. In this article we will contribute with an implementation for Fortran of some of the most widely used software metrics, and we will integrate them in Photran, "a Fortran IDE and Refactoring tool based on Eclipse" [36].

2. Software Metrics

As defined in [18], a software metric is "A function whose inputs are software data and whose output is a single numerical value that can be interpreted as the degree to which software possesses a given attribute that affects its quality". We will focus on a set of well known software metrics that have been used for the last 30 years by programmers and software scientists. All of them will be implemented to be used on Fortran programs.

2.1 Cyclomatic Complexity

In 1977, the "cyclomatic number V(G) of a graph G with v vertices, e edges, and connected components is V(G)=e−n+p " is introduced [24]. The proposal is to "measure and control the number of paths through a program", and it was called Cyclomatic Complexity (CC). In other words, the CC measures "the amount of decision logic in a source code function" [33]. Also, there was a tool to obtain the CC on PDP-10 Fortran programs [24]. One remarkable aspect about the CC is the possibility to determine a number or a threshold to characterize programs such as that suggested CC upper bound per routine: 10

[24, 33]. A set of thresholds were defined [12] related to software risk evaluation, as shown in Table 1. The CC is a widespread software metric even though has its own detractors [30, 31].

Table 1. Cyclomatic Complexity Range Description

Cyclomatic Complexity	Risk Evaluation
1-10	A simple module without much risk
11-20	A more complex module with moderate risk
21-50	A complex module of high risk
51 and greater	An untestable program of very high risk

We have implemented the CC to measure routine's complexity, and taking into account that the CC has different variations to be calculated, our approach is the same as that published in [4]. Each Fortran program is assumed to have a unique entry and exit point in order to simplify the complexity calculations: "For programs with unique entry and exit nodes, this metric is equivalent to one plus the number of decisions and in this work, is equal to the one plus sum of the following constructs: logical if, if-then-else, block-if, block if-then-else's, do loops, while loops, and's, or's, xor's, eqv's, neqv's, twice the number of arithmetic if's, n-1 decision counts for a computed Go To with n statement labels, and n decision counts for a case if with n predicates".

2.2 Halstead Complexity Metrics

A set of metrics derived from source code in order to measure computational complexity was proposed in the 70s [15, 17, 16]. A set of independent properties that could be obtained from an algorithm regardless of the programming language in which it had been written was defined, trying to determine if an algorithm "may possess a general structure which obeys physical laws". The proposed properties were based on classifying program entities as either operators and operands [15]. The properties are:

- Number of different operators occurring in an algorithm η_1.
- Number of different operands occurring in an algorithm η_2.
- The total occurrences of operators in the program N_1.
- The total occurrences of operands in the program N_2.

And the metrics are defined using those properties:

- Unique Entities, also know as Program Vocabulary, η: the number of unique operators plus the number of unique operands in the program. Calculated as $\eta=\eta_1+\eta_2$.

- Accumulated Entities, also know as Program Length, N: the total number of operators plus the total number of operands in the program. Obtained as $N=N_1+N_2$
- Program Volume, V: "consists of the product of the total number of entities occurring, N, and the number of binary digits which would be required to specify each such entity among distinct entities" [15]. Program Volume is calculated as $V=N\log_2\eta$.
- Level, L: "has a maximum value of unity, which will be reached whenever both the number of distinct operators is at its minimum value of two, and no operand has been mentioned more than once" [15]. Program Level is obtained as $L=2/\eta_1*\eta_1/N_2$.
- Difficulty, D: defined as $D=1/L$.
- Effort, E: defined as $E=V*D$

Just like CC, Halstead metrics have different variations to be calculated [6, 4], and our approach is the same as that published in [4].

2.3 Maintainability Index

While Software Maintenance were gaining momentum, software development practices were also maturing. Different authors argue that software maintenance represents the highest cost in the software development process. By 1975, "The total cost of maintaining a widely used program is typically 40 percent or more of the cost of developing it" [5]. In 1979, the software maintenance costs was reported to be 67% [35], in the 1990's this number was about 75% [10], and in the 2000's 90% of the entire project cost was calculated [11]. A maintenance index based on a three dimensional approach was proposed in [8], where the three dimensions are:

- Control structure.
- Information structure.
- Typography, naming and commenting.

The maintenance metric is a result of the analysis of different regression models. "The regression model that seemed most applicable was a four-metric polynomial based on Halstead's effort metric and on metrics measuring extended cyclomatic complexity, lines of code, and number of comments" [8]. As a result, the maintenance index was defined as:

$$171-3.2*\ln(V)-0.23*CC-16.2*\ln(LoC)+Comments$$

where V is the volume as defined in the previous section, CC is the Cyclomatic Complexity, and LoC is the number of Lines of Code. Our implementation is based on a slight modification, resulting in [39]:

$$MI=Max(0,(171-3.2*\ln(V)-0.23*CC-16.2*\ln(LoC))*100/171)$$

3. Fortran Software Metrics Implementation

In order to implement metrics described in the previous section, we have based our development on Photran [36], a multiplatform integrated development environment (IDE) for Fortran based on Eclipse. As an IDE, Photran integrates editing, source navigation, compilation, and debugging into a single tool. Furthermore, as it uses make for compilation, it can work with virtually any existing Fortran compiler. It is also equipped with error parsers, which help to interpret error messages from popular compilers and associate error markers with the appropriate lines of code [36].

Photran infrastructure uses a compiler-like approach to represent a Fortran program by building an Abstract Syntax Tree (AST) as a program representation. An AST "represents the hierarchical syntactic structure of the source program." [1]. Each construct occurring in the source code of the program is denoted on a tree node. It is called "abstract" because the AST does not represent every detail which appears in the real syntax, see Figure 1.

Figure 1. A Complete AST Structure for a helloworld.f90 Program

The Photran AST is based in the Fortran language definition, as expected, but also adds its own information, so that it is possible to modify the AST (which is why it is usually known as *rewritable* AST) an (re)generate the Fortran program.
In order to extract metrics from the source code of the Fortran programs we have built a set of java classes that implement the visitor pattern [14], as shown in Figure 2.

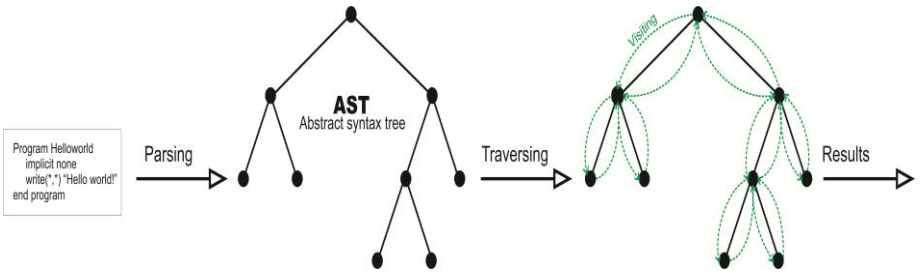
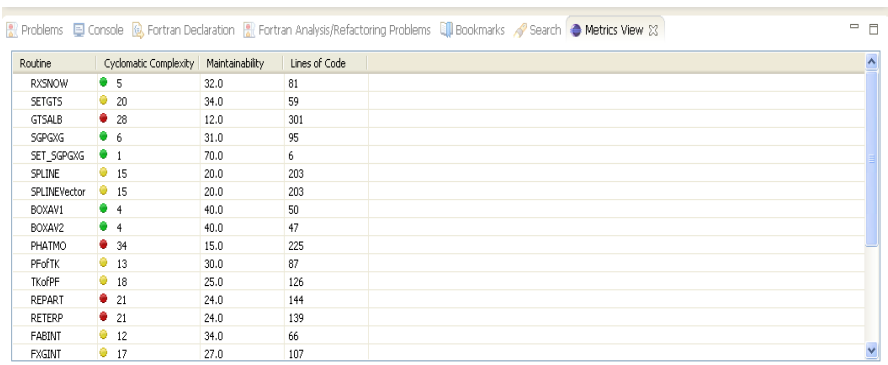


Figure 2. Gathering Fortran Metrics.

While traversing/*visiting* the AST we are able to collect as much information/data as we need. Thus, metrics are easily implemented by collecting data in the AST in a uniform way. This general strategy greatly simplifies the implementation of the metrics presented in this paper as well as new ones, depending on specific project and software analysis needs. Even when some data is nos fully contained in a single AST node, each visitor is implemented so that collects and aggregates data from the proper AST nodes, depending on the specific metric.
After the information about the selected editor is gathered it is shown in a specific IDE view called "Metric View". Thus, we take advantage of the so-called *Views* defined by Photran. The Metric View in Figure 3 shows three specific metrics: Cyclomatic Complexity, Maintainability Index and Source Lines of Code per routine. The information in the view is updated each time a new editor is selected. We have added a small colored circle (red, yellow, green) that is displayed in the CC column indicating the color according to the range shown in Table 1.



Routine	Cyclomatic Complexity	Maintainability	Lines of Code
RXSNOW	5	32.0	81
SETGTS	20	34.0	59
GTSALB	28	12.0	301
SGPGXG	6	31.0	95
SET_SGPGXG	1	70.0	6
SPLINE	15	20.0	203
SPLINEVector	15	20.0	203
BOXAV1	4	40.0	50
BOXAV2	4	40.0	47
PHATMO	34	15.0	225
PFofTK	13	30.0	87
TKofPF	18	25.0	126
REPART	21	24.0	144
RETERP	21	24.0	139
FABINT	12	34.0	66
FXGINT	17	27.0	107

Figure 3. The Metric View: CC, MI, and SLoC per Routine.

The information contained in the Metric View is consistent with the source code shown in the editor as shown in Figure 4. It is expected to aid the programmer to identify the location (routines) of potential problems regarding possible "bad smells" in the source code, such as too complex or too large routines, for example. Colors could be added to other columns (e.g. to the MI column) on demand, and it would change only the Metric View (or, more specifically, the corresponding column in the Metric View).

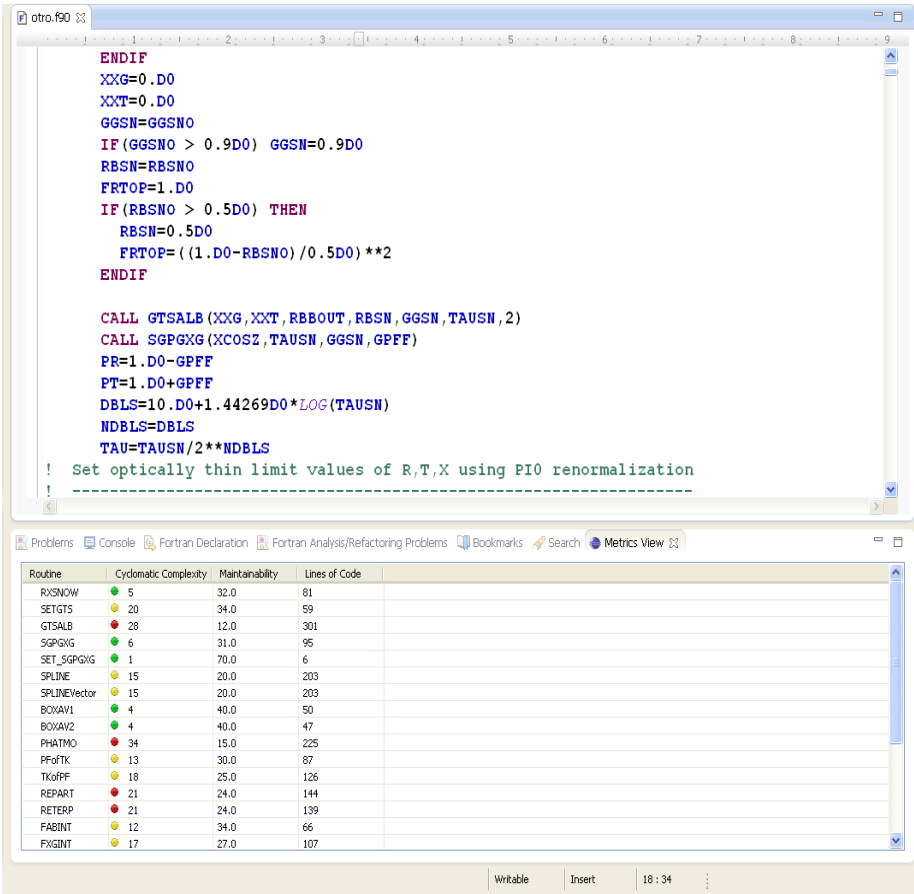


Figure 4. The Selected Fortran Editor and the Corresponding Metric View

4. Related Work

Integrating metrics data into a broadly used IDE is a relatively new idea. On the one hand Microsoft Visual Studio (which was initially known as Visual Studio 97 starting on May 1th, 1997 and passed through at least 8 product versions), has included metrics on its last 4 versions (Visual Studio 2008, 2010, 2012, 2013). On the other hand, Eclipse (from which Photran is derived) does not provide software metric data information in any of its distributions, but there are a set of plugins that allow the user to gather metrics data for Java and other supported programming languages.
Regarding software implemented using Fortran, available tools to obtain software metrics are not integrated to an IDE, and they are stand alone tools [38, 37]. Fortran development *tradition* has been more *naturally* associated to a development tool chain instead of using an IDE (such as Photran).

5. Conclusions and Further Work

We have shown an implementation and integration into a modern IDE for the Fortran programming language of some of the most widely used software metrics. The metric gathering process has been implemented with a compiler-like approach using ASTs. The obtained information has been integrated in a Photran (an Eclipse derivation) view in order to make it available to the programmer while performing all the regular programming activities.
As our main interest is focused on Fortran source code transformation and Fortran Legacy source code handling [25, 32], we will continue working on:

- Adding new metrics to the Metric View.
- Allowing user customize the Metric View.
- Adding a column that shows Fotran obsolescent and deleted features used in the source code.
- Integrating new legacy-aware functionality to the IDE.
- Integrating Static Analysis information as Photran Views.

References

1. Alfred V Aho, Monica S Lam, Ravi Sethi, and Jeffrey D Ullman. Compilers: principles, techniques, & tools, volume 1009. Pearson/Addison Wesley, 2007.
2. J. Backus. The History of Fortran I, II, and III. ACM SIGPLAN Notices, 13(8):165-180, 1978.
3. David W Balmer and Ray J Paul. Casm-the right environment for simulation. Journal of the Operational Research Society, pages 443-452, 1986.
4. Victor R. Basili, Richard W Selby Jr, and T Phillips. Metric analysis and data validation across fortran projects. Software Engineering, IEEE Transactions on, (6):652-663, 1983.
5. Frederick P. Brooks. The Mythical Man-Month: Essays on Software Engineering. Addison-Wesley, 1975.

6. Necdet Bulut, Maurice H. Halstead, and Rudolf Bayer. Experimental validation of a structural property of fortran algorithms. In Proceedings of the 1974 annual conference-Volume 1, pages 207-211. ACM, 1974.
7. Malcolm Cohen. Fortran: A Few Historical Details. http://www.nag.co.uk/nag-ware/NP/doc/fhistory.asp, October 2004.
8. Don Coleman, Dan Ash, Bruce Lowther and Paul Oman. Using metrics to evaluate software system maintainability. Computer, 27(8):44-49, 1994.
9. Microsoft Corporation. Visual Studio . http://msdn.microsoft.com/en-us/library/-bb385914(v=vs.90).aspx, May 1997.
10. A. Eastwood. Firm fires shots at legacy systems. Computing Canada, 19(2):17, 1993.
11. L. Erlikh. Leveraging legacy system dollars for e-business. IT Professional, 2(3):17-23, 2000.
12. John Foreman, David A Fisher, Michael Bray, Kimberly Brune, and Mark Gerken. C4 software technology reference guide-a prototype, 1997.
13. A. FORTRAN. X3. 9-1966. American National Standards Institute Incorporated, New York, 1966.
14. E. Gamma, R. Helm, R. Johnson and J. Vlissides. Design Patterns. Addison-Wesley Reading, MA, 1995.
15. Maurice H. Halstead. Natural laws controlling algorithm structure? ACM Sigplan Notices, 7(2):19-26, 1972.
16. Maurice H. Halstead. Elements of Software Science (Operating and programming systems series). Elsevier Science Inc., 1977.
17. Maurice H. Halstead and Rudolf Bayer. Algorithm dynamics. In Proceedings of the ACM annual conference, pages 126-135. ACM, 1973.
18. E. Iee. IEEE Std 1061-1998. IEEE Standard for a Software Quality Metrics Methodology, 1998.
19. ISO. ANSI/ISO/IEC 1539-1:1997: Information technology - programming languages - Fortran - part 1: Base language.
20. ISO. ANSI/ISO/IEC 1539-1:2004(E): Information technology - Programming languages - Fortran Part 1: Base Language. pages xiv + 569, May 2004.
21. Alan H Karp. Programming for parallelism. Computer, 20(5):43-57, 1987.
22. Donald E Knuth. An empirical study of fortran programs. Software: Practice and Experience, 1(2):105-133, 1971.
23. Meir M Lehman. The programming process. internal IBM report, 1969.
24. Thomas J. McCabe. A complexity measure. Software Engineering, IEEE Transactions on, (4):308-320, 1976.
25. M. Méndez, J. Overbey, and F. G. Tinetti. Legacy fortran software: Applying syntactic metrics to global climate models. In XVIII Congreso Argentino de Ciencias de la Computación, Oct. 2012. Available at https://lidi.info.unlp.edu.ar /~fernando/FortranLegacy/.
26. Michael Metcalf. Has fortran a future? Computer Physics Communications, 38(2):199–210, 1985.
27. Michael Metcalf. The seven ages of fortran. Journal of Computer Science and Technology, 11(1):1-8, 2011.
28. R. Miranda, F. Braunschweig, P. Leitao, R. Neves, F. Martins, and A. Santos. Mohid 2000, a coastal integrated object oriented model. Southampton, UK: WIT Press, Hydraulic Engineering Software VIII, 2000.
29. Miriam Schmidberger and Bernd Brugge. Need of software engineering methods for high performance computing applications. In Parallel and Distributed Computing (ISPDC), 2012 11th International Symposium on, pages 40-46. IEEE, 2012.

30. Martin Shepperd. A critique of cyclomatic complexity as a software metric. Software Engineering Journal, 3(2):30-36, 1988.
31. Martin Shepperd and Darrel C. Ince. A critique of three metrics. Journal of Systems and Software, 26(3):197-210, 1994.
32. F. G. Tinetti and M. Méndez. Fortran legacy software: source code update and possible parallelisation issues. ACM SIGPLAN Fortran Forum, 31(1):5-22, April 2012.
33. Arthur H. Watson, Thomas J. McCabe, and Dolores R. Wallace. Structured testing: A testing methodology using the cyclomatic complexity metric. NIST special Publication, 500(235):1-114, 1996.
34. Maurice V. Wilkes, David J. Wheeler, and Stanley Gill. The preparation of programs for an electronic digital computer.
35. M. V. Zelkowitz, A. C. Shaw, and J. D. Gannon. Principles of software engineering and design. Prentice Hall Professional Technical Reference, 1979.
36. Photran, an Integrated Development Environment and Refactoring Tool for Fortran. http://www.eclipse.org/photran/.
37. SimCon Ltd., FPT and WinFPT Reference Manual - Command-line Commands. http://www.simconglobal.com/fpt_ref_index.html.
38. The Numerical Algorithms Group 2014, NAG Software Developer Tools. Essential Tools Underpinning your Technical Programming. http://www.nag.co.uk /nagware.asp.
39. Code Analysis Team Blog, Maintainability Index Range and Meaning. http://blogs.msdn.com/b/codeanalysis/archive/2007/11/20/maintainability-index-range-and-meaning.aspx.

Design and Implementation of a Faceted Scheme for Publication and Retrieval of Aspects

GRACIELA VIDAL AND SANDRA CASAS

GISP - Instituto de Tecnología Aplicada
Universidad Nacional de la Patagonia Austral
Campus universitario Piloto Lero Rivera s/nro. Río Gallegos Santa Cruz
vidalgracielaunpa@gmail.com, scasas@unpa.edu.ar

Abstract. Aspects in software repositories are needed by users, but they are not available in an explicit manner. Studies on the reuse of aspects have focused on analyzing their dependencies, using inheritance for implementation, and establishing guidelines, and have also focused on aspects that provide quality applications. To date, making aspects available for reuse in software depositories has not been analyzed. Because specification of these modules is crucial for their integration in existing repositories, this paper presents the design and implementation of processes for the publication and retrieval of aspects in a software repository based on a faceted scheme.

1. Introduction

Software reuse is the process by which an organization defines a set of semantic procedures allowing the specification, production, classification, retrieval, and adoption of software artifacts, so that they can be used in different applications [1]. Component-based software development (CBSD) [2] [3] is one of the methodologies that promotes the reuse of software and requires a set of software artifacts to be available for reuse in what is called a *repository*. A software repository is a library in which reusable software components are stored, and it must offer a categorization of these artifacts [4] and should also be designed to provide support that is as automated as possible and that allows users to identify, compare, evaluate, and retrieve components.

A reusable software artifact can be defined as "any component specifically developed for use in more than one context" [5]. For an artifact to be stored in a software repository, general information, information concerning the artifact's functionality, restrictions, requirements, and documentation for further integration and specialization should be provided. Correct specifications and classifications allow users to access the components they require. Some techniques for specification of components that have been applied in CBSD are: faceted schemes [6], ontologies [7], frameworks [8], taxonomies [9], topological methods [10], and structural methods [11].

Aspect-Oriented Software Development (AOSD) [12] has been proposed to encapsulate non-functional requirements (NFRs) [13], also known as crosscutting concerns [14]. The NFRs (logging, synchronization, replication,

etc.) intersect the main functionality of a system, generating so-called scattered and tangled code, which presents several difficulties with regard to maintenance, reuse, etc. Some of the advantages of AOSD are that it can achieve more flexibility and adaptability as well as reuse of components used in system applications.
We have explored software repositories [15], and while they provide a variety of software artifacts that have proven useful for NFRs, these are not implemented as aspects. In order to achieve integration between different types of modules (components and aspects) that are complementary in the application development process, there is a need to include aspects in the repositories.
This paper describes a process for the publication and retrieval of aspects in a software repository. It extends the Unified Framework for Component Software Specification (FUECS) [8] in order to facilitate the definition of a faceted scheme. The implementation of the defined publishing process and retrieval scheme includes a layer for indexing and searching.
The rest of this paper is organized as follows: Section 2 explains the reasons that led to our proposal, Section 3 presents our proposed processes for publishing and recovery, and Section 4 presents our conclusions.

2. Motivations

The reuse of aspects has been approached from different perspectives, including analyzing their dependencies, devising mechanisms for implementing reuse, identifying the requirements and the contexts for reusing an aspect, and focusing on aspects that provide quality applications. A brief description of some of these approaches follows.
In [16], reuse is analyzed based on the dependencies between aspects, and a classification considering two features of these modules is proposed. The first concerns the way they interact with each other, and the second is based on how they display their functionality. An aspect is *orthogonal* if its functionality is independent of other features of the application, *one way* if it depends on some system functionality that may refer to data or services, and *circular* if it and another aspect are mutually dependent, so that these aspects only work together and not individually. The second classification distinguishes autonomous aspects—those that run on their own—and activated aspects that require any part of the active system.
Inheritance is used to implement reuse in AspectJ in [17], and [18] proposes rules and the development of at least three aspects for reuse that eliminate some of the disadvantages of this method. The four rules are:

Rule 1: Pointcuts must have separate declarations and definitions in an abstract aspect.

Rule 2: Since AspectJ does not allow recursive definition of a pointcut, i.e., a pointcut defined in an abstract aspect cannot be redefined, such a pointcut must first be defined and declared and then be assigned to another pointcut with a hook.

Rule 3: A pointcut should not be used by more than one advice. In such cases, the pointcut should be defined using a hook and then assigned to different pointcuts.

Rule 4: Concrete aspects should not contain attributes, methods, or advices that can be reused.

In addition to these rules, at least three aspects must be defined: one for the declaration of pointcuts, a second to declare attributes, methods, and advices, and a third concrete empty aspect. These are then connected through inheritance.

When aspects are valid for a certain context, it is necessary to check the compatibilities and requirements that define their context of use [19]. The use of pre and post conditions in programming design contracts provides an alternative for specifying a reusable aspect.

Finally, [20] focuses on the reuse of properties that add quality to an application rather than on their function, as is usually addressed. These properties are difficult to identify, classify, and catalog, so it is not sufficient to rely on their taxonomy [21] [22]. The article focuses on non-functional requirements [13], requirements oriented toward engineering objectives [23], oriented aspects [12], software reuse [24], and a quality management program.

Software reuse is a desirable feature for all of these approaches; however, to date there has been no analysis of how aspects should be made available for reuse in a software repository. Hence, we have studied the possibility of including these as independent modules implemented in aspect-oriented languages, which involves defining a specification for their publication and retrieval and providing the necessary information for their integration and reuse. In Section 3, we present a faceted scheme for specifying reusable aspects.

3. Publication and retrieval of aspects

Software repositories are critical to CBSD, as these are where the artifacts that are available to be retrieved and reused [25] are stored. For this purpose, a software repository must have capabilities to:

- Store reusable software assets;
- Provide an interface that allows publication, searching, and retrieval of assets;
- Manage their continual updating and maintenance;
- Sort software assets;
- Provide navigation through available assets; and
- Apply search techniques and automated recovery [26]

In addition, repositories must be supported by the definition, specification, and contextualization of each of their artifacts, including the components, classes, and aspects. An aspect is the central abstraction of AOSD and

comprises two modular parts: pointcuts and advices, in addition to attributes and methods. Advices are similar to methods, as they are pieces of code that will be added to the domain at some point (after, before, around), and pointcuts are the expressions that establish the moment or conditions under which the code fragments defined in the advices will be executed for example, a call to a particular method [12]. Example 1 presents code that implements an AspectJ StateManager aspect with the pointcuts removeVariable and addVariable and their capturing calls and executions, and the detailed methods are associated with the before and after advices.

```
public aspect StateManager
{
 pointcut addVariable(): call (* … set*(…));
 pointcut removeVariable():
   execution (* someComplexMethod(…));
 before() : addVariable()
   { … }
 after(): removeVariable()
   { … }
}
```

Example 1. Implementation of aspect StateManager in AspectJ

3.1 Definition of faceted scheme

FUECS, which aims to provide the necessary information to facilitate the development, discovery, and configuration of components, is presented in [8]. It comprises a set of specific fields grouped in forms called *pages*, which are identified with colors. The white page contains general and business information (name, author, version, price, etc.), the yellow page provides categorization (domain, technology, etc.), the blue page describes functionality (processes, tasks, etc.), the green page defines the specified interfaces, and the gray page provides additional information about the quality of the component.
Figure 1 presents our proposed extension to FUECS, which can be applied to aspects with a red page called *Weave* that includes dynamic and static concepts. Dynamic concepts allow the definition of pointcuts, advices, and execution orders when necessary. Static concepts allow the definition of introductions and heritage. Figure 2 displays the specification of the StateManager aspect using the faceted scheme defined in XML.

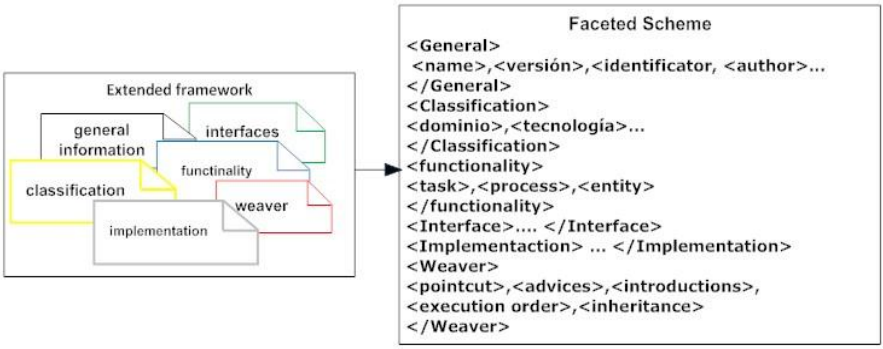


Figure 1. Extended framework and faceted scheme for specification of aspects

```
<aspect>
 <general>
      <name>StateManager</name>
      <version>1.0</version>
 </general>
<classification>
      <domain>Movilidad de Codigo</domain>
      <reuse>lógico</reuse>
      <tecnology>AspectJ</tecnology>
</classification>
<functionality> .... </functionality>
<interfaces> ... </interfaces>
<implementation> .... </implementation>
<weaver>
     <pointcut>addVariable(): call (* …set*(..))</pointcut>
     <pointcut>removeVariable():execution (* someComplexMethod(..))</pointcut>
     <advice>before:addVariable()</advice>
     <advice>after:addVariable()</advice>
</weaver>
</aspect>
```

Figure 2. Faceted scheme applied to StateManager aspect

3.2 Publication and retrieval processes for software artifacts

Figure 3 shows the publishing and retrieval processes for aspects using the defined faceted scheme. The graph illustrates two instances:

- Publication

a. User A completes a form with the specification defined in the faceted aspect source code diagram.

b. The specification of each aspect generates an XML file similar to that in Figure 2.
c. The algorithm analyzes each XML publication and then indexes it, generating a document that is then added to the index.

- Retrieval

d. User B completes a form with the characteristics of the software artifact that is needed.
e. The retrieval algorithm analyzes this query against the document index.
f. The documents that match the conditions and/or restrictions of the query are displayed to the user, who then selects the aspects to be recovered.

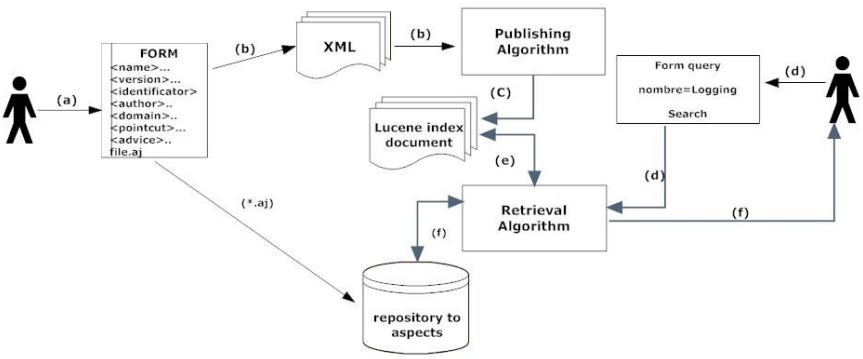


Figure 3. Publication and retrieval processes

3.3 Deployment Tool

We have developed a "proof of concept" style tool for implementing the processes described in Section 3.2. Each specification in the faceted scheme form defined for publication of artifacts is stored in an XML file. Each facet of the form becomes a label in the XML file, and the information in the corresponding field becomes the content for that label. We use Apache Lucene API [27] as the mechanism for transferring these facets and contents in order to provide fast and easy access for the retrieval process. Publication includes an algorithm for the indexing process, which generates an index of documents against which queries can be analyzed. The strategy chosen to select the information for including in these documents allows developers to extract any type of text information from the indexed files.
The algorithm for this tool defines documents consisting of two fields, "tags" and "description", which will store the information from the XML file containing the specification of each aspect. The "tags" field stores the names of the tags and the "description" field stores the content. Example 2 shows the result of this process for specifying a synchronization point.

XML specification	Lucene index document
<aspect> <general> <name>Synchronization</name> <version>1.0<version> <identificator>Sin10<identificator> </general> …. </aspect>	**Tags** **Description** name Synchronization version 1.0 identificator Sin10 … …

Example 2. Document generated from specification XML

Query retrieval can be based on free text such as "logging", "call", or "AspectJ", and it can be made more sophisticated using AND, OR and the wildcards ~ and ^. Example queries are "name = Logging AND technology = AspectJ", "name = Log~", or " technology = AspectJ OR technology = Java". The recovery algorithm we implemented takes this information and generates the query and then analyzes each document in the index. As a result of the query, a structure that stores documents that meet the query conditions is obtained and presented to the user as a first step for retrieval. Example 3 shows a query to be analyzed based on the form submitted by a user.

Query for retrieval	Query against Index
Name: Logging Category: Security	Tags:Name AND Description:Logging OR Tags:Category AND Description: Security

Example 3. Query for retrieval

4. Conclusions

NFRs are available in current software repositories, which indicates that they are requested for reuse by users, but they are not available as aspects. The reuse of aspects has been approached from different perspectives: based on their dependencies, by defining a set of rules, and using inheritance as a mechanism for reuse, to name a few. This means that aspects are reusable and that they are needed, which led us to the analysis of storage along with other components to enable the reuse of more aspects.

This paper proposes as a first step an extension of FUECS for incorporating aspects in software repositories, namely, the incorporation of a specific page in which concepts are defined for aspects.

Our framework defines a faceted scheme for the specification of these modules, with the development of a "proof of concept" style tool to implement the publishing and recovery processes for aspects based on the faceted schemes.

As future work, we plan to test and validate the defined tool. This will consist of two stages—publication and retrieval—and we will publish other components, classes, aspects, and interfaces. In the first stage, we will group different types of software artifacts (components and aspects) to be published. In the second stage, we will define a set of queries based on different criteria for analysis against the index generated by the published specifications for aspects.

References

1. Mili, A., Yacoub, S., Addy, E. & Hafedh, M., Reuse Based Software Engineering, John Wiley & Sons, INC.(2002).
2. González R., Torres. M. Some Implications of Component-Based Development. paper. Bogotá, Colombia: Departamento de Ingeniería de Sistemas Universidad Javeriana; (2005).
3. N. A. y. J. A. C. Jonás A. Montilva C., Component-based software development IV Congreso de Automatización y Control, Maracaibo, Mérida - Venezuela, pp. 9 (2003).
4. S. B. Amandeep Bakshi, Development of a software repository for the precise search and exact retrieval of the components, International Journal of Advanced Research in Computer Science and Software Engineering 3, no. 8,pp 1116-1126 (2013).
5. Kalathipparambil J. and Vasantha, A Mixed Method Approach for Efficient component retrieval from a component repository. Journal of Software Engineering and Applications (2011).
6. Prieto-Diáz, R.: Implementing Faceted Classification for Software Reuse. In: Communications of the ACM 34 (1991).
7. Dragan G., Nima K., Milan M., Ontologies and Software Engineering (2008).
8. Overhage S. Towards a Standardized Specification Framework for Component Development, Discovery, and Configuration. Augsburg University. Dept. of Application Engineering and Business Information Systems (2003).
9. S.-S. Maninder, Goel., Identifying asset type for various reusable component storage/retrieval methods. Proceedings of National Conference on Challenges & Opportunities in Information Technology - RIMT-IET, Mandi Gobindgarh, pp. pp.38-41 (2007).
10. G. Forbes, McCartan C., Ruairi O'Donnell, Niall S. and Leon R. The integration of information retrieval techniques within a software reuse environment. Department of Information Science, Strathclyde University, 26 Richmond St., Glasgow, G1 1XH, UK (2000).
11. Klein, M. and A. Bernstein. Searching for Services on the Semantic Web Using Process Ontologies. The First Semantic Web Working Symposium. Stanford, CA, USA. (2001).
12. Kiczales G., Lamping J., Mendhekar A., et al. Aspect-Oriented Programming. In: Springer-Verlag, editor. Finland: European Conference on Object-Oriented Programming; pp25 (1997).
13. Mylopoulos, J., Chung, L., Nixon, B.: Representing and using nonfunctional requirements: A process-oriented approach. IEEE Trans. Software. Eng. 18 pp.483-497 (1992).
14. Filman, R., et al., Aspect-Oriented Software Development. Addison-Wesley (2004).

15. Vidal G., Casas S. y Marcos C. . Exploration of software repositories and analysis of potential extensions aspects. Journal Scientific Reports NUPA. ISSN 1852-4516 Vol. 5 pp.68-99.
16. Kienzle J., Yu Y., Xiong J. On Composition and Reuse of Aspects. School of Computer Science McGill University. Montreal, QC H3A 2A7. Canada (2003).
17. Web site AspectJ www.eclipse.org/aspectj (2014).
18. S. a. R. Hanenberg, Unland, Using and reusing aspects in AspectJ. Institute for Computer Science - University of Essen, Germany p. 6 (2001).
19. D. Wampler, The challenges of writing reusable and portable aspects in aspectJ: Lessons from contract4j. Aspect Research Associates and New Aspects of Software, pp.9 (2006).
20. Sampaio do Prado Leite J., Yu Y., Liu L., Yu E. and Mylopoulos J. Quality-Based Software Reuse. Departamento de Informática, Pontificia Universidad Católica do Rio de Janeiro, RJ 22453-900, Brasil. Department of Computer Science, University of Toronto, M5S 3E4 Canada. School of Software, Tsinghua University, Beijing, 100084, China. (2005).
21. Sommerville, I.: Software Engineering, 4th Ed. Addison-Wesley (1992).
22. Boehm, B.W., Brown, J.R., Lipow, M.: Quantitative evaluation of software quality. In: ICSE, International Conference on Software Engineering, IEEE Computer Society Press pp. 592–605 (1976).
23. Mylopoulos, J., Chung, L., Yu, E.: From object-oriented to goal-oriented requirements analysis. CACM 42 pp.31–37 (1999).
24. Krueger, C.: Software reuse. ACM Computer Survey 24 pp.131-183 (1992).
25. Nedhal A., Al Saiyd, Intisar A. Al Said, Takrori AHA. Semantic-Based Retrieving Model of Reuse Software Component: Computer Science Department (2010).
26. Kuljit Kaur PK, Jaspreet Bedi, and Hardeep Singh. Towards a Suitable and Systematic Approach for Component Based Software Development. World Academy of Science, Engineering and Technology (2007).
27. McCandless M.,Hatcher E., Gospodnetic O. Lucene in Action. Second Edition: Covers Apache Lucene 3.0. ISBN-10: 1933988177 ISBN-13: 978-1933988177. Manning Publications (2010).

Evolution of a Ranking Algorithm for Scientific Documents in the Computer Science Area

HORACIO KUNA[1], ESTEBAN MARTINI[1], MARTÍN REY[1]

[1] Computer Science Department, School of Exact,
Chemistry and Natural Sciences, National University of Misiones.
hdkuna@gmail.com

Abstract. A meta-searcher is a kind of software system used to retrieve information; it is characterized by operating with results obtained from traditional searching engines and by structured itself with elements tailored and developed for a context of specific functioning. One of the issues which turn to be essential is the fusion and the sorting out of the results which come from the different document sources. This research work presents the evolution of a ranking algorithm developed for a meta- searcher which has the aim to evaluate scientific publications belonging to the area of computer science, application field of the system which is being developed.

Keywords: information retrieval, ranking algorithm, web search, bibliometric indicators.

1. Introduction

In this section the theoretical axes of the current research works are described: Information Retrieval Systems and Metrics for the evaluation of scientific documents used for generating the ranking algorithm.

1.1 Information Retrieval Systems

An IRS (Information Retrieval system) is a process able to store, retrieve and keep information [1], [2]. In [3], it is proposed that an IRS is made up of four fundamental elements: the documents, the user's queries, the way in which these elements are represented and an evaluation function. The IRS models more diffused and spread in the internet are the directories, the search engines and the meta-searchers [4]. Publications exist throughout the literature, among them [5], [6], in which the use of IRS is extended to general contexts as well as particular ones.
In previous works [7], [8] it has revealed the absence of specific IRS which are oriented to scientific documents in the CS area (Computer Science). From then on advancements have been realized in the building of an IRS, particularly, a meta-searcher and its components, considering in every case

the necessary adaptations for its right functioning in a context as the one suggested. The main meta-searcher components developed are [8]:

- Query Management Module: it is in charge of capturing, optimizing and performing the queries which the user enters on the sources of documents integrated to IRS.
- Search Module: it is in charge of performing the queries on the sources to which the meta-searcher Access and retrieve the results of every one of them.
- Result Management Module: it is in charge of obtaining the results lists retrieved in the searches and processing them to present as a unified list to the final user.

Among the components of the latter module, the ranking algorithm stands out; this one is used to establish a sorting out of the scientific articles which will be presented to the user as a result of the performing of a concrete search [7].
In this new presentation, the set of metrics to be used has been extended and corrections have been made in their application formulas to improve the algorithm quality.

1.2 Metrics for the scientific document evaluation

A series of characteristics exist and are widely used to evaluate the quality of scientific documents [9], [10]. These characteristics are quantified from metrics or indicators; most of them are focused on the following properties:

- The quality of publication source.
- The quality of the authors of the publication.
- The publication quality.

Throughout such properties, firstly it is possible to evaluate a scientific document in the first instance on the base of the quality of the source or place in which has been published, distinguishing if this has been a journal or a discipline scientific event, secondly it is possible to evaluate the quality of the publication authors by the acknowledgement their previous publications have obtained and finally the quality of the publication itself understood as the repercussions which has obtained from its presentation.
In the first metrics review, those with a greater use in the latest years stood out [7]. In this paper, the searching of the metrics has been enlarged with the aim to include those which enable to reduce the impact of certain aspects that were limitations of the first ones or at least to enlarge the scope of metrics analyzed and consequently possibly improve the quality of the developed algorithm. The metrics evaluated are shown in table 1; in same one, the metrics previously used and those which were incorporated in the work period covered by the current work are discriminated.
In the case of metrics which allow to value the quality of a determined publication source, specifically, scientific journals, the IF (Impact Factor) [11] has been for a long time the most extended option. In last years, a series of alternative and/or

complimentary metrics have proliferated, i.e. the SJR (SCImago Journal Rank) [12], SNIP (Source Normalized Impact per Paper) and RIP (Raw Impact per Paper) [13] indicators, the EI (Eigenfactor) and the AI (Article Influence) [14]. Metrics which were originally conceived for the evaluation of other aspects have also been found for this criterion. This comes up owing to the practicality of such metrics and has led to their implementation, making the necessary adaptations to build an indicator to be used out of context for which they have been thought in a first time. Such it is the implementation of the H index which is realized by MAS (Microsoft Academic Research) [15].

In all the cases, the metrics evaluate a journal from its articles influence in a determined time window, including in some cases a journal appreciation according to its relationship with others. For the valuation of scientific events or meetings, there is not a similar variety of metrics in terms of quantity or quality. As an alternative, a series of rankings of conferences stand out, these are calculated from a similar analysis that is realized by the metrics applied to journal evaluation. Regarding the CS, the CORE [16] and the ERA rankings are found [17]; along with these again, metrics implementations originally defined for the evaluation of other aspects are also found, but these have been adapted to evaluate conferences by using a different data source for their calculation process. As examples of these, MAS from which it is possible to obtain an implementation of H index for conferences [18], and CiteSeerX using its data in a combined way with the ones of DBLP to generate an IF for scientific events [19].

