

Advances in: Artificial Intelligence, Computing Science and Computer Engineering

**Jesús Figueroa Nazuno
Alexander Gelbukh Khan
Cornelio Yáñez Márquez
Oscar Camacho Nieto
(Eds.)**

Research on Computing Science

**Advances in: Artificial Intelligence, Computing
Science and Computer Engineering**

Vol. 10

Research on Computing Science

Comité Editorial de la Serie / Series Editorial Board

Editores en Jefe / Chief Editors:

Juan Luis Díaz de León Santiago
(México)
Gerhard Ritter (Estados Unidos)
Jean Serra (Francia)
Ulises Cortés (España)

Editores Asociados / Associate Editors:

Jesús Angulo (Francia)
Oscar Camacho (México)
Jihad El-Sana (Israel)
Jesús Figueroa (México)
Alexander Gelbukh (Rusia)
Ioannis Kakadiaris (Estados Unidos)
Serguei Levachkine (Rusia)
Petros Maragos (Grecia)
Julian Padget (Reino Unido)
Miguel Torres (México)
Mateo Valero (España)
Cornelio Yáñez (México)

Coordinación Editorial: / Editorial Coordination

Luis Hernández Lara
Miguel R. Silva Millán

Diseño y Formación: / Design and Format

Leticia Andrade Trujillo
Elvia Cruz Morales
Carmen Rodríguez Aparicio

Diseño de Portada: / Design Front

Ignacio García Araoz

Research on Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 10**, octubre, 2004. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.
Editor Responsable: *Juan Luis Díaz de León Santiago, DISJ 690604*

Research on Computing Science is published by the Computing Research Center of IPN. **Volumen 10**, October, 2004. Printing 500 issues. The authors are responsible of the contents of their articles. All rights reserved. Not part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of Computing Research Center. Printed in Mexico City, October, 2004, in the IPN Graphic Workshop-Publication Office.

Volumen / Volume 10

**Advances in: Artificial Intelligence, Computing
Science and Computer Engineering**

Editores del Volumen / Volume Editors:

Jesús Figueroa Nazuno

Alexander F. Gelbukh

Cornelio Yáñez Márquez

Oscar Camacho Nieto

**Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2004**



ISBN: 970-36-0194-4

ISSN: 1665-9899

Copyright © Instituto Politécnico Nacional 2004

Copyright © by Instituto Politécnico Nacional

Instituto Politécnico Nacional (IPN)

Centro de Investigación en Computación (CIC)

Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal

Unidad Profesional “Adolfo López Mateos”, Zacatenco

07738, México D.F., México

<http://www.ipn.mx>

<http://www.cic.ipn.mx>

Impresiones / Printing: 500

Impreso en México / Printed in Mexico

Preface

Information processing is quickly converting itself into the base of our society, industry, science, and education. It is nowadays perhaps the most rapidly growing, most dynamic and influencing area of scientific research and technological development, actively penetrating in all other research fields and in virtually all areas of human activity.

Responding to the growing demand on timely information on the recent advances in computing science and related areas, we have selected for publication in this volume 43 papers by 96 authors from China, France, Ireland, Mexico, Spain, USA, and Venezuela, which present the latest developments in a wide variety of areas related to information processing and computing. The papers presented in this volume fall into three large thematic categories:

Artificial Intelligence, including such topics as pattern recognition and image processing, natural language processing, and geoprocessing;

Computing Science, including such topics as programming languages, scientific computation, networking and distributed systems, and modeling and simulation;

Computer Engineering, including such topics as digital signal processing and digital system design.

The selection of papers for publication in this volume was based on rigorous reviewing by an international team of experts in computing science and computing engineering.

This volume is the result of the work of many people. The editorial effort resulting in this volume was supported by the ACM (Association for Computing Machinery) and CyT Americas (Science and Technology of the Americas Foundation), whose active help we deeply appreciate. We would like to thank the reviewers for their hard work of selecting the best papers and helping the authors to bring them to perfection. Finally, and most importantly, we thank the authors for their great effort in laboriously obtaining and carefully describing the scientific results presented in this volume and patiently making all the changes requested by our demanding reviewers.

October 2004

Dr. Jesús Figueroa Nazuno
Dr. Alexander Gelbukh Khan
Dr. Cornelio Yáñez Márquez
Dr. Oscar Camacho Nieto

Table of Contents / Índice

	Pág./Page
Pattern Recognition and Image Processing	
Robust Enhancement for Proximal Support Vector Machines <i>Meng Zhang, Gaofeng Wang and Lihua Fu</i>	3
Camera-Based 3D Motion Tracker for Interactive Computer Graphics Animation <i>Alejandro Cornejo and Maria Elena Algorri</i>	17
Optimization of Random Sampling for Character Recognition Using Larges Binaries Strings <i>H. Jiménez Hernández and Jesús Figueroa Nazuno</i>	29
Identification of Structure-Predictability Relations in Time Series with Pattern Recognition Techniques <i>E. Bautista Thompson and Jesús Figueroa Nazuno</i>	43
Mobile System for Regular Polyhedron Recognition and Location <i>Mauricio Gabriel Orozco del Castillo</i>	55
Characterization of the Vascular Network in a Normal Human Fundus Retina <i>J. Silvano Mendoza Mendiola, Edgardo M. Felipe Riverón, Flavio A. Sánchez Garfías, Erick Morales Silva, Jorge I. Ruiz Ramírez and David Silva Leyva</i>	67
A Probabilistic Approach for Adaptive Sizing in Visual Tracking <i>Héctor Barrón, Janeth Cruz and Leopoldo Altamirano</i>	79
New Results on the Lernmatrix Properties <i>F. A. Sánchez Garfías, Juan Luis Díaz de León Santiago and Cornelio Yáñez Márquez</i>	91
Experimental Analysis of the Superposition of Information in Random Memory Spaces <i>H. Jiménez Hernández, A. Ramos Fonseca and Jesús Figueroa Nazuno</i>	103
Continuous Speech Recognition Using Loudness and RO Parameter Segmentation into Syllables Units <i>José Luis Oropeza Rodríguez and Sergio Suárez Guerra</i>	115
Natural Lenguaje Processing	
Ontology-Based Knowledge Management of Chinese Historical Official Title: an Overview of the HOTKB Project <i>Zhu Junwu and Jiang Yi</i>	127

Experiments in Text Categorization Using Term Selection by Distance to Transition Point	139
---	-----

Edgar Moyotl Hernández and Héctor Jiménez Salazar

Using Selectional Preferences for Extending a Synonymous Paraphrasing Method in Steganography	147
---	-----

Hiram Calvo and Igor A. Bolshakov

Representation of Name Sequences in Spanish using Context Free Grammar	157
--	-----

Noé Alejandro Castro Sánchez, José Ángel Vera Félix, Igor A. Bolshakov and Grigori Sidorov

Scientific Computing

Applying Supervised Clustering to Landsat MSS Images into GIS-Application	167
---	-----

M. Torres, M. Moreno, R. Quintero and G. Guzmán

Geographical Objects Representation by Means of Spatial Ontologies	177
--	-----

M. Torres, S. Levachkine, M. Moreno and R. Quintero

Optimization Heuristic to Solve Bipartition Graphs	189
--	-----

V. Landassuri Moreno and Jesús Figueroa Nazuno

Algorithm to Compute the Depth of a Basin Using 8-Connected Skeleton	201
--	-----

G. Guzmán, R. Menchaca, M. Moreno, M. Torres and R. Quintero

Extracting the Parametric Model of Duffing's Oscillator by Using a High Gain Observer	211
---	-----

Carlos Aguilar Ibáñez, Juan Mendoza C., Rafael Martínez G. and Rubén Garrido M.

On a Stabilizable Control Force for the Inverted Pendulum Cart System	221
---	-----

Carlos Aguilar Ibáñez and Octavio Gutiérrez F.

Reconstructing Equations of the Rossler's System from the y-variable	231
--	-----

Carlos Aguilar Ibáñez, Jorge Sánchez H., Fortunato Flores, A. Rafael Martínez G., Rubén Garrido M. And Miguel S. Suárez C.

Efficient Lossless Data Compression Using Advanced Search Operators and Genetic Algorithms	243
--	-----

Angel F. Kuri and Oscar Herrera

Networks and Distributed Systems

BPIMS-WS: Service-Oriented Architecture for Business Processes Management	255
---	-----

Giner Alor Hernández and José Oscar Olmedo Aguirre

General Collaboration Architecture to Work with Geographical Information Systems	265
--	-----

R. Quintero, M. Torres, M. Moreno, C. Carreto and G. Guzmán

Exploiting Features of Collaborative Social Networks in the Design of P2P Applications	277
--	-----

José Mitre and Leandro Navarro Moldes

An Approach for Facilitating Development of Web-Based Information Systems	291
---	-----

Ángel Israel Ortiz Cornejo, Heriberto Cuayáhuatl and Carlos Pérez Corona

A New Swarm Intelligence Approach for Routing Algorithms	305
--	-----

José L. Aguilar

Sub-Game Perfection in Semantic Web Services	315
--	-----

Stephen Kinsella, Juan M. Gómez, Christoph Bussler and Dieter Fensel

Modeling and Simulation

Classification of Time Series Based on their Inner Structures	331
---	-----

H. Solís Estrella, E. Bautista Thompson and Jesús Figueroa Nazuno

Evolutionary Algorithm Based on a Markov Graphical Model Selection of Promising Solutions	341
---	-----

Eunice Ponce de León, Elva Díaz and Felipe Padilla

On Numerical Simulation of Linear Acoustic and Electromagnetic Waves in Unbounded Domains	349
---	-----

Denis Filatov

Digital Signals Processing

Adaptive Multichannel Non Parametric Rank M-Type Filter to Remove Impulsive Noise from Color Images	359
---	-----

Francisco Gallegos Funes, Volodymyr Ponomaryov and Alberto Rosales Silva

Lifting Based DWT for Lossy Image/Signal Compression	373
--	-----

Oleksiy Pogrebnyak, Pablo Manrique Ramírez and Enrique Guzmán Ramírez

Interpolation with Splines and FFT in Wave Signals	387
--	-----

Luis Pastor Sánchez Fernández

Digital Systems Design

An Efficient FPGA Architecture for Block Ciphering in Third Generation Cellular Network	403
---	-----

Tomás Balderas Contreras, René A. Cumplido Parra

Predictive Array Access Cache-PAAC71	415
<i>Alejandro Villar Briones, Oscar Camacho Nieto and Luis Alfonso Villa Vargas</i>	
Instructions-Wake-Up mechanism: Power and Timing Evaluation	429
<i>Marco A. Ramírez, Adrian Cristal, Alexander V. Veidenbaum, Luis Villa and Mateo Valero</i>	
Programming Languages	
Prepare Yourself... The Tiger Hill Be Unleashed Soon	447
<i>María Lucía Barrón Estrada, Ryan Stansifer and Ramón Zatarain Cabada</i>	
Implementing a CPS Compiler for Functional Languages with Zero Overhead Exception Handling	459
<i>Ramón Zatarain Cabada, Ryan Stansifer and María Lucía Barrón Estrada</i>	
MOOL: An Object-Oriented Language with Generics and Modules	471
<i>María Lucia Barrón Estrada, Ramón Zatarain Cabada and Ryan Stansifer</i>	
AID: An Object-Relational Schema Design-Tool	483
<i>Hassan Badir, Etienne Pichat and He Xiaojun</i>	
A Preprocessor Based Parsing System	495
<i>Pradip Peter Dey, Mohammad N. Amin and Thomas M. Gatton</i>	

7

Optimization Heuristics to Solve Bipartition Graphs

V. Landassuri-Moreno & J. Figueroa-Nazuno

Centro de Investigación en Computación
Instituto Politécnico Nacional
México DF, Col. Lindavista, C.P. 07738
vickmlm@hotmail.com, jfn@cic.ipn.mx

Abstract. This paper makes an analysis and a comparison between three heuristics, which give approximations to solution in polynomial time for Optimization problems like the graph bipartition problem. The heuristics uses are: Extreme Optimization (EO) searching in solution space the variable with poorest value for a random update to optimize the problem, Extreme Optimization Distributed (EOD) that is a generalized version of EO searching in a neighborhood and Extreme Optimization Distributed with Small World (EODsw) that searches the variables with poorest value in a neighborhood applying Small World. For the analysis we used Geometric, Random and Small World graphs. The obtained results show us that EOD and EODsw are very similar in their behavior and are better than EO, but EODsw result more competitive than EOD for some classes of problems. Key Words: Optimization, correlation, Small World.

1 Introduction

The Bak-Sneppen model [1] is based over species, it associates one value, which is known as “fitness” taking values between 0 and 1; it is called value of adaptability; the species with the smallest value (poorest degree of adaptation), is selected for a random update, with the purpose of improving its adaptability. Change species adaptation values affects at his neighbors. The adaptability of the neighbors could be modify selecting another random value, warrant work only in the space of solutions; after sufficient number of iterations the system reaches a highly correlated state known as criticality self – organized. The Extreme optimization (EO) is based over the Bak-Sneppen model, only that here, the element with the smallest value, is selected for a random update without any explicative improve for it.

The Extreme Optimization Distributed model (EOD) searches the element with the poorest adaptation inside a locality, then, performs a random update. The Bak-Sneppen model takes the element with the poorest adaptation from a global set, nevertheless EOD takes the element from a locality or a neighborhood.