Table 1. Metrics searched for the evaluation of scientific articles

Properties to be evaluated		**Original metrics**	**New metrics searched**
Type of publication source	Scientific Journals	IF	
		SJR	
			SNIP
			RIP
			EI
			AI
			H Index
	Conferences or Scientific Events	CORE Ranking	
			ERA Ranking
			H Index
			IF
Authors		H Index	
			G Index
			E Index
			W Index
Artricle		AR Index	
		Quatities of citations	

For those metrics which are used for the evaluation of the authors' quality, the H index is the pioneer in this aspect and it is the most used indicator, but it is not free from limitations and for there on, about a dozen of variants have come up [20] whose objective is to cover the inconveniences of the original metric. Among these variants, the G [21], E [22] and W [23] indexes stand out.
To evaluate the quality of a scientific article it was proposed to keep previous metrics, adjusting the AR index [24] so that it can operate on a single document evaluating the citations quantity obtained by the document at the moment of its retrieval by the IRS considering their aging, defining in this way a metric for the publication quality.
The rest of the article is made up of the following layout: the design of the ranking algorithm is described in section 2, its implementation in section 3 and the conclusions in section 4.

2. Design of the ranking algorithm

2.1 Conceptual Model

The ambit of bibliometric indicators is a wide and quite fuzzy field when it is necessary to choose one. This happens because most of the existing metrics have a greater or fewer followers or detractors. In [10] as well as in [13] the authors agree on the facts that there is not a perfect indicator and that the impact of the scientific production cannot be correctly measured by a single indicator because the scientific production is a multidimensional construction.
There are several inconveniences which come up when it is necessary to use bibliometric indicators to evaluate the quality of a specific aspect of a publication. One of the problems is to get to know if it is the same to use either an indicator or another one. In [25] this issue is briefly undertaken and the correlation that exists among a group of scientific journals is studied. The conclusion is that their behavior varies in function to the knowledge area, that is, some indicators are equivalent in certain areas, but nor in others.
Another issue to take into account is how to know which the real value of a metric is. This comes up from the fact that the information used to compute some indicators is homogeneous; therefore the same metric calculated form different databases can have different scores for the same study object. In [26] for example, the variations which exist in H index of a set of authors in relation to the source from which the data are taken are studied. A problem closely linked to the former subject is the degree of overlapping that there is among the data used to calculate the indicators. This is a very difficult limitation to deal with since the indicators are in a continuous development [27]. From the issues previously discussed, it turns to be inaccurate or incomplete to define the quality of some aspect of a scientific publication in function to a single metric.

From these situations, a model (see figure 1) which makes use of a set of indicators to evaluate a specific publication was built. The evaluation is made considering 3 evaluable aspects described in the previous section: the publication source, the documents' authors and the article itself. To evaluate every IRS result, the metrics values which are available for such publication will be obtained, being able to be considered as independent metrics those which are obtained from different sources, this condition allows to a metric to be used for the evaluation of more than a single property of every document.

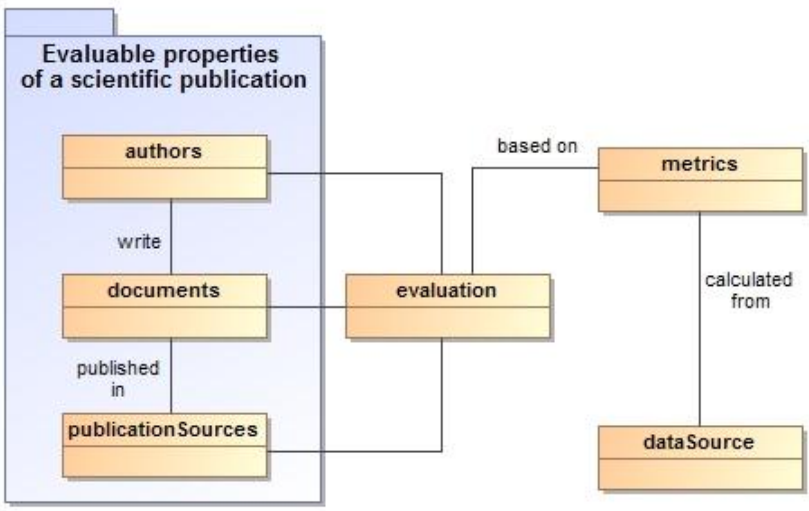


Fig. 1. Structure used for the evaluation of scientific documents

In this way, the model raised takes into account the problems introduced previously and by means of their characteristics allow to minimize its defects, since the holistic approach which is used allows to evaluate the properties of a scientific document from a series of metrics from different origins, without the final evaluation being dependent on some particular metric. Besides the model is greatly scalable, since the integration of a different aspect to evaluate a document could be realized without the need to modify the general structure of the model. Besides and in an analogic way, as the meta-searcher becomes more robust as long as it incorporates databases where it searches articles, the ranking algorithm also becomes more robust as long as it incorporates a major number of metrics and data sources where it obtains its values from.

2.2 Calculation of parameters

In the algorithm evolution it is kept the general formula in which it is defined that the grading of a publication Q results from the addition of 3 parameters: *PS* (Publication Source), *A* (Authors), and *D* (Document), every parameter is multiplied by an adjusting factor α, β and γ (see Formula 1).The adjusting factors are useful to vary the weight of the terms from the importance it is wished every term to be given.

$$Q = \alpha * PS + \beta * A + \gamma * D \quad (1)$$

Every parameter is internally calculated from a number of metrics which is not stablished in a first moment, but which varies in function to those which are found for the publication which is being evaluated. So every parameter (*P)* takes its value from the quotient of the addition of all the metrics *(m)* which are found for such publication above the quantity (*n*) of metrics used (see Formula 2).

$$P = \frac{\sum_1^n m}{n} \quad (2)$$

Since the range of metrics values vary considerably from one to another, it is not possible to use them in a direct way. To be able to integrate all the metrics inside the same calculus and keep the weights balanced among them inside the formula, it is necessary to normalize its values. The normalization proposed has two variants and they can be applied depending of the kind of metric it is being managed. The first variant uses the quotient between the value which such metric has for the publication to be evaluated and the maximum existing value for that metric within the same data source from which it has been obtained (see Formula 3). This kind of normalization is mostly used in the metrics of the *PS* and *A* parameters. The remaining variant is obtained using the logarithm on base 10 of the value of the metric (see Formula 4).

$$m = {}^{m}/_{\max(m)} \quad (3)$$

$$m = \log_{10} m \quad (4)$$

3. Implementation of the ranking algorithm

3.1 Selection and retrieval of metrics to be included in the model

Having the model of evaluation and the method of calculus of the parameters defined, the implementation of the evolution of the ranking algorithm was realized. Firstly, to be able to do this, a selection of metrics to be used for the evaluation of every one of the aspects of a scientific document was carried out.

From the review described in section 1.2, it was determined the existence of a set of metrics which are defined in a theoretical way and which aim to the evaluation of some particular aspect of a scientific publication. Such metrics are calculated from the information which is not easy to gather or retrieve. Taking into account this reality and the meta-searcher performance, it has been opted to use, in this instance, a series of metrics which were previously calculated form the well-known data sources such as Scopus, ISI and DBLP among others. Several different versions of the same metrics which are calculated from different databases have been found. In these situations it was decided to include both of them. In other cases, it was observed that some databases, e.g. MAS, have opted to use indicators to qualify journals or conferences, which were not conceived, originally, for such a purpose, such as H index, besides journal native indicators were used.

Once the implementations of the metrics available were identified, it was necessary to evaluate the probability to generate a method to obtain their values in such a way that they can be used in the ranking algorithm. In all the cases in which it was possible to extract the metrics values, it was decided to use them in a local way, that is, from a IRS internal database with the aim to optimize the times of both operations, query and results' processing.

To achieve this, the research team started by developing software components which allowed to extract their own values for each metric from the web page in which they were published. With the available data, a series of transformation and load processes to achieve the homogeneity in the database were developed.

The set of metrics and data sources, finally selected to evaluate the scientific articles can be seen in Table 2.

Table 2. Metrics incorporated to the model for the calculus of the algorithm and its origin

Properties to be evaluated		Metric used	Data Source
Type of publicaiton source	Scientific Journals	SJR	Scopus
		RIP	Scopus
		SNIP	Scopus
		H Index	Scopus
		AI	ISI
		EI	ISI
		EI	MAS
		H Index	MAS
	Conferences or Scientific Events	ERA	ERA
		CORE	CORE
		IF	CiteSeerX + DBLP
		H Index	MAS
Authors' quality		H Index	ArnetMinner
		G Index	ArnetMinner
		H Index	GS
Article quality		Quantity of Citations	(*)
		Index AR	

(*) Sources used by the search module of the meta-searcher

The software tools which were used in the mentioned tasks were: Java, HTML, XML and JSON languages to extract the content from the web, the module of data integration from the Pentaho (Pentaho Data Integration) business intelligence suite and the PostgreSQL database engine.
The metrics' values corresponding to the evaluation factors of the publication source quality and the authors' quality have been extracted and loaded to the SRI database. The metrics which were used to evaluate an article quality are obtained from every result meta-data which retrieve the IRS when performing a search; in the meantime its values were not stored in the meta-searcher database.

3.2 Validation of the developed algorithm

Once the algorithm was implemented, it was necessary to start with its process of validation for its later incorporation to the SRI. The process fell in stages: the initial one realized by experts and a later one which aims to evaluate statistically the algorithm efficiency by using as a means of performance the correlation which exists between the metrics behavior used and the obtained results.
For the case of the first validation instance, a work outline was designed, similar to the one used in prior research works [7, 8]. A series of queries have been performed by using the meta-searcher and the set of results along with the details of the calculus carried out for the application of the ranking

algorithm has been exported. These results have been evaluated by the query field experts who have determined percentage of effectiveness of the classification realized on the documents retrieved. The percentages of effectiveness obtained regarding the ones obtained in the same query in the original publication were compared with the aim to compare the application of the ranking algorithm evolution in respect to the prior version. With the aim to compare the application of the ranking algorithm as regards the previous version, the effectiveness percentages obtained with the same query in the original publication are compared. The experimentation details and results can be seen in table 3.

Table 3. Results of the first validation instance

Queries performed	Quantity of results processed	Effectiveness evaluated by the experts	Comparison with the original version
data mining AND outliers	60 (20 Google Scholar + 20 IEEEXplore + 20 ACM Digital Library)	78%	+4%
alphanumeric data AND outliers	60 (20 Google Scholar + 20 IEEEXplore + 20 ACM Digital Library)	86%	+5%
scientific production AND metrics	60 (20 Google Scholar + 20 IEEEXplore + 20 ACM Digital Library)	80%	+3%

The second phase of the evaluation was raised from the solution the algorithm could provide to a problem which comes up in the scientific documents evaluation as it is the correlation among the several metrics which are used, for example, those among the quantity of citations and the SJR index in a specific journal. As regards this, what was aimed to determine was if the integration of a set of heterogenic in the algorithm would result in the value to give to a document and consequently its evaluation not determined by a particular indicator.

This validation stage, at the moment of finishing the edition of the current research work, is still in development, owing to the quantity of metrics and documents to evaluate the results which would be available in future publications.

4. Conclusions and future research works

In the current paper an evolution of a ranking algorithm has been raised using it to ponder and sort out scientific documents belonging to the area of computer science. Among the main improvements achieved, the building of a generic model can be stood out. Such model soften some inconveniences that come up when it is necessary to work with bibliometric indicators and it enables the integration of several of them, allowing to evaluate a document in an more robust and integral way. Even though the algorithm is in the second face of evaluation, it has characteristics makes it scalable and flexible and it has positively been validated by experts in the field, making it easy the incorporation of other metrics to the calculus.
As future research works what can be mentioned is to complete the validation phase and if it were necessary to adjust the algorithm, to build the component which applies the algorithm to the results obtained by the meta-searcher and at the same time sort out the results and finally to build the components which capture and update the metrics used in an automatic way.

Bibliography

1. Salton, G., Mcgill, M.: Introduction to Modern Information Retrieval. McGraw-Hill, Inc. (1983).
2. Kowalski, G.: Information Retrieval Systems: Theory and Implementation. Kluwer Academic Publishers, Norwell, MA, USA (1997).
3. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. ACM press New York. (1999).
4. Olivas, J.A.: Búsqueda Eficaz de Información en la Web. Editorial de la Universidad Nacional de La Plata (EDUNLP), La Plata, Buenos Aires, Argentina (2011).
5. Serrano-Guerrero, J., Romero, F.P., Olivas, J.A., de la Mata, J.: BUDI: Architecture for fuzzy search in documental repositories. Mathw. Soft Comput. 16, 71-85 (2009).
6. De la Mata, J., Olivas, J.A., Serrano-Guerrero, J.: Overview of an Agent Based Search Engine Architecture. Presented at the, Las Vegas, USA (2004).
7. Kuna, H., Rey, M., Martini, E., Solonezen, L., Sueldo, R.: Generación de un algoritmo de ranking para documentos científicos del área de las ciencias de la computación. XVIII Congreso Argentino de Ciencias de la Computación (2013).
8. Kuna, H., Rey, M., Martini, E., Solonezen, L., Podkowa, L.: Desarrollo de un Sistema de Recuperación de Información para Publicaciones Científicas del Área de Ciencias de la Computación. ReLAIS. 2, 107-114 (2014).
9. Pendlebury, D.A.: The use and misuse of journal metrics and other citation indicators. Arch. Immunol. Ther. Exp. (Warsz.). 57, 1-11 (2009).
10. Bollen, J., Van de Sompel, H., Hagberg, A., Chute, R.: A Principal Component Analysis of 39 Scientific Impact Measures. PLoS ONE. 4, e6022 (2009).
11. Garfield, E.: Citation Analysis as a Tool in Journal Evaluation: Journals can be ranked by frequency and impact of citations for science policy studies. Science. 178, 471-479 (1972).

12. Gonzalez-Pereira, B., Guerrero-Bote, V., Moya-Anegon, F.: The SJR indicator: A new indicator of journals' scientific prestige. arXiv:0912.4141. (2009).
13. Moed, H.F.: Measuring contextual citation impact of scientific journals. J. Informetr. 4, 265-277 (2010).
14. Bergstrom, C.: Measuring the value and prestige of scholarly journals. Coll Res Libr News. 68, 3146 (2007).
15. Help Center - Microsoft Academic Search, http://academic.research.microsoft.com /About/Help.htm, (2014).
16. CORE: CORE Conference Ranking. Computer Research & Education of Australia, http://core.edu.au/index.php/conference-rankings, (2008).
17. The Australian Research Council: ERA 2012 Journal and Conference Lists, http://www.arc.gov.au/era/era_2012/era_journal_list.htm, (2012).
18. Hirsch, J.E.: An index to quantify an individual's scientific research output. Proc. Natl. Acad. Sci. 102, 16569-16572 (2005).
19. CiteSeerX - Statistics - Venue Impact Factors, http://citeseerx.ist.psu.edu/stats/ venues, (2014).
20. Van Noorden, R.: Metrics: A profusion of measures. Nat. News. 465, 864-866 (2010).
21. Egghe, L.: Theory and practise of the g-index. Scientometrics. 69, 131-152 (2006).
22. Zhang, C.-T.: The e-Index, Complementing the h-Index for Excess Citations. PLoS ONE. 4, e5429 (2009).
23. Wu, Q.: The w-index: A significant improvement of the h-index. ArXiv Prepr. ArXiv08054650. (2008).
24. Jin, B., Liang, L., Rousseau, R., Egghe, L.: The R- and AR-indices: Complementing the h-index. Chin. Sci. Bull. 52, 855-863 (2007).
25. Torres-Salinas, D., Jiménez-Contreras, E.: Introducción y estudio comparativo de los nuevos indicadores de citación sobre revistas científicas en Journal Citation Reports y Scopus. El Prof. Inf. 19, 201-208 (2010).
26. Bar-Ilan, J.: Which h-index? - A comparison of WoS, Scopus and Google Scholar. Scientometrics. 74, 257-271 (2007).
27. Sicilia, M.-A., Sánchez-Alonso, S., García-Barriocanal, E.: Comparing impact factors from two different citation databases: The case of Computer Science. J. Informetr. 5, 698-704 (2011).

A metamodel for assisting dynamic component composition of a Software Product Line

MAXIMILIANO ARIAS, AGUSTINA BUCCELLA, MATÍAS POL'LA, ALEJANDRA CECHICH

GIISCO Research Group,
Departamento de Ingeniería de Sistemas - Facultad de Informática
Neuquén, Argentina
{maximiliano.arias,agustina.buccella,_matias.polla,alejandra.cechich}@fi.uncoma.edu.ar
http://giisco.uncoma.edu.ar

Abstract. Component Based Software Development and Software Product Lines are based on the ability of combining pieces of software as composition units. Assembling an application from existing components is still a matter of study, due to the complexity of the interactions among them and the need for adaptation in some cases. Particularly, performing an automatic composition implies even greater challenges. In this work, we propose a model to assist components with specific metadata using annotations in order to ease the composition process. The model has been instantiated in the context of the Java development platform and validated an instantiation of a Software Product Line component.

Keywords: Software Product Lines, Component based software development, Metadata, Dynamic composition

1. Introduction

In the broad set of reuse-oriented software development paradigms we can find two main paradigms, Software Product Line Engineering (SPLE)[12] and Component Based Software Development (CBSD)[5]. Both paradigms can be combined making a Software Product Line (SPL) can be implemented using reusable components on its variation points. Assembling such components, both to create a product or a new component, is done manually, identifying at least to types of composition [5, 14]. On one hand, if the component is exclusively developed to implement a specific variant point, composition can be done without need for adaptation. On the other hand, a component can be developed for reuse in a more general domain than the one that was firstly developed or it can even be developed by a third party and for a more general purpose. In this case, it is necessary to perform an adapted composition that allows to connect such component with its coupling point. At the same time, the composition process can be affected by a several different issues [14] related to interfaces, operation and exception

incompatibilities. However these issues do not imply that the process cannot be done semi or even fully automatically [16].
In this way, in this work we design, implement and use a metadata system to assist in the dynamic composition of reusable components of a SPL. To do so, we perform an analysis of the elements that participate in the composition process, and create a metadata system definition process to define a system capable of achieving our objective. This paper is organized as follows, in Section 2 we present previous and related works, Sections 3 and 4 we present our proposal, explaining the construction process and a particular implementation. After that, in Section 4 we show a case study performed over previous works. Finally in Section 5 we present the conclusions and future works.

2. Previous and related works

The dynamic composition problem has been analyzed by different approaches [7, 9, 16, 18] using XML configuration files, aspect oriented programming, FODA specifications and even formal languages like ADL (Architecture Description Language).
On the other side, metadata models have been used to assist a great number of software development tasks [2, 3, 4, 8, 10, 15], including testing and validation, natural language description management, catalogs searches, tool's configuration, etc.
In previous works [6, 11, 13] we have developed an SPL for the marine ecology subdomain, working together with the "Instituto de Biología Marina y Pesca Almirante Storni"1 (IMBPAS) and the "Centro Nacional Patagónico2" (CENPAT). This SPL was first implemented using JavaScript [11] and then re-factorized [6, 13] using JAVA3 and Enterprise Java Beans4.
With previous resources we started the creation of reusable components capable of implementing the SPL variabilities and commonalities. These components were defined from a service taxonomy [13] by specializing the ISO/DIS 191195 standard for Geographic Information Systems (GIS).

3. Developing a dynamic composition metadata model

To assist the dynamic composition process of a SPL we propose a metadata model. To do so, we firstly defined a three steps process that allows us to create such a model. Figure 1 shows these steps with a detailed description of the tasks that we need to perform in the second step of the process. We describe this step in more detail due to the existence of a sub-process for defining, generating and recovering the defined metadata.

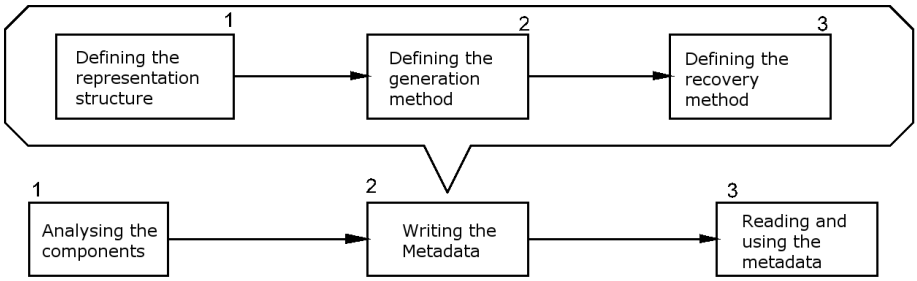


Fig. 1: Process for defining the metadata model for SPL component development.

Each of the steps is described as follows:

1. *Analyzing the components*: In CBSD a component contains a set of interfaces to describe its functionalities and integration requirements. In this way, we can identify two key elements, the component's functional interface [17] and its architectural requirements (or interface) [1]. The first of these elements describes the component's behavior, including all of its public attributes and methods, events that can be triggered and exceptions generated. On the other side, the second element allows the component to describe its architectural needs, which requires the component to be assembled with other components. In this set of requirements we can identify the programming language, entry points, external dependencies (software libraries, other components, databases, etc.), internal dependencies (in case components are composed of other components), etc. The component analysis process must identify each of this elements to use them in the next step.
2. *Writing the metadata*: As the metadata writing process requires a correct definition of the model, we define a general process for such task. Figure 1 shows how this process is organized, identifying three activities:
    a. *Defining the representation structure*: During this activity we must perform an analysis of the problem that we are looking to solve, and trough this we must obtain the set of relevant metadata. Once this set is defined, we must analyze the way we are going to represent this metadata inside the components. There are many ways to achieve this task according to the needs of the problem that the metadata must solve. On one side, the required metadata can be stored outside the components using external files in plain text, XML or another mark-up languages, etc. On the

other side, we can place the metadata inside the component using annotation systems, special comments, or another platform specific technology.

b. *Defining the generation method*: Once the structure is created, we must define how the metadata is going to be generated. This process can be done manually, semi or fully automatically. To define this method we must take into account the platform on which the components are being developed. The representation structure, defined previously, and a technology analysis are used to select or create the generation method.

c. *Defining the recovery method*: Finally, using the representation structure and the generation method we define the recovery method. This method is strongly related to the previous elements and can be necessary the construction of tools to perform it. Like the previous task, we must analyse the available technologies and determine if we can rely on previously developed tools.

3. *Reading and using the metadata*: The final step of the process is to read and use the written metadata and use them to achieve its purpose. This way, the reading process is adapted to the decisions established during the model definition process and its use will be defined by the task which is necessary to fulfill. This task is related to the problem we are trying to solve, for example for automating or assisting in the execution of a process, configuring tools, improving traceability, etc.

4. JAVA implementation

In this section, we describe the implementation of the metadata model definition process previously depicted in Figure 1. As shown in Figure 2, for each step we generate a specific artifact.

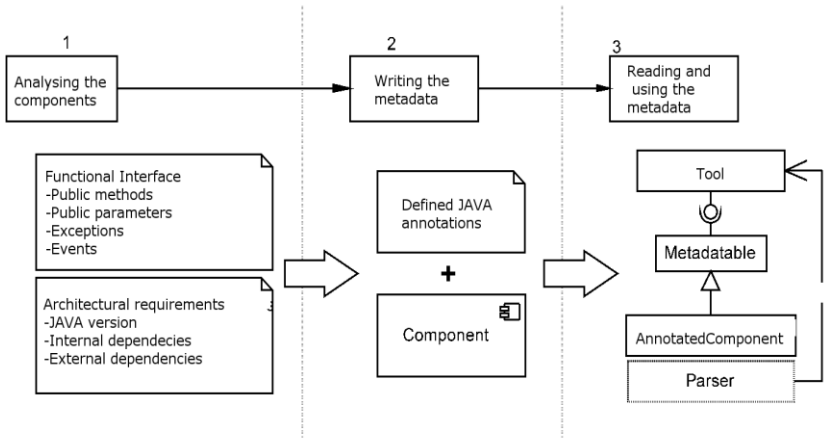


Fig. 2: Applying the metadata system to the dynamic composition problem.

In the first step, *component analysis,* we perform a study of all the elements that can be identified inside a JAVA component. Due to the fact that JAVA does not define a default structure to build the components, all the required information (interface, exceptions, events, etc) is described in the public elements of the classes that implement the component.
If we use the Enterprise Java Beans framework, included in JAVA enterprise edition, the only difference that can be found is that all those public elements are centralized on a single interface file. The architectural dependencies in a JAVA project can be found on the configuration files (properties file, classpath, etc.). These files are generated automatically by many Integrated Development Environments such as Eclipse6 and NetBeans7.
In the second step, *writing the metadata*, we must define a model based on the information collected in the previous step. In this way, we perform the three required activities (Figure 1) in the context of the JAVA platform and the component composition problem. These activities are:

1. *Defining the representation structure*: To assist in the dynamic component's composition problem it is important to have all the public information (related to the communication and configuration of the components) of the involved components. We need to know all the requirements that components need to be assembled with other components.
   Using these criteria, we define a minimal set of data that we need to have interface and public attributes (configuration variables), exceptions, triggered events, minimal JAVA version and other components and libraries it depends on. To structure this metadata we decide to use a two elements schema: a label and the data related to that label. The label is also divided into two elements, a category

and a sub-category. The label allows us to identify the metadata and classify the metadata according to the context. This way, we build the set of metadata shown in Table 1. For example, we can see that the metadata *FuncionalInterface/Method* handles all the information related to the signature of a public method defined inside a component.

Label	Sub-Label	Contained Data
FuncionalInterface	Method	Component's public Methods
FuncionalInterface	PublicParameter	Component's public Parameters
FuncionalInterface	Events	Component's triggered Events
FuncionalInterface	Exceptions	Component's triggered Exceptions
ArchiteturalInterface	JavaVersion	Minimal JAVA version
ArchiteturalInterface	InteralDependencies	Used subcomponents
ArchiteturalInterface	ExternalDependencies	Libraries, other components, databases etc.

Table 1: Set of metadata defined for the case study.

2. *Defining the generation method*: Due to the fact that we are using the JAVA platform, we use a provided generation method using JAVA annotations8. Using this method, we create an annotation for each of the elements defined on Table 1. We assign a descriptive name for each of this elements using the "Meta" prefix and attach an objective to each of them. A JAVA annotation "*objective*" informs the annotations engine which element of the code the annotations annotate (the element that must be written right after the annotation).
3. *Defining the recovery method*: Decisions of the way metadata are generated impact directly on the definition of the recovery method. Due to the fact that the metadata are written inside the components and not in external files we have to design a recovery method. To do so, we develop a parser capable of extracting the metadata and saving them in a JAVA class called "Metadata". The parser only requires that the components implement the "Metadatable" interface. We also develop a tool that fulfills these constraints and allows us to visualize, in a tabular way the metadata inside a component.

In the third step of the process, *reading and using the metadata*, we can observe how we read the metadata using a tool capable of interpreting the "Metadatable" interface assigned to a component (by using the created parser). These tools are related to the problem we are trying to solve, making the extraction independent from the use of the metadata. All the elements previously described (the annotations, "Metadatable" interface and parser) are packed inside a JAR library, allowing them to be imported by any projects that need them.

4.1 Case study

The JAR library was used in the implementation of a new component for the previously developed SPL. This SPL was developed in the context of the Geographic Information Systems (GIS) for the marine ecology sub-domain. It was developed using the Google Web Toolkit9 (GWT) framework, the Open Layers10 library and the GIS extension for postgres, PostGIS11.
The SPL services are implemented using one or more interconnected components, integrated with the Graphic User Interface(GUI) trough a manager called UIManager. The main objective of the component used in this case study is to create a history of maps, using some previously defined components and external libraries. This component is capable of cloning and generating new maps in parallel during a species distribution analysis.
In this way, we must be able to generate and store a history of derived maps. Figure 3 shows a slice of the component diagram for the SPL. In this figure, we can observe the new component MapHistory, that uses the Map Component and used by the UIManager trough the defined interface.

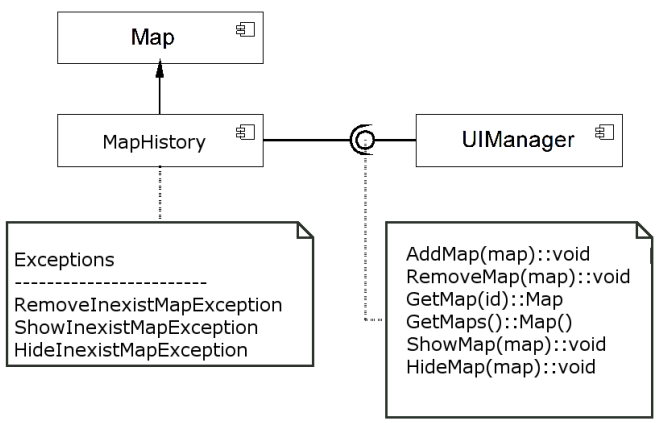


Fig. 3: Part of the component diagram for the SPL, showing the MapHistory component

In order to work correctly, our component uses a set of external libraries that include GWT, OpenLayers, ui-binder12, validation-api13 and others. For the exceptions and events, the component only generates an exception related to the existence of a map that we are trying to show, hide or remove. As a result of applying our process, we obtained an annotated component that can be used by any tool that can read these metadata.

To validate our process, we created a reading tool,which reads and show the embedded metadata in a tabular way. These data can be then used by a dynamic composition tool, letting it know the internal details of a closed component without the need of reverse engineer.

Figure 4, shows the metadata usage process for manual and automatic composition. Here we can observe how the written metadata are read using the parser and then organized for the usage of other tools. On one side, a dynamic composition tool is capable of reading the structure metadata and use it for a compatibility check and adapter generation. On the other side, we can observe the usage of a reading and visualization tool that shows the structured metadata to be used in manual composition.

Even when our process is not yet fully validated, it is easy to realize that using it will reduce the manual analysis effort. In this way, acquiring a component description becomes an easier task. If engineers want to perform an automatic composition, the component description will be structured and organized. It is important to know that as the number of components grow, the complexity of the manual analysis grows along.

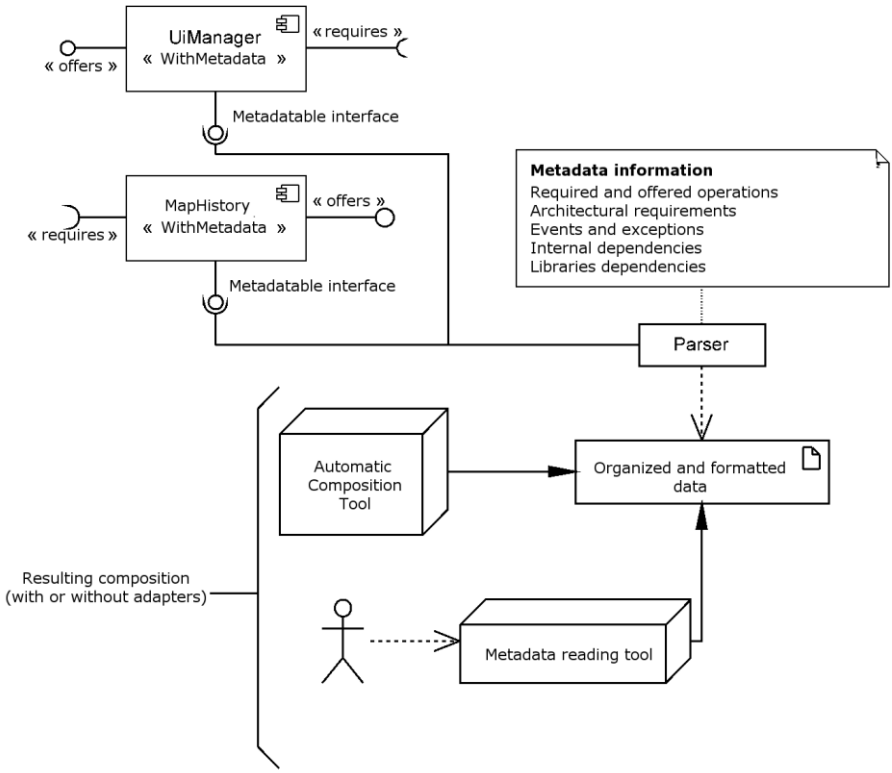


Fig. 4: Reading and usage process for manual and automatic composition.

Using metadata, each component is responsible for declaring and structuring its own information, removing the need for manual analysis. Finally, declaring a public access interface for the metadata, provides a point of access to the component's internal structure, working as self-documentation.

5. Conclusions and future work

We can conclude that a metadata system can assist, in an ad-hoc mode, to many software development process tasks. For this reason we have defined a general process for developing and using them in a consistent way.
In spite of using this process adds additional tasks to developers and it is susceptible to errors when the annotation fields are filled, it simplifies the automatic analysis making such errors be checked and corrected. In this way, we have proposed a starting point to assist to dynamic composition of components.
Therefore as we mentioned earlier, the manual writing task adds to the system a human-error factor. As future work we propose the automation of writing tasks through a code analyser solving the additional correction effort.

References

1. B.Y. Alkazemi. Towards a software product line framework to support cross-domain component composition. In *Computational Science and Its Applications (ICCSA), 2012 12th International Conference on*, pages 130–133, 2012.
2. M Nawaz Brohi and Fakhra Jabeen. Article: A metadata-based framework for object-oriented component testing. *International Journal of Computer Applications*, 41(15):8–18, March 2012. Published by Foundation of Computer Science, New York, USA.
3. John C. Grundy. Aspect-oriented requirements engineering for component-based software systems. In *Proceedings of the 4th IEEE International Symposium on Requirements Engineering*, RE '99, pages 84–91, Washington, DC, USA, 1999. IEEE Computer Society.
4. Mary Jean Harrold, Alessandro Orso, David Rosenblum, Gregg Rothermel, and Mary Lou Soffa. Using component metadata to support the regression testing of component-based software, 2001.
5. George T. Heineman and William T. Councill, editors. *Component-based software engineering: putting the pieces together*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
6. Natalia Huenchuman. Reestructuración de una línea de productos de software para el subdominio de ecología marina, 2013.
7. In-Gyu Kim, Doo-Hwan Bae, and Jang-Eui Hong. A component composition model providing dynamic, flexible, and hierarchical composition of components for supporting software evolution. *Journal of Systems and Software*, 80(11):1797 – 1816, 2007.
8. E. Martins, C.M. Toyota, and R.L. Yanagawa. Constructing self-testable software components. In *Dependable Systems and Networks, 2001. DSN 2001. International Conference on*, pages 151–160, July 2001.
9. Hong Mei, Feng Chen, Qianxiang Wang, and Yao-Dong Feng. Abc/adl: An adl supporting component composition. In *Proceedings of the 4th International Conference on Formal Engineering Methods: Formal Methods and Software Engineering*, ICFEM '02, pages 38–47, London, UK, UK, 2002. Springer-Verlag.
10. Alessandro Orso, Mary Jean Harrold, and David S. Rosenblum. Component metadata for software engineering tasks. In *Revised Papers from the Second International Workshop on Engineering Distributed Objects*, EDO '00, pages 129–144, London, UK, UK, 2001. Springer-Verlag.
11. Patricia Pernich. Diseño e implementación de una arquitectura de líneas de productos para servicios gis, 2011.
12. K. Pohl, G. Bockle, and F. Linden. *Software product line engineering: foundations, principles, and techniques*. Springer, 2005.
13. Matias Pol'la. Instanciación y validación de una línea de productos de software aplicada al subdominio de la ecología marina, 2013.
14. Ian Sommerville. *Software Engineering*. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 9. edition, 2010.

15. Judith A. Stafford and Alexander L. Wolf. Annotating components to support component-based static analyses of software systems. In *IN PROCEEDINGS OF GRACE HOPPER CONFERENCE 2000 (TO APPEAR), HYANNIS*, 2000.
16. G. Succi, R. Wong, E. Liu, and M. Smith. Supporting dynamic composition of components. In *Software Engineering, 2000. Proceedings of the 2000 International Conference on*, pages 787–, 2000.
17. Clemens Szyperski. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition, 1998.
18. Sahil Thaker, Don Batory, David Kitchin, and William Cook. Safe composition of product lines. In *Proceedings of the 6th International Conference on Generative Programming and Component Engineering*, GPCE '07, pages 95–104, New York, NY, USA, 2007. ACM.

XI

Database and Data Mining Workshop

Identifying Featured Articles in Spanish Wikipedia

LIAN POHN, EDGARDO FERRETTI, MARCELO ERRECALDE

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis, Ejército de los Andes 950, San Luis - Argentina
lian_pohn@hotmail.com, {ferretti,merreca}@unsl.edu.ar

Abstract. Information Quality assessment in Wikipedia has become an ever-growing research line in the last years. However, few efforts have been accomplished in Spanish Wikipedia, despite being Spanish, one of the most spoken languages in the world by native speakers. In this respect, we present the first study to automatically assess information quality in Spanish Wikipedia, where Featured Articles identification is evaluated as a binary classification task. Two popular classification approaches like Naive Bayes and Support Vector Machine (SVM) are evaluated with different document representations and vocabulary sizes. The obtained results show that FA identification can be performed with an F1 score of 0.81, when SVM is used as classification algorithm and documents are represented with a binary codification of the bag-of-words model with reduced vocabulary.

Keywords: Wikipedia, Information Quality, Featured Article, Support Vector Machine

1. Introduction

The online encyclopedia Wikipedia is one of the largest and most popular user-generated knowledge sources on the Web. Given the nature of user-generated Web content is commonly suspected of containing low-quality information. In particular, this question also applies to Wikipedia since its authors are heterogeneous and contributions are not reviewed by experts before their publication. Besides, considering the size and the dynamic nature of Wikipedia render a comprehensive manual quality assurance infeasible.
Information Quality (IQ) is a multi-dimensional concept and combines criteria such as accuracy, reliability and relevance. A widely accepted interpretation of IQ is the "fitness for use in a practical application" [1], *i.e.*, the assessment of IQ requires the consideration of context and use case. Particularly, in Wikipedia the context is well-defined, namely by the encyclopedic genre. It forms the ground for Wikipedia's IQ ideal, which has been formalized within the so-called *featured article criteria*.[1] Having a

[1] http://en.wikipedia.org/wiki/Wikipedia:Featured_article_criteria.

formal definition of what constitutes a high-quality article, *i.e.*, a featured article (FA) is a key issue; however, as indicated in [2], at present less than 0.1% of the English Wikipedia articles are labeled as featured.

A variety of approaches to automatically assess quality in Wikipedia has been proposed in the relevant literature. According to our review, there are three main research lines related to IQ assessment in Wikipedia, namely: (i) Featured articles identification [3–6]; (ii) Development of quality measurement metrics [7, 8]; and (iii) Quality flaws detection [9, 10]. In this paper we will concentrate on the first research trend mentioned above.

Since Wikipedia articles are collaboratively written and mainly maintained by volunteers, the rationale behind the idea: "the higher the number of edits and the number of editors; the higher should be an article's quality" is very reaso nable. Indeed, Wilkinson and Huberman [3] provide strong evidence that FA can be distinguished from non-FA by the number of edits and distinct editors. They also found that FA are characterized by a higher degree of cooperation, which is quantified by the number of revisions of the particular Wikipedia discussion pages. This finding agrees with the results reported in [7,11] on that production and cooperation among members of a wiki community allow quality content to emerge.

It is worth mentioning that in order to correctly identify FAs, it is not necessary to analyze complex patterns of author's interaction or edit history, as mentioned above. In fact, there exists a very simple approach [4] which have shown that a single word count feature can compete with sophisticated features when classifying FA and random articles, by achieving an accuracy of 0.96 when classifying an unbalanced test corpus composed by 1554 FA and 9513 random articles. Likewise, a novel approach [6] employs character tri-grams to classify a balanced set of FA and non-FA. With this method, originally applied for writing style analysis, an F1 score of 0.96 for FA identification was achieved.

Most of the aforementioned approaches have been proposed for the English Wikipedia, that containing more than 4500000 articles ranks among the top ten most visited Web sites in the world.[2] With 1110254 articles, Spanish Wikipedia ranks eighth in the list after Dutch, German, Swedish, French, Italian and Russian languages. In spite of being one of the eleven versions containing more than 1000000 articles,[3] few efforts have been made to assess IQ on Spanish Wikipedia. According to our literature review, [12] is the most relevant work related to IQ in Spanish Wikipedia and it can be characterized as belonging to the second main research trend mentioned above, viz. the development of quality measurement metrics.

In [12], Druck *et al.* examine the problem of estimating the quality of new edits in Wikipedia using implicit feedback from the community itself. That is, by observing the community's response to a particular edit, edit's quality can be estimated. The proposed quality metrics are based on the assumption that edits to an article that are retained in subsequent versions of the article

[2] Alexa Internet, Inc., http://www.alexa.com/siteinfo/wikipedia.org

[3] http://meta.wikimedia.org/wiki/List_of_Wikipedias

are thus of high quality, whereas edits that are quickly removed are of low quality. Hence, these community-defined measures of edit quality are learnt in statistical models to predict the quality of a new edit. Quality is predicted using features of the edit itself, the author of the edit, and the article being edited. Besides, a specific analysis of the model parameters is provided to determine which features are the most useful for predicting quality.
Despite the fact that the work performed by Druck *et al.* is a highly valuable step for automatic assessment of IQ in Spanish Wikipedia, the authors state that they originally intended to develop these ideas for the English Wikipedia and due to several consecutive failures in the complete history dump of the English version, they decided to work with the Spanish version.
In this respect, the contribution of our work relies on providing empirical evidence for IQ assessment in Spanish Wikipedia, a research trend not currently explored as it should, despite the practical relevance it has. We present the first study proposed to automatically identify FA in Spanish Wikipedia. Our research question concerns verifying if successful approaches for the English version like word count [4] and style writing [6] also work for the Spanish version, and if not, which changes are needed to accomplish a successful identification. With this aim, in Sect. 2, we formally state the problem faced in this work and we provide further details of existing approaches for the English Wikipedia. Section 3 describes the experimental design and results obtained to answer the research question posed above. Moreover, it compares our findings with results obtained for the English version. Finally, Sect. 4 offers the conclusions and briefly introduces future work.

2. Method

Given the question: Is an article featured or not? we have followed two approaches to answer it; videcelit, *the word count discrimination rule* [4] and *binary classification with character n-grams vectors* [6].
The word count discrimination rule (for the English Wikipedia), consists of clasifying as FA those articles having more than 2000 words. Despite its simplicity, this discrimitation rule achieved an accuracy of 0.96 for an unbalanced corpus (ratio 1:6, featured:non-featured) [4]. Nonetheless, this approach is usually taken as a baseline since it does not really address the challenge of learning the gist of what characterize a FA. As shown in Sect. 3, if a corpus contains FA and non-FA of similar lenghts, then this discrimination rule decreases its classification performance. It is worth mentioning that in our case, we have followed this approach but the threshold value has been set accordingly to Spanish language. The specific details of this approach are described in Sect. 3.1.
An *n*-gram vector of a text *t* is a numeric vector, where each dimension specifies the frequency of its associated *n*-grams in *t*. An *n*-gram in turn is a

substring of *n* tokens of *t*, where a token can be a character, a word, or a part-of-speech (POS) tag.