The Small World (SW) it is a kind of EOD because is based in a neighborhood. “You are only ever six degrees of separation away from anybody else on the planet”, Social networks model six ‘degrees of separation’ (1990) [2], explains that anybody are six persons of separation at most. Small Worlds is a generalized version of this model.

This paper is organized as follows. In section 2 we describe the techniques and the implementation of the used algorithms. In section 3 we present the Graph Bipartition Problem. In section 4 we show the result of the computer simulation of heuristics

2.4 Extreme Optimization Distributed with Small World

There are three fundamental aspects in the study of the network, which have the property to be large and sparse, according James Case [3]:

- (a) The average k over all vertices v of the number of edges incidents on v .
- (b) The average over all connected pairs of different vertices, of the length L of the shortest connecting path.
- (c) The frequency C with which three connected vertices are completely connected.

Extreme optimization distributed with Small World (EODsw) works in the same way EOD with neighborhood, the structural properties of the graphs measures for its characteristically length of path L , average over all vertexes. Moreover grouping coefficient C defined like: suppose that a vertices v has k_v neighbors; then can exist at most $k_v(k_v - 1)/2$ edges between them (this occur when all neighbors of v are connected with all) C denotes the fraction of edges allowed, existence. L is the typical separation measure between two vertex in the graph (one property global), while C is the measure typically grouping from neighborhood (one local property). The regular networks have a big C but small L while the random networks have a big L but small C . There are some networks between this class that have small L and big C . One network with this property improves performance to search because it can find better local optima because high grouping C , but at the same time have a relation with all the system, to compare between this minimum, due to small L . The small world networks used here have a lot of vertex with scarce connection, but not the sufficient to disconnect the graph. Specific is required where guarantees that the random graph are connected. Exist scarce connection, facilities the implementation in distributed ambient, because the number of message interchange are small $O(k)$.

3 The Graph Bipartition Problem

The problem of graph bipartition is a problem that we can to solve with EO. The Graph Bipartition Problem (GBP) is: there is a graph of N vertices, where N is even, have to separate the vertices in two sets of cardinality $N/2$, such that the number of edges that connect both sets "cut size" m , is small.

To approximate this problem for a solution using EO, we associate each vertex v_i one variable x_i can take two values 0, 1, indicated belong to one of two sets. The adaptation function of each variable x_i is defined as follows:

$$f_i = \frac{g_i}{g_i + b_i} \quad (3)$$

Were g_i , is the number of "good" edges that connect 'i' with others vertices inside the same set, and b_i is the number of "bad" edges that connect 'i' with vertices through

partition. For nodes not connected, we associate one $f = 1$. In order to optimize the problem, we can minimize the cut size of the graph.

4 Results of Computer Simulation

We used different type of graphs to prove the heuristics, like this: random graphs, geometric graphs and small world graphs, varying in each one the vertexes and probability or ratio with tow vertexes are connected; the implementation of small world graphs can be found in Ref [8].

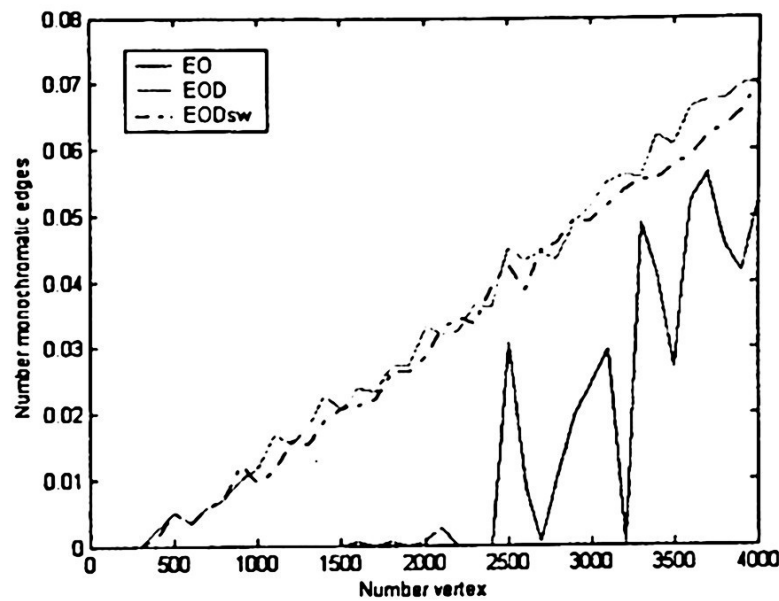


Fig. 1. Geometrics graphs (Number monochromatic edges & number vertex) with 4000 vertex and probability of 0.1

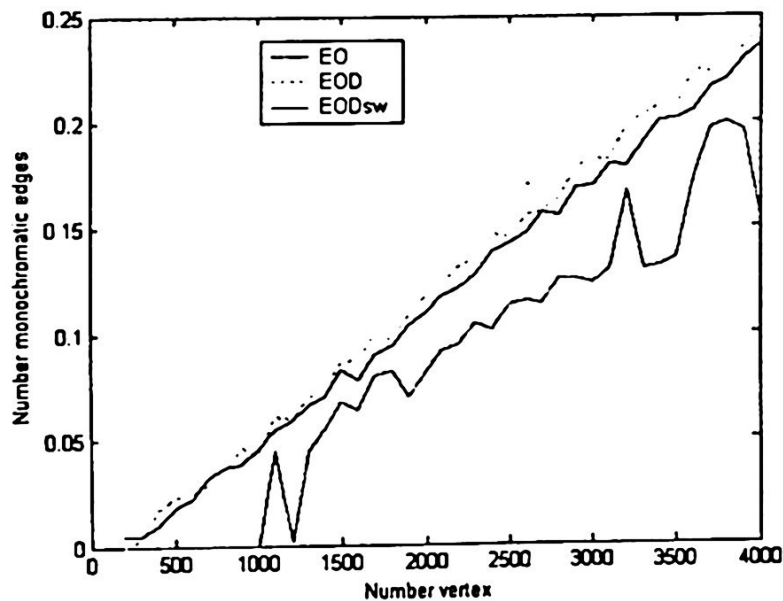


Fig. 2. Random graphs (Number monochromatic edges & number vertex) with 4000 vertex and probability of 0.001

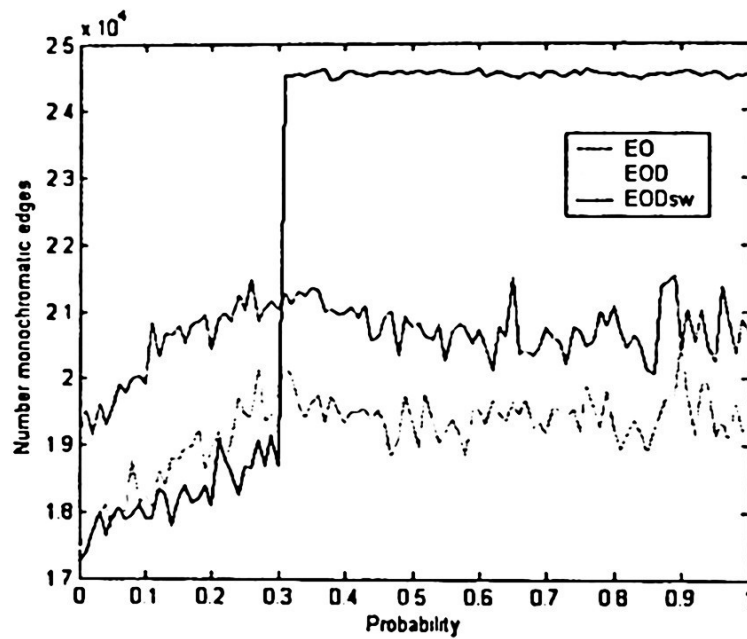


Fig. 3. Small World graph (Number monochromatic edges & Probability) with 2000 vertex

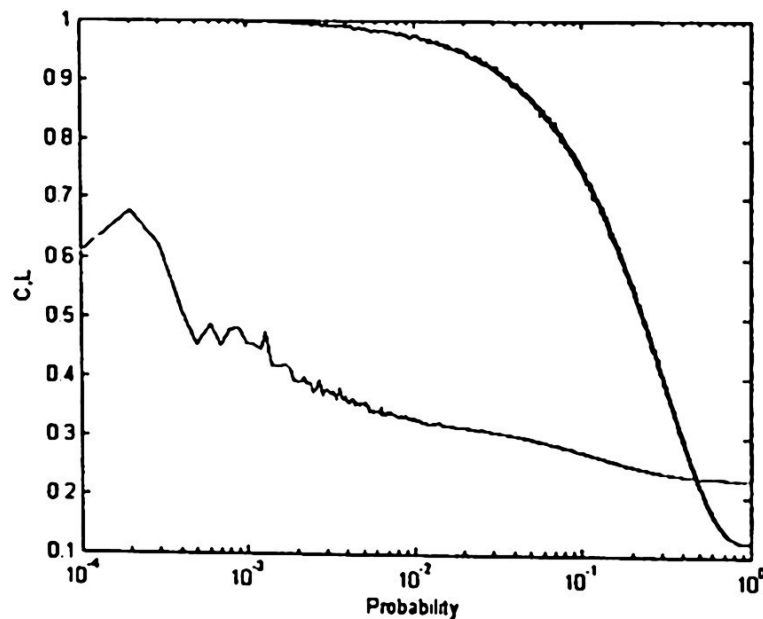


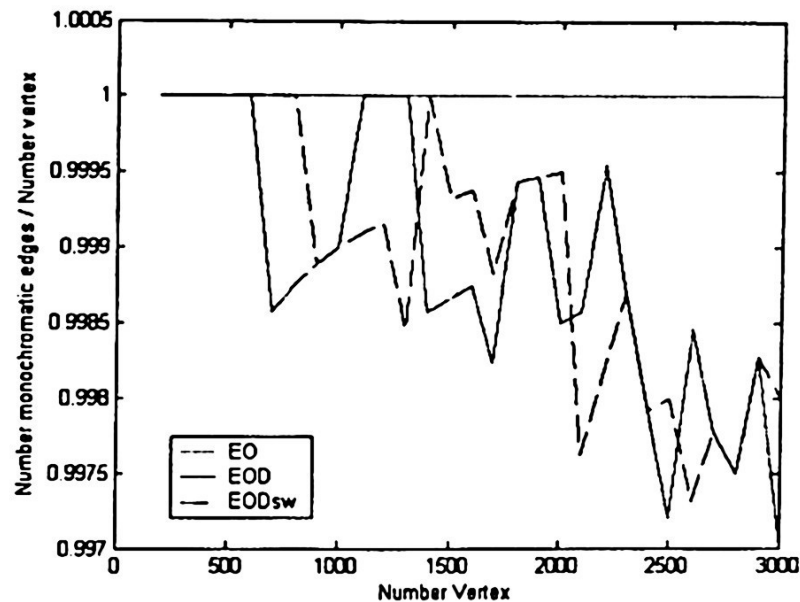
Fig. 4. Small World graph (CL & Probability) with 2000 vertex

In the graphics, it can be observed that EOD and EODsw are very similar, even in fig. 3 EODsw is better than EOD because it will work over SW graph. We can conclude for this part that EOD is better than EO and EODsw is better than EOD over one SW graph; then in normal conditions EOD and EODsw are similar. Now analysis the Bipartition Graph problem:

Table 1. Comparisons between EO, EOD and EODsw. We can see that the values between two distributions are similar, sometimes EODsw is better than EOD sometimes not, but if we follow the behavior of each algorithm with 500, 1000, 1500... we can distinguish how they are varying before 3000 vertex

Geometrics Graphs with 3000 vertex						
Prob/ Radio	Distribution 0.001			Distribution 0.01		
	EO	EOD	EODsw	EO	EOD	EODsw
0.5	1	0.9975	0.9985	0.985	0.85	0.83
0.6	1	0.997	0.998	0.985	0.845	0.835
0.7	1	0.9984	0.998	0.985	0.84	0.825
0.8	1	0.9977	0.999	0.986	0.849	0.82
0.9	1	0.999	0.9975	0.985	0.839	0.83
1.0	1	0.9984	0.999	0.985	0.85	0.83
1.1	1	0.9985	0.9975	0.986	0.845	0.83
1.2	1	0.9975	0.9975	0.986	0.845	0.83
1.3	1	0.9965	0.9974	0.986	0.8455	0.825
1.4	0.999	0.99755	0.999	0.987	0.84	0.83
1.5	0.88	0.999	0.999	0.88	0.83	0.84
1.6	1	0.9992	0.9974	0.985	0.845	0.825

As you can see, for this analysis we use a Geometric graph for the computer simulation varying the distributions of the vertex and the probability or ratio with how vertices are connected, next some representative figures from Table 1.