The *Term Frequency* * *Inverse Document Frequency* weighting scheme, commonly abbreviated as *TF-IDF*, was used for weighting the vector components. The term frequency $TF_{d,i}$ of the *i*-th term of the document *d* is the frequency of occurrence of the given term within the given text. Thus, *TF* is a text-specific statistic and it varies from one document to another, attempting to measure the importance of the term within a given document. On the other hand, the *Inverse Document Frequency* (*IDF*) is a global statistic and it characterizes a given term within an entire collection of *N* training documents. It is a measure of the *Document Frequency* (*DF*) of a given term *i* over the given collection (*i.e.*, it calculates how widely the term *i* is distributed), and hence of how likely the term is to occur within any given document. The purpose is to sub-estimate those terms that occur in many of the documents of the collection and, therefore, which are not relevant (when a term DF_i occurs in the *N* documents of the collection, its *IDF* value is equal to 0). In order to allow for variation in document size, the weight is usually normalized. The purpose and effect of weight normalization, is that the weight of a term in a given document (*i.e.*, its importance) should depend on its frequency of occurrence with respect to the other terms of the same document, not on its absolute frequency of occurrence. Weighting a term by its absolute frequency would obviously tend to favour longer documents.

The vector is called binarized if the occurrence or non-occurrence of an *n*-gram is counted as 1 and 0, respectively. In particular, in [6] several *n*-gram vectors where evaluated to illustrate how writing style matters with respect to our classification task of FA versus non-FA identification. From the three experimental setups performed in [6], in our work we only address one, viz. to evaluate a classifier by tenfold cross validation within a single Wikipedia domain. As stated in [6], the rationale of this experiment is to minimize the influence of topical discrimination, which can occur when articles of more than one domain are shuffled.

1.1 Terms Selection: the Information Gain Method

The number of terms of any given collection of texts of medium size may be approximately ten of thousands. It is very important to optimize the list of terms that identify the collection. This optimization is focused to reduce the number of terms eliminating those with poor information. For computational efficiency reasons, in space and time, the study of methods for reducing the numbers of terms in the vocabulary results of great interest. Moreover, some of these techniques help to improve the results of categorization in determined data sets, once noisy vocabulary is eliminated.

There are several methods for selecting the terms to remove [13], in our work, we have employed the Information Gain (IG) method [14]. IG measures the amount of information which contributes a term for the

prediction of a category, as a function of its presence or absence in a given text. The IG value of a term *i* is calculated as indicated in (1), where *m* is the number of existing categories, Pr(c_j) the probability that a text belongs to the category *j*, Pr(*i*) the probability of occurrence of the term *i* in the text, Pr($c_j|i$) the probability that a text belongs to the category *j* given that the term *i* occurs in the text, and Pr($c_j|\neg i$) is the probability that a text belongs to the category *j* given that the term *i* does not occur ($\neg i$ indicates no occurrence of term *i*). Once calculated the IG_i value for all the terms, those terms with the highest values are selected because they are the most relevant for the category selection.

$$\begin{aligned} IG_i = & -\sum_{j=1}^{m} \Pr(c_j) \log \Pr(c_j) \\ & + \Pr(i) \sum_{j=1}^{m} \Pr(c_j|i) \log \Pr(c_j|i) \\ & + \Pr(\neg i) \sum_{j=1}^{m} \Pr(c_j|\neg i) \log \Pr(c_j|\neg i) \end{aligned} \tag{1}$$

3. Analysis

In this section, we report on the experiments performed to assess the effectiveness of the above-mentioned approaches for FA identification when articles from several domains are shuffled.

1.2 Experimental Design

Due to the lack of a standard corpus related to the study we have performed, we created two corpora, namely: a balanced corpus and an unbalanced corpus. It is worth noting that "balanced" means that FA and non-FA articles were selected with almost similar document lengths. In a similar manner, "unbalanced" refers to the fact that non-FA articles were randomly selected without considering their average lengths. Both corpora are *balanced* in the traditional sense, *i.e.*, the positive (FA) and negative (non-FA) class contain the same number of documents. In particular, the balanced corpus contains 714 articles in each category and the unbalanced one has 942 articles in each category as well.[4] It is ensured that non-FA articles belonging to the balanced corpus has more than 800 words.

The articles belong to the snapshot of the Spanish Wikipedia from 8th, July 2013. Featured articles were identified by searching for files in the dump that

[4] https://dl.dropboxusercontent.com/u/71979810/Corpus.tar.xz

contained the FA template in the Wikitext. As negative class, we used non-FA that were selected from among the remaining articles in the dump. Wikitext files were parsed[5] to get their corresponding plain texts, viz. without symbols belonging to the *MediaWiki Markup Language*. To get the character *n*-grams from the plain texts, we programmed our own Java application given that our experiments were intended to try several *n*-gram features in the document models. In particular, *n*-grams with $n \in \{3, 4, 5\}$ were extracted for all the plain texts, and were used as features with a binary document model (*bnn* codification from the SMART nomenclature [15]) and an *ntc* TF-IDF weighting scheme (see also [15]).

As stated in [6], POS *n*-gram vectors and character *n*-gram vectors are writing-style-related since they capture intrinsics of an author's text synthesis traits. Likewise, character *n*-grams unveil preferences for sentence transitions as well as the utilization of stopwords, adverbs, and punctuation. In particular, in articles written in English, character tri-grams have shown to be very discriminative features for writing style analysis [16]. That is why, in [6], character tri-grams vectors are used to represent the articles.

In our case, given that we are working with the Spanish version of Wikipedia, also 4-grams and 5-grams were evaluated. This is due to the fact that many stopwords and adverbs are characterized better in Spanish with larger character *n*-grams ($n > 3$). For instance, many adverbs of place like: *aquí*, *allí*, *allá*, *abajo*, *cerca*, *lejos*, *atrás*, etc. are fully encompassed in 4-grams or 5-grams and the same occurs for many other kind of adverbs.

To perform the experiments we have used the WEKA Data Mining Software [17], including its SVM-wrapper for LIBSVM [18]. All the results presented in Table 1 are average values obtained by applying tenfold cross-validation. For SVM, results have been obtained with the linear kernel. Parameter $C = 2^5$ was experimentally derived ranging its values in the set $\{2^{-5}, 2^{-3}, 2^{-1}, \ldots, 2^{13}, 2^{15}\}$. It is also a good theoretical compromise value, since having lower values for this parameter gets wider margins for the hyperplane drew by the classifier, thus allowing more misclassified documents. Conversely, having high penalty values (*e.g.*, $C = 2^{15}$), may yield in an over-fitting of the model and hence a poor capability of generalization of the classifier.

1.3 Results

To begin with, we evaluated the word count discrimination rule. In this context, classification performs as follows: each article having more words than a certain discrimination threshold empirically derived, is predicted as featured and as non-featured instead. As it can be observed in Fig. 1, for the unbalanced test set this rule achieved an F1 score of 0.91, when the threshold was set to 3070 words. Likewise, for the balanced test set it achieved an F1 score of 0.66 for a discrimination threshold of 955 words. Naturally, when

[5] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor.

FA and non-FA articles' average length are similar, the performance of this discrimination rule decreases. Likewise, for the unbalanced setting, it was expected that for the Spanish version the discrimination value would be greater than 2000 words (as reported in [4]) since articles in this language tend to be longer than in English.

As mentioned in Sect. 2, the word count discrimination rule has been taken as a baseline since it does not really address the challenge of learning the gist of what characterize a FA. In opposition, having an explicit document model for the articles (in our case, *n*-gram vectors and bag-of-words (BOW)) should help in capturing those aspects characterizing a FA. In this respect, Table 1 shows all the results we have obtained from our different experimental settings.

Last four columns of Table 1 presents the results for NB and SVM classifiers with TF-IDF and binary document models for the unbalanced corpus. As expected, SVM performs slightly better than NB in both document models, and the performance of both classifiers clearly improves for the case of the binary vector. It is worth mentioning than for the binary representation of documents, both classifiers perform as good as the baseline of F1 = 0.91, and that SVM performs best for all the space features, achieving the highest F1 score of 0.94 for 3-gram and 4-gram features. This finding agrees to that previously mentioned, *i.e.*, if the document model is appropriate in characterizing FAs, then a binary classifier performs better than the baseline.

Given that FA discrimination is an easy task for the unbalanced corpus, we did not perform any further kind of analysis for this corpus, like performing an operating point analysis on the parameters of SVM classifier or verifying the impact that vocabulary size reduction has in classification performance.

Regarding the balanced corpus, the binary document model also outperforms the *ntc* TD-IDF weighting scheme, but in this case, the differences in performance are not that significant than that for the unbalanced corpus; 19% of average improvement (over all features) for the unbalanced setting against 13% of the balanced setting, considering the SVM classifier with full vocabulary size. Besides, SVM is the best performing classifier when applied to a binary vector representation with full vocabulary size, being 0.8 the best F1 score achieved when 4-grams are used as features.

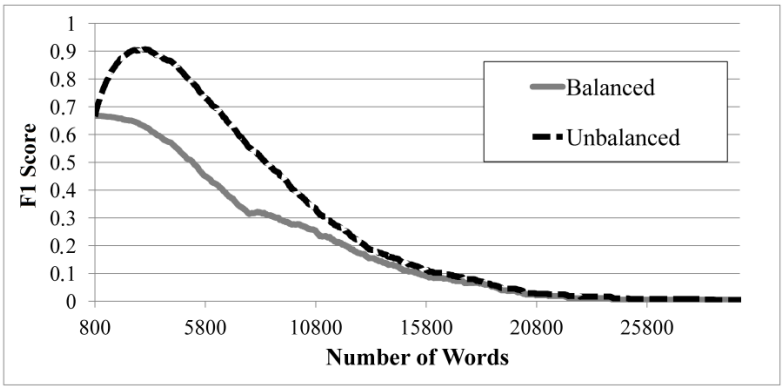


Fig. 1. Experimental setting of the word-count discrimination threshold. Y axis presents F1 score of classification performance with respect to the number of words used as discrimination threshold (X axis).

Table 1. F1 performance values for all the combinations of features and vocabulary sizes with both classification approaches on balanced and unbalanced test sets

Corpora	**Balanced test set**								**Unbalanced test set**			
Weighting	**TF-IDF**				**Binary**				**TF-IDF**		**Binary**	
Classifier	**SVM**		**NB**		**SVM**		**NB**		**SVM**	**NB**	**SVM**	**NB**
Vocabulary / *Features*	100%	10%	100%	10%	100%	10%	100%	10%	100%	100%	100%	100%
3-grams	0.69	0.68	0.68	0.75	0.77	0.73	0.73	0.75	0.79	0.77	0.94	0.90
4-grams	0.70	0.73	0.61	0.71	0.80	0.74	0.73	0.74	0.78	0.75	0.94	0.91
5-grams	0.70	0.71	0.64	0.69	0.79	0.70	0.72	0.73	0.78	0.76	0.93	0.91
BOW	0.69	0.72	0.66	0.78	0.77	0.81	0.70	0.74	0.77	0.76	0.93	0.90

Despite the fact that F1 = 0.8 is a good result, if compared to the performance value of F1 = 0.96 reported in [6], the result is not that satisfactory. That is why, we performed an operating point analysis on SVM different kernels and its related parameters. We found out that no improvement is achieved for the binary vector representation with full vocabulary size. This fact indicates that the classification problem we are facing is not linearly separable, and the kernel tricks do not help at all given the input (features) space we are using. That is why, we carried out the same experiments for a reduced space of features selecting them by the IG method. As mentioned above in Sect. 2.1, there are studies which have shown that reducing the vocabulary can help the classification performance in some application domains. Besides, in our particular case, reducing the input space of the classifier could help when trying other kernels than the linear one, *i.e.*, the feature space obtained by the kernels from a reduced input space could be more prone to be linearly separable than in the original case.

Table 1 shows the results for a reduced vocabulary of 10%, for the balanced corpus only. As it can be observed, NB increases its performance for the *ntc* TF-IDF document model with reduced vocabulary and a slightly improvement is also achieved for the binary vector representation. Given that NB is a statistical classifier which obtains/approximates the a-priori/conditional probabilities from the training set, one noisy vocabulary is removed, its classification performance increases. SVM also increases its performance with a reduced vocabulary for the *ntc* TF-IDF document model but the improvement is less important than for NB. Conversely, for the binary document model with reduced vocabulary SVM did not improve its performance except for the case when BOW is used as features. The F1 score achieved for this case is 0.81.

Finally, in order to explore if F1 = 0.81 could be improved, for the binary document model with reduced vocabulary we performed an operating point analysis with the RBF kernel and the polynomial kernel as well. The obtained results were similar to the ones achieved by the linear kernel. In particular, γ values very close to zero (*e.g.*, 2^{-5} or lower) and $C = 2^5$ (or higher) reported the best values for the RBF kernel, thus yielding a configuration quite close to a linear kernel. Similarly, $d = 3$, $r = 1$ and $C = 2^5$ (or higher) reported an F1 = 0.81, for the polynomial kernel, hence no improvement was accomplished.

It is well known that increasing γ and d parameters from the RBF and polynomial kernels allow a more flexible decision boundary. A more flexible decision boundary in the input space means having more dimensions in the feature space generated by the kernels. Besides, setting $r = 1$ helps learning since this parameter role is the same that setting the bias $b = 1$ in Artificial Neural Networks. Thus, based on the theoretical properties of the kernels and given that the empirical performance achieved is not better than a linear kernel, we can conclude that the document models we are using should be improved in o rder to get a better performance close to that of F1 = 0.96, achieved for the English Wikipedia.

Finally, it is worth mentioning that considering the reduced vocabularies for the BOW feature, both document codifications have the words: *2012*, *2011*, *2010*, *nacionales*, *república*, *participación*, *presidencia*, *sede* and *partido*, ranked in the first twenty positions. In our view, this is due to the fact that *History* domain is the only one with more than 100 FA in the Wikipedia snapshot we are using. Hence, words and years related to history, would naturally be the most discriminative terms to distinguish FA versus non-FA. Likewise, for the *ntc* codification of TD-IDF, ranked among the first ten positions we also find the terms: *enlaces*, *externos*, *notas* and *bibliografía* which refer to a proper structure of an article, which is accomplished by each FA.

4. Conclusions

In this work we have evaluated two approaches for FA identification in Spanish Wikipedia, videlicet, the word count discrimination rule and binary classification with character *n*-gram vectors. These approaches, originally proposed for the English Wikipedia have shown their good performances (see [4] and [6]). One contribution of our work is presenting the first empirical comparison of the above-mentioned approaches for English and Spanish Wikipedias.

Given the basic principle underlying the word count discrimination rule, when the discrimination threshold is properly set, it performs well for corpora where average lengths of FA and non-FA are dissimilar. Tri-grams vectors have been proven to be very effective for FA discrimination in the English Wikipedia but for the Spanish version, BOW and *n*-grams with $n > 3$ performed better in general. As mentioned above, this can be due to the case that in Spanish many kind of adverbs are fully encompassed in 4-grams or 5-grams. The best F1 scores achieved were 0.8 and 0.81, when SVM is used as classification algorithm, documents are represented with a binary codification, and 4-grams and bag-of-words are used as features, respectively.

In order to have a proper explanation on the poor performance achieved with the (most popular) *ntc* weighting scheme, we should evaluate the 20 combinations resulting from the SMART codification nomenclature. As shown in [19], the best weighting scheme heavily relies upon the collection used. This issue will be explored as future work. Moreover, we also plan to continue evaluating existing approaches for the English Wikipedia on the Spanish Wikipedia, but on all the research trends mentioned in the introductory section.

To conclude, it is worth mentioning that an important "by-product" of the empirical study performed and the conclusions derived in this work, is the compilation of a corpus to study classification techniques for FA vs. non-FA discrimination for the Spanish Wikipedia. To the best of our knowledge, this is the first existing corpus for this classification task for Spanish language.

5. Acknowledgments

This work has been funded by the European Commission as part of the WIQ-EI project (project no. 269180) within the FP7 People Programme. The authors thank the Universidad Nacional de San Luis from which receive continuous support (PROICO 30312).

References

1. Wang, R., Strong, D.: Beyond accuracy: what data quality means to data consumers. Journal of management information systems **12**(4), 5-33 (1996).
2. Anderka, M., Stein, B.: A breakdown of quality flaws in Wikipedia. In: 2nd joint WICOW/AIRWeb workshop on Web quality (WebQuality'12), pp. 11-18. ACM (2012).
3. Wilkinson, D., Huberman, B.: Cooperation and quality in Wikipedia. In: 3th international symposium on wikis and open collaboration, pp.157-164. ACM (2007).
4. Blumenstock, J.: Size matters: word count as a measure of quality on Wikipedia. In: 17th international conference on World Wide Web, pp.1095-1096. ACM (2008).
5. Lex, E., Völske, M., Errecalde, M., Ferretti, E., Cagnina, L., Horn, C., Stein, B., Granitzer, M.: Measuring the quality of web content using factual information. In: 2nd joint WICOW/AIRWeb workshop on Web quality (WebQuality), pp. 7-10. ACM (2012).
6. Lipka, N., Stein, B.: Identifying featured articles in Wikipedia: writing style matters. In: 19th international conference on World Wide Web, pp. 1147-1148. ACM (2010).
7. Lih, A.: Wikipedia as participatory journalism: reliable sources? Metrics for evaluating collaborative media as a news resource. In: Proceedings of the 5th international symposium on online journalism, pp.16-17 (2004).
8. Stvilia, B., Twidale, M., Smith, L., Gasser, L.: Assessing information quality of a community-based encyclopedia. In: Proceedings of the 10th international conference on information quality (ICIQ'05), pp. 442-454 MIT (2005).
9. Anderka, M., Stein, B., Lipka, N.: Detection of text quality flaws as a one-class classification problem. In: Proceedings of the 20th ACM international conference on information and knowledge management (CIKM'11), pp. 2313-2316 ACM (2011).
10. Anderka, M., Stein, B., Lipka, N.: Predicting Quality Flaws in User-generated Content: The Case of Wikipedia. In: 35rd annual international ACM SIGIR conference on research and development in information retrieval, pp. 981-990 ACM (2012).
11. Anthony, D., Smith, S., Williamson, T.: Reputation and reliability in collective goods: The case of the online encyclopedia wikipedia. Rationality & Society **21**(3) 283-306 (2009).
12. Druck, G., Miklau, G., McCallum, A.: Learning to predict the quality of contributions to wikipedia. WikiAI **8** 7-12 (2008).
13. Yang, Y., Pedersen, J.O.: A comparative study on feature selection in text categorization. In: 14th International Conference on Machine Learning, pp.412-420 (1997).
14. Lewis, D.D., Ringuette, M.: A comparison of two learning algorithms for text classification. In: 3rd Annual Symposium on Document Analysis and Information Retrieval, pp. 81-93 (1994).
15. Salton, G.: The Smart Retrieval System: Experiments in Automatic Document Processing. Prentice Hall (1971).

16. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for Information Science and Technology **60**(3) 538-556 (2009).
17. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations **11**(1) (2009).
18. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** 27:1-27:27 (2011).
19. Ferretti, E., Errecalde, M., Rosso, P.: The Influence of Semantics in Text Categorisation: A Comparative Study using the k Nearest Neighbours Method. In Prasad, B., ed.: IICAI, IICAI (2005) 749-768.

IX

Architecture, Nets and Operating Systems Workshop

WDM Network Design with Node Protection
An approach based on MOACO

CARLOS BELLINO, ADITARDO VÁZQUEZ, AND DIEGO P. PINTO-ROA

Facultad Politécnica UNA,
Av. Mariscal Estigarribia, San Lorenzo, Paraguay
{carlosbellino, vazquez.aditardo, dppintoroa}@gmail.com
http://www.pol.una.py/

Abstract. This work studies the survivable optical network design problem subject to simple node failure where disruption by network reconfiguration degrades the Quality of Service. The impact of the number of reconfigurations when a node fails over other objective functions is critical. Therefore, a Multi-Objective Ant Colony Optimization (MOACO) Algorithm is proposed, which tries to find the best network design as well as the primary and back-up multicast trees considering a multicast request set. The MOACO algorithm simultaneously minimizes the network design cost, the maximum end-to-end optical delay, the total number of reconfigurations and the maximum number of reconfigurations. The experimental results over different instances show the benefits of the proposed approach front to two state-of-the-art protection approaches based on total or partial reconfiguration.

Keywords: Optical Networks, Multicast, Node Failure, Protection, Reconfiguration, Multi-objective Optimization, Ant Colony Optimization.

1. Introduction

Optical networks have evolved substantially to cope with the growth in traffic demand. [1]. High demand services for the last decade are multimedia and video conferencing among others. In this context, major architectural changes are given by (i) the WDM (Wavelength Division Multiplexing solution that has expanded the capacity of the optical fiber, and (ii) OXC (Optical Cross Connects) devices with which network scalability is achieved [1, 2].

Several services demand multicast transmissions, which are transmitted on the network through light trees that connect one source node to other destinations nodes. Calculating light tree is known as the problem of MRWA (Multicast Routing and Wavelength Assignment). MRWA is a NP-complete problem and optical communications focus, in which RWA is a particular case where there is a source node and a single destination node [3].

This paper addresses the problem of WDM networks design and protection against single node failures considering static multicast traffic in a multi-fiber network architecture without wavelength conversion and at the same time

improve the quality of service and total cost of the network. The problem addressed is called OM-PNF (Optical Multicast and Protection against Node Failures) [4]. The OM-PNF looks to calculated for each multicast request the primary light tree and a set of backup light trees for each failed node. In at time of network reconfiguration data losses are caused by unnecessary disruptions. These reconfigurations are generated by a network design in which the cost of it is prioritized. The OM-PNF has been discussed in literature without studying the problem of minimizing the reconfiguration [5, 4].

In this context, the main contribution of this paper is to propose a system based on multi-objective ant colonies (MOACO, Multi-Objective Ant Colony Optimization) for calculating primary light trees and backups [6] in which it seeks to minimize not only the cost of network design approach but also the Quality of Service given in the end to end maximum delay, destinations total number of reconfigurations and destinations maximum number of reconfigurations; these last two are critical to the disruption of services.

The rest of the article is organized as follows: The mathematical formulation of the problem is discussed in Section 2. In Section 3 the proposal of this work is presented. Experimental tests are discussed in Section 4. Finally, conclusions and future work are presented in Section 5.

2. Problem Formulation

For a correct reading of the mathematical formulation of the problem based on [4] the following symbols is proposed:

$\|.\|$	Cardinality of a set.
V	Set of network nodes.
E	Set of network links.
$(i,j) \in E$	Link from node i to the node j; $i,j \in V$.
$G = (V,E)$	Network graph formed by V and E.
$V(i) \subset V$	Subset of nodes; $V(i) = V - \{i\}$.
$E(i) \subset E$	Subset of links; $E(i) = E - \{(i,j),(j,i): \forall j \in V\}$.
$G(i) \subset G$	Subgraph network; $G(i) = (V(i), E(i))$.
$s \in V$	Source node of a multicast request.
$d \in V$	Destination node of a multicast request.
D	Set of multicast destinations nodes.
$m = (s, D)$	Multicast request.
M	Set of multicast requests; $M = \{m_1, m_2, .., m_{\|M\|}\}$
f_{ij}^{k}	k-th fiber optic in link (i,j).
$\ell_{ij}^{k\lambda}$	Light link consisting of the λ-th wavelength in the k-th fiber optic in link (i,j).
t_m	Tree light to a request m; $t_m = (V', \ell)$ with $V' \subseteq V$ and $\ell = \{ \ell_{ij}^{k\lambda} \}$ with $V' = \{i_1, i_2, ..., i_n\}$ where $s \in V'$, $D \subseteq V'$.
p_m^{sd}	Light path between node s and a node d in t_m; $p_m^{s,d} = (V'', \ell)$ where $V'' \subseteq V'$, $V'' = \{i_1 = s, i_2, ..., i_n = d\}$.

$\lambda_{i,j}^{m}$ Wavelength assigned to t_m in link *(i,j)*.
T Tree set light to set M; $T = \{t_1, t_2, \ldots, t_{|M|}\}$.
T_p Set of primary light trees for all M in graph G.
T_s^i Set of secondary light tree to set M in sub-graph *G(i)*.
X_{ij}^{msd} Variable indicator, if $\ell_{ij}^{k\lambda} \in p_m^{sd}$ then $X_{ij}^{msd} = 1$, otherwise $X_{ij}^{msd} = 0$.
$R_{i,j}$ Distance between node i and j node in kilometers.
$R = [R_{ij}]$ Matrix distance between nodes.
$K \in N$ Maximum number of optical fibers per link.
$F_{i,j} \leq K$ Number assigned to optical fiber link *(i,j)*.
F_{ij}^{k} Variable indicator, if link *(i,j)* has assigned k-th optical fiber then $F_{ij}^{k} = 1$, otherwise $F_{ij}^{k} = 0$.
$F = \left[f_{ij}^{k} \right]$ Fiber optic vector location.
A_{ij}^{k} Variable indicator, if the k-th fiber optic in link (i,j) has allocated an amplifier then $A_{ij}^{k} = 1$, otherwise $A_{ij}^{k} = 0$.
$A = [A_{ij}^{k}]$ Amplifiers vector location.
B_i Number of light splitters in node *i*.
$B = [B_i]$ Light splitters vector location.
Γ Solution to OM-PNF problem; $\Gamma = \{A, B, F, T_p, T_1^s, T_2^s, \ldots, T^{|V|}{}_s\}$.
Λ Maximum number of wavelengths carrying WDM system.
$C_{i,j}$ Cost of link *(i,j)*.
C_i Cost of node *i*.
c_f Cost of optical fiber per kilometers.
c_a Cost of amplifiers per kilometers.
c_b Cost of a light splitter.
c_w Cost of a WDM system (MUX, DEMUX).
c_w^p Cost of a WDM port.
c_o^p Cost of a OXC port.
$S_{ij}^{k\lambda}$ Signal strength of light link $\ell_{ij}^{k\lambda}$.
$N_w(i)$ Number of multiplexers/demultiplexers in node *i*.
$N_w^p(i)$ Number of multiplexer port in node i.
$N_o^p(i)$ Number of OXC ports in node i.
$N_b(i)$ Number of light splitters in node i.
O_s Max output ports of a light splitter (fan-out).

Given a graph G which represents the optical network and a set of multicast requests M, the OM-PNF problem can be treated as a Multi-objective Optimization Problem [7] where the best Γ solution is sought, and it simultaneously seeks to minimize the cost of network f_1, the maximum delay f_2, the total number of adjustment reconfigurations f_3, and the maximum

number of adjustment reconfigurations f_4, subject to constraints of the continuity of light e_1, wavelength used e_2, receiving minimum power e_3, maximum number of optical fibers per link e_4 and maximum number of branches per node in a light tree e_5.
At this point it should be noted that the reconfiguration of primary light trees is done for two reasons: (1) when a node $i \in t_m$ fails, and (2) when a resources optimization is needed. In the latter case it is called adjustment reconfiguration which we want to minimize as Quality of Service.

Objective Functions

- Network Cost [8, 4]:

$$f_1(\Gamma) = \sum_{(i,j)\in E} C_{ij} + \sum_{i\in V} C_i \tag{1}$$

- where

$$C_{ij} = (c_f + c_a) \cdot R_{ij} \cdot F_{ij} \tag{2}$$

$$F_{ij} = \sum_{k=1}^{K} F_{ij}^k \tag{3}$$

$$C_i = c_w \cdot N_w(i) + c_w \cdot N_w(i) + c_o^p \cdot N_o^p(i) + c_b \cdot N_b(i) \tag{4}$$

- Maximum Delay [9, 4]:

$$f_2(\Gamma) = \max\{\Delta T : T \in \Gamma\} \tag{5}$$

- where Δ_T is the maximum delay for the whole light trees T set:

(6)

$$\Delta_T = \max_{t_m \in T} \left\{ \max_{d \in D_m} \left\{ \sum_{(i,j) \in E} X_{ij}^{m,s,d} \cdot R_{ij} \right\} \right\}, \forall T \in \Gamma$$

- Total Number of Adjustment Reconfigurations:

$$f_3(\Gamma) = \sum_{m \in M} \sum_{d \in m} \sum_{i \in E} Z_{md}^i \tag{7}$$

- where $Z_{md}^i = 1$ if $p_m^{sd} \neq \rho_m^{sd}$ with $p_m^{sd} \in \tau_m \in T_p$ and $i \neq s, i \notin D$, otherwise $Z_{md}^i = 0$.

- Maximum Number of Adjustment Reconfigurations:

$$f_4(\Gamma) = \max_{m \in M} \sum_{d \in m} \sum_{i \in E} Z_{md}^i \tag{8}$$

The details of network components are as follows:

- Number of multiplexers and de-multiplexers:

$$N_w(i) = \sum_{j \in V} (F_{ij} + F_{ji}) \tag{9}$$

- Number of WDM ports (multiplexers and de-multiplexers):

$$N_w^p(i) = \Lambda \cdot \sum_{j \in V} (F_{ij} + F_{ji}) \tag{10}$$

- Number of ports in optical switches:

$$N_o^p(i) = N_w^p(i) \tag{11}$$

Constraints

- e_1 - Wavelength continuity: each tree light t must use the same wavelength on all links that belong to it.

$$\lambda_{ij}^{m} = \lambda, \quad \forall (i,j) \in t, \forall t \in T \tag{12}$$

- e_2 - Wavelength Use: whether $\ell_{ij}^{k\lambda}$ a t_m light tree, any two light trees t_{ma} y t_{mb} They cannot use the same wavelength λ in the same optic fiber f_{ij}^{k} .

$$\ell_{ijm_a}^{k\lambda} = \ell_{ijm_b}^{k\lambda}, \forall t \in \Gamma \tag{13}$$

- e_3 - Minimum power on links: each light link $\ell_{ij}^{k\lambda} \in t$ must transmit an $S_{ij}^{k\lambda}$ greater than or equal to a power S_{min} given the sensitivity of the optical sensors in the receiver.

$$S_{ij}^{k\lambda} \geq S_{\min}, \ \forall \ell_{ij}^{k\lambda} \in t, \forall t \in \Gamma \tag{14}$$

- e_4 - Maximum number of optical fibers per link: each link $(i,j) \in E$ may be assigned at most K optical fibers.

$$F_{ij} \leq K, \forall (i,j) \in E \tag{15}$$

- e_5 - Max fan-out: each light tree t may have a maximum number of branches in each network node; this is subject to the number of output ports of each light splitter.

$$\left| \left\{ \ell_{ij_1}^{k\lambda}, \ell_{ij_2}^{k\lambda}, ..., \ell_{ij_n}^{k\lambda} \right\} \right| \leq O_s, \forall i \in V \tag{16}$$

- where $N_i = \{j_1, j_2, ..., j_n\}$ are n adjacent nodes to node i; therefore, $\ell_{ij}^{k\lambda}$ are light links that leave node i and reach adjacent nodes N_i.

3. Proposed Solution

Several approaches to MOP problems have been proposed in the literature [10]. The algorithm proposed in this paper is based on MOACO [6] for the OM-PNF problem, which uses a colony of H ants and a pheromones matrix τ to build in every generation w solutions Γ. Each solution Γ consists of a primary light tree and |V| protection/secondary light trees. For each multicast request, a fail in node i will activate protection light trees T_s^i, simultaneously, the primary light trees T_p will be disabled according to the policy used. Basically, the ant progressively build a tree of light from the node s to reach all nodes $d \in D$ used as a means of decision pheromone matrix τ and visibilities η. During processing the ant adds light links to t_m while checks whether you need to add light splitters, amplifiers and optical fibers. Once it reaches all destinations it proceeds to eliminate those light links that makes free branches.

3.1 Proposed Approach Protection

Once a primary light tree $t_m \in T_p$ is built, for each network node i secondary light trees must be constructed. State of the art works as [5] propose two approaches, total and partial reconfiguration, which generate particular solutions. We propose a general approach based on Hybrid Reconfiguration for building secondary light trees, which exploits the benefits of above approaches.

In partial reconfiguration search space is much lower while in total reconfiguration the chance to find less expensive solutions is much higher. The approach of hybrid reconfiguration receives as a parameter a sub-graph *G(i)* which is composed by nodes and links of the original graph, except the node that we assume failed and its links to adjacent nodes, a set of requests M for the current solution, the set of primary trees of active sessions, the failed node to generate the corresponding protection and returns the subset of primary trees M_i to be reconfigured. Calculating set M_i has two parts: (a) resources sessions traversing the failed node are released and these sessions are added to the subset of sessions M_i to reconfigure. (b) A random selection of other sessions that are unrelated to the failed node, to also be added to the sub-set M_i it is then performed. Note that if the set $M_i = M$ is a total reconfiguration and if $M_i = \{m;\ t_m \in T^i_s\}$ is a partial reconfiguration, which are particular cases of hybrid reconfiguration approach.

3.2 Proposed Objective Functions

The work of the state of the art [4] proposes not only minimize the cost of network design but also the quality of service considering only the maximum delay end to end. This paper also proposes the Quality of Service given in the

packet loss caused by disruptions during reconfiguration of the primary light trees. In this context, it is considering total reconfiguration of the solution and the maximum perceived by a multicast session as a measure of fairness.

In order to guide the MOACO rapid convergence is used as heuristics the local visibility η; which it consists of the visibility distance $\eta^d = [\eta_{ij}{}^d]$ associated to cost and delay, optical fiber visibility $\eta^\lambda = [\eta_{ij}{}^\lambda]$associated to cost, and reconfiguration visibility $\eta^r = [\lambda_{ij}{}^r]$ associated with the disruption. In the latter case visibility is the weight that is given to links used in the primary tree in order to re-use similar main routes in the protection.

4. Experimental Tests

The simulations have been developed in National Science Foundation Network (NSFNet) topology that consists of 14 nodes and 42 links [10]. The algorithms were implemented in Java programming language and it is executed on a computer with the following characteristics: Intel Core i7-2670QM CPU 2.2GHz, 4GB RAM and ArchLinux 64bit as OS. They have been used 25 different loads to the tests summarized in Table 1. The network architecture and setting used in tests are summarized in Table 2.

Table 1: Test Scenarios

Number of Sessions per node	Number of destinations per session
4	3
6	6
8	8
10	10
12	13

Table 2: Experimental Parameters

Network Architecture	Symbol	Details
Number of wavelengths	λ	10
Maximum number of Light Splitters	O_s	3
Minimum power	S_{min}	0.33
ACO Settings		
Number of ants	H	40

% Pheromone evaporation	p	0.1
Importance of pheromones	α	1
Importance of Visibility	β_1	Random(0,100)
Importance of Visibility	β_2	Random(0,100)
Importance of Visibility	β_3	Random(0,100)
Pseudo-random Probability	0.9	see [6]
Fiber Optical Cost	c_f	0.1 per kilometer
Optical Amplifiers Cost	c_a	0.6 per kilometer
WDM System Cost	c_w	6
WDM Port Cost	c_w^p	0.5
OXC Port Cost	c_o^p	0.6
Light Splitter Cost	c_b	2

A set Alg = {MOACO-PNF-RT, MOACO-PNF-RP y MOACO-PNF-HI} of implemented algorithms for testing is defined, where RT is the total reconfiguration, RP is the partial reconfiguration and HI is the hybrid reconfiguration. For each multicast group M, sets of solutions were calculated. For each algorithm a $\in$ Alg, 10 runs were performed. For every algorithm 10 set of non-dominated $Y^a = \{Y^a_1, Y^a_2,...,Y^a_{10}\}$ calculated one was obtained per each run. For each algorithm is obtained an approximate Pareto set F^a_{known} by removing dominated solutions Y^a. The stopping criterion for each run was 30 minutes, experimentally adopted. In brief 10 runs * 3 algorithms * 25 multicast groups * 1 Topology were performed; totaling 375 hours of experimental tests. The distances between adjacent nodes were obtained from [8].

Results are presented using the metric comparison of dominance between Pareto fronts [7] which are detailed below.

Total reconfiguration performance:

The results demonstrate certain dominance on partial solutions and hybrid reconfiguration, only when each destination session takes every available node in the network (Broadcast). This dominance as shown in Figure 1 is substantially greater on the partial reconfiguration than over hybrid reconfiguration solutions.

Partial reconfiguration performance:

The set of solutions of partial reconfiguration (Figure 2) completely dominates total reconfiguration solutions only in cases where the number of destinations per session is three. While the dominance of hybrid reconfiguration solutions, ranging from twelve to seventy-five percent in cases where the number of possible destinations is not the highest.

Hybrid reconfiguration performance:

The solutions provided by the hybrid reconfiguration (Figure 3) denote a wide dominance over total reconfiguration solutions in cases where the number of destinations is less than the maximum number of possible destinations, while this dominance is reduced when the number of destinations per session is maximum, but, occurring the opposite on partial reconfiguration solutions. This demonstrates a wide dominance only in some cases in which the number of destinations per session is the maximum number of possible destinations.

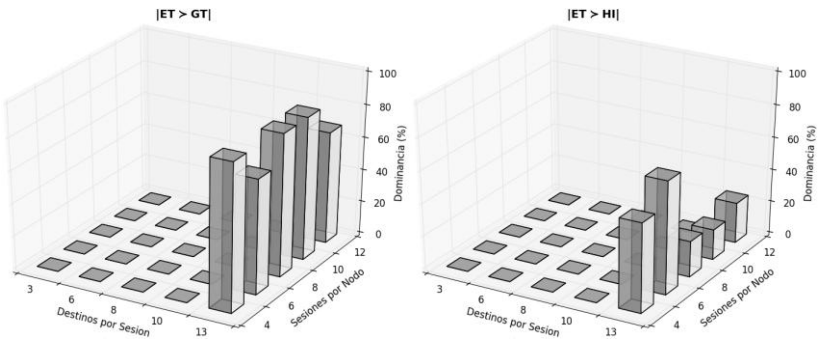


Fig. 1: Percentage dominance of RT (Total reconfiguration) solutions over RP (Partial Reconfiguration) and HI (Hybrid Reconfiguration) solutions.

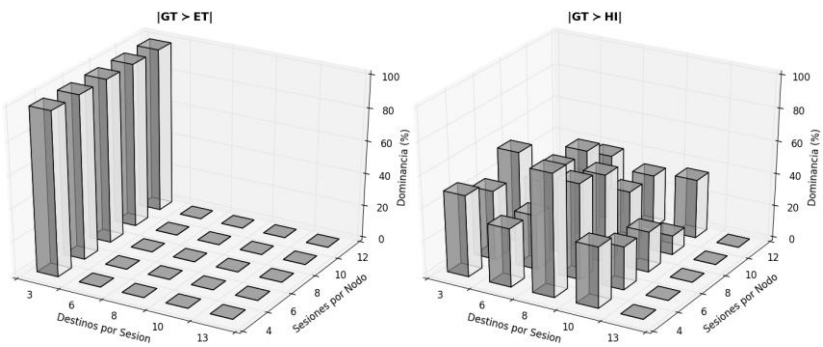


Fig. 2: Percentage dominance of RP (Partial Reconfiguration) solutions over RT (Total Reconfiguration) and HI (Hybrid Reconfiguration) solutions.

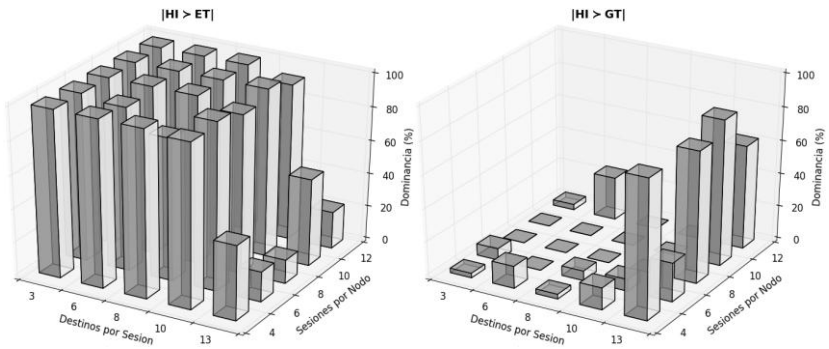


Fig. 3: Percentage dominance of HI (Hybrid Reconfiguration) solutions over RT (Total Reconfiguration) and RP (Partial Reconfiguration) solutions.

5. Conclusions and Future Work

This paper has developed a MOACO algorithm for WDM networks design for static multicast traffic subject to recovery against single node failure considering partial, total and hybrid reconfiguration. Note that reconfiguration has been considered as a critical problem given that is associated with data loss by disrupting. The proposed algorithm designs the network, primary and backup routes. Studies indicate that hybrid reconfiguration is promising to outperform the total and partial reconfiguration.

As future work, it can be considered the problem of wavelength converters location and further testing in other topologies.

References

1. C. S. R. Murthy, WDM optical networks: Concepts, design, and algorithms, P. H. PTR, Ed., New Jersey, 2002.
2. B. Mukherjee, “WDM optical communication network: Progress and challenges,” IEEE J. Select. Areas Commun., vol. 18, pp. 1810–1824, Sept. 1989.
3. Y. Xin and G. N. Rouskas, “Multicast routing under optical layer constraints,” in INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 4. IEEE, 2004, pp. 2731–2742.
4. A. Vazquez, D. P. Pinto-Roa, and E. Davalos, “Optical multicast with protection against node failure an approach based on MOACO,” in Computing Conference (CLEI), 2013 XXXIX Latin American, Oct 2013, pp. 1–9.
5. K. Luekijna and C. Saivichit, “Multicast traffic reconfiguration in WDM network for single node failure design,” in Advanced Communication Technology, The 9th International Conference on, vol. 3. IEEE, 2007, pp. 1833–1838.

6. M. Schaerer and B. Barán, “A multiobjective ant colony system for vehicle routing problems with time windows,” in Proc. Twenty first IASTED International Conference on Applied Informatics, Insbruck, Austria, 2003, pp. 97–102.
7. C. A. Coello Coello, B. Lamont, Gary, and D. A. van Veldhouzen, Evolutionary algorithms for solving multi-objective problems. Springer. 2007.
8. D. P. Pinto-Roa, “Multi-objective Design of WDM Networks. An approach based on Evolutionary Algorithm,” Ph.D. dissertation, Polytechnic School UNA, San Lorenzo - Paraguay, September 2011.
9. B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, “Some principles for designing a wide-area WDM optical network,” Networking, IEEE/ACM Transactions on, vol. 4, no. 5, pp. 684–696, 1996.
10. A. Arteta, B. Barán, and D. P. Pinto-Roa, “Routing and wavelength assignment over WDM optical networks: a comparison between MOACOS and classical approaches,” in Proceedings of the 4th international IFIP/ACM Latin American conference on Networking. ACM, 2007, pp. 53–63.

Evaluation of Scheduling Algorithms on an Asymmetric Multicore Prototype System[1]

ADRIÁN POUSA[1], JUAN CARLOS SAEZ[2], ARMANDO DE GIUSTI[1], MANUEL PRIETO MATIAS[2]

[1] III-LIDI, School of Computer Science, National University of La Plata, Argentina
{apousa,degiusti}@lidi.info.unlp.edu.ar

[2] Complutense University of Madrid, Spain
{jcsaezal,mpmatias}@ucm.es

Abstract. Single-ISA (instruction set architecture) asymmetric multicore processors (AMPs) were shown to deliver higher performance per watt and area than symmetric CMPs (Chip Multi-Processors) for applications with diverse architectural requirements. AMPs combine complex and high performance *fast cores* with low power *slow core*. Previous work has proposed various scheduling algorithms for AMPs, which have been mostly evaluated by means of emulated asymmetric platforms or simulators. In this article, we carry out the experimental analysis of some state-of-the-art scheduling schemes for AMPs on the Intel QuickIA, an asymmetric multicore prototype system. For the analysis, we implemented these schemes in a real operating system, and evaluated them using multi-program workloads. A key aspect of the implementation of these algorithms is the methodology proposed to approximate at run time the relative benefit that an application obtains by running on a *fast core* with respect to a *slow core.*

Keywords: AMP, Asymmetric Multicore Processors, process scheduling, operating system.

1. Introduction

Asymmetric multicore processors (AMPs) integrate in a single chip several types of cores with different characteristics (frequency, microarchitecture or power consumption), but with the same instruction set architecture. Previous research has shown that AMPs offer better performance per watt and area unit than symmetric multicores [9,1]. It has also been shown that using two

[1] This paper is part of the existing collaboration between UNLP and UCM, Project B/024918/09 - Training in Advanced Computing, approved by the Spanish Agency for International Development Cooperation (AECID).

types of cores only ("fast" and "slow") in the chip results in a very efficient utilization of processor area and considerably simplifies the design [9]. The recent appearance of the ARM big.LITTLE [2] processor and the Intel QuickIA prototype [4] demonstrates that main manufacturers are interested in this type of asymmetric designs.

Despite the advantages of AMPs, these pose significant challenges to the operating system. One of the most significant challenges is how to efficiently distribute *fast core cycles* among the applications that share the system. Most proposals in this area seek to maximize global performance or throughput [9,17,12,8]. To make this possible, the scheduler must map to the *fast cores* those applications that benefit the most when run on fast cores versus *slow cores* [17,8,12]. In this article, we refer to this benefit as the application *speedup* on the AMP.

Other relevant objectives, such as providing *fairness* in the distribution of fast core cycles or supporting priorities, have received less attention. Notably, a few scheduling proposals aim to improve *fairness* in AMP environments by distributing *fast cores* equally among the various applications [3,10] or by ensuring that all applications experience the same performance *slowdown* when sharing the system relative to running alone on the platform.

Most scheduling schemes proposed for AMPs have been evaluated on emulated asymmetric platforms (e.g., cores with different frequencies) or using simulators. In this paper, we analyze *fairness* and *throughput* delivered by some of the state-of-the art scheduling algorithms for AMPs on the Intel QuickIA [4]. One of the most significant challenges when implementing these algorithms on a real operating system, such as the one carried out for this work is designing a mechanism to determine *speedup* at run time. In this article, we propose a systematic methodology to obtain *speedup* estimation models.

The rest of the paper is organized as follows: Section 2 presents the metrics used for the experimental analysis. Section 3 describes the schedulers that were analyzed. Section 4 outlines the process for generating the speedup estimation model. Section 5 presents and analyzes the experimental results obtained on the real asymmetric system. Finally, Section 6 presents related work, and Section 7 concludes.

2. Metrics

To quantify throughput on AMPs, we opted to use Aggregate speedup, which is defined as follows:

$$Aggregate\ Speedup = \sum_{i=1}^{n} \frac{CT_{Slow,i}}{CT_{Sched,i}} - 1 \quad \textbf{(1)}$$

where n is the number of applications in the workload, $CT_{Slow,i}$ is the completion time of application i when it runs alone in the AMP and uses

small cores only, and $CT_{Sched,i}$ is the completion time of application *i* under a given scheduler.
Furthermore, we define a scheduler to be fair if equal-priority applications suffer the same slowdown due to sharing the system with respect to the situation in which the whole system is available to each application [11,5][2]. Therefore, we opted to use this definition and employ the unfairness metric [11,5], which is defined as follows:

$$Unfairness = \frac{MAX(Slowdown_1, \dots, Slowdown_n)}{MIN(Slowdown_1, \dots, Slowdown_n)} \quad (2)$$

where $Slowdown_i = CT_{Sched,i}/CT_{Fast,i}$, and $CT_{Fast,i}$ is the completion time of application *i* when running alone in the AMP (with all the fast cores available).

3. Scheduling Algorithms for AMPs

In this paper, we carry out an experimental analysis of some of the state-of-the-art scheduling algorithms for AMPs, such as RR (Round Robin), Prop-SP (Proportional-SPeedup), HSP (High-SPeedup) and A-DWRR.
For this analysis, we have implemented these algorithms in the Linux kernel.

RR [3,12] distributes *fast cores* equally among all workload applications without knowing their *speedups*. Previous research has shown that this algorithm has a more deterministic behavior and its performance is better on AMPs than that of default schedulers in current operating systems [13]. For this reason, RR has been widely used as baseline for comparison with other algorithms.
Prop-SP [16] is an algorithm designed to improve the *throughput-fairness* trade-off. To do so, Prop-SP assigns to each application a fraction of *fast core* time that is proportional to the application's dynamic weight, defined as the product of its static weight and its net speedup (*speedup* minus one). The static weight is derived from the priority of the application, which is set by the user. The speedup is approximated at run time by the scheduler by means of estimation models (with no interaction from the user). We will refer to this implementation of the algorithm as Prop-SP (dynamic). The effectiveness of this algorithm strongly depends on the accuracy of speedup predictions, so for the sake of completeness, we compared it with a static variant of the algorithm, – Prop-SP (static), – which uses *speedup* values measured offline (i.e., ratio of completion times on both core types).

[2] Due to the existence of shared resources in a multicore system (cache levels, cores of various types, and so forth), this concept of fairness is more appropriate for multicores than the classic concepts of fairness that are based solely on CPU time granted to applications [11,5].

HSP (High-SPeedup) [12,8] optimizes *throughput* by devoting *fast cores* to run applications in the workload that experience the greatest fast-to-slow *speedups*. Clearly, this algorithm is inherently unfair, since it makes no effort to distribute *fast cores* among applications. As with Prop-SP, two variants of the algorithm are analyzed: HSP (static), where the scheduler is fed with average speedups for each application obtained offline, and HSP (dynamic), where the *speedup* is estimated at run time.

A-DWRR [10] aims to deliver *fairness* on an asymmetric system. The algorithm uses a notion of CPU time, extended to AMPs, known as *scaled* CPU time. Using this time notion, the amounts of CPU cycles used by an application on a fast core have a greater weight than those used on a *slow core*. To ensure fairness, A-DWRR attempts to even out scaled CPU time among threads by factoring in their priorities as well. Note that A-DWRR assumes a constant fast-to-slow performance ratio when computing the scaled CPU time for each thread and not the actual *speedup* of each application.

It is worth highlighting that all of our implementations of these scheduling algorithms exhibit some common aspects. Firstly, they all keep *fast cores* busy to maximize their utilization. Secondly, when mapping threads to cores, all algorithms trigger thread swaps between different cores types. Thirdly, all algorithms have been implemented as separate scheduling classes in the Linux kernel.

4. Speedup estimation

In this section, we detail the process used for generating an estimation model to determine the *speedup* of a sequential application on the Intel QuickIA prototype [4]. This prototype is a dual socket system that integrates two different multicore processors: a dual core Intel Atom N330 - 1.6Ghz- and a quad core Intel Xeon E5450 -1.2Ghz- . To minimize the effect of shared-resource contention in the experimental analysis, we have opted to disable two cores in the Intel Xeon processor, so that no core shares last level cache with another core. Therefore, we have an asymmetric multicore configuration with two high-performance *fast cores* (Xeon) and two low-power *slow cores* (Atom).

Our goal is to obtain a performance model making it possible for the scheduler to approximate the *speedup* of a sequential application using hardware counters on the core where the application is currently running. For a sequential application, *speedup* is also known as *Speedup Factor (SF)*, and it is defined as: $\frac{IPS_{Fast}}{IPS_{Slow}}$ where IPS_{Fast} and IPS_{Slow} are the ratios of instructions per second achieved by the application in a *fast core* and a *slow core*, respectively.

Several researchers have tried to estimate the SF by looking for correlations between the SF and different performance metrics that can be captured using the processor's hardware counters. In particular, it has been shown that, when the cores in the system differ only in their frequency, the SF shows a negative

correlation to the last level cache miss rate [12,8]. Similarly, Koufaty et al [8] detected that, in systems whose cores have different instruction retirement rates, there are two factors that have a negative correlation to the SF [8]. The first factor is the application's degree of memory intensity, which can be approximated by using the last-level-cache miss rate or the last-level-cache access rate. The second factor is related to stalls in the processor's pipeline front-end due to branch misprediction rate. Both memory-

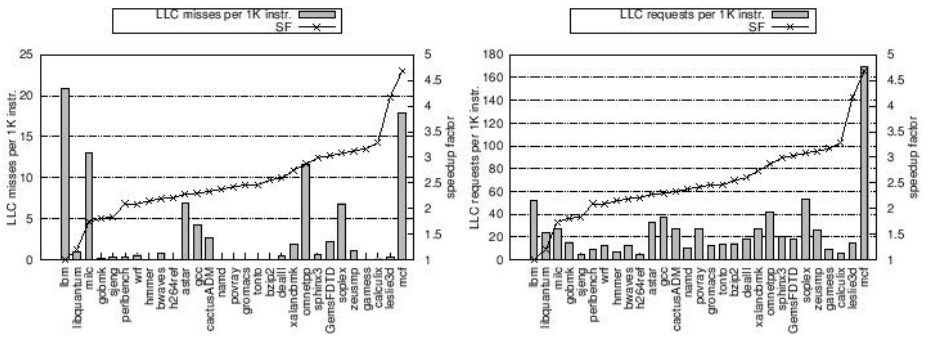


Figure 1: Relation between SF and some performance metrics on the Intel Xeon

intensive applications (high last level cache miss rate) as well as those with frequent branch miss prediction usually experience low speedup factors.

To illustrate the complexity associated with SF estimation on the Intel QuickIA, Figure 1 shows the relationship between the SF and some of the aforementioned performance metrics. Unlike what happens on emulated asymmetric systems, no clear correlations between the performance metrics and the SF are observed for this platform.

In search of a systematic methodology to design estimation models in complex scenarios, such as the one studied in this article, we explored various data mining techniques by means of the WEKA tool [6]. WEKA provides multiple methods to infer relationships between a set of observations (or input attributes) and a target variable. For the SF estimation problem, observations correspond to various performance parameters, and the target variable is the SF. Among the methods that were considered, we detected that additive regression allows obtaining a relatively accurate estimation model for this platform.

The methodology proposed for generating the estimation model in the asymmetric platform is a variant of that presented in [14] and can be summarized with the following steps:

(1) Select a set of representative sequential applications, AP, and a set of performance metrics, M.

(2) Run the applications in AP on both types of cores to obtain their SF and performance metrics in M using the processor's hardware

counters. To carry out this task, we used an in-house performance-monitoring tool . [3]

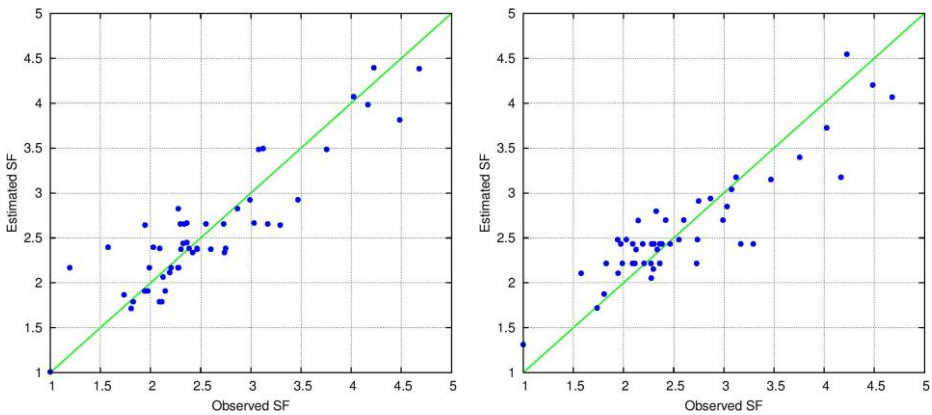


Figure 2: Observed and predicted SFs for a subset of SPEC CPU2006 and CPU2000 benchmarks. Each point represents the SF of a benchmark. The closer a point is to the diagonal, the better the SF estimation. The correlation coefficient for the estimation on both core types is 0.9.

(3) After collecting the necessary performance monitoring data over time for the various applications, compute the average value of each performance metric in M for the entire execution. An application's SF can be approximated with the ratio of instructions per cycle achieved by the application on both core types or, alternatively, with the ratio of completion times observed on both core types.

(4) The results obtained in the previous step are provided to WEKA, which generates two SF estimation models based on additive regression – one to estimate the SF from the fast core and another one to do the same from the slow core. Notably, when generating the models, WEKA automatically discards those metrics in M that are less relevant for the model. This makes it possible to significantly reduce the model complexity.

Figure 2 shows the comparison between observed and estimated SF in both types of cores for a subset of SPEC CPU2006 and SPEC CPU2000 applications on the Intel QuickIA. The generated models require the scheduler to monitor the following metrics at runtime: instructions per cycle,

[3] The implementations of the schedulers analyzed in this paper make intensive use of the kernel API exposed by the tool. Thanks to this API, any scheduler implemented in the Linux kernel can easily collect per-thread performance monitoring data via hardware counters.

last-level-cache misses and accesses per 1K instructions, mispredicted branches per 1K instructions and ITLB/DTLB misses per 1M instructions.
The main difference between the methodology proposed in [15] at that described above is the fact that the former imposes a restriction on the maximum amount of metrics that can be monitored online to approximate the model whereas the latter does not. Specifically, in the previously proposed methodology, generated estimation models could only depend upon metrics that can be monitored simultaneously using the available hardware counters in the platform. Notably, each core in the Intel QuickIA platform only features two general-purpose (configurable) performance counters and three fixed-function hardware counters. This small amount of hardware counters dramatically reduces the number of performance metrics that can be monitored at a time. Specifically, the metric set specified above cannot be monitored simultaneously on any core of the Intel QuickIA. To overcome this issue, we turned to event multiplexing in the scheduler implementation. In particular, we broke down the entire metric set into different subsets with compatible metrics that can be monitored simultaneously. At run time, the scheduler samples different subset of metrics in a round-robin fashion. This makes it possible to collect the values for the various metrics the model depends upon with various sampling rounds.

5. Experiments

In this section, we analyze *throughput* and *fairness* for the scheduling algorithms described in Section 3 running on the Intel QuickIA prototype.
To carry out a thorough analysis of the various scheduling policies, we built several multi-program workloads, consisting of sequential applications from the SPEC CPU2006 suite. To select representative workloads, we sorted the various applications taking their SF into account – H (high), M (medium) or L (low). The first column in Table 1 shows the composition of the selected workloads. For instance, the 3H-1M workloads consists of three sequential applications with a high SF (*calculix, GemsFDTD* and *bzip2*) and another application with a medium SF (*h264ref*).

Categories	***Benchmarks***
4H	calculix, gamess, GemsFDTD, bzip2
3H-1M	calculix, GemsFDTD, bzip2, h264ref
3H-1L_A	gamess, GemsFDTD, bzip2, sjeng
3H-1L_B	calculix, gamess, sphinx3, sjeng
2H-2M	gamess, soplex, povray, h264ref
2H-2L_A	mcf, calculix, sjeng, gobmk
2H-2L_B	gamess, sphinx3, gobmk, libquantum
1H-1M-2L_A	mcf, h264ref, sjeng, gobmk
1H-1M-2L_B	calculix, gromacs, sjeng, libquantum
2M-2L_A	namd, h264ref, gobmk, libquantum
2M-2L_B	gromacs, povray, sjeng, libquantum

Table 1: Multi-application workloads consisting of sequential applications.

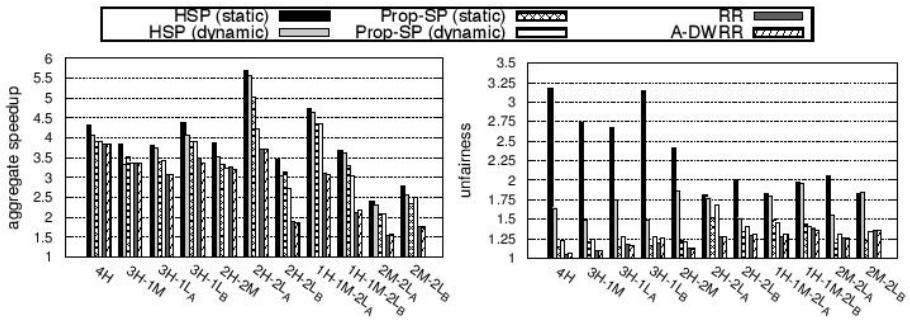


Figure 3: Aggregate Speedup and unfairness for the various scheduling algorithms.

Figure 3 shows the results for the considered workloads. The results show that the HSP scheduler obtains the highest aggregate speedups, but at the cost of degrading fairness significantly for all workloads. RR, by contrast, offers the best fairness values in many cases. However, an even distribution of *fast core cycles among applications* does not always ensure the same performance degradation (slowdown) for all applications [15]. The A-DWRR scheduler behaves in a very similar way, since in the considered scenario (one thread per core), balancing per-thread scaled CPU times is achieved by fair sharing *fast cores* among applications (as RR does). Overall, our results also show that the RR and A-DWRR algorithms are not capable of obtaining aggregate speedups as high as the other algorithms counterparts. This throughput degradation stems from the fact that the diversity of speedups among applications is not taken into account [3, 12].

The results also reveal that Prop-SP achieves a good fairness-throughput (aggregate speedup) trade-off. This algorithm achieves a higher aggregate speedup than RR and A-DWRR for all workloads. In addition, for some workloads such as 3H-1L_A, 3H-1M or 2M-2L_A, Prop-SP offers a better fairness-throughput trade-off than those algorithms (higher aggregate speedup for similar fairness figures). These results also show that Prop-SP has a much fairer behavior than HSP and obtains throughput values that are closer to HSP than the other algorithms.

Lastly, we focus on the dynamic variants of HSP and Prop-SP. In most cases, the dynamic version of HSP achieves better fairness (lower unfairness values) than its static counterpart. This is because HSP (dynamic) makes sure that the threads with the highest SF phases are always mapped to a *fast core*. Since an application may go through different SF phases over time, HSP (dynamic) migrates applications between cores every so often. This indirectly translates into better fairness in many cases, since multiple applications have the opportunity of running on a *fast core* for some time. On the other hand, we have observed that in the dynamic version of Prop-SP, fairness and throughput values can get worse in the presence of SF mispredictions.

6. Related work

Most existing schedulers for AMPs strive to maximize the system throughput. Those algorithms designed for workloads that include sequential applications only [9, 3,8,17,13,18] try to optimize system performance by running on the *fast cores* those programs with a higher SF. To improve the overall performance in workloads that include multithreaded programs, some schedulers use *fast cores* in an AMP as accelerators for sequential phases in parallel applications [1,12].

A large body of scheduling proposals for AMPs has been evaluated using simulators [3,18,7]. Other authors, instead, have resorted to implementing their proposals in a real operating system but then evaluating them on emulated asymmetric platforms [14,8,10]. In this article, we implemented scheduling algorithms in a real OS and evaluated them using actual asymmetric hardware. Previous research in the area has shown that implementations on real systems allow detecting significant problems in algorithms that have been previously evaluated using simulators. This is the case, for example, of the conclusions drawn by Shelepov et al [17,13] when implementing the IPC-driven algorithm [3]. The authors showed that this type of algorithms, which require sampling the IPC in both types of cores [9,3], is subject to inaccuracies when calculating the SF and they may introduce significant overheads at run time compared to that of scheduling strategies based on SF estimation models, such as our implementations of HSP [8,12] or Prop-SP [16].

7. Conclusions

In this article, we have carried out an experimental analysis of some of the state-of-the-art scheduling algorithms for AMPs: RR, HSP, Prop-SP and A-DWRR. To this end, we implemented these algorithms in the Linux kernel and evaluated their throughput and fairness using various multi-program workloads running on the Intel QuickIA prototype. A methodology for estimating the speedup of a sequential application on AMPs (i.e., the relative benefit obtained by the application when running on a *fast core* versus a *slow core*) is also proposed.

The results obtained indicate that the HSP scheduler, which uses *fast cores* to run applications with greater speedup, obtains the best throughput results but that comes at the expense of serious fairness degradation. On the other hand, RR and A-DWRR obtain good fairness results but significantly degrade global throughput, since they do not factor in applications' fast-to-slow speedups. Finally, Prop-SP is capable of making efficient use of the AMP while offering a better fairness-throughput trade-off than the rest of the algorithms for a wide range of multi-program workloads.

7. Acknowledgments

This work has been supported by the Spanish government through the research contract TIN2012-32180 and the European Network of Excellence HiPEAC. We thank David Koufaty and Intel Labs for allowing us to experiment with the asymmetric prototype system.

References

1. Annavaram, M., et al.: Mitigating Amdahl's Law through EPI Throttling. In: Proc. of ISCA'05. pp. 298{309 (2005).
2. ARM: Benefits of the big.LITTLE Architecture (2012).
3. Becchi, M., Crowley, P.: Dynamic Thread Assignment on Heterogeneous Multiprocessor Architectures. In: Proc. of CF '06. pp. 29{40 (2006).
4. Chitlur, N., et al.: QuickIA: Exploring heterogeneous architectures on real prototypes. In: in Proc. of HPCA '12. pp. 1 {8 (Feb 2012).
5. Ebrahimi, E., et al.: Fairness via source throttling: a con gurable and high performance fairness substrate for multi-core memory systems. ASPLOS '10 (2010).
6. Hall, M., et al.: The WEKA data mining software: an update. SIGKDD Explor. Newsl. 11, 10{18 (2009).
7. Joao, J.A., et al.: Utility-based acceleration of multithreaded applications on asymmetric CMPs. In: Proc. of ISCA '13. pp. 154{165 (2013).
8. Koufaty, D., Reddy, D., Hahn, S.: Bias Scheduling in Heterogeneous Multi-core Architectures. In: Proc. of Eurosys '10 (2010).
9. Kumar, R., et al.: Single-ISA Heterogeneous Multi-Core Architectures for Multithreaded Workload Performance. In: Proc. of ISCA '04 (2004).
10. Li, T., et al.: Operating system support for overlapping-ISA heterogeneous multicore architectures. In: HPCA'10. pp. 1{12 (2010).
11. Mutlu, O., Moscibroda, T.: Stall-time fair memory access scheduling for chip multiprocessors. In: Proc. of MICRO '07 (2007).
12. Saez, J.C., et al.: A Comprehensive Scheduler for Asymmetric Multicore Systems. In: Proc. of ACM Eurosys '10 (2010).
13. Saez, J.C., et al.: Leveraging workload diversity through OS scheduling to maximize performance on single-ISA heterogeneous multicore systems. J. Parallel Distrib. Comput. 71, 114{131 (January 2011).
14. Saez, J.C., et al.: Leveraging core specialization via OS scheduling to improve performance on asymmetric multicore systems. ACM TOCS 30(2) (Apr 2012).
15. Saez, J.C., et al.: Delivering fairness and priority enforcement on asymmetric multicore systems via OS scheduling. In: Proceedings of the ACM SIGMETRICS'13. pp. 343{344 (2013).
16. Saez, J.C., et al.: Exploring the throughput-fairness trade-off on asymmetric multicore systems. In: To appear in 2nd Workshop on Runtime and Operating Systems for the Many-core Era (ROME), held in conjunction with Euro-Par 2014 (2014).
17. Shelepov, D., et al.: HASS: a Scheduler for Heterogeneous Multicore Systems. ACM SIGOPS OSR 43(2) (2009).
18. Van Craeynest, K., et al.: Scheduling heterogeneous multi-cores through performance impact estimation (PIE). pp. 213{224. ISCA '12 (2012).

IPv6: Comparison of mobile environments in academic networks

CARLOS TAFFERNABERRY[1], SEBASTIÁN TOBAR[1], GUSTAVO MERCADO[1], JOEL NOGUERA[1], CRISTIAN PEREZ MONTE[1], RAÚL MORALEJO[1], SANTIAGO PEREZ[1]

[1] Grupo GridTICs - Departamento de Electrónica - Universidad Tecnológica Nacional - Facultad Regional Mendoza - Rodríguez 273, Ciudad Mendoza
CP (M5502AJE) República Argentina
{carlos.taffernaberry, sebastian.tobar, gustavo.mercado, joel.noguera, cristian.perez, raul.moralejo, santiago.perez}@gridtics.frm.utn.edu.ar

Abstract. The use of academic networks is generating important changes in research and education fields, providing new tools that bring us closer to other scientific and educational worldwide communities. Thus, investigations are carried out between work teams that are geographically distant. This allows better interaction and support between researchers, advanced scientific research collaboration, etc. In this paper, a comparison between two different IP mobility alternatives available in version 6 is presented. Standards evaluated are Mobility IPv6 (MIPv6) and Proxy Mobility IPv6 (PMIPv6). A comparison of performance, advantages, disadvantages, configuration, ease of handling and installation is made in the first instance. Afterwards the behavior of MIPv6 and PMIPv6 handover is evaluated for traditional and real time applications. Finally, comparisons and a conclusion are made.

Keywords: Mobile IP, IPv6, MIPv6, PMIPv6

1. Introduction

By the end of the twentieth century internet use became massive, which introduced changes in all areas of our lives. However, soon after its incorporation into everyday life, an emptiness and dissatisfaction in the scientific, academic and research community was felt. Advances in technology infrastructure were developed on academic networks to re install the hopes in these communities. That allowed the exclusive use of tools and applications to improve and increase their activities.
Nowadays these networks are known as advanced academic and research networks. Its main feature is to allow geographically distant research and scholars communities working together through collaborative mechanisms,

sharing information and resources over a series of high speed interconnected networks.
With the emergence of mobile devices using different wireless technologies, the model of Internet connectivity changed. Currently when a user travels between different networks (roaming), each of the new networks visited provides a different IP address, so the user cannot keep the session open for an application during his displacement. The goal of Mobile IP [1] is assign to the mobile user an unique address, regardless of the network where it is, allowing him to keep, for example, the session for applications.
After 30 years, the version 4 of the Internet Protocol (IP) can no longer provide scalability by the gradual depletion of available IP addresses, due to the current growth rate of nodes connected to the network [2].
The lifetime of IPv4 has been extended for a few years thanks to techniques such as address reuse with translation (NAT) [3] and classless routing (CIDR) [3]; IPv6 in this period grew and finally established as the successor to IPv4.
Some features of IPv6 [4] are: expanded addressing capability, quality of service (QoS), autoconfiguration (Neighbor Discovery), end to end connectivity, simplified header format and mobility support.
The conjunction between academic networks, mobility and IPv6 is a powerful tool to achieve the objective of start up collaborative research.
There are worldwide educational networks using IPv6, such as Internet 2 [5], GÉANT2 [6], Clara [7]. In our country there is a network called Innova Red [8], which provides national institutions access to global advanced networks through the Clara network.
Within the UTN the RUT2 [9] (Network Technology University) interconnect all the the Regional Faculties.
The ACyTNet initiative (Academic Network of Scientific and Technological Mendoza) is a network that joins, at this moment, CONICET Mendoza, INA National Institute for Water Resources and UTN GridTICS FRM pretending to be the Advanced Academic Network of metropolitan institutions in Mendoza.

1.1 IP Mobility

Network Mobility is a concept by whom a node is able to move from one network to another without losing it's current connection and this change should be transparent to the user.

The main entities involved in a mobile architecture [1] are:

- Mobile Node (MN): Device that moves frequently between different networks.
- Home Agent (HA): This device maintains information about the MN current location. It's a router, generally, located in the home network of the MN.
- Foreign Agent (FA): Device located in the foreign network that stores information about MNs. It coordinates, using messages, with the HA to provide mobility.

- Correspondent Node (CN): Any device, mobile or not, that communicates with the MN.

Additionally, IPv4 and IPv6 Mobility implementations have the following differences:

- IPv6 mobility does not need FA nodes. Self-address configuration and neighbor discovery are used, both unique features of IPv6 protocol.
- Mobility packets in IPv4 that travels from the HA to the MN must be encapsulated. This feature is not required in IPv6.
- IPv6 Mobility avoids triangular routing.

The rest of the paper is organized as follow: two approaches to mobility, with a brief explanation of them, are introduced in the section 2. The test environment used to conduct the experiment is described in the section 3. The way the experiment was conducted and results is found in Section 4. Finally, in the last Section are the conclusions reached.

2. Covered Approaches

There are two approaches related to mobile IPv6 networks management.

2.1 Traditional

The IPv6 mobility is managed by the participating nodes, by exchanging mobility messages between the MN and the HA. Therefore the kernel of both nodes must be prepared for it. The procedure is explained as follows:
A MN can have two addresses, a local HoA (Home-of-Address), and, in case of being in a foreign network, a dynamic CoA (Care-of-Address). If the MN is on its local network, packets will continue using conventional routing rules with its HoA. If it is in a foreign network, and a CN wish to communicate with it, it will initially use the HoA. These packets are intercepted by the HA, which manages a table with information linking HoAs with CoAs addresses, as well as tunnels from the local network of the HoAs to the foreign networks of the CoAs. The HA redirects these packets destined to the MN through a tunnel, carrying a new IP header with the CoA address, that permits encapsulate the original header with the HoA address. At the end of the tunnel, packets are decapsulated by the MN removing the IP header added before.

2.2 Mobility managed by the Network

This approach, known as NETLMM (Network-based Localized Mobility Management) [10], makes possible to implement mobility in IPv6 nodes without involve the MN and CN nodes in the exchanging mobility messages. It is an advantage over traditional mobility because does not require changes

in the software behavior of these nodes. This is the case of the PMIPv6 protocol [11]. The main entities in the PMIPv6-NETLMM infrastructure are:

- Local Mobility Anchor (LMA): Is a HA with proxy properties. It's the responsible for the MN to be accessible. Topologically is the origin point (anchor point) for the origin network prefixes (home network prefix(s)) of the MN.
- Mobile Access Gateway (MAG): Usually is a router that manages the mobility in behalf of the MN, located in the local network of the MN. It is responsible of detecting the movements of the MN to and from the local network. It is involved in the MN registration in the LMA.

There may be multiple LMA in a Proxy Mobile domain, each giving service to different groups of MN.
From the perspective of each MN the entire PMIP domain seems like a single link. The network ensures that the MN does not detect any change with respect to the network layer, even if it changes its attachment point to the network.

3. Implementation

After a comparison between different free operating systems, GNU/Linux Distribution Fedora Core 14 was selected, mainly because the kernel in this distribution had the module MIP6 mobility [12] compiled. Also, it was necessary to run a MIP service in user mode to complete the IPv6 mobility support.
There are several alternatives for MIPv6 service implementations. UMIP [13] was selected due to a wide support and frequent improvement to his code.
For PMIPv6 service currently exists three options: OPMIP [14], OAI PMIPv6 [15] and UMIP [16] plus a patch to support PMIPv6. The first implementation is not very mature, has only two years of work. The second, OAI PMIPv6 is implemented over UMIP, and has specific hardware requirements that hinder and limit its implementation. That is why we decided to install UMIP and apply the necessary patches to give Proxy Mobile support.

3.1 MIPv6 deployment

In the ACyTNet network environment, a test bed was set up with five nodes, as shown in Figure 1. According to the detailed in Section 3, the GNU/Linux Distribution Fedora Core 14 Operating system was installed on all nodes.

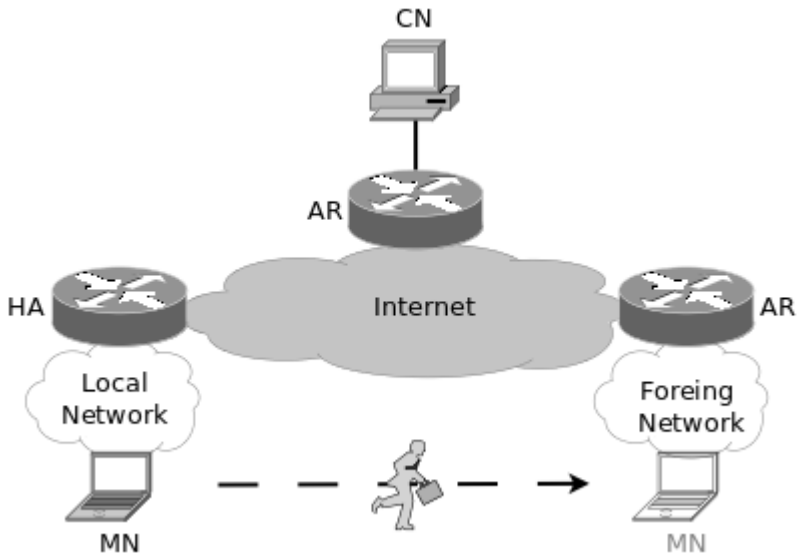


Figure 1. Mipv6 scenario.

The UMIP service was installed in the MN and HA nodes, running in user space. This service was used to manage the tunnels and links of each node. The MN and HA kernels were responsible for handling IPv6 mobility specific extension headers. The UMIP service makes extensive use of XFRM [17] framework, manipulating the IPv6 headers.
Finally, IPSec [18] support was added, modifying the configuration, to protect the data flow. The MN and HA node settings were modified and also security associations were created in both operating systems.

3.2 PMIPv6 deployment

The test bed was reconfigured to deploy a PMIPv6 domain, as shown in Figure 2.
Due to PMIPv6 characteristics, the MN and CN operating systems version were not relevant. The chosen implementation, PMIPv6 UMIP version 0.4, needed a software patch to acquire PMIPv6 functionalities.
The node previously used as HA, assumed the LMA role, while the other routers assumed MAGs function. The MN and CN remained unchanged.

As well as in MIPv6, there was a single executable which meets PMIPv6 different roles; depending on the configuration file associated.

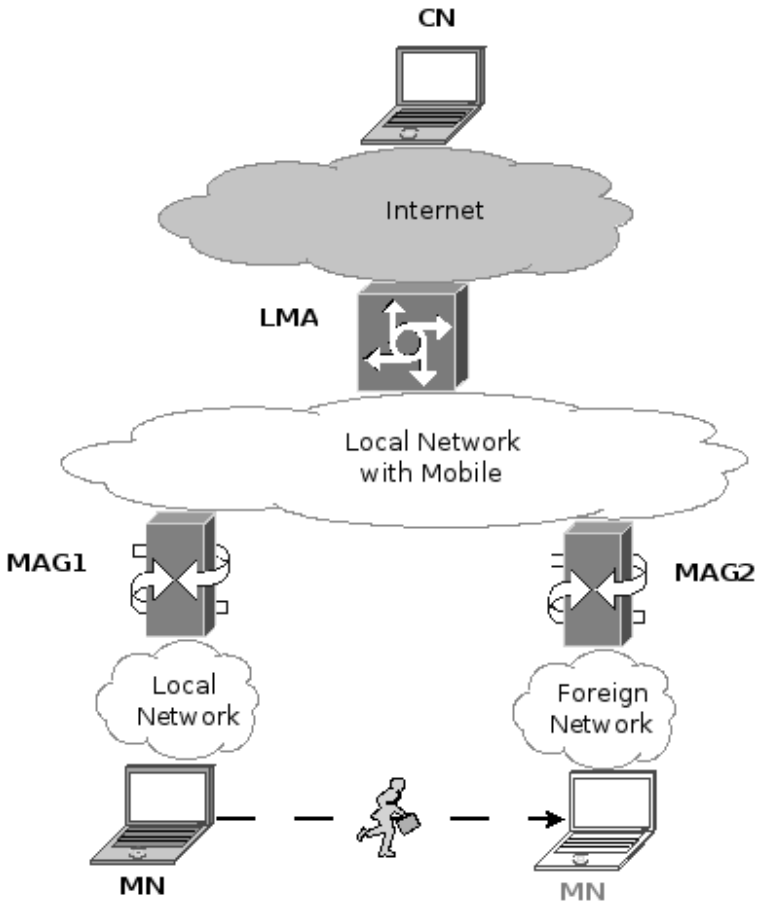


Figure 2. Test environment used by PMIPv6

The LMA configuration for PMIPv6 was the same used for the MIPv6 HA. As said, in PMIPv6 the LMA is the MN topological anchor point.

4. Experimentation y Results

The test consisted of running applications on the MN node. Afterwards, while the applications were still running, the MN was moved from the home network to a foreign network. The MN handover was triggered by this change in the anchor point.
The applications under test were ping, FTP, SSH and real time traffic.
A topology without mobility was used as a reference scenario to compare the measurements.

4.1 Scenario without Mobility

The reference scenario was assembled at the GridTICs laboratory, in the UTN FRM, as part of the ACyTNet. The network architecture and addressing used is observed in Figure 3.

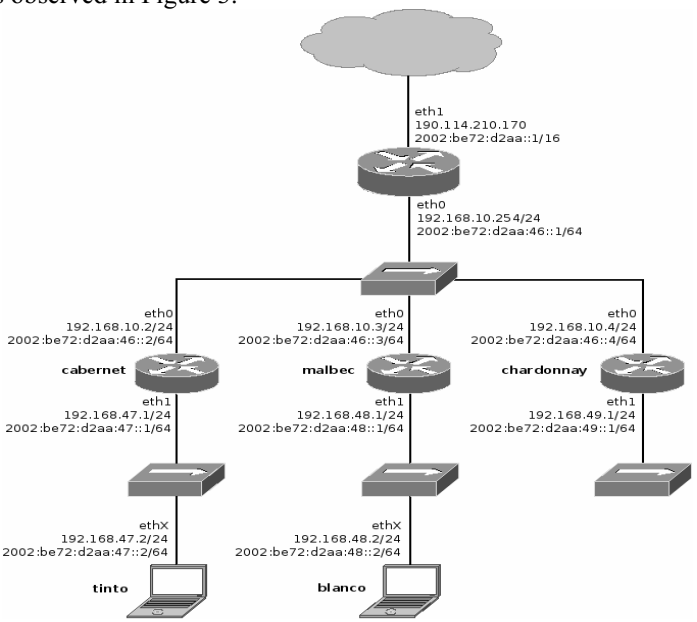


Figure 3. Reference scenario

About MIPv6, as long as the MN was located at the home network, there was not any topological difference between the reference and the MIPv6 scenario. Once the MN was in a foreign network, a tunnel had to be created between the MN and the HA, adding one hop to the topology
The topology remained unmodified either when the MN was attached to the home or foreign network, when PMIPv6 was used. Also, there always existed a tunnel in this scenario.

4.2 Results

Basic functionality:
Both mobile protocols (MIPv6 and PMIPv6) worked as expected. During the handover, the MN conserved his IP address and established sessions.
Channel capacity:
A comparative of the bandwidth available for data transmission between MIPv6, PMIPv6 and reference scenario was carried out. This test, the Iperf [19] tool was used. The time configured to be used in the test was 10

seconds, while the number of concurrent clients was 1, 2 and 10 for full duplex communication. A graphic with these results can be seen in Figure 4.

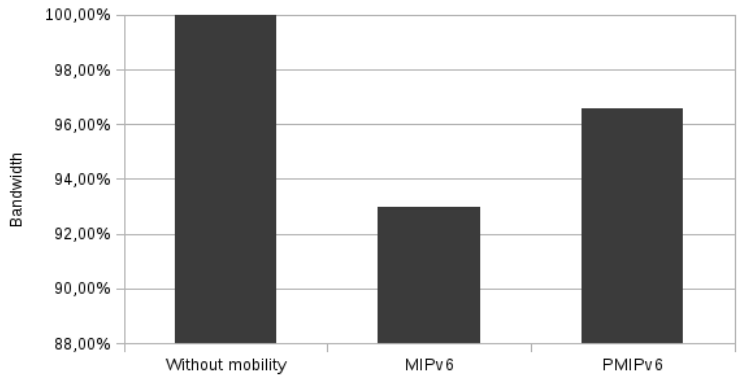


Figure 4. Bandwidth comparison

RTT analysis. IPSec impact:
The ping tool was used to test the average round trip time between the MN and the CN. The packet size used was 56, 512 and 1024 bytes, along with flooding option (-f).

Obviously, the lowest RTT time was obtained using the reference scenario, without mobility. PMIPv6 took a slightly longer RTT, due to the overhead introduced by the tunnels used. With a considerably longer RTT time, caused by a triangular routing, appears MIPv6. Finally, MIPv6 with IPSec had the worst RTT, which was strongly affected by the processing overhead of the security algorithms. This benchmark can be seen in Figure 5.

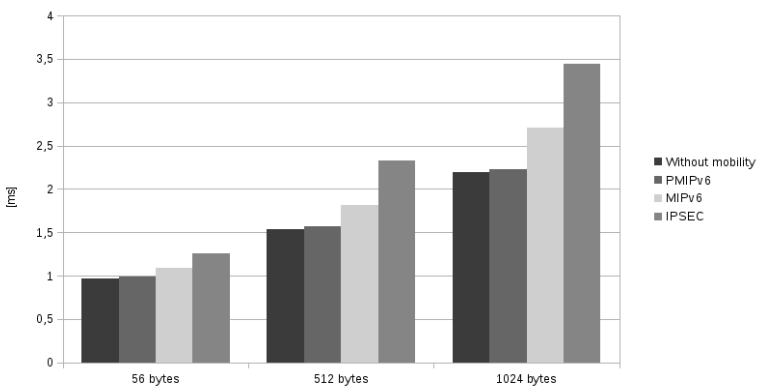


Figure 5. RTT comparison

Handover Time:

Several considerations must be taken into account, regarding handover time measurements. This time depends on different variables not related to the mobility protocol under test [20][21].

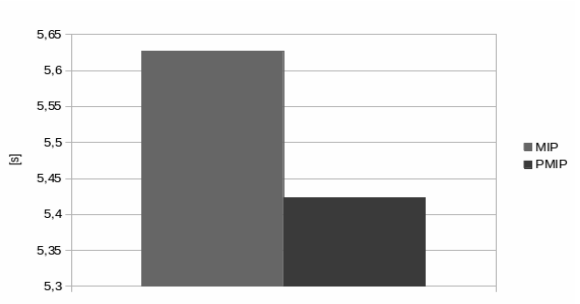


Figure 6. Handover Comparison

The handover time in PMIPv6 and MIPv6 scenarios was evaluated in this comparison. As can be seen in Figure 6, this time is shorter for PMIPv6. This situation is related to the signaling driven by the LMA and MAGs instead of the CN or MN nodes.

5. Conclusions

It has been shown that the implementation of both protocols behaved properly in terms of functionality. Also, PMIPv6 allowed the use as MN of any IPv6 node without mobility extensions. This feature permitted a great diversity of mobile devices acting as MN; especially those that could not modify its operating system, such as smart phones, tablets, notebooks with proprietary software, etc.

The performance of MIPv6 was better than PMIPv6 while the MN was in the HN. It was because the data was always exchanged in “IPv6 in IPv6” tunnels for PMIPv6, causing overhead. If the MN was in a foreign network the performance was slightly higher for PMIPv6 protocol.

The security feature could not be compared because the PMIPv6 implementation did not have support to send encrypted signaling or data.

Handover measurements showed a shorter time for the PMIPv6 protocol when the MN moved from one network to another. In both protocols, the TCP session was not lost.