a) Probability 0.001

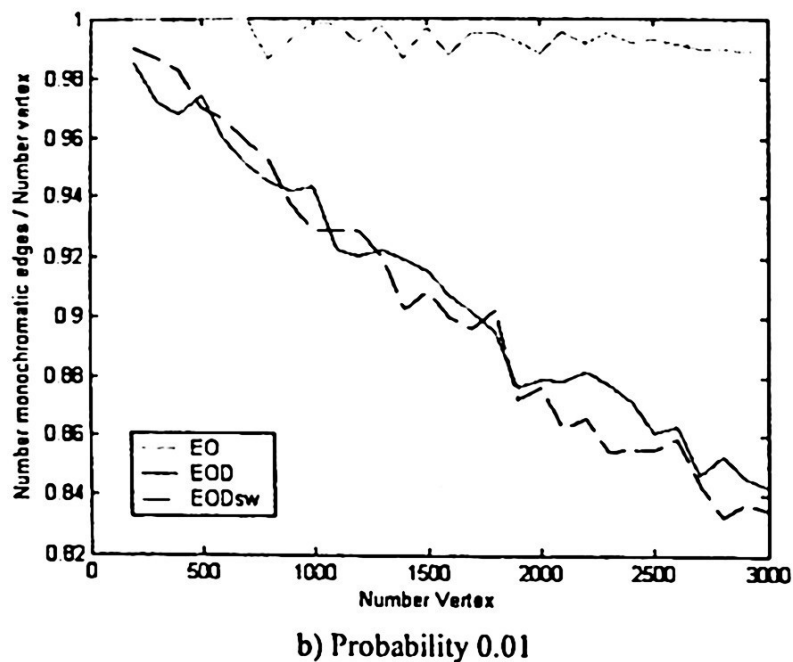
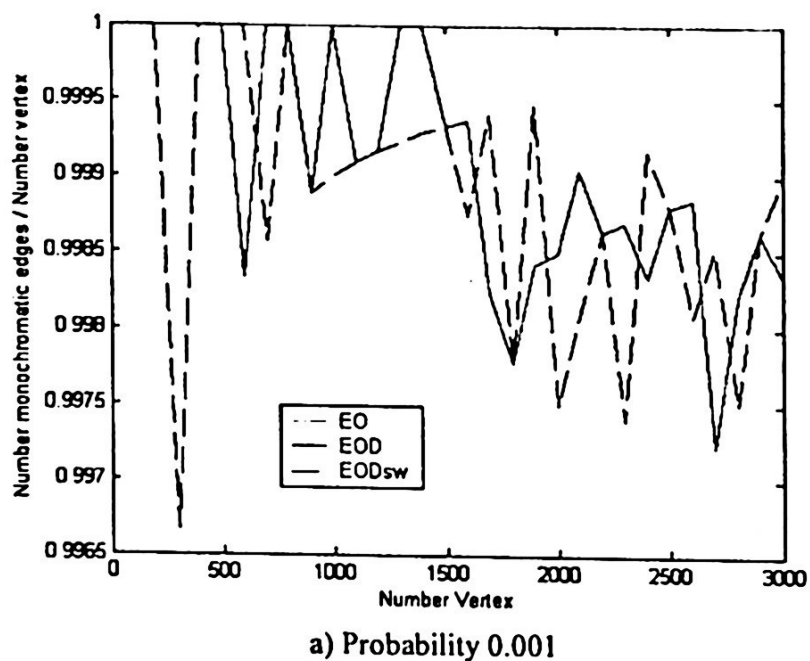
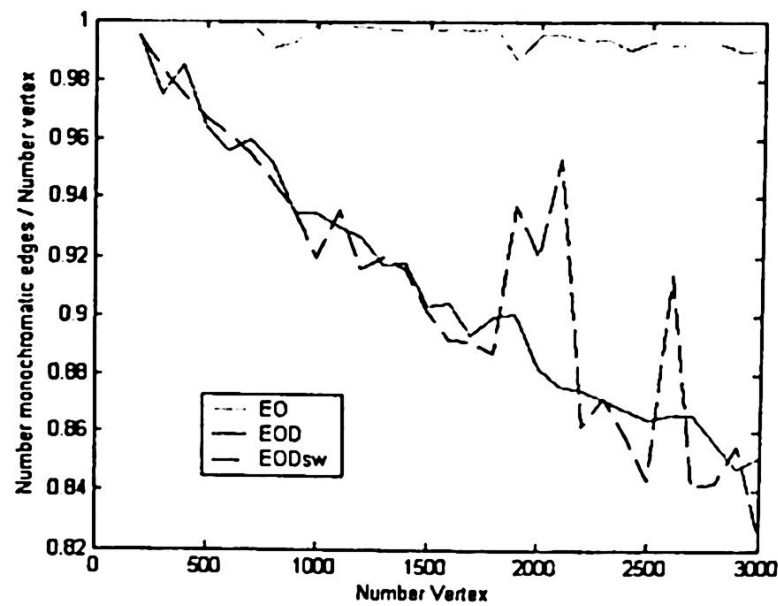


Fig. 5. Bipartition graphs problem (Number monochromatic edges / Number vertex & Number Vertex) with 3000 vertex and distribution 0.6

In fig. 5 EO don't follow the others, EOD and EODsw varying but they maintain a similar behavior.



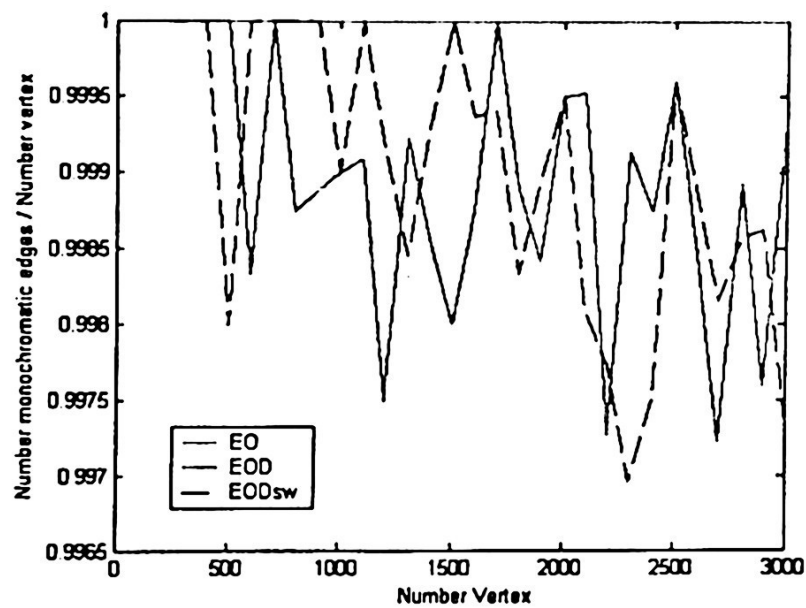


b) Probability 0.01

Fig. 6. Bipartition graphs problem (Number monochromatic edges / Number vertex) with 3000 vertex and distribution 1

This figure are very similar to fig. 5, OE remains on top of the graphic while EOD and EODsw maintain their behavior and decreases continuously. Some times EOD is better than EODsw and sometimes EODsw is better.