Finally, it's clear that both protocols were designed for distinct goals. MIPv6 ensures global reachability, sacrificing speed and requiring a modified MN. On the other hand, PMIPv6 offers a higher handover speed, limiting the MN mobility only to networks with such service.

References

1. James Solomon “Mobile IP: The Internet Unplugged” 1st. Edition. ISBN: 978-0138562465 (1997)
2. Informe LACNIC, “Distribuciones/Asignaciones IPv4, espacio disponible y pronósticos “http://www.lacnic.net/web/lacnic/reporte-direcciones-ipv4
3. Kevin Fall - Richard Stevens “TCP/IP Illustrated, Volume 1: The Protocols” 2nd. Edition. ISBN 978-0321336316 (2011)
4. Peter Loshin “IPv6: Theory, Protocol and Practice”, Morgan Kaufmann, Segunda Edición. ISBN 1-55860-810-9 (2004)
5. The Internet2 Net – http://www.internet2.edu
6. The high-bandwidth, academic Internet serving Europe’s research and education community - http://www.geant2.net
7. Cooperación Latinoamericana de Redes Avanzadas - http://www.redclara.net
8. Red Nacional de Investigación y Educacion Argentina - http://www.innova-red.net
9. Red Universitaria Tecnológica (RUT2) - http://www.utn.edu.ar/virtual/rut/index.html
10. RFC 4831 - J. Kempf, Ed. ”Goals for Network-Based Localized Mobility Management (NETLMM)”, August 2008 - http://tools.ietf.org/html/rfc4831
11. RFC 5213 - S. Gundavelli, Ed.” Proxy Mobile IPv6”, August 2008 - http://tools.ietf.org/html/rfc5213
12. Fedora Project: la evolución del código abierto. http://es.redhat.com/resourcelibrary/ articles/the-fedora-project-open-source-evolved
13. “Umip Source Code” - http://git.umip.org/
14. OPMIP - http://helios.av.it.pt/projects/opmip
15. OAI - http://www.openairinterface.org/
16. "How to use our set of PMIPv6 patches" - http://www.umip.org/contrib/umip-pmipv6.html#config
17. “Framework XFRM” - http://git.kernel.org/cgit/linux/kernel/git/shemminger/iproute2.git/
18. RFC 4301 - S. Kent, K. Seo - “Security Architecture for the Internet Protocol”, Dicember 2005 - http://tools.ietf.org/html/rfc4301
19. Iperf - https://iperf.fr/
20. RFC 4861 - T. Narten, E. Nordmark, W. Simpson, H. Soliman, “Neighbor Discovery for IP version 6 (IPv6)” September 2007 - http://tools.ietf.org/html/rfc4861
21. DUNMORE, Martin; PAGTZIS, Theo; EDWARDS, Chris. Mobile IPv6 handovers: performance analysis and evaluation. 6NET Consortium, Deliverable D, 2005, vol. 4.

VI

Innovation in Software Systems Workshop

The Challenge of Being a Product Owner PO´s Responsibilities in Agile Projects

MARCELO ESTAYNO[1] AND JUDITH MELES [2]

[1]Universidad Tecnológica Nacional - Facultad Regional Buenos Aires
Buenos Aires, Argentina. mestayno@gmail.com
[2] Universidad Tecnológica Nacional- Facultad Regional Córdoba y Villa María
Córdoba, Argentina. jmeles@gmail.com

Abstract. In Agile methodologies, Scrum in particular, the Product Owner role (PO) is often assumed by a person with solid knowledge about users, market, competitors and future trends for the domain or the type of system is being developed. The PO´s challenge is to build a product vision and then place it into user stories that will help you convey that vision to the development team that will build the software. This paper presents the problems related to the creation, prioritization, validation and acceptance of the product obtained from the user stories. This requires expertise, generic and specific skills, for which the PO is not always prepared.

Keywords: agile projetcs; resnponsibilities; user stories; skills; roles.

1. Introduction

Agile development methods family evolved from the known iterative and incremental life cycles. They emerged from the belief that a closer contact with the social reality and the state of learning, innovation and change products driven development would yield better results.
The main purpose of these methods is to achieve a balance between disciplined processes that propose an excessive workload and bureaucracy and the lack of process that leads to develop software chaotically.
The alternative offered by Agile Methods proposes a radical change that emphasizes a rethinking of cultural, communicational and social aspects, which are the main causes of failures in software development projects.
The Agile manifesto [1] reflects the agreement of most of the industry leaders that propose agile practices. This manifesto contains values and principles that those who adhere to it agree to respect.
This manifesto is focused on privileging what is of value to the customer, ie, frequent deliveries of working software. It proposes a collaborative environment between client / user and team members and accepting attitude to changes that arise during the development of the software product. In this regard it emphasizes to give more relevance to the people involved in the software development than tools and processes used to build it.

The importance and the impact that the presence and collaboration client to achieve project success instead of the extensive and unprofitable negotiations to defend a contract is also highlighted.
The most challenging proposal of the agile manifesto is evaluating every action, every decision, and every choice with the rod to deliver "business value" who asks us for the software, which at the same time feels part of the project.
Currently there exists a large number of agile methods, which subscribe to the agile manifesto. Some of these methods defined practices for agile development, agile modeling and others for agile project management. In this paper we will refer to SCRUM.
Scrum is a framework that define practices for agile project management. While it may be used for projects of different purposes, emphasis will be placed on software development projects.
Scrum proposes an iterative and incremental life cycle, used to frequently release software versions, bounded in functionality, at the end of each iteration being called "Sprint". Each sprint is a fixed period, between two and four weeks, framed in "timeboxing" feature, meaning that the time is fixed.
While the proposal is that at the end of each sprint a potentially shippable product is obtained, i.e. software capable to be installed and used, many organizations choose not release new product functionality at the end of each sprint. Instead, they prefer to combine the result of several sprints on one release.
Next, the Scrum proposal is described in panning mode. The framework consists of three roles, three deliverables and four meetings or ceremonies.
Defined roles that make up the Scrum Team are:

- Scrum Master: is the servant leader whose most important responsibilities is to ensure compliance with the rules of Scrum, as well, to assist the rest of the team and act as a neutral facilitator, concerned with promoting a collaborative environment and guide the team to continuous improvement and self-reliance.
- Product Owner: the owner of the product, the main voice of the customer, who is responsible for its construction and to maximize business value. This role, which is the main object of this work, will be expanded in later sections.
- Development Team: The development team is responsible for building the product defined by the Product Owner. The team in Scrum is "multi-functional" - has all the skills and abilities necessary to deliver a potentially shippable product at the end of each iteration. It is a "self-organized" (self-managed) team, with a high degree of autonomy and responsibility. In Scrum, teams are self-organizing rather than being led by a project leader.

The Scrum framework proposes the generation of three deliverables:

- Product Backlog: the list of features that are expected to have the software product. These features are prioritized and may be expressed at different levels of granularity. Importantly is a dynamic list that is constantly evolving and refining.
- Sprint Backlog: List of elements taken from the top of the Product Backlog, which contains slice of features that the team agreed to be built during an iteration (Sprint). These characteristics should be prioritized by the Product Owner and estimated by the team.

- Product Increment: Software version potentially shippable in the production environment, obtained as a result of executing a sprint.

It proposes four types of events or ceremonies:

- Sprint Planning Meeting: this is the first ceremony, which aims to get the definition of the product increment to be released in this sprint. In the first part of the meeting, the "what" meeting, the purpose of the sprint and the Sprint Backlog is defined. In the second part of the meeting, the "how" meeting, the team estimates the features and can break them down into tasks that need to be done to develop them.
- Daily Meeting: This meeting joins the team, standing where possible. All should attend, the recommended duration is approximately 15 minutes. The main purpose is give daily visibility to the team and synchronize it. Each member in turn must answer three questions: What did you do since the last meeting? What will you do for the next meeting? What impediments arose?
- Sprint Review Meeting: also known as "Sprint Demo" is the time in which the team presents the product owner the product increment obtained during sprint execution. It is made at the end of the sprint, its purpose is to obtain feedback from the PO about the product. The Product Owner will the functionalities developed and will determine which ones to accept and which not.
- Sprint Retrospective Meeting: This is the last meeting that takes place in a sprint, the purpose is to obtain feedback on the process used, review strengths and weaknesses and define actions to take to improve the quality of the process.

While Scrum proposes 4 ceremonies, there is a fifth ceremony, although not explicitly defined in the framework, which is important and necessary. In this event called Grooming, the Product Owner has a fundamental role, which is the creation, preparation and maintenance of the Product Backlog. This meeting does not have a predefined fixed moment of realization, however, we suggest doing it at the following moments: to define the Product Backlog for the first time, before each Sprint Planning and anytime that the development team considered necessary during a Sprint Execution, to understand and refine the requirements being developed.

Scrum suggests a low level of ceremony in the process, which means that few practices were defined, however not to be confused agile with weak or permissive because the method is strict compliance with their definitions and rules.

Because Scrum is a "management framework", it does not define any guideline on issues related to product engineering, and it gives no details on how to build the software, assuming that the development team knows, it is trained and able to solve this mission.

The differences that can be highlighted between this agile management approach from the traditional project management are strongly supported by the commitment, self-organization, self-management and collaborative effort. These abilities and skills must be developed in people for obtaining expected results.

By focusing our attention on the case of the Product Owner role and set of responsibilities assigned to him, questions arise as to whether to assume this role who really know how to connect the business needs with development of software product? How to translate strategies and return to investment in user

stories? How to prioritize user stories to maximize the business value? These are some of the questions, and this is the challenge.

2. Definition of Product Owner profile

Mayer in his recent book [2] called the Product Owner (PO) as the "what voice" a unique voice, possibly channeling many voices, with the aim of defining "well-formed outcomes" and prioritize the highest possible value in a given context. The voice of the "what" is also responsible for the expenditure and any go/no-go decision.
Product Owner profile meets several skills. The PO as a change agent, acting as a catalyst for change in the organization, enabling the creation of value through projects and products. The PO creates a connection between how business should look in the future and their current status.
The PO is a key facilitator in the organization, to join the customer and the business community with the agile development team.
The PO has a challenge in her work, it must create the confidence necessary for the information flow stream between the worlds that tries to link. Acting as an interface between stakeholders, in many cases he must define agreements between conflicting interests. Also PO should continuously collaborate and provide permanent feedback in both directions.
As part of the set of competencies that must a PO must develop is the communicator one. Communication is the main facilitator of learning vehicle and facilitator for the team to communicate effectively, it is vital that the PO be present existing filters in all individuals when making the communication effective. These filters are personal values, beliefs, interests, expectations and previous experiences. [3] When we communicate we are subjective and our tendency is to move away from messages that conflict with our ideas and beliefs. We tend to hear only what they want to hear, we put more attention to the things that interest us.
In this context and as a summary way, it show the picture of Figure 1, where the PO act as a "bridge" or "link" between the problem space and the solution space, placed in what we call coexistence space. The PO must understand both worlds, business world and development world, and should make a translator between these two worlds. The PO must have the overview of the final product.

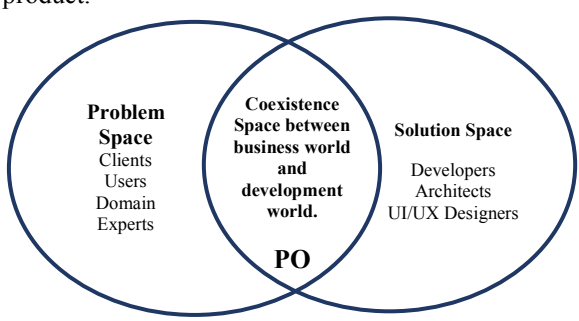


Figure 1. Product Owner (PO) with his interconnection space

3. Description of Product Responsibilities of Product Owner

After an analysis of Schwaber [4] Schwaber [5] Rubin [6] and Nir [3] proposals, this section summarized the set of responsibilities assigned to a Product Owner in an agile project context.

3.1 Define the product

Elicitation allows the product definition, it is a process of analytical and free-flowing communication.
Elicitation is to understand the needs rather than desires capture. It is a collaborative effort that is well suited to development proposed by the Agile Manifesto [1] and occurs at the ceremony above described as Grooming.
This is a discovery work, not just a passive collection, which assumes that the product is already defined and you just have to capture it. On the contrary, the creation occurs through which Pichon Rivière [7] called "dialectic spiral". As showed in Figure 2 it is represented as an inverted cone with a base, an apex and the dialectic spiral. At the base: emerging, manifestos or "explicit" content is located. At the apex: The basic or universal "implicit" situations. The spiral chart the dialectical movement of inquiry and clarification ranging from the explicit to the implicit, in order to explicit it. Make explicit the implicit is what specifically is called "interpretation". This responsibility can be considered a major role attributed to the PO.
These responsibilities in the context of the defined processes relapsed between the tasks to be undertaken by the role known as Functional Analyst or Business Analyst.

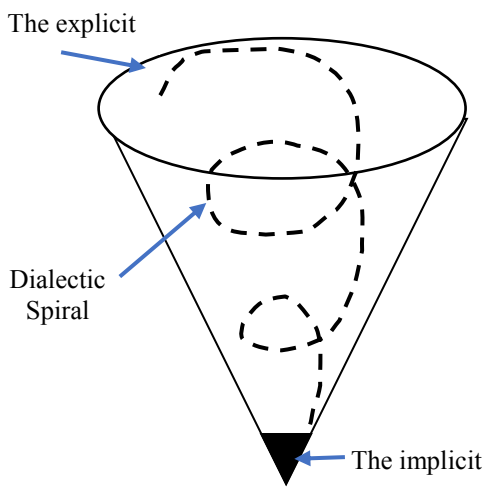


Figure 2. Dialectic Spiral: analysis is makes explicit the implicit.

However, what most of agile methodologies currently propose, as a facilitating tool of this dialectical process? Propose the use of "user stories".

User stories do not require a full specification of the "up-front" solution, instead stimulate a productive and fluid dialogue among stakeholders, seeking the best solution during the software construction.
According to Albert Mehrabian [8], human communication consists of three parts:

1. 7%: The content (words, what is said)
2. 38%: The voice tone
3. 55%: The facial and body expressions

Hence the foundation of one of the agile manifesto principles of favoring the face-to-face contact between stakeholders to achieve a solid, comprehensive and meaningful communication.
The user stories reflect a user's need, a product description, a mechanism to defer a conversation, as well as a planning item.
The 3C´s model, proposed by Ron Jeffries [9] to distinguish "social" user stories from the "documentary" requirements practices, presents these three components:

- **Card:** The most visible manifestation of the user story, but not the most important, it must "capture" the requirement, not "specify it". The user story must be able to be written on a card or post-it standard size. If not enough space will be a sign that we push the boundaries and communicating too much information that should be shared face to face. Suggested syntax to express the User Stories allow us to answer these questions:
  - *Who*, represents the role that need the functionality;
  - *What*, which enables the segmentation of product functionality;
  - *Why*, communicate the usefulness of functionality from a business perspective.

  Therefore syntax is expressed as follows:
  As a <role name>, I want to<activity>so that<business value>[1]

As a Driver I want to search for a destination from his coordinates to know the way to reach the desired destination.

Figure 3. Shows a *user story* example for a GPS software.

- **Conversation:** All user story should have a conversation with the Product Owner. A face to face communication exchange not only information but also experiences, thoughts, opinions and feelings. They are the most important part of the user story. The detail you get from the user story should be a result of this conversation, not a replacement for it.
- **Confirmation**: It serves to determine what the product owner expected, ie for the development team knows what to build. They are tests that communicate and document details and that can be used to determine whether a history is complete.

During the discovery process and product definition, it be necessary to work with different levels of granularity that are closely related to the size and level of detail of what we define. This raises a hierarchy that facilitates the requirements work.
To follow, it introduces the hierarchy proposal by Rubin [6], which is coincident with the presented by Leffingwell [11]. They raise the Epic to indicate stories that need several months to develop. The term refers to the idea of stories like The Lord of the Rings or War and Peace. Epics are markers containing a large collection of stories that will be detailed later.
The next level of story is one that has a size that exceeds the duration of the sprint, and often called Feature. Finally, the smallest form is the user story, often also called implementable story, because it can entirely develop it in one sprint.
I finally called Theme to the collection of related user stories. Topics provide a convenient way to group user stories that have something in common.
Working with different levels of granularity for requirements reduce the amount of early specifications and the need to make decisions when there is not enough information to do so, it also reduces management overhead for large products.
Write requirements as user stories it is a PO work, who should not limited to just read their stories, but instead should ask members of the development team, make sure they understood. If necessary he must use any resources it deems to ensure understanding, such as concept maps, charts, drawings.
Prioritization is an outstanding allocation of responsibilities sets, the PO must understand the stakeholders' needs and know the business to can communicate to the development team what to build and in what order.
As a result of prioritization activity, the product backlog is managed, organizing the most important items on top of the stack and the minor at the bottom. Similarly, the prioritized and next to enter in the sprint backlog elements should be prepared and have a user story level of granularity, while lower priority items can be defined with a greater level of granularity in the form of epics or themes.

3.2 Participate in Planning

PO is a key participant in all levels of planning [4] present in an organization. The more global level, known as portfolio level. At this level you determine what software products work and in what order, but in the long term.
The next level is the product planning, also known as envisioning, and it is the activity where the essence is captured and an approximate plan for the definition of the product is created. It begins with the creation of the product vision, followed by the creation of high-level definition of product backlog.
Then, it continues the release level planning, whose purpose is to communicate with the precision that is reasonably possible when the product version will be delivered, which features will include and how much it will cost.
Finally, short-term planning, which is sprint planning, during which the PO works with the development team to choose a set of product backlog items the team can realistically deliver at the end of the sprint.
In this case the responsibilities previously described, in the context of "traditional" softwaredevelopment, lies in Project Leader, Project Manager,

Product Manager and/or Program Manager roles. Organizations have invested in the training of the people who play these roles.

3.3 Define acceptance criteria

The acceptance criteria are satisfied conditions specific to each user story under which the product must meet the functional1 and nonfunctional2 requirements. The PO must ensure that the acceptance criteria are specified and are built and executed tests that verify compliance with these criteria in the product. In this sense PO acts partly as a business analyst and partly as a test analyst.
Without these criteria, the team would have an incomplete understanding of the product backlog item. For this reason, many teams include a clear acceptance criteria setting-up as part of the definition of ready criteria.

3.4 Manage the Economy

PO is responsible for ensuring that consistently good economic decisions related to the product Backlog, Sprints and Releases were taken.
In a release level PO should make permanent trade off to scope, budget, dates and quality, as it receives information flowing during product development. The compensations made at begin of the release, may not be adequate to the presence of new information that arrives during the release execution.
PO, reviewing the outstanding items in the product backlog, decide that the cost of creating them is greater than the value delivered and subsequently will remove those items.
PO can also decide to change the delivery cadence, which was defined e.g. every 5 sprints, releasing product versions at the end of each sprint.
At sprint level, PO will also manage the economy, ensuring a good return on investment (ROI) is delivered at the end of each sprint.
Regarding the vision of the economy on the product backlog, as PO is who prioritizes, if economic conditions change, priorities in product backlog will change as well.

3.5 Accept the product

Acceptance tests are intended to demonstrate that an application is acceptable to the Product Owner who has been responsible for guiding the system development, Leffingwell [2]. This means that it is he who should run the acceptance tests and confirm they have reached acceptance criteria.
But PO may decide to run the acceptance tests himself, or require expert user assistance to help confirm that the functionality meets the conditions of satisfaction. Nevertheless, PO is the one who must give final view on whether the developed items form the product backlog, meet their expectations.

---

[1]**Functional Requirements**: refer to the expected behavior for the software, commonly referred to as the “what”.
[2]**Nonfunctional Requirements**: refer to the conditions or non-behavioral skills, the "how" system must meet. Also known as quality requirements.

To summarize, most of the work done by PO can be broadly summarized as the creation and increase of value and the reduction and elimination of business costs.

4. Who asumes the Product Owner Role?

In most organizations do not work with Scrum, are unlikely to have a role labeled under the name of "product owner". Therefore, who can assume this role? To address this issue, it is convenient to previously identify the various scenarios that may occur when developing software, these scenarios represent the circumstances in which the teams conforms and what kind of organizations they represent.
Who should be PO depends on what kind of development effort is involved and of each specific organization. Then Rubin approach is taken [2] to identify contexts of development for which it is determined the most appropriate person to assume the role.

A. Internal Development: to develop software belonging to an internal area of a company, the right person to assume the PO role must be who will benefit from the development, i.e. a representative of the business area. For example, if the software product to develop is one for the production management, an authorized person of the area should be the PO. Some organizations that have not yet learned the importance of having a person committed with day to day product development, can ask someone in the system area assume daily responsibilities of a PO. Since it is known that the IT person is not entitled to make important decisions, organizations that do this, have covered the role in a confusing and inefficient way
B. Development of a commercial product "off the shelf": In this scenario a company builds a product and then sell to external customers. Here who assumes the PO role is an employee of the company acting as the voice of the real customer. Often, this person comes from product management or product marketing areas, such as a product manager or project manager.
C. Components Development: some companies are engaged in the construction of components that are part of solutions, then joining them will be used to build complete value solutions for customer. These teams development focus is to a more specific level, that's the reason PO in these cases is a person with a technical background, which is able to define and prioritize these issues in their backlogs. In fact it is a considerably more technical PO profile than the other scenarios.
D. Development of an outsourced product: this is the case when a company X hires another company to develop a software product. The person assuming the PO role must be from company X, i.e. the company that is hiring and paying for the solution and will receive the benefits. Eventually can be assigned to an internal person of the company hired to act as a proxy and certainly the development team and the Scrum Master will be of the contracted company.

5. Expected skills for the Product Owner

As stated in Section II, Definition of the PO profile is defined so far in the literature agile. From investigation done for this work, a set of skills that complement the expected profile is proposed. They are outlined below:
Convert tacit knowledge into explicit knowledge: Tacit knowledge is personal or individual self-knowledge, this knowledge is deeply embedded in the person´s mind and largely related to his practical experience. Explicit knowledge, the other kind of knowledge, characterized as more formal and systematic, which can be transmitted to the development team. Nonaka and Takeuchi, [5].
Technical Lexicon Management: Among other competences PO must have, it is a lexicon and understanding of terms that are probably more typical software development to their area of expertise. The competition to integrate a technology-based work team is also required.

6. Conclusions

The work done, it derives that responsibilities originally attributed to other roles in traditional development projects based on defined processes, have been transferred to the Product Owner role, proposed by Scrum in agile project management framework.
You can mention the definition, management and product prioritization actions in traditional management are Functional Analyst responsibilities and planning and economy management actions lies in the figure of the Project Leader or Project Manager. The definition of acceptance tests tasks than in traditional projects lies on a Test Analyst
The PO acts as a focal point for strategic and tactical management of the product. It is a strategic role for SCRUM and it can also be its "Achilles heel", given the risk of becoming the weakest link in the value chain of software development process. This risk is latent until the need to train those who will assume this role in agile projects is not recognized.
Clearly the role of Product Owner role is a full-time commitment, with significant responsibilities. In fact one might wonder, after reading all the abilities attributed to it, if one person really can make all ones.
When choosing the person to assume this role, it is important to keep in mind not only who should be PO, but also and especially who can be.
It is also important to consider who should receive a very extensive prior training. This training should be included as part of the project costs and time.

References

1. M. Cohn, User Stories Applied, Boston: Addison Wesley, 2004.
2. D. Leffingwell, Agile Software Requeriments, Boston: Addison Wesley, 2011.
3. Authors: The Agile Manifesto, "Manifesto for Agile Software Development," 2001. [Online]. Available: http://agilemanifesto.org/. [Accessed 18 Abril 2014].

4. P. R. Enrique, El proceso grupal, Buenos Aires: Nueva Visión, 1981.
5. R. Jeffries, "XProgramming.com," 30 08 2001. [Online]. Available: http://xprogramming.com/articles/expcardconversationconfirmation/. [Accessed 25 02 2014].
6. A. Mehrabian, Silent messages., Oxford: Wadsworth, 1971.
7. H. Nonaka I & Takeuchi, The knowledge-creating company: How Japanese Create the Dynamics of Innovations, New York-Oxford: Oxford University Press, 1995.
8. T. Mayer, The Peoples´s Scrum- Agile Ideas for Revolutionary Transformation, San Francisco: Dymaxicon, 2013.
9. M. Nir, Agile Product Owner Secrets, London: Sapir Consulting, 2014.
10. K. Rubin, Essential Scrum- A practical guide to the most popular agile process, Michigan: Addison Wesley, 2013.
11. K. &. B. M. Schwaber, Agile Software Development with Scrum, London: Prentice Hall, 2001.
12. K. &. S. J. Schwaber, "The Scrum Primer - The Definitive Guide to Scrum: The Rules of the Game," Julio 2013. [Online]. Available: https://www.scrum.org/Portals/0/Documents/Scrum%20Guides/2013/Scrum-Guide.pdf. [Accessed 12 07 2014].

A Support System for the Diagnosis of Balance Pathologies

AUGUSTO VILLA MONTE[1], FACUNDO QUIROGA[1], FRANCO RONCHETTI[1], CÉSAR ESTREBOU[1], LAURA LANZARINI[1], PEDRO M. ESTELRRICH[23], CLAUS ESTELRRICH[3], RAIMUNDO GIANNECCHINI[4]

[1] Instituto de Investigación en Informática LIDI,
Facultad de Informática, Universidad Nacional de La Plata
[2] Facultad de Ciencias Médicas, Universidad Nacional de La Plata
[3] Centro Otorrinolaringológico La Plata
[4] Facultad de Ingeniería, Universidad Nacional de La Plata
La Plata, Buenos Aires, Argentina

{avillamonte, fquiroga, fronchetti, cesarest, laural}@lidi.info.unlp.edu.ar
martinestelrrich@hotmail.com, clausestelrrich@yahoo.com.ar, giannerai@yahoo.com.ar

Abstract. Electronystagmography is one of the most widely used diagnostic studies for detecting balance dysfunction. There are various methods that can be used to carry out this diagnostic test, gyratory stimulation being the least invasive and most physiological for the patient. The procedure is based on measuring (analyzing) eye movements in search of certain patterns called nystagmuses.
In this article, we introduce a hardware and software system for carrying out this type of studies that allows the healthcare professional to acquire, view, store, and manage results. The system also provides an intelligent method based on neural networks that can detect nystagmus patterns to help healthcare professionals make a diagnosis. The system is currently being used at a medical office for the detection of balance disorders.
Even though there are similar systems that are commercially available, these are usually very expensive due to their hardware equipment requirements and use of proprietary technology. The system presented here was developed nationally at a very low cost and can be easily adapted to future changes in technology.

Keywords: medical diagnosis, balance pathologies, nystagmus, neural networks, peak detection.

1. Introduction

Three complex sensorial systems are involved in the anatomo-physiological mechanism that helps keep balance [2]: the visual system, that relates the body to its immediate surrounding environment (external coordinates); the somatosensory system, that sends information to the brain about the position of each of the different parts of the body in relation to the others (internal

coordinates, proprioception); and the vestibular system, that detects body accelerations and movements. These systems produce a large volume of information that is instantaneously processed by the central nervous system through its complex nerve pathways. Then, from the central control areas, nerve impulses are sent that result in muscle contractions that help keep stability and control body position.

Any disease or disorder, either in any of these systems or their connecting pathways, may cause an alteration in static balance (instability) or dynamic balance (vertigo). Balance disorders [1] mainly appear as dizziness that is usually accompanied by nausea, visual disorders and discomfort. The study of patients with this type of disorders is complex and encompasses many medical specialization areas. Typically, a general clinical exam, a basic neurological exam, an otorhinolaryngological exam, an ophthalmological exam and a neuro-otological exam are carried out.

The neuro-otological exam is a full neuro-physiological test of cranial nerve VIII (vestibulocochlear nerve), which also includes a battery of objective tests to study the vestibular root. In particular, electronystagmography (ENG) tests are of special interest. This type of tests consists in graphically recording specific eye movements (nystagmuses) (Figure 1) with the following special characteristics: slow deviation movement of the gaze axis to the right or to the left, followed by a rapid movement to return to the rest position or straight gaze.

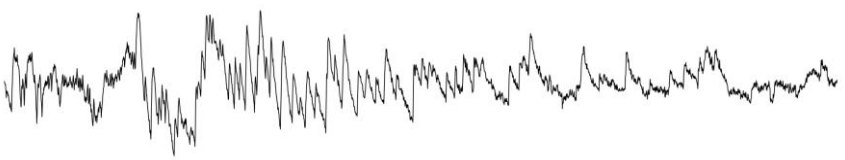


Figure 1. Typical study signal with right nystagmuses.

The occurrence of these movements is achieved in diagnostic [6] tests by means of various stimuli: changes in position, head and/or entire body rotation, as well as vestibular receptor calorization or retinal stimulation. Once the nystagmuses occur, they are identified and assessed based on various parameters, such as latency, frequency, amplitude, slow component velocity, rapid component velocity, and direction (Figure 2). This task used to be performed manually and carried a significant level of subjectivity on the part of the expert; however, computer analysis can provide an objective, automated method to assess nystagmuses with which such subjectivity can be avoided.

In the following sections, a system to assist in the early detection of balance disorders will be presented; this system was developed in the Country and has been operating for several years at an otorhinolaryngology center in the city of La Plata, Argentina. This system measures the eye movement generated when the patient is rotated while sitting on a chair. This method is

less invasive than other alternative procedures, such as introducing hot water inside the ear of the patient, but more costly due to equipment requirements.
Since it was first installed, the system has undergone several technology updates to replace outdated hardware, with the consequent adaptation of the software developed. Some of the software improvements that have been implemented include an intelligent method based on neural networks aimed at detecting eye pathologies in an adaptive manner (as detailed in Section 3). Since this is a nondeterministic method, the strategy can be adapted for new technologies, thus accommodating various signals from several converter boards.

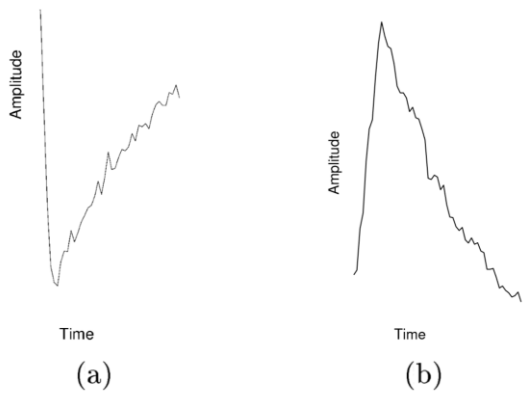


Figure 2. Parts of a signal corresponding to left and right nystagmuses, respectively. The charts show variation in amplitude versus time. In both cases, two nystagmus phases can be identified: during the first one, the fast phase, amplitude goes up or down abruptly based on nystagmus direction. During the second one, the slow phase, amplitude slowly increases or decreases.

Several tests are carried out, with durations between 20 and 60 seconds, including biological calibration, right optokinetic test, left optokinetic test, spontaneous nystagmus in the sitting position test, per and post-rotational nystagmus to the left, per and post-rotational nystagmus to the right, and pendular test.
This paper is organized as follows: Section 2 describes the system, Section 3 details the nystagmus recognition method, and Section 4 discusses the conclusions.

2. System Description

The system consists of two well-defined subsystems: a hardware system and a software system. Figure 3 shows a simplified diagram. The first subsystem includes:

- A chair, designed by an electronic engineer, where the patient sits while the test is carried out.
- A horizontal LED bar showing a light that the patient must follow with his/her gaze.
- A module that allows the healthcare professional to control the movement of both the chair and the light, and shows the corresponding velocity.
- Four electrodes that are placed on the face of the patient to capture eye movement signals.
- An A/D converter board for data acquisition.
- A basic computer, where the software application is installed.

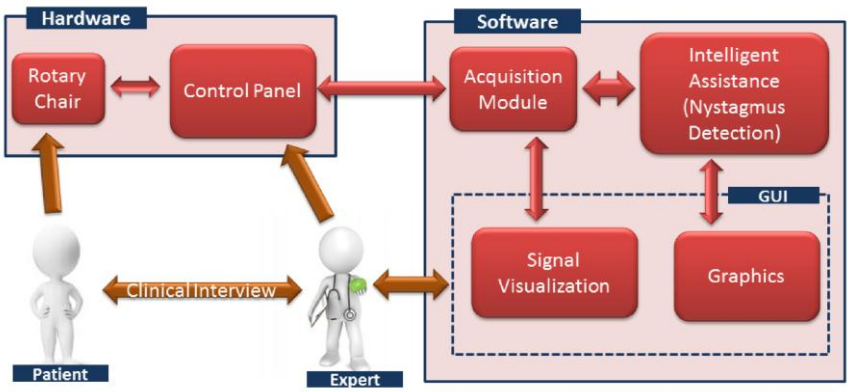


Figure 3. General diagram of the entire system.

As regards the software subsystem, it includes:

- A controller for the board.
- Signal acquisition, processing, and storage.
- The database that includes all information pertaining to the patient, including tests and diagnosis carried out.
- The algorithm for detecting nystagmuses.
- The interface for the physician.

Rotatory chair. The patient sits on a chair that can spin for the optikinectic studies, achieving a speed of up to 30° per second. The chair contains four electrodes which are placed on each side of the eyes and in the middle of them. By measuring the changes in the electrical field within the eye, the amount of rotation experienced it experiences can be measured during the study. Since the voltage signals captured by the electrode have very low intensity, an amplifier was attached to the output of the system.

Visual Interface. The visual portion is one of the most important components, since it is used by the physician to monitor signal acquisition in real time. Figure 4 shows the window used to view the signal. From top to bottom, the three channels being tested can be seen in it: both eyes combined, right eye, and left eye. The fourth channel corresponds to the velocity of the chair or movement of the light, depending on the type of test. Between tests, the physician can have the patient rest while asking him/her about symptoms and entering the information to the system.

Once the test is finished, it can be re-run as it was originally recorded to observe signal acquisition in real time (scroll) without the patient being present. This allows the healthcare professional to free the patient and analyze the signal in greater detail (as if the patient were right there on the rotating chair), taking as much time as necessary to reach an accurate diagnosis. Also, the acquired signal can be viewed in full, applying slight scale improvements and smoothing it. This window is shown in Figure 5. As it can be observed, on the upper part of the first three channels there are blue markings that correspond to automatically identified nystagmuses; the physician can then manually add or remove them.
Using the identified nystagmuses, calculations are carried out to build tables and charts that are part of an automated pre-diagnosis. This consists of a text created with default fill-in phrases that the specialist completes to issue the final diagnosis for the patient. Figure 6 shows one of the charts generated by the system.

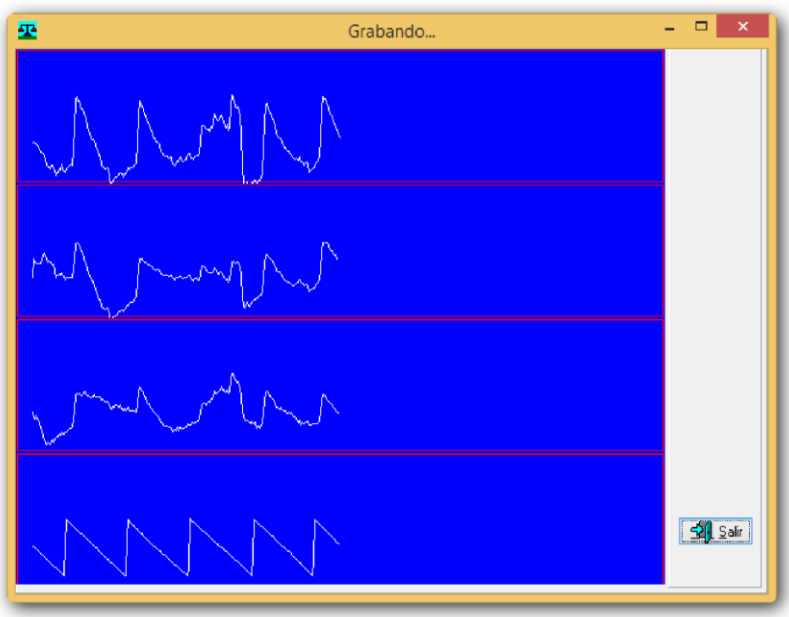


Figure 4. Signal acquisition and scroll window.

3. Automatic Identification of Nystagmus

As mentioned above, one of the main goals of the system is to help analyze the signals recorded during the various types of tests that can be performed with the chair; in this sense, it is of interest that the system can automatically identify the nystagmuses in the signals. This functionality helps to mantain a high level of objectivity in nystagmus identification, regardless of the healthcare professional that is in charge of carrying out the test and his/her particular personal situation on any given day. It also streamlines the test itself, which allows providing a more efficient service to the patient and saves specialist time, who can therefore focus solely on the diagnosis. Additionally, the subsystem responsible for carrying out this analysis should be separate from the capture hardware, so that when the latter is updated, there is no need to re-calibrate the system.

Since signals do not have a fixed number of nystagmuses, and these can appear at different positions and with different characteristics, there are two variables that are essential for the automated analysis of the signal: which signal sections correspond to the nystagmus phenomenon in the patient, and the direction of a nystagmus, i.e., whether any given nystagmus is a left or right nystagmus.

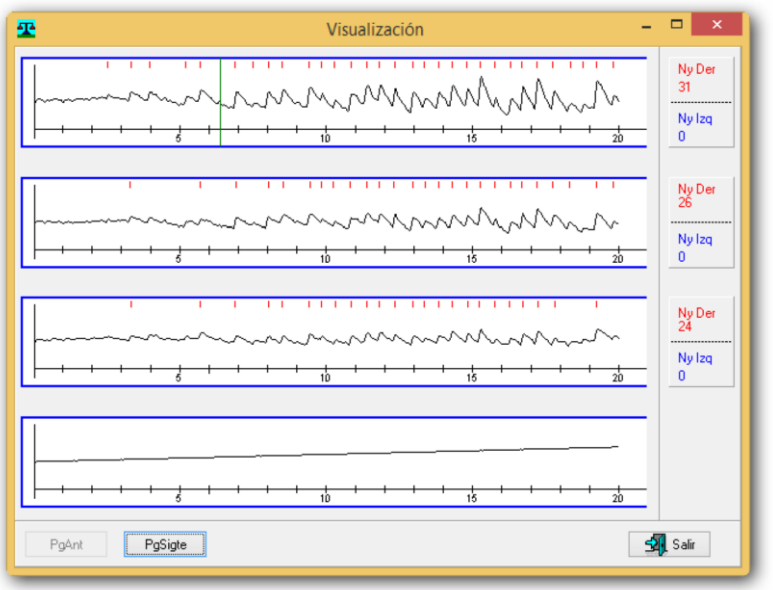


Figure 5. Signal visualization and marking window.

Using signal processing and automated learning techniques, a model was generated to analyze these signals. This model, based on artificial neural

networks, allows processing signal intervals and calculating the probability of an interval corresponding to a nystagmus and its direction.
In order to apply the model, a peak detection algorithm is used that allows proposing signal intervals as potential nystagmuses to be analyzed. For each potential nystagmus, the model calculates the probability of it being an actual nystagmus and, if the value obtained is above a preset threshold, the corresponding interval is marked as a nystagmus.
In the following subsections, the process used to generate the model and then apply it is described.

3.1 Generation of the Model for Nystagmus Detection

The detection model is based on ProbSom [3], a method developed for recognizing individuals based on voice features.
With the purpose of obtaining a large number and variety of sample nystagmuses to train the model, the features that define the nystagmuses were analyzed and, from them, a database of artificial nystagmus signals was generated. These artificial nystagmuses were fed to the ProbSom for it to learn their features and be able to identify them when presented with a new signal.
For the recognition process to be more effective, instead of using nystagmus data directly, a representation of these data based on their gradient and a windowing scheme was designed.

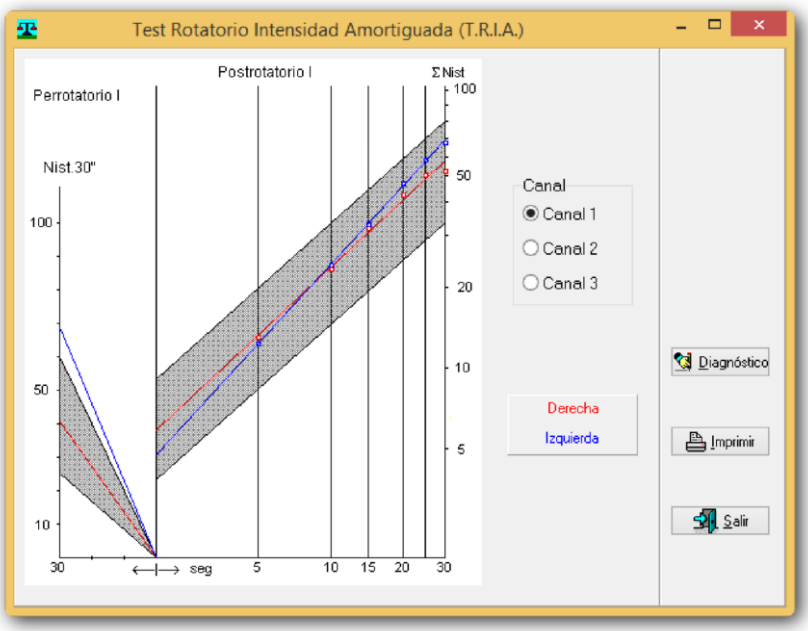


Figure 6. Rotatory Intensity-Damping Test window.

Then, after pre-processing the nystagmus signal, its gradient is calculated to obtain the invariance value corresponding to the mean amplitude of the signal and have a measure of its movement. The resulting gradient vector has as many elements as original sample points are in the signal.
Since ProbSom losses all gradient sequence information, i.e., it only analyzes the properties of the signal regardless of their order, a windowing scheme is used to include this temporal relation to signal properties. Thus, based on a signal that has M samples, V -sized windows can be generated by taking V consecutive samples from the signal. In particular, for a signal with M samples and V -sized windows, a total of $M - V + 1$ windows are generated.

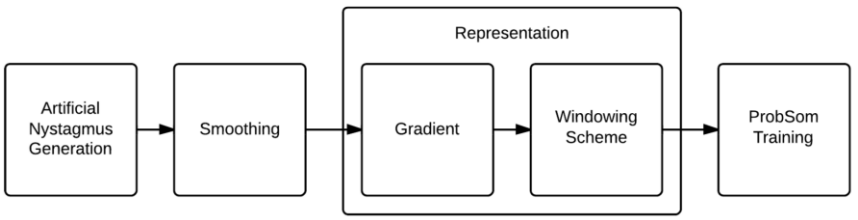


Figure 7. Diagram representing the process used to train the nystagmus recognition model.

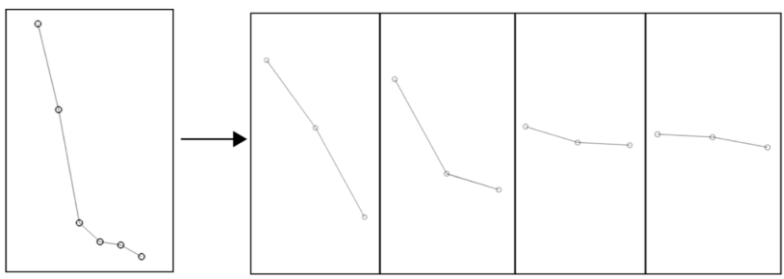


Figure 8. Generation, from signal gradient, of $M - V + 1 = 4$ windows for a signal with $M = 6$ and window size $V = 3$.

As already mentioned, the recognition model used to detect if any given portion of the signal corresponds to a nystagmus is based on ProbSom. The essential idea of this method consists in training a Self-Organizing Map (SOM) [4], a type of associative neural network, with small nystagmus portions, so that the model can learn their features. Then, each neuron on the network is associated to one of these nystagmus features, and each feature is in turn associated to a class (left nystagmus, right nystagmus, non-nystagmus). In particular, the probability of a nystagmus belonging to one of the classes is determined, given that it presents a certain feature.
In this case, each feature consists in the signal having some of the windows mentioned in the previous section.

This training process is carried out using model nystagmuses; in particular, our system uses artificial nystagmuses. Then, during the classification stage of a real signal, the model will be able to take a portion of that signal and decide whether it corresponds to a nystagmus or not and, if it does, it will be able to decide what type of nystagmus it is.

3.2 Application of the Nystagmus Detection Model

With the previous model to detect if a portion of a signal corresponds to a nystagmus and if identified nystagmuses are left or right nystagmuses, the entire signal can be processed to detect all nystagmuses present in it.
Detection consists of three parts. First, the signal is pre-processed to remove undesired noise and inter-subject differences, as well as those differences caused by varying experimental conditions. Then, a traditional peak detection algorithm is run to find maxima and minima within the signal, and, using this information, the signal is segmented into intervals that represent potential nystagmuses. Finally, the recognition model processes each of these intervals to validate if they effectively correspond to a nystagmus and, if they do, it determines their directions. These steps are described in detail below.

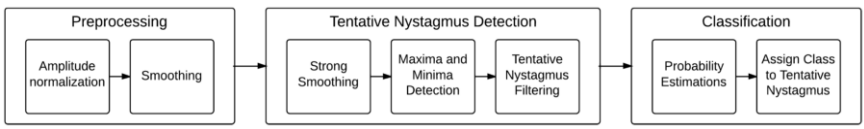


Figure 9. Three stages of the nystagmus detection process in a signal.

The signals originated from electronystagmographies present systematic differences caused by inter-subject variations and variations in the conditions for carrying out the test, such as mean room lighting. Therefore, signal amplitude is normalized for the interval [−1,1] in order to standardize the meaning of this variable.

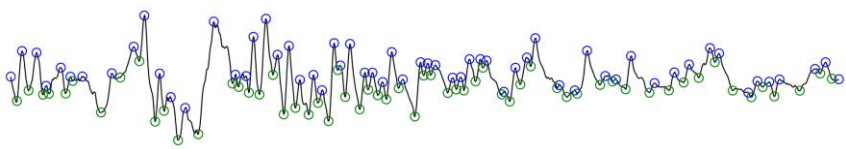


Figure 10. Maxima and minima detected in a signal that has undergone two smoothing processes.

Additionally, the signal contains noise from errors that are inherent to the capture system, as well as undesirable movements made by the patient. Since the removal of such noise at this stage is also important to prevent the

deformation of the natural peaks present in the signal, a Savitzky-Golay filter [5] was applied in a window of size w = 13, with a polynom of order n = 2.
It should be noted that in this case, the entire signal is pre-processed, not just a portion that corresponds to a nystagmus.
After the signal has been normalized, it is segmented into potential nystagmus, which will be then confirmed or discarded by the detection model. Segmentation is based on the detection of peaks, using them as indicative of the beginning and the endo of a nystagmus, which helps save calculation time, and filtering those signal portions that do no correspond to a nystagmus.
Thus, signal minima and maxima are identified by means of a traditional peak detection algorithm. However, before this step, the signal is strongly smoothed once again to increase to remove spurious maxima and minima, leaving only the most salient peaks.
Using these maxima and minima, potential intervals are identified that could correspond to nystagmuses in the original, non-smoothed signal. In the case of nystagmuses to the right, intervals with two minima around a maximum are detected, and the opposite for nystagmuses to the left, thus generating two sets of potential nystagmuses, one for each type. These intervals are filtered so that they do not overlap.

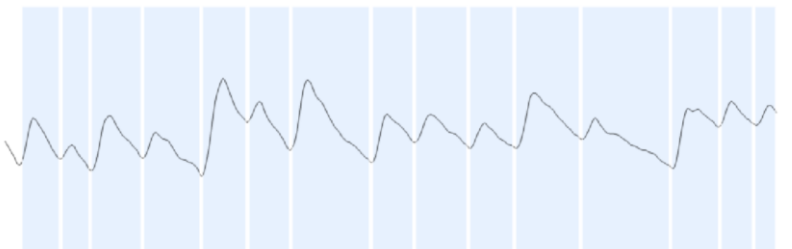


Figure 11. Potential nystagmuses to the right (blue rectangles).

For the application of the classification model, the segmented signal portions corresponding to potential nystagmuses are pre-processed and the representation described in Subsection 3.1 is calculated for each portion. Then, the ProbSom model uses these representations to calculate the probability that these signal portions belong to the left nystagmus or right nystagmus classes, or to sort them as belonging to any of the rejection classes.
It should be noted that only potential nystagmuses to the right, i.e., nystagmuses that are formed by a maximum peak between two minimum peaks, are considered for classification as right nystagmuses, and vice-versa for left ones.
Thus, in the case of potential right nystagmuses, the system confirms that they are indeed nystagmuses if the probability calculated by ProbSom for the

class right nystagmus is at least double than for the rest of the classes. The same criterion is applied for potential left nystagmuses.

4. Conclusions

A system that helps diagnose balance pathologies has been presented. It is being successfully used at a doctor's office, assisting in performing of dozens of tests each month.

The system allows the physician to capture and store the signals recorded during an electronystagmography and associate them to a patient and a set of studies so that patient history can then be reviewed with information from all the tests that he/she has undergone.

Additionally, the signals captured during the electronystagmography are automatically analyzed to identify nystagmuses and this information is presented to the health-care professional to help him/her reach a faster and more objective diagnosis. Using this intelligent processing, the values obtained in the test are compared to those corresponding to normal reference values, so that the physician can quickly determine if the patient suffers some pathology. This processing is independent from the hardware equipment used for the capture.

It should be noted that the system is a national development with a low associated cost, versus the significantly higher costs of other commercially available products with similar functionalities that are produced abroad.

References

1. Brandt, T., Daroff, R.B.: The multisensory physiological and pathological vertigo syndromes. Ann Neurol 7(3), 195-203 (1980).
2. Claussen, C.F., Bergmann, J.M., Bertora, G.: Equilibriometría y Tinnitología Práctica (2009).
3. Estrebou, C., Lanzarini, L., Hasperue, W.: Voice recognition based on probabilistic SOM. In: Latinamerican Informatics Conference. CLEI 2010. Paraguay. October 2010. (2010).
4. Kohonen, T.: Self-organizing maps, vol. 30. Springer (2001).
5. Savitzky, A., Golay, M.J.: Smoothing and differentiation of data by simplified least squares procedures. Analytical chemistry 36(8), 1627-1639 (1964).
6. Serra, A., Leigh, R.J.: Diagnostic value of nystagmus: spontaneous and induced ocular oscillations. J Neurol Neurosurg Psychiatry 73(6), 615-8 (2002).

V

Signal Processing and Real-Time Systems Workshop

Emotional Status Focused on Stimuli by Applying Brain-Machine Interface

JORGE IERACHE[1], FACUNDO NERVO[1], GUSTAVO PEREIRA[1], JUAN IRIBARREN[1]

Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica (ISIER)[1]
Facultad de Informática Ciencias de la Comunicación y Técnicas Especiales
Universidad de Morón, Cabildo 134, (B1708JPD) Morón, Buenos Aires, Argentina
54 11 5627 200 int 189
jierache@yahoo.com.ar

Abstract: This article shows the initial outcomes in developing a recording and reading system of emotional status records of an individual by using Brain-Machine Interface (BMI) focusing on arousal and meditation when faced to audiovisual stimuli.

Keywords: Emotional Control, Brain Machine Interface, Bio-Electrical Signal, Human Machine Interfaces.

1. Introduction

Over the last twenty years [1], the development of BMI has increased, thus allowing communication between the user's mental and cognitive functions to be processed and classified in order to be interpreted by applications or specific devices. The uses of BMI can range from robot control [2], wheelchairs, computer control, domotics [3] to video games [4]. Prior research [5], [6], [7] focused on the user's electrical biopotentials through electromiogram, electroencephalogram and electrooculogram which are electrical biosignals generated by activity patterns of the user's muscles, brain and eyes. In this case, we work on the readings of the user's emotions, focusing on arousal and meditation. The research of BMI interfaces is developed in a multidisciplinary, scientific field with a range of uses from computing, domotics to robotics and entertainment [8]. Different papers were presented appealing to electroencephalogram (EEG) signaling used in mental commands such as moving a computer's pointer [9], [10] based on Brain-Machine Interface (BMI). Saulnier et. al. [11] implement a speed control based on electromiograms and infer the user's stress level by electroencephalograms and thus, influence the social behavior of domestic robots, particularly a robot vacuum. Recent research by Mitsukura [12], using Neurocam [13], allows to capture images the user observes from a threshold of emotional status. Other Neurowear [14] applications, such as Necomimi [15] (Fig. 2.a) and Shippo [16] (Fig. 2.b) link a device on the user's head to the control of cat ears or animal tails which, according to emotional status, adopt different positions.

Fig. 1. Neurocam

Fig. 2.a. Necomimi

Fig 2.b. Shippo

EEG sensors together with BMI also open the possibility to diverse software uses allowing to directly obtain signals from the user's brain waves in order to know taste, sensations and emotional status. Some examples of the abovementioned can be observed in the applications: ZEN TUNES (Fig. 3.a) is an iPhone app which analyzes brain waves emitted when listening to music and produces a musical graph based on the "relaxation" and "focus" states of the listener. The reproduction list is unique to the listener since it organizes the information according to how each song is perceived by the user and it can be understood as his/her personality. NEURO TURNTABLE (Fig. 3.b) is a music player, which reads neural waves and plays music only when the user concentrates. The music starts when the user concentrates and automatically stops when he/she starts talking to someone or thinks about something different. There is a kind of unconscious dialogue between the user and the application, in which the user realizes he/she could not actually be focused on the music, even when thinking he/she was.

Fig. 3.a. ZEN TUNES

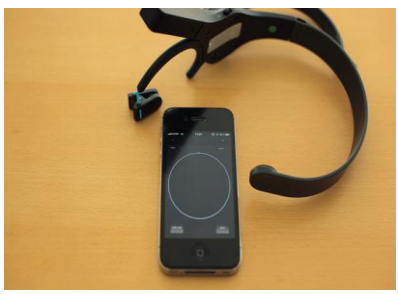


Fig. 3.b. Neuro Turntable

Brain Disco is a project in which DJs must keep the audience's attention that is measured through brain wave sensors in order to continue playing music. If the average attention span remains lower than a determined threshold for some time, the DJ is expelled. NeuroSky has developed a family of headbands called MindWave which are designed so that developers can quickly create applications monitoring electroencephalogram signals. The MindWave family is divided into MindWave and MindWave Mobile.

MindWave is designed to run on PC and Mac while the Mobile version is also compatible with mobile devices running on IOS and Android operating systems.In the context of emotion capturing tools, the manufacturer of BMI EMOTIV EPOC provides a set of libraries to being able to work with the three Suites (Expressive, Affective, and Cognitive). The manufacturer, in its online shop in the section "application/neurofeedback", also offers the application Mind WorkStation from Transparent Corp [17]. It is an all-in-one set of tools for professional development in "brainwave synchronization" (Brainwave Entertainment) fully compatible with EMOTIV EPOC and with which real-time emotions can be, among many other things, charted and exported to a spreadsheet file with a specific representation determined by the manufacturer which, in our research, was not suitable since we were not able to record the reading and storage of emotional data from BMI in correspondence with the external stimulus provided to the user.

2. Problem

In order to capture and record the emotional values according to the stimuli provided in an editing software, it was necessary to develop an application allowing the search for global patterns from different users when faced to audiovisual stimuli.

3. Solution description

As the basis of the proposed solution, the experiences from the prior research works focusing on the execution of commands by using BMI from EMOTIV [18], [19] were taken into consideration; this BMI (Fig. 4.a) was used with a NXT robot control (Forward, Backward, Left, Right). In CACIC 2013 real demonstrations of device control (DVD player, music player, lamp) were performed by using BMI from Emotiv based on the Framework developed for device control [20].

Fig. 4.a. BMI Emotiv

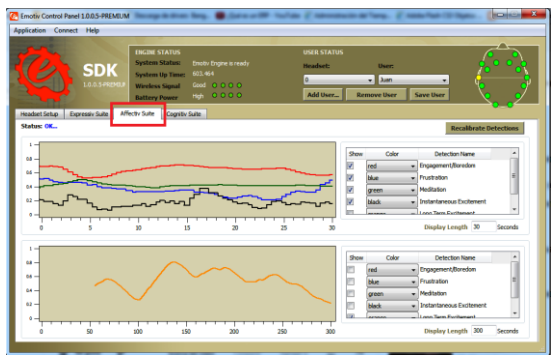


Fig. 4.b. Affective Suite

BMI (Brain Machine Interface) EMOTIV EPOC, articulated with its SDK features a control panel to create the user and log the profile; it also helps to visualize the connection status of sensors and represents different record patterns (**expressive**, **affective** and **cognitive)**. In particular, our current research development focuses on the **affective** record pattern, which verifies different moods. The affective suite (Fig. 4.b) allows the visualization of certain emotional mood changes according to time.

The recorded emotional moods are: Commitment/Boredom, Frustration, Meditation, Instant Arousal, long-term Arousal. Two schemes can also be visualized; one short-term 30-second scheme and one long-term scheme allowing the visualization of any of the emotional moods for at least 300 seconds (5 minutes). In both graphics there exists the possibility of choosing the moods the user prefers to analyze and also picking any particular color.

The developed application allows the recording and storage of emotional signals read with EMOTIV EPOC during a certain period of time while the user is stimulated with an external source (audiovisual presentation) and then, it performs an analysis in order to find patterns by linking the stimuli which produced changes in the user's biosignals. Thus, the user logs the duration of the session in which generated emotions will be recorded from external stimuli of a presentation file; for the case trials, two stimuli were set up: a black picture and an audio of a telephone recording. Then, when logging the session in the application, the emotions are recorded with the developed application; these emotions detected by BMI are linked to the audio or image shown at that moment and stored in a database until reaching the duration of the session defined by the user. After finishing the session, it is created a spreadsheet file with the recorded values for further analysis to find patterns arising from the stimulus/recorded emotion relation and have a unique identifier of the session number and a time marker (HH:MM:SS) linked to each recording at every instant, creating the session file. In Figure 5,a conceptual diagram of the described process can be observed.

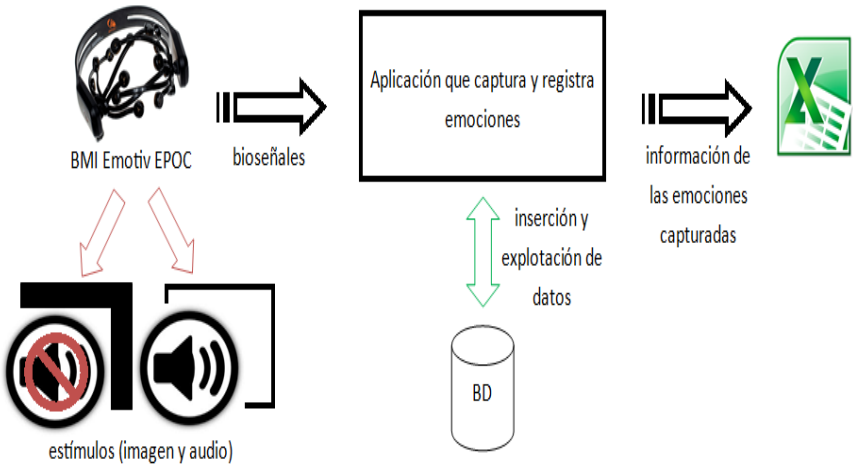


Fig. 5. Integration of BMI-Emotiv and the emotion recording and storage application.

.

The application was developed in .NET C# with WinForms and Entity Framework with SQLite as database, on a three-layer architecture (Client-Business-Data). For the connection to BMI, the API from Emotiv [21] was used; it is an application programming interface or, more specifically, a library of functions offered by EMOTIV so that developers can write software applications which work with the BMI device and the detection suites.To develop the application, the library DotNetEmotivSDK.dll, which is a library wrapper written in C but specifically developed for .NET, was used. Figure 6 represents the component diagram, which shows the interaction between the client software (ClienteWF) and the components related to emotion recording and storage (LogicaEmotiv, DotNetEmotivSDK.dll) and spreadsheet file generation (ClosedXML.dll)

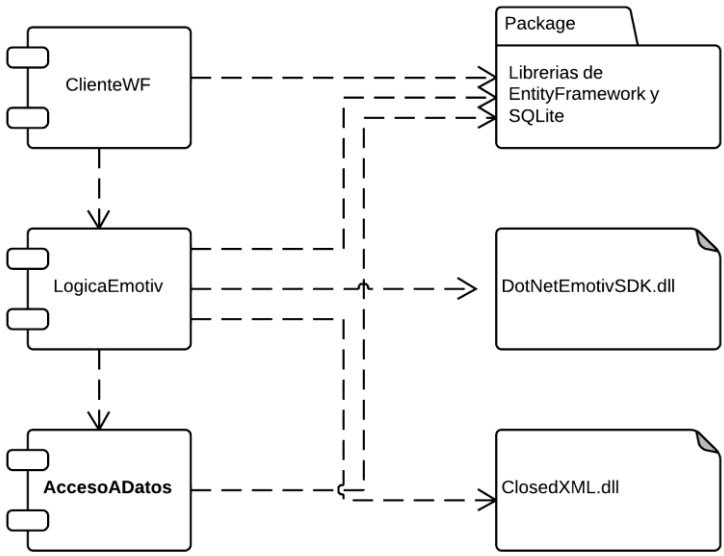


Fig. 6. Component diagram

Figure 7 shows the class diagram with properties and methods used to connect all the solution components. The class **AdministratorEmotiv** (**AdministradorEmotiv**) stands out since it verifies that the emotion recording time is no longer than the session, records the stimulus from the readings EPOC would detect and assigns the values related to emotions so that the class **StorageBaseEmotion** (**RepositorioBaseEmocion)** enters the values represented by the class **Emotion** assigned in each reading.

Fig. 7. Class diagram

Figure 8 shows the application sequence diagram from the beginning of a new recording session, logged in by the user, where the class **AdministratorEmotiv** extracts the maximum value from the session number; it records the stimulus from the emotion signal readings while the time elapsed is shorter than the test time. Via the class **StorageBaseEmotion** (**RepositorioBaseEmocion**) the values obtained from each biosignal reading are inserted until the creation of a spreadsheet file with the data from the recently executed session.

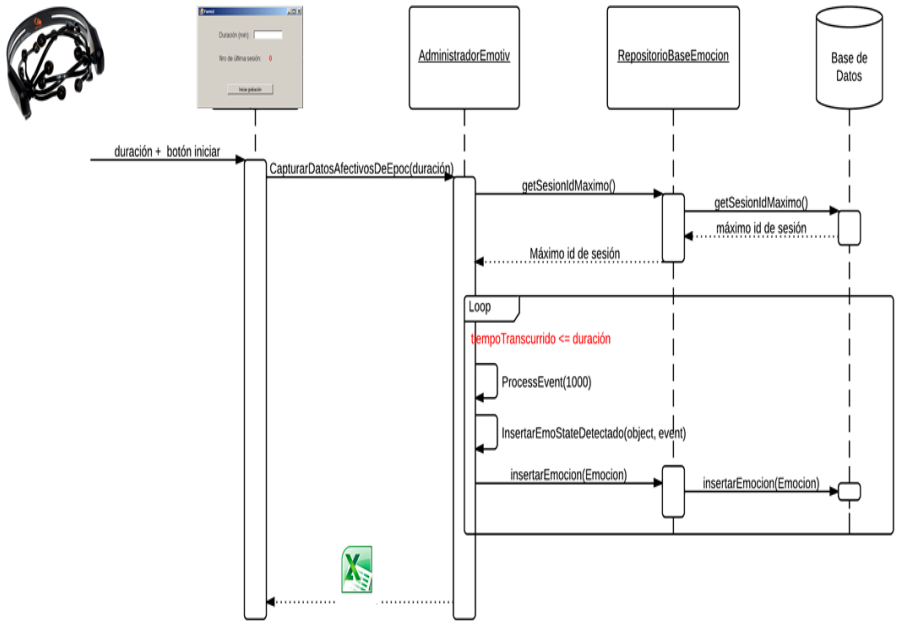


Fig. 8. Application sequence diagram

4. Trials and Obtained outcomes

In order to perform the trials, it was used a room without sound from the exterior or light so that the user would concentrate and avoid external stimuli that would disrupt the trial and allow uncontrolled or exterior-related emotions. During the trial, two PCs were placed; the first one executed the session creating the audiovisual stimuli through headphones and speakers, and the second one ran the application recording the values read by Emotiv EPOC. A series of trials [22] were performed between two male individuals with an average age of twenty-seven both independently performing a total of eight tests. The stimulation session used in each test was performed for a total duration of four minutes (240 seconds) and with full-screen visualization. During the first minute (0 - 60 seconds), a black, soundless image is observed (stimulus A). During the next two minutes (61 - 180 seconds), an audio recording of a telephone call to 911 is played [23]; the call was made by a teenager asking for help due to a domestic violence incident; this recording is paired with a white image (stimulus B). During the last minute (181 - 240 seconds), a black, soundless picture is shown again (stimulus A). The short-term arousal values and the meditation values were considered by default for the test recordings. The synchronization for execution corresponds to the

beginning of the session generated by the individual. Consequently, a graphic from one of the tests which has been randomly chosen (Fig. 9) is shown to describe how the emotion values vary during the period of time with the stimuli used in the test. The periods of time each stimulus lasted during the test are shown in it.

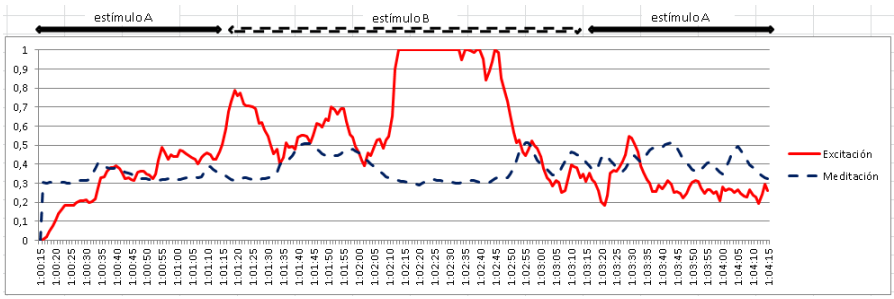


Fig. 9. Representation of the recorded values in a session

Stimulus A: absence of sound/audio and black image (60 seconds).
Stimulus B: 911 phone call and white image (120 seconds).

In the representation, during stimulus A, a slightly increasing arousal value and a stable meditation value can be observed. When stimulus B starts, the arousal increased due to the contrast from the black to white image and from the absence of sound to the phone call. Then, during the most dramatic moment of the phone call (the teenager is crying because the mother is being beaten), the arousal reaches its highest value for about 30 seconds. Near the end of the test, when stimulus A is shown, the arousal value decreased and remained unchanged. Figure 10 represents the arousal results obtained during four of the tests performed on an individual.

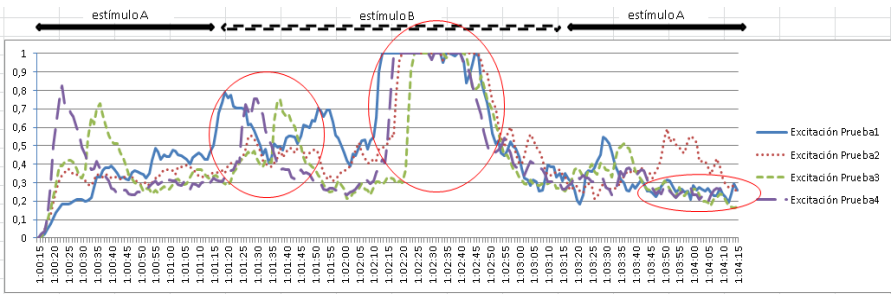


Fig. 10 Arousal results obtained from the sessions

By having all the arousal results from the test chosen for analysis represented, it can be observed that after stimulus A, when stimulus B starts, the arousal value increases for some seconds; in some cases, it can be

observed that this happens a few moments before; in others, it happens moments later, but almost all reach the same value. Then, during the most dramatic moment of the call, it can be observed that the individual's maximum arousal value is reached for a mean period of time of 15 seconds. During the last part of the test, a decrease in the user's arousal can be observed.
Of the tests performed, we have concluded that general patterns of emotional behaviour related to stimuli can be found, especially when related to an individual's arousal. Lastly, figure 11 represents the mean arousal and meditation values of each individual, with the values obtained from the eight performed tests.

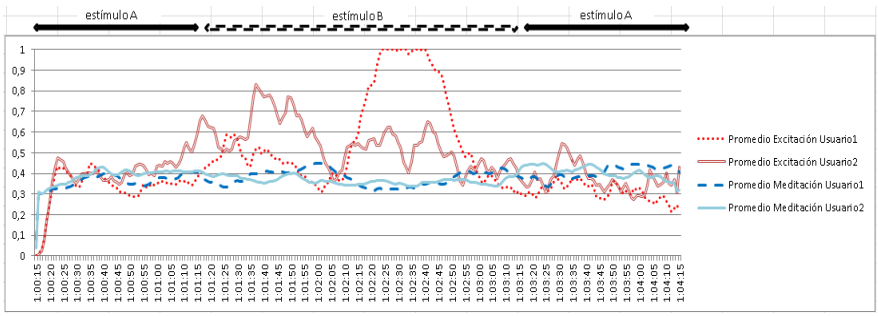


Fig. 11. Mean meditation and arousal values obtained from the sessions chosen for both individuals

Finally, from the analysis of the eight trial sessions, it can be concluded that with a specific sound stimulus (in our case, a real recording of a phone call asking for help) it is possible to repeat an arousal value pattern during a period of time with different users considering that the values obtained by each individual are independent.

5. Conclusions and Future Lines of Research

Even though the signal values obtained from each individual are independent, it is not possible to anticipate that two users would generate equal emotional values since each of them responds with their emotional level. However, the aim of recording and representing the values according to the stimuli the individual faced was achieved, thus proving that similar patterns are emotionally repeated in both individuals.

In the context of future lines of research, it is intended to expand the application functionalities with the development of a framework allowing multi-media stimuli integration, improving record collection and exploitation, and automatically finding patterns. Other research areas focus on the

emotional learning of robots, devices with answer to the user's emotional state.

6. Acknowledgments

This research is financially supported by *Universidad de Morón* as part of PID 01-001/14, which is conducted at the *Instituto de Sistemas Inteligentes y Enseñanza Experimental de la Robótica (ISIER)*.

References

1. Hamadicharef, "Brain Computer Interface (BCI) Literature- A bibliometric study", in 10th International Conference on Information Science, Signal Processing and their Applications, Kuala Lumpur, 2010, pp. 626-629.
2. Ierache, J., Dittler, M. García-Martínez, R., "Control de Robots con Basado en Bioseñales". XII Workshop de Investigadores en Ciencias de la Computación WICC 2010: Universidad Nacional de la Patagonia San Juan Bosco, Calafate, Santa Cruz, Argentina. 2010, ISBN 978-950-34-0652-6, pag 641
3. Ierache., J, Pereira.,G, Sattolo., J, Iribarren Aplicación de interfases lectoras de bioseñales en el contexto de la domótica XV Workshop de Investigadores en Ciencias de la Computación 2013 Facultad de Ciencia y Tecnología Universidad Auto noma de Entre Ríos (UADER), ISBN: 9789872817961.
4. http://edugamesresearch.com/blog/tag/bci/,
5. Ierache, J., Dittler M., Pereira G., García Martínez R.,(2009) "Robot Control on the basis of Bio-electrical signals" XV Congreso Argentino de Ciencias de la Computación 2009, Universidad Nacional de Jujuy, Facultad de Ingeniería, ISBN 978-897-24068-3-9 ,pag 30.
6. Ierache Jorge, Pereira Gustavo, Iribarren Juan, Sattolo Iris, "Robot Control on the Basis of Bio-electrical Signals" : "International Conference on Robot Intelligence Technology and Applications" (RiTA 2012) Gwangju, Korea on December 16-18, 2012. Series Advances in Intelligent and Soft Computing of Springer.
7. Ierache., J, Pereira.,G, Sattolo.,I , Guerrero., A, D'Altto J, Iribarren,. J. Control vía Internet de un Robot ubicado en un sitio remoto aplicando una Interfase Cerebro-Máquina". XVII Congreso Argentino de Ciencias de la Computación 2011, Universidad Nacional de La Plata, Facultad de Informática, ISBN 978-950-34-0756-1, paginas 1373-1382.
8. M. A. Lebedev and M. A. L. Nicolelis, "Brainmachine interfaces: Past, present and future," Trends in Neurosciences, vol. 29, no. 9, pp. 536-546, September 2006.
9. R. Wolpaw, D. J. McFarland, and T. M. Vaughan, "Brain-computer interface research at the Wadsworth center," IEEE Trans. Rehab. Eng., vol. 8, pp. 222–226, 2000.
10. J. del R Millán, "Brain-computer interfaces," in Handbook of Brain Theory and Neural Networks, 2nd ed, M.A. Arbib, Ed. Cambridge, MA: MIT Press, 2002.
11. Paul Saulnier, Ehud Sharlin, and Saul Greenberg. Using Bio-electrical Signals to Influence the Social Behaviours of Domesticated Robots. HRI'09, 2009, USA.ACM 978-1-60558-404-1/09/03.
12. http://www.st.keio.ac.jp/kyurizukai/english/11_mitsukura_e/pdf/mitsukura11_e.pdf
13. http://neurowear.com/projects_detail/neurocam.html
14. http://neurowear.com/news/index.html

15. http://www.necomimi.com/
16. http://neurowear.com/projects_detail/shippo.html
17. Mind Workstation http://emotiv.com/store/app/?cat=121
18. http://www.emotiv.com/
19. Ierache Jorge, Pereira Gustavo, Iribarren Juan del articulo "Demostración de los resultados en la integración de Interfases Lectoras de Bioseñales aplicadas al Control de un Robot" VII Congreso Educación en Tecnología y Tecnología en Educación 2012 Universidad Nacional del Noroeste de la Provincia de Buenos Aires. UNNOBA, 2012, demos educativas. ISBN 978-987-28186-3-0.
20. Ierache., J, Pereira., G, J, Iribarren Framework for Brain Computer Interface implemented to control devices in the context of home automation XIX Congreso Argentino de Ciencias de la Computación 2013, CAECE Mar del Plata, ISBN 978-897-23963-1-2.
21. API Emotiv EPOC https://emotiv.com/epoc/develop.php
22. Sitio ISIER en Facebook https://facebook.com/isierum
23. Archivos de sesión https://googledrive.com/host/0B5HvIVl8A-ROYTUxTHNYakN1OFU

III

Computer Security Workshop

Trivium vs. Trivium Toy

ANTONIO CASTRO LECHTALER[1,2], MARCELO CIPRIANO[1], EDITH GARCÍA[1], JULIO CÉSAR LIPORACE[1], ARIEL MAIORANO[1], EDUARDO MALVACIO[1]

[1]EST – Facultad de Ingeniería - Instituto Universitario del Ejército
[2]FCE - Universidad de Buenos Aires

{acastro, marcelocipriano}@iese.edu.ar
{edithxgarcia, jcliporace, maiorano, edumalvacio}@gmail.com

Abstract. We present the characteristic and minimal polynomials of the linear algorithms Trivium and Trivium Toy. We show the different cycles and minimum lengths obtained. The existence of initial states determining short cycles is verified. Finally, linear Trivium Toy is shown to be as cryptologically secure as the linear Trivium algorithm.

Keywords: Trivium, Trivium Toy, cycles, periods, weak initial states**.**

1. Introduction

1.1 Pseudo-random Sequences

Pseudo-random sequences with long cycles and high linear complexity are widely used in the fields of communications and cryptology.
Until recently, these sequences were generated with simple algorithms, using *Linear Feedback Shift Registers (LFSR)* – alone or combined – along with certain non-linear components.
Currently, their design has become more complex. Nonetheless, two cryptological properties should be carefully observed and controlled: the length of the cycle and the linear complexity. Generators with short periods, small cycles or low linear complexity are cryptanalized and then broken. Hence, algorithm design must verify that it achieves an acceptable limit value of the minimum period.

1.2 Trivium and Trivium Toy

The stream cipher Trivium – an e-STREAM finalist- has successfully endured every cryptological attack so far. [2, 6, 13]. However, its minimum period has not been determined neither by design nor by cryptanalysis, remaining as an open problem to this date.

2. Overview

2.1 Feedback Shift Register (FSR)

Let the polynomial *f(x):*

$$f(x) = c_0x^0 + c_1x^1 + c_2x^2 + \ldots\ldots + c_{n-1}x^{n-1} + x^n \; c_i \in \{0,1\} \quad (1)$$

be an n^{th} degree characteristic polynomial over *GF(2)*.
A sequence $\boldsymbol{s} = \{s_i\}$ is a length *n LFSR* sequence generated by *f(x)* if it satisfies the following linear recurrence relation:

$$s_{n+k} = \sum_{i=0}^{n-1} c_i s_{k+i} \quad (k=0;1;2;...) \quad (2)$$

Note that if the first *n* bits of ***s*** belong to the initial state, the register corresponds to a feedback polynomial (*feedback function*).
If, on the other hand, ***s*** begins with the fed bits, except for those in the initial state, the characteristic polynomial is considered a *feedforward function*.
The polynomial *f(x)* can also be interpreted as a linear Boolean function

$$f: \{0,1\}^n \rightarrow \{0,1\} \quad (3)$$

$$f(x_0; x_1;..; x_{n-1}) = c_0x_0 + c_1x_1 + ... + c_{n-1}x_{n-1} \quad (4)$$

If the resulting function is non-linear, it is considered a N*on-Linear Feedback Shift Register (NLFSR).*

$$(s_0; s_1; \ldots; s_{n-1}) \qquad s_i \in \{0,1\} \quad (5)$$

where s_i is the initial state of the LFSR generating the sequence ***s***.
Given any polynomial *f(x)* of degree *n*, the reciprocal polynomial *f*(x)* is defined as

$$f^*(x) = x^n f(x^{-1}) \quad (6)$$

2.2 Properties of m-sequences

If *f(x)* is a *primitive polynomial*[1], ***s*** is an *m-sequence,* thus ***s*** has a maximum cycle of 2^n*-1;* i.e., given any initial state (except when all values equal 0), all sequences belong to the same cycle.
If *f(x)* is not primitive, different initial states generate cycles smaller than 2^n*-1*. [7]

[1] A polynomial $f(x)$ over GF(2), irreducible of degree n, is *primitive* if the least positive integer m such that $f(x) \mid (x^m + 1)$ is $m = 2^n\text{-}1$.

A minimal polynomial of ***s*** is the polynomial of the smallest degree generating ***s***. If *m(x)* is the minimal polynomial of ***s***, then *m(x)* divides *f(x)*.
The *linear complexity* of ***s*** *(LC(**s**))* is the degree of the minimal polynomial *m(x)*. In general, *m(x)* can be found using the *Berlekamp-Massey* algorithm, taking *2LC(**s**)* consecutive bits [10].
S(f(x)) is defined as the set of all binary sequences which satisfy the recurrence relation determined by *f(x)*.
The *order of f(x)* is defined as the least positive integer *e* such that $f(x) \mid x^e + 1$.
The period of a sequence ***s*** equals the order of its minimal polynomial. It is the least integer *p* such that $s_n = s_{n+p}$ for every positive *n*.
The array $(s_0; s_1;...; s_{p-1})$ is the cycle of the sequence ***s*** and its size is equal to *p*.

2.3 Linear Trivium

The stream algorithm TRIVIUM was created by Christophe De Cannière and Bart Preneel. It was designed to generate at least 2^{64} bits, using an *80-bit* secret key and an initialization vector (IV) of also *80* bits [3].
It consists of three combined NLFSRs. The first register controls the second, the second controls the third, and the last one controls the first.
The core idea behind the design focuses on using the principles of block cipher design to create equivalent components in stream ciphers.
Three parts can be clearly identified in the design:

- A linear part originated by a *96-bit* sub-generator which consists of three linear feedforward and feedback registers.
- An interleave process *in threes* of the linear Trivium sub-generator [8].
- A non-linear part obtained from AND operations in the linear Trivium.

The output consists of three combined non-linear shift registers of lengths 93, 84, and 111 in which particular positions are selected to obtain a key bit stream. Whereas no efficient attack has successfully broken the generator, its period remains undetermined [11, 12].
A complete description is given by the following simple pseudo-code:

```
INPUT: s0, s1,..,s287 initial state, integer n., si
{0,1}.

OUTPUT: binary sequence {kt}

    1.Initialization.

    1.1 t1 ← s65 ⊕ s92
    1.2 t2 ← s161 ⊕ s176
    1.3 t3 ←s242 ⊕ s287
    2.While ( t<n ) do the following:
          2.1   kt ←t1 ⊕ t2  ⊕ t3
          2.2   t1 ←t1 ⊕ s90 ⊗ s91 ⊕ s170
```

2.3 $t_2 \leftarrow t_2 \oplus s_{174} \otimes s_{175} \oplus s_{263}$

2.4 $t_3 \leftarrow t_3 \oplus s_{285} \otimes s_{286} \oplus s_{68}$

2.5 $(s_0; s_1; \ldots; s_{92}) \leftarrow (t_3; s_0; \ldots; s_{91})$

2.6 $(s_{93}; s_{94}; \ldots; s_{176}) \leftarrow (t_1; s_{93}; \ldots; s_{175})$

2.7 $(s_{177}; s_{178}; \ldots; s_{287}) \leftarrow (t_2; s_{177}; \ldots; s_{285})$

3.Return $\{k_t\}$

Note that $\oplus$ is the XOR operation and $\otimes$ the AND operation.

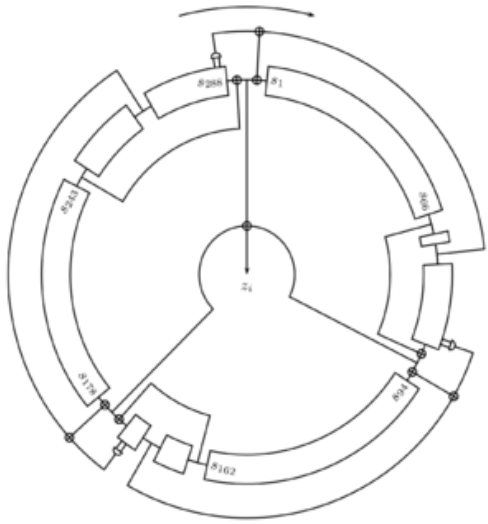


Fig.1: Original Trivium diagram

The linear Trivium algorithm follows the same procedure with the exception of the AND operations which are omitted. Terms $s_{90} \otimes s_{91}$, $s_{174} \otimes s_{175}$ and $s_{285} \otimes s_{286}$ are eli-minated.

2.4 Linear Trivium Toy

In [1] we present a reduced model of the Trivium algorithm. The reduced model - decimated by 3 is based on previous work by Yun Tian et al, who developed an extended model of the TRIVIUM structure [14].
The model consists of three NLFSRs - X, Y, and Z - of lengths 31, 28 and 37 in the following states:

(7)

$$X(31): \quad X_0, X_1, \ldots\ldots\ldots, X_{30}$$
$$Y(28): \quad Y_0, Y_1, \ldots\ldots\ldots\ldots, Y_{27}$$
$$Z(37): \quad Z_0, Z_1, \ldots\ldots\ldots\ldots, Z_{36}$$

Being the feedback of each register; i.e. the bit input in each:

$$\begin{aligned} X_0&: Z_{21} \oplus Z_{36} \oplus Z_{35} \otimes Z_{34} \oplus X_{22} \\ Y_0&: X_{21} \oplus X_{30} \oplus X_{29} \otimes X_{28} \oplus Y_{25} \\ Z_0&: Y_{22} \oplus Y_{27} \oplus Y_{26} \otimes Y_{25} \oplus Z_{28} \end{aligned} \tag{8}$$

and the key bit stream:

$$K_t: \qquad X_{21} \oplus X_{30} \oplus Y_{22} \oplus Y_{27} \oplus Z_{21} \oplus Z_{36} \tag{9}$$

In a stream cipher each plaintext bit is encrypted one at a time with the corresponding bit of the key bit stream, to give a bit of the ciphertext stream.

$$C_t = P_t \oplus K_t \tag{10}$$

where C_t is the cipher bit and P_t is the plaintext bit.

The linear Trivium Toy algorithm consists of the same equations shown in (8) omitting the AND operations. Terms $Z_{35} \otimes Z_{34}$, $X_{29} \otimes X_{28}$ *and* $Y_{26} \otimes Y_{25}$ are eliminated.

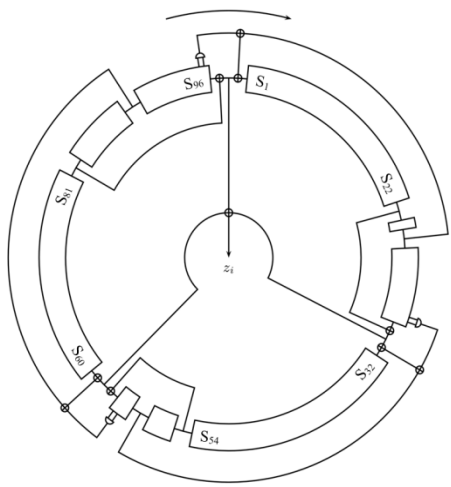


Fig.2: Trivium Toy diagram.