Next our last figure, his behavior is similar to the other figures; the figures from Table 1 that we don't graphics have a similar behavior like presente here.



a) Probability 0.001

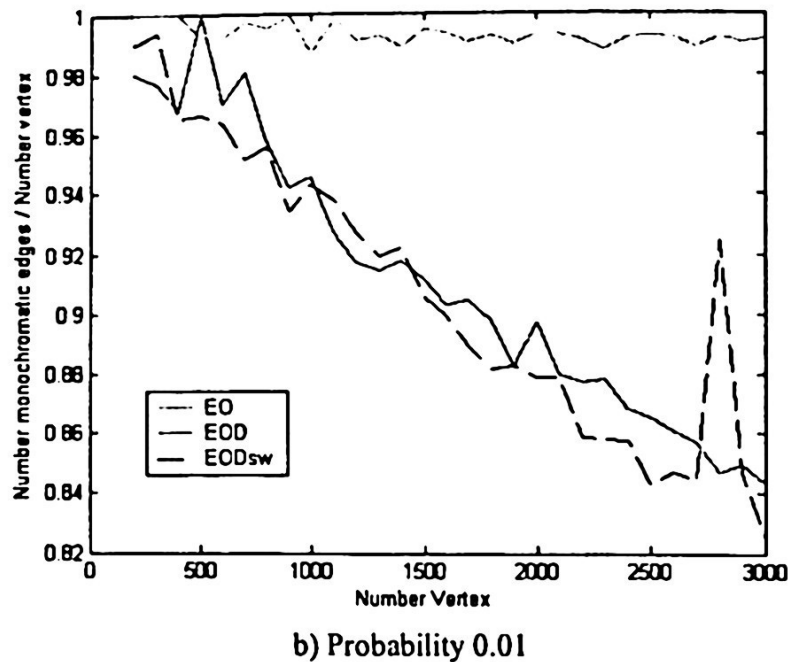


Fig. 7. Bipartition Graphs Problem (Number monochromatic edges / Number vertex) with 3000 vertex and Distribution 1.6

5 Conclusions

In this work we reviewed three algorithms; first we implemented them and made a comparison to find the best. Then we talk about the Bipartition Graphs Problem and see that sometimes EODsw is better and sometimes EOD is better. Concluding EODsw and EOD are similar to solve the Bipartition Graph Problem, we can apply the specific algorithm for our necessities like in the figure 7) a) 1300 vertex EOD is better than EODsw.

We want to do this comparative to determine that if we apply EODsw we went to obtain better results than EOD and EO, but we didn't, the results obtained tell us that both algorithms in general are better than EO, and not only in bipartition problem. If we have a SW graph, we can take advantage from EODsw, but if we are in bipartition problem we can't. To bipartition problem the used graph is not important, hence is not important the graph's distributions, because when we apply the algorithm and move the vertex to another positions in the graph, we change the topology, the only thing that import us is the connection of the vertex to another vertex, it does not matter if exist a neighborhood in the graph.

In the future works, the same problem it can be solve for three, four, five, et cetera partitions, to try to find if EODsw is better to the other, maybe it can be possible because if we have more partitions we would translate the neighborhood to the partitions.

6 References

1. Stefan Boettcher, Allon Percus "Nature's way of optimizing". *Artificial Intelligent* 119 (2000) 275-286. ELSEVIER.
2. D.J. Watts, "Small Worlds: The Dynamics of Networks Between Order and Randomness". Princeton University Press, Princeton, NJ, 1999.
3. Case, James. "The Continuing Appeal of Small World Networks". *Community Lecture* 2001. SIAM News, Vol 34, Number 9.
4. A. Broder, R. Kumar, F. Maghoul, P. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph structure of the web". *Proceedings of the 9th WWW Conference*, Amsterdam, 15-19 May, 2002, 309.
5. S. Bornholdt, H.G. Schuster. "Handbook of graphs and Networks from the Genome to the Internet". Wiley-VCH, Germany, 2003.
6. Kleinberg, Jon. "The Small-World Phenomenon: An Algorithmic Perspective". Department of Computer Science. Cornell University. Ithaca, NY.
7. Kleinberg, Jon. "Small-World Phenomena and the Dynamics of Information". Department of Computer Science. Cornell University. Ithaca, NY, 14853.
8. D. J. Watts, S. H. Strogatz "Collective dynamics of 'small-world' networks". *Nature* Vol 393, 4/June/1998, pp. 440.
9. Buchanan Marrk. "Nexus, Small Worlds and the Groundbreaking Science of Networks". W. Norton & Company. New York. 2002.

Algorithm to Compute the Depth of a Basin using 8-Connected Skeleton

G. Guzmán, R. Menchaca, M. Moreno, M. Torres and R. Quintero

Geoprocessing Laboratory (GEOLAB)

Centre for Computing Research (CIC)

Mexico City, Mexico

{jguzmanl, rmen, marcomoreno, mtorres, quintero}@cic.ipn.mx

Abstract. In this paper, we present a method to compute the depth of a basin using its 8-connected skeleton. The proposed algorithm consists of two stages: First, the preprocessing stage, which is based on transforming the source image into a binary format. The second stage is denominated quantifying that is used to find the skeleton and its variables to determine the depth of the basin. Although in most cases the algorithm provides good results, we found several pathological situations. However, these cases can be solved by means of a crossing point fusion in conjunction with morphological opening algorithms. Our approach uses as input geo- image that contains only information about the studied basin. Moreover, the image contains the entire trajectory of the basin. This algorithm can be used in diverse areas of Geocomputation; similarly, by making some modifications in the parameters to obtain the features. In addition, the algorithm can be applied to other problems related to the identification and quantification of cartographic objects.

Keywords: Image Processing, Skeleton, Mathematical Morphology, Geo-Image.

1 Introduction

Previous works [1-3], propose algorithms to extract specific information of high resolution raster images, for instance; highways, rivers, populations, among others. Usually, these algorithms (automatic or semiautomatic) are used to process geo-images. In addition, the output image only contains the pixels that compose the desired objects, denominated raster to vector algorithms. These works avoid a manual digitalization of the data. Nevertheless, in Geocomputation it is frequently required to know quantitative information about the geographic data set, hence, having data in raster or vector format is commonly not enough.

The determination of the depth of a basin is a typical example in hydrology, geomorphology and cartography; because from this datum other parameters can be obtained, e.g. flood risk, bifurcation ratio [9], water flow, etc.

We develop an algorithm to determine, the depth of a basin, following the method formulated by Horton [4]. The advantages of our technique are a minimum computing

cost and the final result is the same that obtained using the Horton method. Up-to-date there is not an automatic method or algorithm to compute this.

The paper is organized as follows: in section 2 the mathematical foundations of the algorithm are sketched out, section 3 explains in detail the proposed algorithm, in section 4 we present the obtained results, and finally in section 5 we outline the conclusions and future work of the present method.