3. Characteristic Polynomial of the Linear Trivium Sub-generator and the Linear Trivium Toy

3.1 Feedforward and Feedback Functions of the Linear Trivium Sub-generator

The *feedforward* and *feedback* functions ($f_i(x)$ and $g_i(x)$ respectively) in their reciprocal form (6), define the Trivium sub-generator [2] and determine their characteristic polynomial $p(x)$:

$$p(x) = \prod_i f^*_{i\,(x)} + \prod_i g^*_{i\,(x)} \tag{11}$$

$$f_i^* \begin{cases} f_1^* = 1 + x^9 \\ f_2^* = 1 + x^5 \\ f_3^* = 1 + x^{15} \end{cases} \tag{12}$$

$$g_i^* \begin{cases} g_1^* = x^{31} + x^8 \\ g_2^* = x^{28} + x^2 \\ g_3^* = x^{37} + x^8 \end{cases} \tag{13}$$

$$\begin{aligned} p(x) = {} & x^{96} + x^{73} + x^{70} + x^{67} + x^{47} + x^{44} + x^{41} + x^{29} + x^{24} + x^{20} \\ & + x^{18} + x^{15} + x^{14} + x^9 + x^5 + 1 \end{aligned} \tag{14}$$

The polynomial is not irreducible, i.e., it can be expressed as a product of two polynomials such that:

$$p(x) = q(x) * r(x) \tag{15}$$

$$q(x)=(x+1)^3 \tag{16}$$

$$\begin{aligned} r(x) = {} & x^{93} + x^{92} + x^{89} + x^{88} + x^{85} + x^{81} + x^{80} + x^{77} + x^{76} + x^{73} \\ & + x^{72} + x^{70} + x^{68} + x^{67} + x^{44} + x^{43} + x^{41} + x^{39} \\ & + x^{38} + x^{35} + x^{34} + x^{31} + x^{30} + x^{27} + x^{25} + x^{23} \\ & + x^{20} + x^{19} + x^{17} + x^{14} + x^{13} + x^{12} + x^9 + x^8 \\ & + x^6 + x^4 + x + 1 \end{aligned} \tag{17}$$

where $r(x)$ is a primitive polynomial.

3.2 Feedforward and Feedback Functions of the Linear Trivium Toy

Due to [1], consider the X register. The feedforward of the Z_{21} position corresponds to x^{22} of f_3, and the feedback X_{22} corresponds to x^{23} of g_1.

For the Y register, the feed-forward of the X_{21} position corresponds to x^{22} of f_1; and the feedback is Y_{25}, corresponding to x^{26} of g_2.

For the Z register, the feed-forward of the Y_{22} position corresponds to x^{23} of f_2; and its feedback is Z_{28}, corresponding to x^{29} of g_3.

$$f^*_i(x) = \begin{cases} f^*_1(x) = (x^{-22} + x^{-31}) * (x^{31}) = (x^9 + 1) \\ f^*_2(x) = (x^{-23} + x^{-28}) * (x^{28}) = (x^5 + 1) \\ f^*_3(x) = (x^{-22} + x^{-37}) * (x^{37}) = (x^{15} + 1) \end{cases} \quad (18)$$

$$g^*_i(x) = \begin{cases} g^*_1(x) = (x^{-23} + 1) * (x^{31}) = x^8 + x^{31} \\ g^*_2(x) = (x^{-26} + 1) * (x^{28}) = x^2 + x^{28} \\ g^*_3(x) = (x^{-29} + 1) * (x^{37}) = x^8 + x^{37} \end{cases} \quad (19)$$

As explained above, the *feedforward* and *feedback* functions ($f_i(x)$ and $g_i(x)$ respectively) define the linear *Trivium Toy* and determine its characteristic polynomial $p(x)$.

The *characteristic polynomial* of the linear Trivium Toy is obtained by applying formulaes (6) and (11) to $f^*_i(x)$ and $g^*i(x)$. The resulting $p(x)$ is the same as the polynomial of the linear sub-generator of Trivium, as well as the one obtained in formula (14).

4. Calculating Sequences and Periods of the Linear Trivium Toy

4.1 Background

In order to establish the main results of this section, the following theorems must be considered [9]:

Theorem 1: Let f(x) = $\prod_i f_i^{b_i}$ where the $f_i(x)$ are distinct irreducible polynomials over $GF(2)$ and b_i are positive integers. Then:

$$S\big(f(x)\big) = S(f_1(x)^{b_1}) + S(f_2(x)^{b_2}) + \cdots + S(f_n(x)^{b_n}) \quad (20)$$

Define $S(f_1(x)^{b_1}) + S(f_2(x)^{b_2}) + \cdots + S(f_n(x)^{b_n})$ to be the set of all sequences $\boldsymbol{s}_1 + \boldsymbol{s}_2 + \ldots + \boldsymbol{s}_n$ with $\boldsymbol{s}_i \in S(f_i(x)^{b(i)})$.

Theorem 2: for each i = 1; 2;…; n, let s_i be a linear recurring sequence in $GF(2)$ with a minimal polynomial $f_i(x) \in GF(2)[x]$ and a least period p_i.
If the polynomials $f_1(x), f_2(x), \ldots, f_n(x)$ are pair-wise relatively prime, then the

least period of $s_1 + s_2 + \ldots + s_n$ is equal to the least common multiple of $p_1; p_2; \ldots; p_n$.

Theorem 3: let $f(x) = (g(x))^b$ with $g(x) \in GF(2)[x]$ irreducible over $GF(2)$, $g(0) \neq 0$, degree $(g(x)) = k$, *order* $(g(x)) = e$, and b a positive integer. Let t be the smallest integer with $2^t \geq b$. Then, $S(f(x))$ contains the following numbers of sequences with least periods: one sequence with least period *1*, 2^k-1 sequences with least period e, and for $b \geq 2$, $2^{2^j k} - 2^{2^{j-1} k}$ sequences with least period $e*2^j$ $(j=1;..;t-1)$, and $2^{kb} - 2^{2^{t-1} k}$ sequences with least period $e*2^t$.

4.2 Linear Trivium Toy Sequences and Periods

Formula (15) shows that the characteristic polynomial $p(x)$ of the Linear Trivium Toy is reducible. Thus, different initial states yield different Least Periods or cycles. Theorems 1 to 3 are applied to obtain the following values:

For $q(x) = (x+1)^3$ from (16), given that it is not primitive:

Number of Sequences	Least Period
2	1
2	2
4	4

Table 1: Number of sequences and least periods for $q(x)$.

For the primitive $r(x)$ in (17), a null trivial sequence is obtained and the rest of all possible sequences of maximum length are shown in the following table:

Number of Sequences	Least Period
1	1
2^{93}-1	2^{93}-1

Table 2: Number of sequences and least periods for r(x).

Thus, for the polynomial $p(x)$:

Number of Sequences	Least Period
2	1
2	2
4	4

$2*(2^{93}-1)$	$2^{93}-1$
$2*(2^{93}-1)$	$2*(2^{93}-1)$
$4*(2^{93}-1)$	$4*(2^{93}-1)$

Table 3: Number of sequences and least periods for *p(x)*.

It can be observed that there are 8 sequences with short periods (of length 1, 2 and 4 bits). Hence, these sequences have been generated by weak initial states.

5. Calculating Sequences and Periods of the Linear Trivium

5.1 Feedforward and Feedback Functions of Linear Trivium with interleave process

The feedforward and feedback functions defining the linear Trivium -i.e., the sub-generator and the interleave process- are:

$$f^*_i(x) = \begin{cases} f^*_1(x) = 1 + x^{27} \\ f^*_2(x) = 1 + x^{15} \\ f^*_3(x) = 1 + x^{45} \end{cases} \tag{21}$$

$$g^*_i(x) = \begin{cases} g^*_1(x) = x^{93} + x^{24} \\ g^*_2(x) = x^{84} + x^{6} \\ g^*_3(x) = x^{111} + x^{24} \end{cases} \tag{22}$$

Given that the characteristic polynomial of the linear Trivium takes the form in (11) but with the functions shown in (21) and (22), the characteristic polynomial *p(x)* is:

$$p(x) = x^{288} + x^{219} + x^{210} + x^{201} + x^{141} + x^{132} + x^{123} + x^{87} + x^{72} + x^{60} + x^{54} + x^{45} + x^{42} + x^{27} + x^{15} + 1 \tag{23}$$

The polynomial is not irreducible, that is, it can be expressed as the product of four polynomials such that:

$$p(x) = q(x) * s(x) * t(x) * u(x) \tag{24}$$

$$q(x)=(x+1)^3 \tag{25}$$

$$s(x)=(x^2+x+1)^3 \tag{26}$$

$$t(x) = x^{93} + x^{90} + x^{87} + x^{86} + x^{84} + x^{83} + x^{82} + x^{81} + x^{80} + x^{79} + x^{78} + x^{77} + x^{74} + x^{72} + x^{71} + x^{70} + x^{67} + x^{65} + x^{63} + x^{62} + x^{51} + x^{44} + x^{41} + x^{38} + x^{35} + x^{34} + x^{31} + x^{29} + x^{27} + x^{25} + x^{24} + x^{21} + x^{19} + x^{17} + x^{16} + x^{15} + x^{11} + x^{9} + +x^{8} + x^{6} + x^{5} + x + 1 \quad (27)$$

$$u(x) = x^{186} + x^{180} + x^{179} + x^{175} + x^{174} + x^{173} + x^{172} + x^{167} + x^{166} + x^{164} + x^{163} + x^{162} + x^{161} + x^{160} + x^{158} + x^{157} + x^{152} + x^{151} + x^{150} + x^{149} + x^{148} + x^{147} + x^{144} + x^{142} + x^{141} + x^{140} + x^{139} + x^{138} + x^{135} + x^{130} + x^{129} + x^{127} + x^{124} + x^{121} + x^{120} + x^{116} + x^{115} + x^{113} + x^{110} + x^{109} + x^{107} + x^{101} + x^{100} + x^{98} + x^{96} + x^{95} + x^{91} + x^{90} + x^{88} + x^{86} + x^{80} + x^{78} + x^{75} + x^{74} + x^{71} + x^{70} + x^{69} + x^{66} + x^{64} + x^{61} + x^{58} + x^{53} + x^{52} + x^{50} + x^{48} + x^{46} + x^{45} + x^{44} + x^{43} + x^{42} + x^{41} + x^{40} + x^{39} + x^{38} + x^{32} + x^{31} + x^{29} + x^{26} + x^{22} + x^{21} + x^{19} + x^{17} + x^{16} + x^{9} + x^{8} + x^{7} + x^{6} + x^{5} + x^{2} + x + 1 \quad (28)$$

5.2 Linear Trivium Sequences and Periods

The characteristic polynomial *p(x)* yields different sequences and length cycles, depending on the initial states of the registers.

For $q(x) = (x+1)^3$, the same values of table 1 are obtained. For the polynomial $s(x) = (x^2 + x + 1)^3$ from (26), the following values are obtained:

Number of Sequences	Least Period
1	1
3	3
12	6
48	12

Table 4: Number of sequences and least periods for s*(x)*.

For *t(x)* is primitive:

Number of Sequences	Least Period
1	1
2^{93}-1	2^{93}-1

Table 5: Number of sequences and least periods for *t(x)*.

And, for *u(x)* irreducible but not primitive:

Number of Sequences	Least Period
1	1
2^{186} -1	$3*(2^{93}-1)$

Table 6: Number of sequences and least periods for u*(x)*.

The characteristic polynomial *p(x)* of the Linear Trivium obtained yields the values:

Number of Sequences	Least Period
2	1
2	2
6	3
4	4
54	6
444	12
2b	b
2b	2b
8a+6b+8ab	3b
4b	4b
56a+54b+56ab	6b
448a+636b+256ab	12b

Table 7: Number of sequences and least periods for *p(x)*.

Note: For clarity, values have been replaced with a = $(2^{186}-1)$ and b = $(2^{93}-1)$

Tables *3* and *7* show that the cycles of the linear Trivium Toy and the linear Trivium have the same order of magnitude, with a difference in the maximum length between them of a factor of *3*. In other words, the difference observed is linear and not exponential or of some other type, indicating that their recursion lengths or linear complexities are comparable.

In the case of the linear Trivium, note the existence of *512* short cycle sequences, among these *512*, *444* sequences producing cycles of size *12*. Thus, the existence of weak initial states can be verified.

6. Conclusion

This work shows a linear equivalence between the Linear Trivium and the Linear Trivium Toy generators. The complexity of both algorithms only differs in one linear factor and their minimum periods are both of the order of

2^{93}. In addition, the number of sequences in the Linear Trivium with short periods rises significantly in comparison to the Linear Toy, leading to a considerable increase of weak initial states.

7. Future Research

Further work shall explore AND operations in the generators, analyzing them as NLFSR [4, 5] or as the combination of linear filters (feedforward and feedback) with non-linear inputs. The authors of the stream cipher Trivium restricted their scope to linear expressions. Advancing their analysis to more complex forms seems a reasonable direction to pursue.

References

1. Castro Lechtaler, A.; Cipriano, M.; García, E.; Liporace, J.; Maiorano, A.; Malvacio, E. "Model Design for a Reduced Variant of a Trivium Type Stream Cipher" In XVIII Congreso Argentino de Ciencias de la Computación, ISBN. 978-987-23963-1-2. Pg. 1483-1491. Mar del Plata, Argentina, 2013.
2. De Canniére, C. and Preneel, B. "TRIVIUM A Stream Cipher Construction Inspired by Block Cipher Design Principles". In Workshop on Stream Ciphers Revisited (SASC2006), 2006.
3. De Canniére, C. and Preneel, B. "TRIVIUM Specifications". eSTREAM, ECRYPT Stream Cipher Project, Report. 2008.
4. Dubrova, E. "A List of Maximum-Period NLFSRs", Cryptology ePrint Archive, Report 2012/166, March 2012, http://eprint.iacr.org/2012/166
5. Dubrova, E. "A Scalable Method for Constructing Galois NLFSRs with Period $2^n - 1$ using Cross-joint Pairs". Technical Report 2011/632, Cryptology ePrint Archive, November 2011. http://eprint.iacr.org/2011/632.
6. eSTREAM: eSTREAM – The ECRYPT Stream Cipher Project: http://www.ecrypt.eu.org/stream/
7. Golomb. "Shift Register Sequences". Aegean Park Press, 1982.
8. Gong, Guang. "Theory and Applications of q-ary Interleaved Sequences". IEEE Transactions on Information Theory. Vol. 41 No. 2. March 1995.
9. Lidl R., Niederreiter H. "Introduction to Finite Fields and their Applications". Cambridge University Press, 1986.
10. Massey, J.L. "Shift-Register Synthesis and BCH Decoding". IEEE Transactions on Information Theory 15, 1969.
11. Maximov, A. and Biryukov, A. "Two Trivial Attacks on Trivium", Selected Areas in Cryptography, Lecture Notes in Computer Science, Vol.4876, Springer, 2007.
12. McDonald, C. and Pieprzyk, C. "Attacking Bivium with MiniSat", Cryptology ePrint Archive,Report 2007/040, 2007.
13. Raddum, H."Cryptanalytic Results on Trivium", eSTREAM, ECRYPT Stream Cipher Project, Report 2006/039, 2006.
14. Yun Tian, Gongliang Chen, Jianhua Li: "On the Design of Trivium". IACR Cryptology ePrint Archive 2009.

III

Innovation in Computer Science Education Workshop

Tools for discovering vocations towards Computer Science

SONIA V. RUEDA, ANDREA COHEN, TELMA DELLADIO
SEBASTIAN GOTTIFREDI, LUCIANO H. TAMARGO

Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur
Avenida Alem 1253, (B8000BCP), Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136
{svr, ac, td, sg, lt}@cs.uns.edu.ar

Abstract This work describes the tools evaluated by the Department of Computer Science and Engineering from the Universidad Nacional del Sur, in order to improve the promulgation of its careers and wake the high school students interest in Computer Science. It also presents some application alternatives based on these goals, and proposes future lines of work in terms on them.

Keywords: early vocations discovery, academic offer promulgation, virtual worlds construction, visual programming environments, computational thinking development.

1. Introduction

Over the past 10 years the Department of Computer Science and Engineering (DCIC) of the *Universidad Nacional del Sur* (UNS) has been intensively participating in activities linking the university with the high school level. One of the goals of such activities is to provide information to students that are about to finish high school and to those that are starting their universitary studies, in order to ease the transition between these two environments and help in their adaptation process. Another goal is to improve the promulgation of the careers offered by the DCIC and wake the interest of high school students in topics related to Computer Science.

Having the second goal in mind, several projects have been designed and implemented where the participants are high school and university teachers, high school students, and tutors from the DCIC. As part of these projects, the DCIC offers informative talks in schools, organizes conferences by software industry professionals aimed at presenting job opportunities, coordinates visits from students to different software enterprises, and offers a training workshop for programming contests, as well as courses for developing skills in the resolution of problems. In particular, the last activity is specifically oriented to students that are about to finish high school. In addition, the

DCIC, as every other academic department in the UNS, participates in the "*Muestra Anual de Carreras"*, an event aimed at presenting the careers offered by different academic institutions, where students from Bahía Blanca, and also from nearby cities and towns, attend.
Some projects are carried out under specific contracts signed between the DCIC and different high schools. These are mostly academic institutions where students graduate as Bachelors in Informatics, whose intention is to stimulate students to pursue higher studies in Computer Science. In particular, those schools include programming courses in their curricula. Therefore, our proposal is to incorporate the use of new tools for generating motivating activities for the students while having a positive impact on their learning process.
The DCIC also organizes activities with schools that do not have an Informatics orientation. In most cases, their curricula includes some contents related to informatics, but only limited to teaching the students how to use application software, especially office suites. Hence, students do not learn anything about programming. Moreover, frequently, they do not know about the incumbencies of the Computer Science universitary careers nor the wide range of job opportunities associated with them. In that context, the DCIC intends to provide information about the professional activities of its graduates and generate a space where high school students can have a first approach to programming.
In all cases, the incorporation of programming activities in the schools curricula allows for the development of *computational thinking*. This capacity benefits the resolution of problems, not only in Computer Science but in any discipline related to science and technology.
There exist several tools and different strategies that can be used for introducing computational thinking in workshops and short courses, and also for teaching full courses. In general, these alternatives are not exclusionary but, in contrast, they complement each other and can be applied in sequence starting with the simple ones and moving forward to the most versatile, which require more training.
The following section characterizes computational thinking. Then, we describe the tools evaluated by the DCIC in order to contribute to the development of this capacity and improve the promulgation of its careers. Following that, we present some activities designed with the aim to achieve our goal. Finally, we comment on future lines of work.

2. Computational Thinking

Computational thinking is a concept introduced by Jeanette M. Wing in [5]. This notion seeks to characterize the resolution of any kind of problem using Computer Science tools. Having to solve a particular problem, we might ask: "How difficult is it to solve?" and "What's the best way to solve it?". Computer Science rests on solid theoretical foundations to answer such questions precisely. For instance, for determining the difficulty of a problem,

we need to account for the expressive power of the underlying architecture, *i.e.*, the computing device that will run the solution. Thus, we must consider the machine's instruction set, its resource constraints (memory, disk, etc.) and its operating environment.

According to this, computational thinking implies thinking recursively, parallel processing, interpreting code as data and data as code, making type checks, understanding both the virtues and the dangers of the different constructs, as well as their associated cost and expressive power. Moreover, computational thinking is using abstraction and decomposition when tackling a large complex task or designing a large complex system. It is having the confidence that we can safely use, modify, and influence a large complex system without needing to understand its every detail. It is judging a program not just for correctness and efficiency but for aesthetics, simplicity of design and elegance; it is planning, learning and scheduling to find solutions in the presence of uncertainty.

All the above mentioned elements can be used for resolving any kind of problem, and Computer Science provide methods and guidelines for using them adequately. To illustrate this, let us consider the following everyday examples. When a girl goes to school in the morning, she puts in her backpack the things she needs for the day; that is prefetching and catching. Similarly, when a boy loses his keys, you suggest he mentally retrace his steps until he remembers where he put them; that's backtracking. When a young man decides which line to stand in at the supermarket, he's making a performance analysis on a multi-server system.

Nowadays, it is possible to observe the influence of computational thinking in other disciplines. For instance, machine learning has transformed statistics; statistical learning is being used for large scale problems, in terms of both data size and dimension, something that was unimaginable only a few years ago. Another example is the influence of computational thinking in biology, not only observed in the exploration of vast amounts of sequence data looking for patterns, but also in the use of algorithms and data structures for representing the structure and behavior of biological agents such as proteins. Therefore, computational biology is changing the way biologists think. Similarly, computational game theory is changing the way economists think; nanocomputing, the way chemists think; and quantum computing, the way physicists think.

As a result, computational thinking can be seen as a fundamental ability that every individual, in addition to Computer Science students and graduates, should develop. Hence, it is really important that students at all echelons start working on the development of this skill [10]. Furthermore, several studies show that the influence of computational thinking in courses taken by high school and university students has increased their problem solving capabilities [6, 7] as well as their interest in Computer Science [9].

One of the most successful approaches for engaging high school students into computational thinking is the design and development of virtual worlds or interactive games [8]. While simulations of the real world must comply with the laws and principles of nature and physics, everything is possible when

creating virtual worlds. Therefore, students can create different scenarios and model the behavior of their characters limited only by their imagination [14]. The following section describes a series of tools that provide a suited environment for the development of this kind of applications, while also helping students to further develop their computational thinking skills.

3. Tools

In this section we analyze several tools that allow for the proposal of activities suitable for detecting vocations in Computer Science. In this sense, the studied tools present approaches that clearly favor the computational thinking in young students. This in turn benefits the students problem solving, project design and idea retransition capabilities. Therefore, bringing these kind of tools to the classroom provides a clear contribution, even for those students that do not aim for careers related to Informatics.
Every tool described in this section is freely available and provides constructs to teach programming concepts and to design activities for exploring problem solving tasks. Most of them can be used by 12 year old students (in some cases, even younger students), and some of them are integrated with collaborative repositories where users can share their work and activities. Almost every tool can be easily used with little knowledge, even though some of them have more complex interfaces than others. Finally, it can be pointed out that there are remarkable differences among the technical requirements of these tools.

3.1 Scratch

Scratch is a programming environment that can be used to create animations in order to develop interactive stories or videogames [1]. Animations are assembled by establishing the behavior of several objects involved in the scenario. In turn, the behavior of each object is programmed using a Logo based instruction set. Each instruction can be seen as block, and provides the interface to be connected with other "blocks". Then, these instruction blocks can be latched together to determine the order in which they will execute. The full instruction repertoire provides constructs for movements, looks, controls, data, operators, sensors, events, pencils, and sounds.
Even when the object behavior programming depends on the instruction repertoire, the first steps with this tool are really simple: there is no explicit syntax or notation to learn, and the student can immediately observe the effects of executing programs.
As mentioned before, programming in Scratch consists in latching instruction blocks. A sequence of latched blocks can be named and integrated to the block repertoire. Thus, in Scratch, programs can be imported, exported and latched together to produce more complex animations. This particular feature inspired the name of the tool, since it can be related to the "scratching" technique used by the Disc Jockeys to play and reuse music.

This tool was built as part of a 2003 project in the Media Laboratory of the MIT, whose motto was "*Imagine, Build and Share"*. Scratch is supported by the National Science Foundation, Intel Foundation, Microsoft, MacArthur Foundation, LEGO Foundation, Code-To-Learn Foundation, Google and Dell, among others. Its working environment can be installed in Windows, Mac and Linux.

3.2 Blockly

Blockly is a visual programming environment strongly influenced by Scratch, but with several distinctive features. As in Scratch, instructions are chosen from a repertoire and there is no need to learn specific syntactic notation. Blockly runs directly in a web browser, and its programs can be exported to mainstream programming languages like JavaScript or Python. It is open source and very scalable. Currently, besides its educational perspective, it has been used for several kind of projects like robotics applications.

Under the motto of "*Teach the Hour of Code in your classroom*" Code.org provides a web site with several tutorials and challenges involving Blockly. In particular, this site provides a comprehensive database with programming lesson references and testimonies from famous people about programming [2].

The first tutorial was designed to be solved in approximately one hour by a person with no previous programming experience. Every subsequent challenge takes into consideration that the previous activity was successfully resolved. Each problem is verbally presented; howerver, Blockly provides a graphical interface where the user can visualize the challenge through images.

To solve some challenges the user has to program the behavior of a given character to achieve some goals. For this, the user points which path this character should traverse and which actions should he perform alongside. The movement blocks provide primitives to advance and rotate, the action blocks allow to fill a hole or grab something from the floor, and the control blocks allow conditional and repetitive structures. Other challenges are related to shape drawing using rotation, advancing, repeating, coloring and brushing tools. At every moment it is possible to inspect the JavaScript code that Blockly generates from the latched blocks in a program.

Code.org is a NGO that in 2013 promoted a campaign to motivate the inclusion of programming lectures in primary and secondary education. The main goal of this movement is to familiarize the students with computational thinking, and to increase the number of candidates pursuing Computer Science studies. This is also motivated by a projection pointing out that by 2020 there will be thousands of empty jobs in Informatics related disciplines.

3.3 Alice

Alice provides a programming environment integrated with a 3D visual engine. It allows for the creation of animations, videogames and simulations via *drag-and-dropping* objects into a scenario and modifying their properties.

At any time, when the user is building the virtual world, Alice shows the instructions that were generated to capture the user actions. This allows to clearly visualize the relation between the objects behavior and the generated code.
This tool was conceived to be mainly used in educational environments. Thus, it provides a teacher with a considerable amount of resources that can be used to design and adapt activities to the different education levels. The official web site contains several documents and training videos, providing extensive support to newcomers and experts.
In Argentina, the *Sadosky Foundation* promotes the use of Alice in high schools in the context of the project "*Vocaciones en TIC*". This project aims at motivating young students into developing programming skills. For this purpose, it intends to use Alice to introduce concepts of object oriented programming and contribute to the development of computational thinking. The *Sadosky Foundation* offers lectures of 40 hours in order to train teachers in these regards. Even though building animations in Alice is easier than in a conventional programming language, the training needed by students and teachers using this tool to reach these goals is noteworthy. As a final remark, it is worth to mention that Alice can be installed in Windows, Mac and Linux.

3.4 Pilas Engine

Pilas Engine [12] is a videogame programming environment based in Python. This tool aims to provide a simplified development interface for young students going through their first steps into programming. The users do not need to have previous programming knowledge. Pilas is inspired in professional videogame development tools like Cocos2d or Pygame, and tool to learn the basics of computational programming like Logo. Differently from Scratch, Blockly or Alice, the users of Pilas program using Python instructions. Nevertheless, Pilas provides a simplified set of primitives, allowing the user to considerably abstract from Python details.
This tool provides instructions to add actors, modify actors, add behavior to an actor, or even to make an actor interactive. The environment allows to see how these instructions modify the scene in real-time.
Pilas is free under the LGPL license and can be used under Windows, Linux and Mac. Unlike the other tools studied in this section, Pilas software and its full documentation is in Spanish.
This tool is promoted to learn computational programming in the context of the initiative *program.ar* [11]. This initiative was motivated by the Argentinian National State to promote Computer Science among young individuals, and to provide public awareness of this discipline and its importance in the modern world.

3.5 Algodoo

Algodoo is a 2D sandbox physics simulator, where it is possible to create simulation scenes which can be used to experiment with components that are constrained by the physics laws. A scene is built using intuitive tools to draw components such as circles, triangles, polygons, gears, ropes, liquids, motors, axis, and lasers (among others). Algodoo allows the user to easily interact with these components with classical drawing tools and to configure their physical features such as speed, mass, refraction or attraction. When a scene is executed, the user can see how the components interact with each other following the physics laws. In addition, while a scene executes, the user can dynamically interact with the components, applying forces, moving them or cutting them. Thus, Algodoo provides tools to design games or physical challenges that should be resolved by an interacting user.
Therefore, similarly to the other tools presented in this section, in order to use Algodoo the user does neither need programming nor formal physics knowledge. This can be clearly seen in the firsts lessons and tutorials proposed for Algodoo, where it is possible to create interesting scenarios by only using intuitive notions and common sense. Notwithstanding this, Algodoo provides the means to create extremely complex scenarios where advanced physics concepts are used. In addition, Algodoo provides a scripting language called *Thyme,* which allows to attach specialized behavior to some components in a scene.
Algodoo was born in 2008 as part of the master studies of Emil Ernerfeldt [3], and its physics engine is based on the linear resolution model for restrictions called SPOOK [4]. Currently, Algodoo is under the direction of *Algoryx Simulation AB*. The tool can be used in Windows and Mac, and has tablet support. Algodoo has a big community where users share their simulation scenes (there are over 50.000 of them available). The main difference between Algoodo and the other tools presented in this section is that it focuses on physics and not in computational programming. Nevertheless, given its expressive power, Algodoo is widely used to develop math and problem solving capabilities.

4. Application Alternatives

As mentioned in the Introduction, the analysis performed in this work is focused on the design of activities contributing to the development of computational thinking in high school students. Clearly, students attending to schools with an Informatics orientation will develop different skills than those attending to schools teaching limited (or none) contents related to computer science.
For the latter case, the DCIC is currently offering workshops where students can program during an hour using *Blockly*, in the context of the activities proposed by *code.org*. These workshops are held at the different schools or at the DCIC's laboratories, depending on the case. Then, after attending the

workshop, students motivated with programming can continue using the tool autonomously in order to overcome more complex challenges. In addition, they can experiment with similar tools like Scratch, or participate in the training workshop for programming contests that the DCIC has been offering for the past three years.

For those schools with an Informatics orientation, teachers from the DCIC are offering workshops for high school teachers in order to update the syllabus of their Programming courses. In should be noted that the impact in these cases is higher, because high school teachers propagate the knowledge on the use of the tools to their students.

Each workshop is divided into two stages; the first one focuses on the presentation of a specific tool, proposing concrete activities to be implemented in the classroom. The second stage requires a more active participation of the high school teachers, and it is expected that they share their experiences and results.

In this context, it is possible to choose between a Pilas Engine workshop or an Algodoo workshop; Mathematics and Physics teachers are also encouraged to participate in the latter. Both tools can be presented incrementally. Teachers and students can begin to use them starting from previously generated scenarios, and learn how to use the primitives by modifying the existing objects. Then, as they acquire more experience, they will be able to address more ambitious challenges.

Both workshops rely on the fact that the complexity of programming is related to the simplicity of the programming language and the programming environment. When going through the first stages of learning about programming, the strictness of the language's syntax and semantics can be an obstacle. Taking this into account, the manipulation of objects in an interactive environment, as offered by the above mentioned tools, facilitates the process of building programs.

As the complexity of challenges arise, the difficulty of problems is less affected by the syntax of the language and thus, programming in an interactive way without a previous design stimulates creativity; however, this can generate bad programming habits leading to a somewhat chaotic methodology.

It is important to note that the strategies adopted when constructing virtual worlds, with a flexible goal in mind, may not be adequate when addressing other kind of problems with precisely specified requirements. If a workshop is not only oriented at giving a first approach to what programming is, but also to learn about the software development process, it does not suffice to stimulate the students creativity; it is also important to propose quality criteria and a methodology for guiding the development process following them. Therefore, it is required to incorporate some content on software engineering into the workshops, as commented in the following section.

Until now, we have not proposed any activities using Alice. Notwithstanding this, we have decided to analyze it since it is one of the tools chosen by the *Sadosky Foundation* in their "*Vocaciones en TIC"* program. As a result, we have decided to use the other tools described in this paper given the simplicity of their environments, their ability to gradually increment the level

of complexity of challenges, and the possibility of designing activities to present them sequentially.

5. Future Work

Our experiences so far have been really positive and encouraging. Our work with high schools having an orientation towards Informatics is growing yearly, enriched by new activities. Notwithstanding this, it is necessary to move further, widening our connections with other institutions from cities and towns near Bahía Blanca. Another important aspect to be considered at the management level concerns the consolidation of projects enabling to provide certificates for the teachers attending to the workshops. Also, it is crucial to perform a follow up of first-year students at the UNS after participating in adaptation activities and using the tools described in the previous sections.

Finally, from an academic point of view, as mentioned before, the tools presented in this work are well suited for teaching programming through the construction of virtual worlds. It is important to continue observing the learning process after the first approach, in order to evaluate the development of abstraction capabilities and logic reasoning. On the other hand, since Computer Science does not only involve programming, it is necessary to consider the addition of other contents that favor the computational thinking; for instance, incorporating software engineering topics as the complexity of the addressed problems increases.

Regarding high schools that do not offer an Informatics orientation, to which most students attend, the proposal of offering an hour of contact with programming (the *hour of code*) has been extremely satisfying. The students are enthusiastic when overcoming the different challenges, which are organized with an increasing level of difficulty so that they require more effort and concentration. However, it is only a first approach and, although it allows to provide the students an insight of what programming is, it is hard to discover a vocation towards Computer Science after only one hour. We expect that, after this first approach, the students will be motivated to participate in other activities organized by the DCIC with the aim to bring them closer to Computer Science.

The fulfillment of each project requires a great effort in terms of management, both for the high schools and the DCIC. Also, regardless of the orientations adopted by the different schools, it is really important to evaluate the impact of the proposed initiatives, with respect to concrete goals. For that purpose, it is necessary to reach a continuity of activities over time, which we believe is our biggest challenge.

References

1. Scratch: Programming for All Resnick M., Maloney J, Monroy-Hernandez A., Rusk N., Eastmond E., Brennan K, Millner A., Rosenbaum E., Silver J., Silverman B., Kafai Y. Communications of the ACM (Nov 2009), Vol. 52, No. 11.
2. http://www.code.org
3. Emil Ernerfeldt Master Thesis 2008, Dept. Computing Science, Umeå University, Sweden.
4. Claude Lacoursière's PhD Thesis 2007, Dept. Computing Science, Umeå University, Sweden.
5. Computational Thinking Wing J. M. Communications of the ACM (Mar 2006) Vol. 49, No. 3
6. Computational thinking for youth in practice. Lee I., Martin F., Denner J., Coulter B., Allan W., Erickson J., Malyn-Smith J., Werner L. ACM Inroads (Mar 2011) Vol. 2 Issue 1, pp 32 -37.
7. Bringing computational thinking to K-12: what is Involved and what is the role of the computer science education community? Barr V., Stephenson C. ACM Inroads (Mar 2011) Vol. 2 Issue 1, pp 48 -54.
8. Scalable game design and the development of a checklist for getting computational thinking into public schools. Repenning A., Webb D., Ioannidou A. SIGCSE '10 Proceedings of the 41st ACM technical symposium on Computer science education, pp. 265- 269.
9. A multidisciplinary approach towards computational thinking for science majors. Hambrusch S., Hoffmann C., Korb J.T., Haugan M., Hosking A.L. SIGCSE '09 Proceedings of the 40th ACM technical symposium on Computer science education, pp. 183-187.
10. Computational thinking and thinking about computing. Wing J. M. Phil. Trans. R. Soc. A 28 October 2008 vol. 366 no. 1881, pp. 3717-3725.
11. http://program.ar
12. http://pilas-engine.com.ar
13. Alice, Greenfoot, and Scratch - A Discussion. Utting I., Cooper S. Kölling M., Maloney J., Resnick M. ACM Transactions on Computing Education (Nov 2010). Vol. 10 Issue 4, Article No. 17.
14. "El rol de la simulación simbólica en la Teoría Constructivista" Sonia Rueda Ateneo de Profesores, IV Congreso Argentino de Ciencias de la Computación. Neuquén-Argentina. (Octubre 1998).

Design of a Game Based on Tangible Interaction for Teaching Programming

ARTOLA VERÓNICA[1, 2], SANZ CECILIA[1], GORGA GLADYS[1], PESADO PATRICIA[1, 3]

{vartola, csanz, ggorga, ppesado}@lidi.info.unlp.edu.ar
1 Institute of Research in Computer Science III – LIDI. School of Computer Science. National University of La Plata
2 Fellow, CONICET (Type 1 Fellowship)
3 Scientific Research Agency of the Province of Buenos Aires (CICPBA)

Summary. In this article, we present an educational game that is based on tangible interaction and was designed to help students understand key concepts in introductory programming courses in Computer Science. The game, called EPIT (Programming Teaching Using Tangible Interaction), allows reflecting on and analyzing possible solutions for a problem that involves the use of various data structures, included in the curricula for these subjects. It is a collaborative game in which students work in groups of 4. They are required to go through three stages, where they need to use their cognitive skills, such as analysis, comparison and comprehension. In the second and third stages, these skills are specifically oriented to collaboration and joint reflection. We present here the background for the creation of this game, its design, and the plan for its use in the coming months. Expected results and conclusions are detailed.

Keywords: Tangible Interaction, Programming Teaching, Collaborative Work

1. Introduction

The history of the interface evolves, diversifies and specializes with time. There are currently a very wide range of interface types, and many interfaces affect interaction models. Thus, several authors talk about new interaction models that form new paradigms for the individual-computer interaction area (Bowman, 1997) (Välkkynen, 2003) (Canny, 2006) (Heim, 2007) (Oramas, 2010). Interaction based on mobile devices, interaction based on cooperative and collaborative interfaces, multi-touch interfaces, augmented reality, tangible interaction, etc. Each of these, with their limitations and advantages, adapts to different environments and scenarios (Ishii, 2006).

The use of tangible interaction in educational scenarios has been gaining significance, and it has been the focus of several research studies (O'Malley, 2004) (Price, 2008) (Marshall, 2007) (Manches, 2009) (Zufferey, 2009) (Guisen, 2011) (Sanz, 2012). Tangible interaction proposes a strong coupling between digital information and its tangible representation. Through the physical manipulation of tangible representations, the digital representation is altered. Physical shapes are used both for representing and controlling their digitales counterparts (see Fig. 1).

The interaction model used in tangibles interfaces is called *MCRpd*, for *Model-Control-Representation* (*physical and digital*). The components in this model are the same as those in *MVC* [1], but the view element, called representation, is divided in two sub-components: physical representation and digital representation. *MCRpd* shows the integration between the physical representation and the control (Ullmer, 2001).

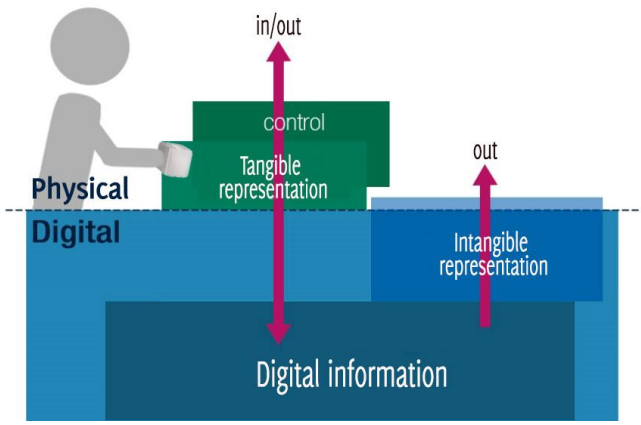


Figure 1. – Tangible interfaces interaction model (adapted from Ishii, 2006).

Koleva et al. (Koleva, 2003) differentiated various types of tangible interfaces in terms of "level of coherence", that is, they propose analyzing if the physical and digital representations are viewed as a common object that exists in the physical and digital realms or if they are seen as independent objects that are temporarily interrelated. The lowest level of coherence in their scheme corresponds to general use tools, where physical objects that are

[1] MVC (Model-View Control). Information systems architecture. It is based on three main dimensions: The *Model* corresponding to the *information*, the *View* corresponding to the *presentation* or interaction with the user, and the *Control* corresponding to the *behavior*. *(Weitzenfeld, 2004).*

used to handle any number of digital objects can be found. For example the mouse, that controls various different functions (such as menus, scrolling bars, windows, check boxes) at different times. As opposed to this, there are physical objects, such as input devices, where each object is used to carry one and only one function. The greater level of coherence (physical-digital coupling) in the scheme presented in Koleva (2003) creates the illusion that the physical and digital representations are the same object (O'Malley, 2004).

Among tangible interaction proposals, those based on active surfaces are already commonly used, be it as interactive boards or as horizontal tabletops (Marco et al, 2010).

Tabletops are computationally enhanced horizontal surfaces, also known as interactive tables. In general, to use tangible interaction on a tabletop, a number of fiducial markers are used which, when recognized, trigger certain events and actions within the application. A fiducial marker is an image that is attached to the base of a physical object and allows identifying the object through a visual detection system. These markers can provide information such as identity, position, and orientation.

The game presented in this article is based on tangible interaction on a tabletop and uses markers as method for detecting the physical objects that are being used.

2. Tangible Interaction for Programming Teaching

Throughout the years, researchers have developed various systems oriented to offering simpler environments for teaching programming, many of theme based on tangible interaction. These systems allow users to write programs by using physical objects instead of a keyboard and mouse. Some representative examples of this are Papert's Logo proposal [2] (Chakraborty, 1999) and Perlman's "Button Box" and "Slot Machine"[3] input devices

[2] Seymour Papert: (Pretoria, South Africa, 1928) Pioneer of artificial intelligence, inventor of the Logo programming language in 1968. He is considered as a renowned computer scientist, mathematician and educator. He worked with educational psychologist Jean Piaget at the University of Geneva between 1959 and 1963. http://www.papert.org/

[3] Radia Perlman: (Portsmouth, Virginia, USA, 1951) is a software designer and network engineer, expert in security, known as the Mother of the Internet. She currently works for Intel US, for which she has more than 47 patents issued.

(McNerney, 2004). Below, other projects related to the teaching of programming using tangible interaction are described.

AlgoBlocks (Suzuki and Kato, 1995) is a project that adopts the use of blocks as interaction medium. Each block has its own special semantics, and users can write programs by connecting blocks to guide a submarine through a labyrinth.

Electronic Block (Wyeth and Wyeth, 2001) (Wyeth and Purchase, 2002) uses building blocks for programming. It consists of three types of building blocks: sensor blocks as input, logic blocks to carry out logical computation, and behavior blocks as output. It is designed for pre-school children, so its syntax is simple and easy to handle.

Tern and *Quetzal* are tangible programming languages that use blocks that represent specific semantics. Users connect blocks to create programs. Once the program is created, a camera is manually operated to capture an image of the sequence of blocks. This image is transferred to a computer to identify the program that has been developed and use it to control virtual functions or a real robot (Horn and Jacob, 2007) (Horn et al, 2008) (Horn et al, 2009).

Wang et al. (2011) have created *T-Maze,* then *TanPro-Kit* (Wang, 2013), and also *E-blocks* (Wang, 2013). All of these are tangible programming systems that use blocks that, when connected, result in real-time movements within a labyrinth. They work with different types of blocks – start block, end block, direction block, and sensor block. When blocks are connected, the corresponding result is shown both on the computer screen as well as in the tangible blocks themselves as feedback to indicate if the action carried out is correct.

Dialando (Smith, 2010) is yet another tangible programming system that uses Arduino controllers connected to a notebook via Bluetooth. *Dialando*'s interface design is based on the rotation of a disc. The disc represents the direction in which an avatar on the screen will move.

Toque (Tarkan, 2010) allows creating programs by using the *Wiimote*[4] and *Nunchuck*[5] controllers as input devices. Programs are based on recipes with chef animations that are shown on the screen and are dynamically controlled. It uses programming concepts such as events, objects, primitives, functions, arguments and loops.

All these systems are aimed at providing students with a road to move forward, in layers, towards increasingly authentic programming environments.

3. Design of a Game Based on TI for Teaching Programming

The game on which this paper focuses has been planned from the experience of the authors in the area of tangible interaction and education. In Artola (2013), the development of ITCol (Tangible Interaction for Collaboration) was presented, which is a detective game based on TI and focused on experiencing collaborative work in the context of a post-graduate course of the School of Computer Science of the National University of La Plata (UNLP). Work was assessed by means of a process that involved validating game dynamics, the application itself, the interaction model and the usefulness of the application based on the educational objective proposed. Many sessions were carried out, results were analyzed, and the following was observed:

- There was a collaborative process that took place, and it benefited from tangible interaction through the use of the tabletop. This tabletop did not disperse nor distract the attention of participants; they tackled their task in a natural manner where technology was not a barrier or distracting element, but rather an integration element that accompanied them throughout the task.
- Working with real objects and the computer application was motivational for the students.