2 Mathematical Background

In this section, we present some important concepts of digital geometry. Let $p = (x, y)$ and $q = (u, v)$ be points of Z^2 . The following metrics are commonly used:

$$\begin{aligned} d_1 &= |x - u| + |y - v| \\ d_4 &= \max(|x - u|, |y - v|) \\ d_{6R} &= \max(|x - u|, |y - v|, |x - u + y - v|) \\ d_{6L} &= \max(|x - u|, |y - v|, |x - u - y + v|) \end{aligned} \quad (1)$$

Each one of those metrics $d_k, k \in \{4, 6L, 6R, 8\}$, lead to a none reflective symmetric relation N_k defined by $(p, q) \in N_k \Leftrightarrow d_k(p, q) = 1$. The structure (Z^2, N_k) usually is named k -connected graph. We say that points p, q are k -neighbors if $(p, q) \in N_k$, and $N_k(p)$ denotes all k -neighbors of p . In this paper, we use a 8-connectivity, that is, $k = 8$.

A subset $R \subseteq Z^2$ is k -connected if for any $p, q \in R$, exists at least one sequence of points in $R, p = a_1, a_2, \dots, a_n = q$, where a_i and a_{i+1} are k -neighbors for $i = 1, \dots, n-1$. Let S the 8-connected skeleton of a region R , and p any point that satisfies the condition $p \in S$, we have:

1. If $N_8(p) = 1$, p is a terminal point, TP.
2. If $N_8(p) = 2$, p is an internal point, IP.
3. If $N_8(p) \geq 3$, p is a cross point, CP.
4. If $N_8(p) = 3$, p is a strict triad, TE.

3 Proposed Algorithm

As we stated previously, the developed method consists of two stages: one for preprocessing and another for quantification. The main purpose of the *preprocessing*

stage is to take a color (RGB values) or gray level image and generate as output a binary image using this convention: an intensity of 1 is assigned to object pixels; whereas an intensity of 0 is used for none object pixels. In the *quantification* stage, we compute two variables to determine the depth of the basin, the number of terminal points and the number of crossing points. Both stages are shown in Figure 1.

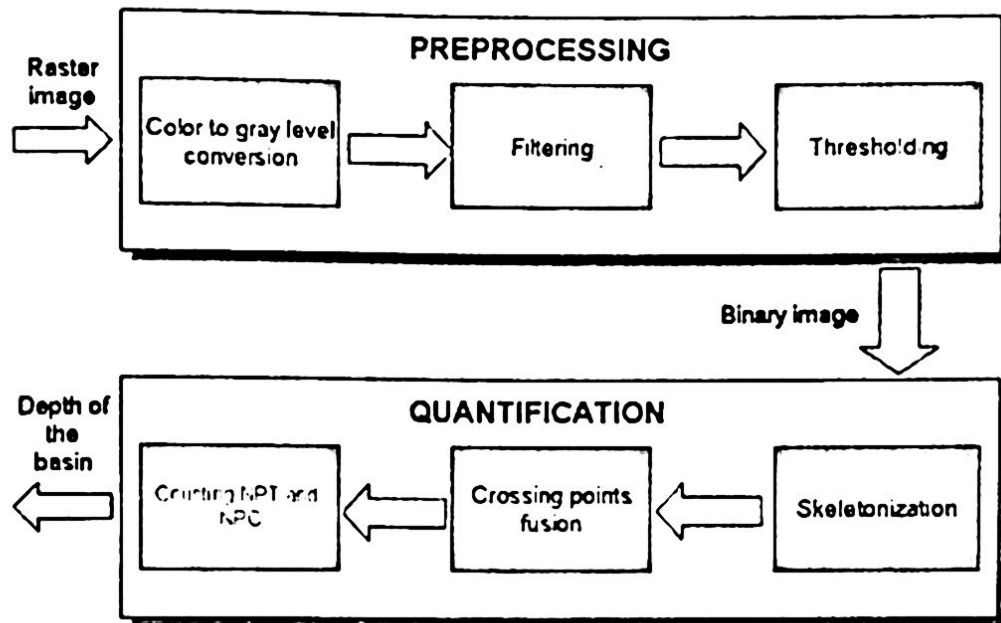


Fig. 1. Preprocessing and quantification stages

1. *Color to gray level conversion.* If the pixels are in RGB format, we apply a conversion using Equation 2 for each pixel of the original image.

$$\text{gray-level} = \frac{299 \cdot R + 587 \cdot G + 114 \cdot B}{1000} \quad (2)$$

Where R, G, B are the red, green and blue components respectively of a pixel p in an image.

We use a conversion, based on the YIQ model because the loss of descriptive information is minimal in this case. Figure 2 shows an example using this formula.

2. *Filtering.* To decrease the *probable* noise contained in the digital image, we apply a smoothing algorithm, for example, the median filter with a mask of 3 x 3 pixels. It is not required to remove all noise in the contour of the *basin* (watershed), since the performance of the skeletonization algorithm is not affected by the presence of structural noise and this algorithm guarantees removing any possible parasite branches. Also, some other technique of smoothing can be used, such as ordering or average filter [8].