[4] Wiimote (Wii Remote) is the main controller of Nintendo's Wii gaming console. It can detect movements in the space and point to objects on the screen.

[5] Nunchuk is an expansion for Wii's wireless controller. Its name comes from the Nunchaku, a martial arts weapon, because when connected to the Wiimote, the resulting device looks like it.

- The distribution of the participants around a table favored collaboration and dialogue.
- The cognitive skills required for the proposed methodology were those necessary to tackle a collaborative process (communication, joint analysis, negotiation, etc.).

Based on the results obtained, the use of the template developed for ITCOL was considered as basis for the development of other educational games oriented to the teaching of various topics. In particular, the game developed here reuses the ITCOL template by adapting it to its own educational objectives.

3.1. Motivation for Creating EPIT

Students taking the subjects Algorithms, Data and Programs and Programming I, corresponding to the first year of studies in Computer Science at the School of Computer Science of the UNLP, have difficulty in understanding and applying the most basic data structures, such as arrays and simple lists), that are taught in those courses. In particular, the implementation of programs that use operations on these structures are rather complex for students, since these are their first steps in the development of algorithmic solutions. In these courses, students must learn to solve problems using a computer. First, the focus is for them to be able to master the use of control structures, and then they are required to be able to represent real-world objects through various data types. Ultimately, the end goal is that students can build comprehensive solutions using a programming language, where they must select relevant control and data structures to represent the different data involved in the problem, and they must be able to perform operations on them.

EPIT is aimed at helping students practice the analysis of solutions in a collaborative fashion. It proposes the exercise of discussing and negotiating which solution, among several of them, is the correct one, and why. It tests the level of understanding of the students in relation to the topics taught in the course, as well as their analysis and collaboration skills, so that the group can find the solution to the problem at hand. In the following section, EPIT functionalities are discussed in detail.

3.2. EPIT Functionality

EPIT is a computer application with the dynamics of a game that allows the participation of a group of students and is operated on a tabletop, which is the interaction space.

Given the size of the tabletop that is available to play the game and its interaction surface, group work is organized for groups with 3-4 members.

The game is organized in three stages, each of them with a specific goal. These stages are described below, as well as their relation to the educational objective.

Stage 1: during this stage, participants interact at the table one at a time. Each member of the group receives an object that represents a module (procedure or function) that carries out a specific task. When the object is placed on the table, EPIT shows a description of the objective for the module and a number of possible implementations for it. The student must choose which of the codes is appropriate for the objective described.

After all players have selected the code that they consider to be appropriate for the module they received, they can move forward to the next stage.

There is a time limit for the stage. If this limit is reached and the team has not finished yet, they can request additional time; doing so will degrade the performance of the team.

Stage 2: the team gathers around the table. The application presents a problem to be solved.

The objective of this stage is selecting a solution to the general problem from a set of possible solutions. It should be noted that only one of the solutions is the correct one; all others have errors of various types. Additionally, the team must make decisions in relation to the modules that were individually selected during the first stage, in order to negotiate and come to a consensus to determine if such selections were appropriate or if there are errors in the modules. This is aimed at adding a social component to the learning process and encouraging students to make decisions together, as a team, so that all players can discuss and analyze the topics involved through communication.

During this stage, interaction occurs through 3 types of physical objects:

- Solution Objects:

These objects represent main programs. When they are placed on the tabletop, a program code is shown. The team must select the solution object that represents the correct solution (see Fig. 2).

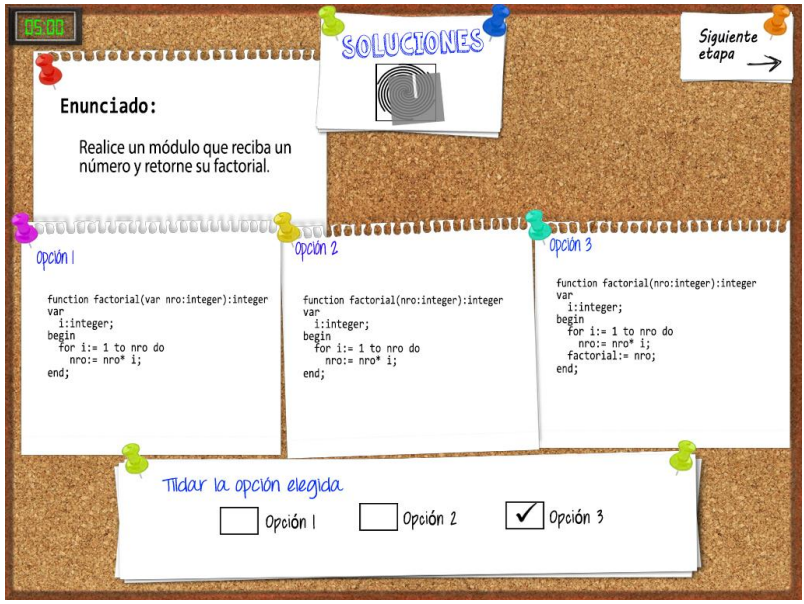


Figure 2. Example of the screen when an object representing a module is placed on the tabletop.

- Error Objects:

This type of objects represents error categories that may have a code (e.g., errors when selecting control structures, errors when passing parameters, logical errors, etc.). These objects gain meaning when used in combination with solution objects. If the students decide to place on the tabletop both a solution object and an error object, EPIT shows information/suggestions related to the errors corresponding to the category selected and the solution in question (represented by the object that was placed on the tabletop). If there is no relation between the solution and the error objects, the application displays this information.

- Module Objects:

These are the objects used in the first stage. This time, when they are placed on the table, only the code selected by the student as the appropriate code is shown. They are used as a reminder for the team to be able to know the individual selections of each of its members.

During this stage, the students are required to make decisions in relation to the following: selecting the implementation for each module from Stage 1, and selecting the implementation of a general solution that uses

those modules. To this end, they must discuss the tasks they carried out individually, and compare and analyze the different solutions to the general problem. If the team suspects that there might be an error in the solutions, they can decide to relate the code to an error category, using the corresponding objects mentioned above (by placing them on the tabletop). However, there is a set number of relations allowed, and each interaction of this type degrades team performance.

As in Stage 1, there is a time limit to complete the task, i.e., to select the correct implementation for the problem and decide if the modules that it uses (selected by each student individually) are correct or not.

Stage 3:

During this stage, the students, as a group, define their answers to the problem presented in Stage 2. They must place the solution object (the one they consider to be the correct solution to the problem) on the tabletop, as well as those modules they consider to have some error (see Fig. 3)

Figure 3. Image showing the screen corresponding to Stage 3. Interaction areas are marked with a spiral inside a square, which is the icon used throughout the application.

3.3. Specific Case for the Game

Using the game dynamics described before, a specific case was created. It should be noted that this practical case can be changed in order to use the game to solve different problems. In particular, the problem selected was one that required students to use simple lists and vectors and apply operations for creating lists (always adding the new module at the end), removing the

occurrence of certain values from the list, inserting elements in a vector, and removing the occurrence of certain values from a vector. The problem was presented as a selection process to award the best players in a local soccer tournament. To do so, there is information about the players and the goals they scored in each game.

3.4. Technical Aspects for Implementing the Game

The prototype for the application prototype was developed on *Adobe Flash Builder* 4.5 and *ActionScript 3.0* (*AS3*), with *Adobe AIR* as run environment. *GIMP* was used for graphics.

For fiducial marker detection, the *ReacTIVision* library (Kaltenbrunner and Bencina, 2007) was used. *ReacTIVision* communicates through the *UDP* port using the *TUIO* protocol (Kaltenbrunner et al., 2005).

For the configuration of the game, an *XML* file was used whose schema was designed specially to allow the generation of new activities.

A tabletop called *VisionAR* is used. For the framework of this tabletop, Medium Density Fiberboard (MDF) and acrylic were used. Its design facilitates assembly/disassembly operations, which is useful for moving the tabletop from one location to another. Figure 4 shows the parts of the *VisionAr* framework, which is based on the *NikVision*[6] model.

a) b) c)

Figure 4- a) VisionAr framework b) VsionAR with its projector, which is on an adjustable stand, and an inner mirror c) in operation.

[6] NikVision is a tabletop developed by the group GIGA AffectiveLab of the University of Zaragoza, Spain. NikVision was specially designed to be used by young children. Interaction with NIKVision occurs through the manipulation of conventional toys placed on the tabletop surface.

4. Expected Results and Study Case Planning

EPIT will be used in the context of a study case that will help validate it. This study case will be part of the course Programming I mentioned above, and it will be implemented in October this year. Students will be invited to participate in this activity, which will be organized in work sessions with 4 students each. The goal of the activity will be the integration of the topics studied before as a general review before mid-term examinations for the course. There is a study in relation to frequent mistakes made by students when implementing the operations involved in the problem used for the study case. Thus, for each module that will be individually assigned to the students, the correct solution will be presented together with distractor solutions that include the errors described in that study. The same will be done for the general solution to the problem, which will have to be selected by the team of students after discussing or reflecting upon the options presented.

During the sessions, there will be two observers that will record the process through a semi-structured form. Aspects pertaining to the following areas will be analyzed:

Contents Domain: errors that students can detect and those they cannot, reasoning proposed, analysis capabilities present in relation to the problem, ability to read code and understand the problem being solved, and so forth.

Collaborative Process: progression of the process in general and in relation to communication, ability to reach consensus and negotiate solutions, joint decision making, strategies to reach the objective, organization, etc.

Application and Proposed Interaction Methodology: the operation of the application will be analyzed in terms of usability and its tangible interaction interface on the tabletop. Even though there are already several results in relation to the use of tangible interaction on tabletops from previous work carried out by the authors (Sanz, 2012), (Baldassarri, 2011), (Artola, 2013), the goal is to expand the studied sample using this population of university students. Additionally, specific aspects of EPIT will be considered, for example, in relation to the physical objects that will be used for the application and their link to the events they produce.

5. Conclusions and Future Work

In this paper, we have presented the design for a game called EPIT that is based on tangible interaction on a tabletop. It is a collaborative game whose

general purpose is facilitating the teaching of programming. In particular, a problem related to the use of data structures such as vectors and simple lists has been posed to analyze a study case in the subject Programming I of the School of Computer Science of the UNLP.

It is our belief that this game is a positive contribution to the educational scenario due to several aspects:

a. Students of the first year of the career do not have experience in working together to find solutions to computer science-related problems. Since they are just starting their road in professional training, we think it is important to encourage them to carry out this type of activities to strengthen their analytical capabilities, understand other points of view or reasoning paths, express their ideas in front of others, and identify correct solutions and errors.
b. Approaching the topics taught in this course through a game allows working on the contents with a strategy that is different from those traditionally used in these courses. It is expected to help students understand the topics covered by EPIT, and to work as a supplement to traditional practices carried out.
c. Educators can observe their students' reasoning process in more detail and detect errors in concepts or practices in relation to the problem being solved. Therefore, group communication will be observed and recorded for each group of students, as well as their comments, difficulties, success and errors.
d. Motivation can encourage computer science students to use novelty applications or applications that are out of their reach, such as the use of tangible interaction on a tabletop. Thus, their motivation as future professional in the area is expected to strengthen, so that they can notice and get interested in research projects that are being carried out at the School.
e. Closer connection with their educators from a scenario that is not that of the traditional dialogue-based expositional classroom, so that better communication is achieved as well as the possibility of exposing their difficulties in relation to the topics being taught.

References

1. Artola V., Sanz, C. y Giacomantone J. (2013). *Interacción tangible en aplicaciones educativas. Diseño e implementación de un prototipo basado en este paradigma de interacción orientado al aprendizaje*

colaborativo. Tesis de Licenciatura en Sistemas, Universidad Nacional de La Plata, Argentina.

2. Baldassarri S., Marco J., Sanz C., Guisen M.A., De Giusti A.E., Cerezo E. (2011) Interacción Tangible para Desarrollar Competencias Comunicacionales en Educación Especial. XII Congreso Internacional Interacción Persona-Ordenador, ISBN: 978-84-9281-234-9, pp. 341-344. Lisbon, Portugal.
3. Bowman D., Hodges, L. F. (1997) "An Evaluation of Techniques for Grabbing and Manipulating Remote Objects in Immersive Virtual Environments". Proceedings of 1997 pp 35–38. Symp. On Interactive 3D Graphics. Providence, RI, USA.
4. Canny J. (2006) "The Future of Human Computer Interaction". ACM Queue, Vol. 4 No. 6. pp24-32.
5. Chakraborty, A., Graebner, R., & Stocky, T. (1999). LOGO.
6. Guisen A., Baldasarri S., Sanz C., Marco J., De Giusti A., Cerezo E. (2011). "Herramienta de apoyo basada en Interacción Tangible para el desarrollo de competencias comunicacionales en usuarios de CAA". VI Congreso Iberoamericano de Tecnologías de Apoyo a la Discapacidad (IBERDISCAP 2011). pp. 301-308. V.1. Palma de Mallorca, Spain.
7. Heim S. (2007) "The Resonant Interface: HCI Foundations for Interaction Design." ISBN-13: 978-0321375964
8. Horn, M.S., y Jacob, R.J.K., (2007). Designing Tangible Programming Languages for Classroom Use. Proceedings of TEI'07 First International Conference on Tangible and Embedded Interaction.
9. Horn, M. S., Solovey, E. T., & Jacob, R. J. (2008). Tangible programming and informal science learning: making TUIs work for museums. InProceedings of the 7th international conference on Interaction design and children (pp. 194-201). ACM.
10. Horn, M. S., Solovey, E. T., Crouser, R. J., y Jacob, R. J. (2009). Comparing the use of tangible and graphical programming languages for informal science education. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 975-984). ACM.
11. Ishii H. (2006). Tangible User Interfaces. CHI 2006 Workshop.
12. Kaltenbrunner M., Bencina R. (2007) "ReacTIVision: a computer-vision framework for table-based tangible interaction". En TEI '07: Proceedings of the 1st international conference on Tangible and embedded interaction, pages 69-74, New York, NY, USA. ACM.
13. Kaltenbrunner, M., Bovermann, T., Bencina, R., Costanza, E. (2005) "TUIO - A Protocol for Table-Top Tangible User Interfaces". En Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, Vannes, France.
14. Koleva B., Benford S., Kher Hui N., Rodden T. (2003). "A Framework for Tangible User Interfaces". Physical Interaction (PI03) - Workshop on

Real World User Interfaces, a workshop at the Mobile HCI Conference 2003. Udine, Italy.

15. Manches A., O'Malley, C., Benford, S. (2009). The role of physical representations in solving number problems: A comparison of young children's use of physical and virtual materials. Computers & Education 54, 622-640.
16. Marco, J., Baldassarri, S. & Cerezo, E. (2010). Bridging the Gap between Children and Tabletop Designers. In Proceedings of the 9th international Conference on interaction Design and Children. Spain: IDC '10. p. 98-107.
17. Marshall, P. (2007). Do tangible interfaces enhance learning?. TEI'07. Baton Rouge, LA, USA
18. McNerney, T. S. (2004). From turtles to Tangible Programming Bricks: explorations in physical language design. *Personal and Ubiquitous Computing*,*8*(5), 326-337.
19. O'Malley C. (2004). "Literature Review in Learning with Tangible Technologies". NESTA Futurelab.
20. Oramas Mogrovejo J. A. (2010) "Diseño e implementación de un sistema para la manipulación de objetos virtuales por medio de un lenguaje dactilológico". Accessed at: http://www.dspace.espol.edu.ec/handle/ 123456789/8689 (Accessed in 2013).
21. Price S. (2008). A representation approach to conceptualizing tangible learning environments. TEI'08, Bonn, Germany.
22. Sanz C., Baldassarri S., Guisen A., Marco J., Cerezo E., De Giusti A. (2012) ACoTI: herramienta de interacción tangible para el desarrollo de competencias comunicacionales en usuarios de comunicación alternativa. Primeros resultados de su evaluación. VII Congreso de Tecnología en Educación y Educación en Tecnología. TE&ET. ISBN: 978-987-28186-0-9. Pág.226-233. Junín/Pergamino, Buenos Aires, Argentina
23. Smith, A. C. (2010). Dialando: Tangible programming for the novice with Scratch, Processing and Arduino.
24. Suzuki H. & Kato H. (1995) **Interaction-level support for collaborative learning: AlgoBlock—an open programming language**. Proceeding CSCL '95 The first international conference on Computer support for collaborative learning. pp 349-355. L. Erlbaum Associates Inc. Hillsdale, NJ, USA
25. Tarkan S., Sazawal V., Druin A., Golub E., Bonsignore E.M., Greg W., Atrash Z. 2010. Toque: Designing a Cooking- Based Programming Language For and With Children. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (The Atlanta, Georgia, USA, April 10 - 15, 2010). CHI '10. 2417-2436

26. Ullmer B., Ishii H. (2001). "Emerging frameworks for tangible user interfaces", in Carroll, J. ed. Human Computer Interaction in the New Millenium, Allison. Wesley, 579-601.
27. Välkkynen P., Korhonen I. et al. (2003) "A user interaction paradigm for physical browsing and near-object control based on tags". Physical Interaction (PI03) - Workshop on Real World User Interfaces", collocated with Mobile HCI Conference 2003. Udine (Italy).
28. Wang, D.L., Zhang, C., Wang, H.A. (2011). T-Maze: a tangible programming tool for children. In Proceedings of the 10th International Conference on Interaction Design and Children (Ann Arbor, USA, June 20-23, 2011). IDC '11. ACM, New York, NY, 127-135.
29. Wang, D., Qi, Y., Zhang, Y., & Wang, T. (2013). TanPro-kit: a tangible programming tool for children. In Proceedings of the 12th International Conference on Interaction Design and Children (pp. 344-347). ACM.
30. Wang, D., Zhang, Y., & Chen, S. (2013). E-Block: A Tangible Programming Tool with Graphical Blocks. Mathematical Problems in Engineering, 2013.
31. Weitzenfeld A., (2004) Ingeniería de software orientada a objetos con UML, Java e internet. Cengage (Thomson) Learning (ISBN 970-686-190-4)
32. Wyeth P. y Wyeth G., (2001) "Electronic blocks: tangible programming elements for preschoolers," inProceedings of the International Conference on Human-Computer Interaction (INTERACT '01), pp. 496–503, 2001.
33. Wyeth P. y Purchase H. C., (2002) "Tangible programming elements for young children," in Proceedings of the Extended Abstracts on Human Factors in Computing Systems (CHI '02), pp. 774–775, ACM Press, 2002.
34. Zufferey G., Jermann P. Lucchi A., Dillenbourg, P. (2009) "TinkerSheets: Using Paper Forms to Control and Visualize Tangible Simulations" In Proceedings of the Third International Conference on Tangible and Embedded Interaction (TEI'09), pp. 377-384
35. Schweikardt, E. and Gross, M.D. The robot is the program: interacting with roBlocks. In *Proc. Conference on Tangible and Embedded Interaction TEI'08,* (2008).

T-World: a graphical, flexible and portable environment for teaching and research on intelligent agents

SERGIO BURDISSO, GUILLERMO AGUIRRE, MARCELO ERRECALDE

Universidad Nacional de San Luis, Ejército de los Andes 950,
San Luis - Argentina
sergio.burdisso@gmail.com, gaguirre@unsl.edu.ar, merreca@unsl.edu.ar

Abstract. Testbeds have proven to be useful tools for Artificial Intelligence research, particularly when we are interested in studying Intelligent Agents. A well-known testbed is *Tileworld*, which consists of a simulated robot agent and a simulated environment which is both dynamic and unpredictable. Both, the agent and the environment, are highly parameterized, enabling the user to set different scenarios and experimentally investigate how the different architectures and control strategies of the agents behave in the different environments. However, little attention has been devoted to provide implementations of *Tileworld* that take advantage of the full potential this testbed might offer as a tool for research and education. In this context, this paper introduces *T-World*, a graphical, flexible and portable platform composed by a testbed, a simulated 3D environment and a Web-based GUI. *T-World* testbed is based on several improvements of the classic *Tileworld* and provides a visually appealing cross-platform work environment in which agents can be coded in any programming language.

Keywords: testbeds, *Tileworld*, Artificial Intelligence, Intelligent Agents.

1. Introduction

The most widely accepted definition of the term "artificial agent" [6] remarks the relevance of the environment to explain what an agent is; the type of environment in which an agent acts is extremely important when we need to perform a characterization of it. It is possible to ensure that the key requirement for an agent to exist is to be situated in an environment in which to act and perceive. The term *percept* refers to the agent's perceptual inputs at any given instant. An agent's percept sequence is the complete history of everything the agent has ever perceived. In general, it is said that an agent's choice of action at any given instant depends on the entire percept sequence observed to date. Thus, it is possible to characterize an agent's behavior as an agent function that maps any given percept sequence to an action.

Controlled experimentation [2] is a popular methodology used in scientific research; it allows us to measure and study how different program

characteristics and environmental settings impact a given system performance. For instance, in the design of central processing units (CPUs), the designer could make use of tools for controlled experimentation such as benchmarks or testbeds to find the settings that lead to the best performance. Thus, if the interest is in processing speed as measure of performance, then matrix multiplication is a good benchmark. Good performance on matrix multiplication problems will allow to predict good performance on the larger class of numeric tasks for which the processor is being designed.

An early benchmark task for AI planning programs was the so-called *blocks world problem*, first described by Gerald Sussman as part of his PhD work [10]. In the problem, three blocks (labeled A, B, and C) are located on a table. The agent must stack blocks (moving one block at a time) so that C is on table, B is on C and A is on B. For instance, if the problem starts as shown in Figure 1(1), we can see that reaching

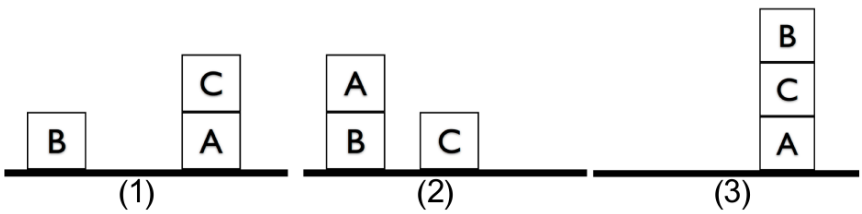


Figure 1. The blocks-world problem.

the goal state is not as simple as separating it into two sub-goals: I) get A on B and II) get B on C. If we first achieve sub-goal I (see Figure 1(2)), to be able to achieve sub-goal II the planner needs to undo the already achieved sub-goal I, and if instead the planner starts with sub-goal II the most efficient solution is to move B. But again, the planner cannot pursue sub-goal I without undoing sub-goal II (see Figure 1(3)). This simple example exposes what is known as *the Sussman anomaly*, which helped many researchers elucidate how their planners worked. It was popular because like matrix multiplication, it was representative of an important class of problems, those involving interactions among conjunctive sub-goals, and it was very easy to describe.

Researchers and teachers in the field of artificial agents have long been debating about the relevance of using these kinds of tools for controlled experimentation, which help to understand and analyze the phenomena that arise from combining different types of agents and environments. This is not a trivial task, if we take into consideration that these tools should be easy to describe and understand while also being capable of representing most of the challenges that can be found in real world problems. Additionally, such tools should allow the manipulation of different environmental properties such as dynamism, uncertainty, partial observability, etc. so that the problem to be solved by the agent enables the study a wider range of agent types, ranging

from a reactive behavior with minimum response times, to a proactive behavior with high-level reasoning skills and capabilities such as long-term planning.

A well-known testbed that seems to show many of these desired properties is the *Tileworld*, which has been widely used in research projects allowing researchers the evaluation of several agent architectures and reasoning strategies. Unfortunately, *Tileworld* as an educational and experimentation tool for students taking courses related to intelligent agents has not had the same popularity and dissemination. From our perspective, one of the reasons for this flaw has been the lack of user-friendly, graphical and interactive environments that meet the graphic standards students commonly find in videogames today. These environments should be portable, easy to use and provide support for teaching as well as all the tools for scientific research that have been traditionally used.

In this context, the main contribution of this work is to describe and introduce *T-World*, a portable, flexible and graphical platform which in its core consists of a testbed based on several improvements of the classic *Tileworld*. *T-World* was developed from scratch with the aim to be used for both, research and educational purposes.

The rest of the article is organized as follows. Section 2 describes the classic *Tileworld* testbed and Section 3 shows some experiences with the use of *Tileworld* in Artificial Intelligence research. Our platform, *T-World*, is introduced and described in Section 4 remarking how different types of environments can be implemented in this platform. Finally, in Section 5 some conclusions and opportunities of future work are outlined.

2. The Tileworld

The *Tileworld* [5] was first introduced as a testbed to experimentally evaluate agent architectures. The *Tileworld* is a chessboard-like grid on which there are *agents*, *tiles*, *obstacles*, and *holes* [4]. An agent (or robot) is able to move up, down, left, or right, one cell at a time. A tile is a unit square which behaves like a tile: it slides, and rows of tiles can be pushed by the agent. An obstacle is a group of grid cells which are immovable. A hole is a group of grid cells, each of which can be "filled in" by a tile when the tile is moved on top of the hole cell; the tile and hole cell disappear, leaving a blank cell. If a hole becomes completely filled, the agent gets points for filling it in. The agent knows ahead of time how valuable the hole is; its overall goal is to get as many points as possible by filling in holes. A *Tileworld* simulation takes place dynamically: it begins in a state which is randomly generated by the simulator according to a set of parameters, and changes continually overtime. Objects (holes, tiles, and obstacles) appear and disappear at rates determined by parameters set by the researcher, while at the same time the agent moves around and pushes tiles into holes. Among these parameters, user is able to set the frequency at which objects (holes, tiles and obstacles) can appear and

disappear; also it is possible to set whether holes have either hard timeouts or gradually decay in value. In *Tileworld* holes appear in randomly selected empty squares, and exist for a time interval known as their *life expectancy*, unless they disappear prematurely due to the agents actions. The actual time for which a hole exists is its *lifetime*. The interval between the appearance of one hole and the next is known as the *gestation period*. Each hole has a specific value, its *score*. Life-expectancies, gestation periods, and scores are taken from independent random distributions. The agent performance is measured by running a *Tileworld* simulation for a predetermined time and measuring the amount of holes successfully filled by the agent. Therefore, the agent's performance measure (utility *u*) on a given simulation, *r*, is defined as follows:

$$u(r) = \frac{\text{number of holes filled by the agent on r}}{\text{total number of holes on r}}$$

This measure gives a normalized performance measure ranging from 0 (i.e. the agent hasn't filled any hole) to 1 (the agent has successfully filled every hole). Beyond its simplicity, the *Tileworld* enables the study of important capabilities of agents, such as the ability to react to unpredictable changes and to consider new opportunities. For example, suppose an agent is pushing a tile towards a hole (Figure 2 (a)), when suddenly the hole disappears (Figure 2 (b)). At this point the agent's goal becomes meaningless and the best the agent could do is, if he is aware of the change, to reschedule its initial goal. To illustrate what "recognize opportunities" would mean in this example, suppose the agent is under the same circumstances as described in Figure 2 (a) and suddenly a hole appears on the right side as shown in Figure 2 (c). It is more likely for the agent to fill in the new hole (since he has to push the tile only one cell) rather than the old one; therefore it is probably better for the agent to change the goal to fill the new just-created hole.

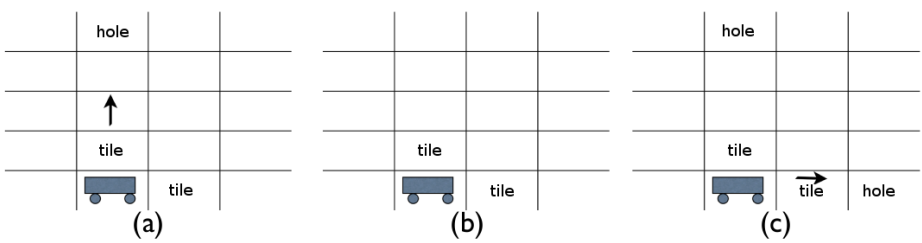


Figure 2. Consideration of opportunities

In more detail, the *Tileworld* provides a set of "knobs" which can be adjusted to control the evolution of a simulation. The main knobs are: a) *Dynamism*, the rate at which new holes appear; b) *Hostility*, the rate at which obstacles

appear; c) *Variability of utility*, difference in hole scores; d) *Variability of difficulty*, differences in hole sizes and distances from tiles; and e) *Hard/Soft bounds*, holes having either hard timeouts or gradually decaying in value. By adjusting the knobs, one can allow conditions to vary from something resembling an unconstrained football field to something like a crowded maze, or from a fixed puzzle to constantly changing chaos. For each set of parameter settings, an agent can be tested on tens or hundreds of randomly-generated runs automatically. Agents can be compared by running them on the same set of pseudo-random worlds.

3. Experiences with the use of Tileworld

In this section two research papers based on the use of *Tileworld* are briefly reviewed. These papers are of interest since they help us to highlight the manner in which experiments are carried out, how different simulation parameter settings produce an impact on the performance of agents, and how these differences can be graphically monitored by the researcher.

Evaluation of PRS on the Tileworld. In [3], Kenny and Georgeff (K&G) evaluated, among other things, feasibility of developing systems for empirical measurement of agent performance that are stable, sensitive, and capable of revealing the effect of "high-level" agent characteristics such as *commitment*. K&G establish some useful concepts that also have been included in our research, such as the already defined holes life-expectancies and gestation periods.

The experimental system was based upon the PRS real-time reasoning system [1] operating within the *Tileworld* environment. To be able to measure the agent performance over this type of dynamic environment, the experimental parameter that has the most fundamental influence upon agent effectiveness is the rate γ at which world changes. The *Tileworld* and K&G agents both measure the time by an abstract clock. They execute synchronously, with the ratio of their clock rates set by γ. Therefore, the γ value allows the dynamism of the world, as perceived by the agent, to be varied over a wide range.

K&G defined an agent's effectiveness ε to be its score divided by the maximum possible score it could have achieved. This gave a measure of performance that was largely independent of game length, and proved to be stable and reproducible. By plotting ε as a function of γ and analyzing the obtained effectiveness curves they identify three different sections: as γ increases, hole life-expectancies decrease reciprocally, as measured by the agent's clock. Eventually holes start to disappear before the agent has filled them, and ε drops below 1. Then, this decline in effectiveness has a sudden onset and is initially steep, with ε falling from 0.9 to 0.5 for a factor of 2 increase of γ. Finally, as γ increases further the decline in ε becomes more gradual and eventually asymptotically approaches zero. K&G used this relationship between these two values to characterize different agent types.

Markov decision processes and BDI models in the Tileworld. In [8], Simari and Parsons (S&P) were concerned with decision making in autonomous agents and, in particular, the tradeoff between the optimal solution provided by *Markov Decision Processes* (MDPs) and the more tractable approximation provided by the *Belief-Desire-Intention* (BDI) model. To carry out this task they developed BDI agents and a MDP model for the *Tileworld* domain. S&P adopted the simplified version of the *Tileworld* created by Martijn Schut [7]. The simplifications to the model are: tiles are omitted, so an agent can score points simply by moving to a hole; agents have perfect, zero-cost knowledge of the state of the world; and agents build correct and complete plans for visiting a single hole (they do not plan tours for visiting more than one hole).

The BDI agents for the *Tileworld* are implemented as follows: the agent's beliefs consist of its perceptions of the locations of holes in the world. Various parameters dictate how accurate these beliefs were: *accessibility* (how many positions the agent can see from where it is standing), *dynamism* (how many steps the world takes for each step of the agent), *planning cost* (how many steps the agent must spend in order to build a plan), and the agent's *intention reconsideration strategy*. This last parameter is one of the most important because it has a great influence on how efficient the agent will be; if intentions are reconsidered too often or too soon, this could lead to a waste of effort.

For an MDP model, the world is modeled by taking into account every possible action in every possible state. For the simplified *Tileworld*, this means that for a world of size n (that is, an $n \times n$ grid) there is a set of 8 actions, n^2 possible positions for the agent, and 2^{n^2} possible configurations of holes. This last number is obtained by considering that every position in the grid may contain a hole or not. A state in this world consists of a pair (P, H), where $P = (i, j)$, $0 \leq i, j \leq n - 1$, and H represents a given configuration of holes on the grid. Thus, the total number of states in this case is $n^2 2^{n^2}$. These values show that even for a very small *Tileworld*, the MDP model requires an intractable amount of resources in order to compute an optimal policy.

Comparing these approaches in a parameterized way, making use of the *Tileworld* knobs, they concluded that for small worlds, the MDP approach –effectively giving an agent a complete conditional plan– outperforms the BDI approach even if the agent has a full set of linear plans that it switches between. However, when the size of the world becomes larger, the performance of the MDP approximations becomes poorer, and the heuristic nature of the BDI planner is able to deal with the increase in world size without difficulty.

4. The *T-World* platform

The design and development of *T-World* (tworld-ai.com), within our research group, initially arose from the need for a *Tileworld* version capable of dealing

with both, education and research. Therefore, in order for *T-world* to meet these requirements, it had to be portable and flexible to allow researchers and students to use it independently of any programming languages, operating system, frameworks or libraries –that is, users should code their agent programs in any programming language and platform they want or need. Additionally, aiming to encourage students to use the tool, *T-World* had to be a visually appealing, user-friendly, graphical and interactive environment and met the graphic standards students commonly found in videogames today.

In *T-World*, simulations take place in a three dimensional environment (see Figure 3), objects and robots act and move in a realistic and continuous (not discrete) fashion; For instance, users are able to watch the reactions of robots when colliding with obstacles or pushing tiles, tiles falling and filling the holes, etc.

Figure 3. Snapshot of a *T-World* simulation

The entire *T-World* logic was completely coded in Javascript since web browsers do not natively support any other programming language. JavaScript is a high level, dynamic, untyped, and interpreted programming language. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles.

The most common use of JavaScript is to add client-side behavior to HTML pages, which is also known as Dynamic HTML (DHTML). Scripts are embedded in or included from HTML pages and interact with the Document Object Model (DOM) of the page. Some simple examples of this usage are animation of page elements or the creation of interactive content, like games. Since JavaScript code can run locally in a user's browser, the browser can respond to user actions quickly, making an application more responsive. Furthermore, JavaScript code can detect user actions that HTML alone cannot, such as individual keystrokes. In order to run JavaScript code, web browsers make use of what is known as a JavaScript engine, which is an interpreter that interprets the source code and executes the script accordingly.

4.1. Environment configuration

Given that this paper focuses on the potential use of *T-World* in Artificial Intelligence education as a teaching tool, this subsection describes the different types of environments that can be created using our tool. These types of environment are categorized along the dimensions introduced in the leading textbook in Artificial Intelligence [6]: *observability*, *number of agents*, *determinism*, *dynamism* and *knowledge of the environment*. Users are able to choose and set these parameters through GUI components such as sliders and spinners.

Fully observable vs *partially observable*: if an agent's sensors give it access to the complete state of the environment at each point in time, then it is said that the task environment is fully observable. Fully observable environments are convenient because the agent does not need to maintain any internal state to keep track of the world. An environment might be partially observable because of noisy and inaccurate sensors or because parts of the state are simply missing from the sensor data. In *T-World* a partially observable environment can be configured using two different approaches: a) *Observability bound*, number of cells a robot is allowed to see around; b) *Noise generator*, percentage of noise while perceiving obstacles, tiles and holes.

Single agent vs *multi-agent*: as discussed in [6], in some cases may be *convenient* to view a given entity as an agent, and there are other situations in which entities *must* be viewed as agents. In case an user needs a task environment in which multiple agents (robots) are going to perform in, *T-World* provides three different options: a) *competitive* multi-agent environment, every robot in the simulation has its own score and they all have to compete against each other in order to obtain the highest score; b) *cooperative* multi-agent environment, all robots in the simulation share the same score, so they all have to coordinate actions in order to maximize the score points; c) *competitive* and *cooperative* multi-agent environment; there are multiple teams of robots, all team members share the same score (i.e. each team has its own score) and different teams must compete against each other.

Deterministic vs *stochastic*: if the next state of the environment is completely determined by the current state and the action executed by the agent, then it is said the environment is *deterministic*; otherwise, it is *stochastic*. In principle, an agent need not worry about uncertainty in a fully observable, deterministic environment. If the environment is partially observable, however, then it could appear to be stochastic. Most real situations are so complex that it is impossible to keep track of all the unobserved aspects; for practical purposes, they should be treated as stochastic. It is said an environment is *uncertain* if it is not fully observable or not deterministic. In *T-World*, users are able to create stochastic environments by means of a user-defined stochastic model of actions, in which the user specify all possible outcomes of an intended action and quantify the uncertainty about outcomes in terms of probabilities; Additionally, the user can choose among a set of predefined stochastic models of actions or manually define a new one.

Static vs *dynamic*: if the environment can change while an agent is deliberating, then it is said the environment is *dynamic* for that agent; otherwise, it is *static*. Static environments are easy to deal with because the agent need not keep looking at the world while it is deciding on an action, nor need it worry about the passage of time. Dynamic environments, on the other hand, are continuously asking the agent what it wants to do; if it hasn't decided yet, that counts as deciding to do nothing. If the environment itself does not change with the passage of time but the agent's performance score does, then we say the environment is *semi-dynamic*. In *T-world* is possible to choose any of these three options. Additionally, if the task environment is static, *T-World* provides a world editor in which the user can graphically draw the initial state.

Known vs *unknown*: strictly speaking, this distinction refers not to the environment itself but to the agent's (or designer's) state of knowledge about the "laws of physics" of the environment. In a known environment, the outcomes (or outcome probabilities if the environment is stochastic) for all actions are given. Obviously, if the environment is unknown, the agent will have to learn how it works in order to make good decisions. Note that the distinction between known and unknown environments is not the same as the one between fully and partially observable environments. It is quite possible for a known environment to be partially observable. Conversely, an unknown environment can be fully observable.

4.2. Agent program

In *T-World* all agents perceive the environment in the same manner, percepts are sent as data messages formatted in either JSON, XML or Prolog facts[1] depending on what the user needs are. The percept message carries three types of information: a) external to the agent, such as the location of all

1 The JSON and XML formats are open standard data-interchange format that uses human-readable text to transmit data objects consisting of attribute–value pairs and are supported by all popular programming languages.

objects and other agents in the world, time elapsed, etc.; b) relative to the agent itself, such as energy levels, score, current location, etc.; and c) built-in knowledge, those fixed values previously set by the user to control the evolution of the simulation, such as probabilities of actions outcomes, dimensions of the grid, costs of actions, etc.

Thus, users need only to code their agents reasoning engines (also known as agent programs) in any programming language, receiving the agent's percepts message from *T-World* via a socket. Note that the fact that percepts are sent to the agent programs using a standard format, gives *T-World* the property of being flexible, since all popular programming language support JSON and XML.

5. Conclusions and future work

Taking into consideration the important role of testbeds in the study and evaluation of different types of intelligent agents and the important role we believe these tools could play in Artificial Intelligence education as teaching tools, this article proposes, introduces and briefly describes the *T-World* platform. In order to make *T-World* portable, we decided to implement it as a web page so that anyone was able to use it via Internet through a web browser (tworld-ai.com). *T-World* is flexible, agent programs can be coded in any programming language since percepts are sent using a standard format, such as JSON and XML. Finally, with the aim of encouraging students to use the tool, *T-World* was developed to be visually appealing, user-friendly, graphical and interactive platform in which simulations take place in a three dimensional world.

Among the immediate uses of the platform, we will integrate *T-World* in different projects within our research group [9]. With respect to education, we are considering the use of the platform in an “Artificial Intelligence” course of our University.

References

1. Michael P. Georgeff and Francois Felix Ingrand. Decision-making in an embedded reasoning system. In Proceedings of the 11th international joint conference on Artificial intelligence-Volume 2, pages 972-978. MKP Inc., 1989.
2. Steve Hanks, Martha E. Pollack, and Paul R. Cohen. Benchmarks, test beds, controlled experimentation, and the design of agent architectures. AI Magazine, 14(4):17-42, 1993.
3. D Kinny and M George. Commitment and efectiveness of situated agents. In IJCAI-91, pages 82-88, 1991.
4. Michael Lees. A history of the tileworld agent testbed. Technical report, School of Computer Science and Inf. Tech, University of Nottingham. UK, 2002.
5. Martha E. Pollack and Marc Ringuette. Introducing the tileworld: Experimentally evaluating agent architectures. In In Proceedings of the Eighth National Conference on Artificial Intelligence, pages 183-189, 1990.

6. Stuart J. Russell and Peter Norvig. Artificial Intelligence - A Modern Approach (3. internat. ed.). Pearson Education, 2010.
7. Martijn Schut and Michael Wooldridge. Intention reconsideration in complex environments. In Proceedings of the fourth international conference on Autonomous agents, pages 209-216. ACM, 2000.
8. Gerardo I Simari and Simon D Parsons. Markov Decision Processes and the Belief-Desire-Intention Model: Bridging the Gap for Autonomous Agents. Springer, 2011.
9. Cecilia Sosa-Toranzo, Marcelo Errecalde, and Edgardo Ferretti. On the use of agreement technologies for multi-criteria decision making within a BDI agent. In 14th Ibero-American Conference on Artificial Intelligence (IBERAMIA), November 2014.
10. Gerald Jay Sussman. A Computer Model of Skill Acquisition. PhD thesis, Artificial intelligence lab, MIT, 1973.

Esta edición de 150 ejemplares
se terminó de imprimir en Estudiocentro,
Bolívar, Buenos Aires, Argentina,
en el mes de octubre de 2015.

RedUNCI

Its objectives are:
Coordinate academic activities related to the improvement of the teachers' training as well as the curricular update and the use of shared resources to assist the development of both the Computer Sciences careers and the Technology careers in Argentina" and "To establish a cooperative framework for the development of Postgraduate activities in Computer Sciences and Technology, in order to optimize the assignation and use of the resources.

RedUNCI:
This Network was formally created through an Agreement signed in November 1996 by five National Universities (UNSL, UBA, UNLP, UNS y UNCPBA), during the second edition of CACIC. Actually 45 argentine Universities are active members of this network.

Regular Activities of the RedUNCI
Arrangement of an Annual Congress on Computer Science (CACIC) since 1995. Arrangement of an Annual Workshop for Researchers on Computer Science (WICC) since 1999. Meetings for university professors of Computer Science, for Postgraduate Dissertators and for specialists in certain areas, to promote the debate of common interest topics. Publication of *the Journal on Computer Science &Technology* by agreement with ISTEC (Iberoamerican Science and Technology Education Consortium). Annual Congress on Technology in Education and Education in Technologies (TE&ET) since 2006. Publication of the *Iberoamerican Journal of Technology in Education and Education in Technology*, since 2007.

Edulp
Editorial
de la Universidad
de La Plata

ISBN 978-987-1985-71-5
9 789871 985715