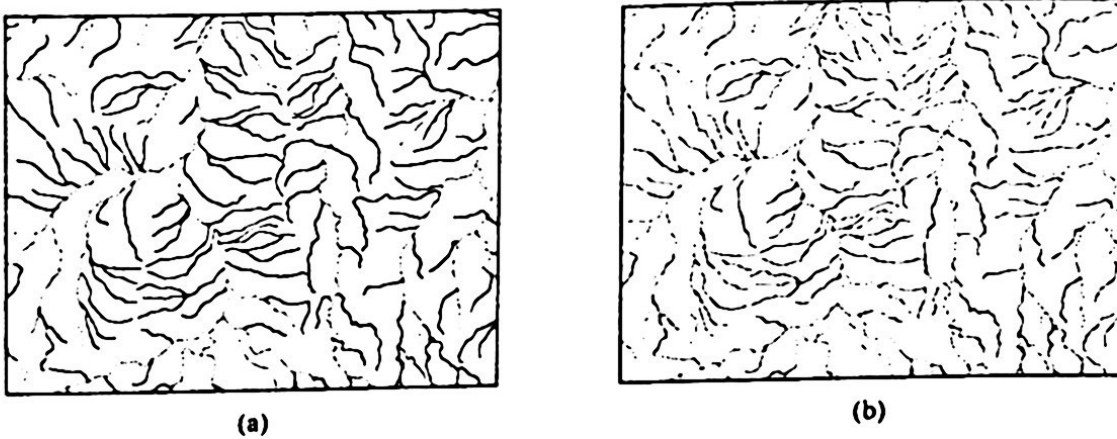


Fig. 2. (a) Input color image. (b) Gray level image generated using YIQ model

3. *Thresholding*. The purpose of this part is to accurately identify all object regions in the image. We can use one of different techniques, which generally are based on the Otsu method [6]. The selected algorithm [7] makes use of probability techniques, which allow us to choose a suitable threshold for the thresholding algorithm. The binary image that we obtained at the output of the *preprocessing* stage is used as input for the *quantification*.
4. *Skeletonization*. This process generates the 8-connected skeleton of the source image. To do this, we use the thinning algorithm developed by Díaz [5]. This method removes all contour pixels without breaking the connectivity of the region. This algorithm is based on two conditions: terminal point rule and an own set of templates that decides if a pixel can be or not removed from the image. Basically, this algorithm consists of the next steps:
 - Let R be a region, compute the contour of this region using the next criterion: a pixel $p \in R$ is a contour pixel if at least one k' -neighbor q , where $q \notin R$, exists. It is very important to use dual metric; otherwise parasites braches or ruptures in the connectivity may exist in the final skeleton. As we need an 8-connected skeleton the dual metric used is 4-connected.
 - Using the template shown on Figure 3, each contour pixel is examined to determine if it can be removed or not. If the pixel and its neighbors don't match any of the templates we proceed to eliminate it.
 - If no pixel was removed in the previous step, the region is the desired skeleton, otherwise, return to first step.

$$A_1 = \begin{bmatrix} 1 & 0 & a \\ 0 & p & b \\ e & d & c \end{bmatrix}$$

$$A_5 = \begin{bmatrix} X & 0 & X \\ 1 & p & 1 \\ X & 0 & X \end{bmatrix}$$

$$\begin{aligned}
 A_2 &= \begin{bmatrix} e & 0 & 1 \\ d & p & 0 \\ c & b & a \end{bmatrix} & A_6 &= \begin{bmatrix} X & 1 & X \\ 0 & p & 0 \\ X & 1 & X \end{bmatrix} \\
 A_3 &= \begin{bmatrix} c & d & e \\ b & p & 0 \\ a & 0 & 1 \end{bmatrix} & A_7 &= \begin{bmatrix} 0 & f & 0 \\ i & p & g \\ 0 & h & 0 \end{bmatrix} \\
 A_4 &= \begin{bmatrix} a & b & c \\ 0 & p & d \\ 1 & 0 & e \end{bmatrix} & & a+b+c+d+e > 1 \\
 & & & f+g+h+i = 1
 \end{aligned}$$

Fig. 3. Templates set to generate 8-connected algorithm

5. *Crossing points fusion.* Due to basin topology, it is possible to find cases in which the skeleton algorithm breaks a single cross point into two cross points. This problem can be easily solved using the crossing points fusion algorithm [10].

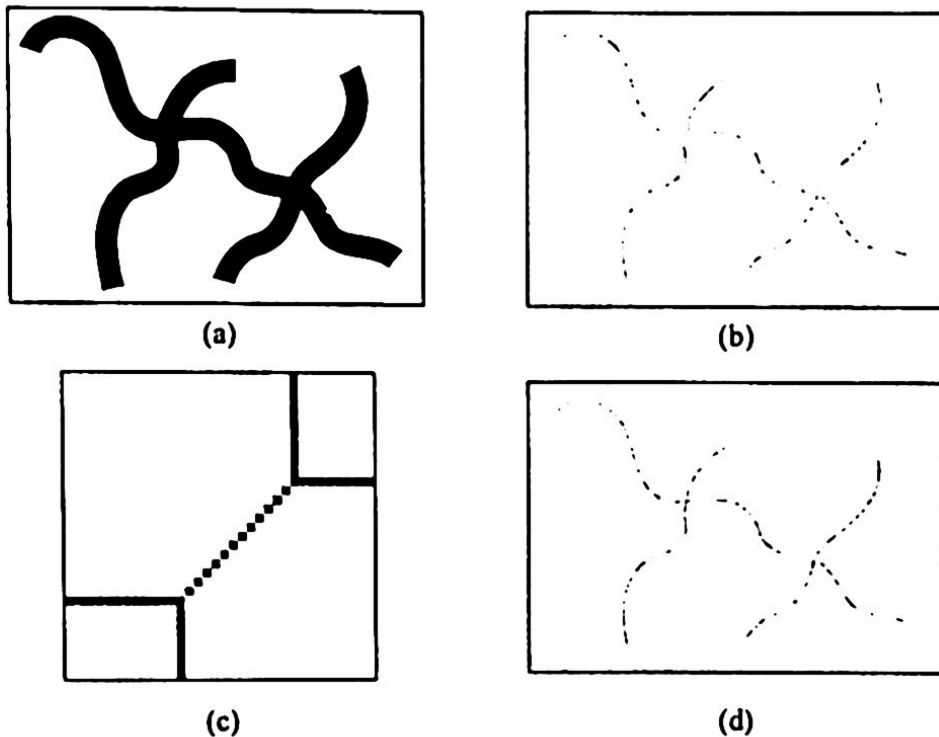


Fig. 4. Description of crossing points fusion. (a) Original image. (b) Initial skeleton (c) Amplified region of the skeleton showing the two crossing points generated by the algorithm. (d) Result of fusion algorithm

Consider the objects presented in the Figure 4(a), its corresponding skeleton that is shown in Figure 4(b), and an amplified section of the image that is shown in Figure 4(c). It is possible to appreciate that we obtain two cross point instead of one. This pathological case can be corrected using this criterion: if two cross points

(CP) are separated be at most $\pm (0.5)W$ pixels¹, applying the Euclidean distance, these points and arcs of the image are joined in one cross point. The result appears in Figure 4(d).

6. *Terminal and cross points counting.* Here, we proceed to count the two variables that allow us calculate the depth of a basin. The algorithm makes a top – down analysis and proceeds to classify object pixel according to Equation 3.

$$\begin{aligned} TP &= \{p \in S \mid N_s(p) = 1\} \\ CP &= \{p \in S \mid N_s(p) \geq 3\} \end{aligned} \quad (3)$$

Where *TP* denotes a terminal point, and *CP* a cross point. An arc is a k-connected sequence of pixels delimited by two terminal points. Finally, the steps involved in our algorithm are:

- a. Find the starting terminal point: First, associate the binary image to a Digital Elevation Model map for each basin pixel. Second, find the longest arc in the image, the length of the arc is the number of pixels contained in it. For the largest arc, the initial point is the terminal point of this arc with higher altitude; the other terminal point is the end terminal point.
- b. If *NCP* (number of cross points) is zero, then finish, the depth of the basin is 1.
- c. Review each arc contained in the skeleton, starting with the arc at the terminal point to which it belongs.
- d. Assign a weight of 1 to all terminal points (*TP*), except to the end terminal point.
- e. Find the weight for each cross point using the next rule: In a basin, all cross points are denominated bifurcations. These bifurcations define the intersection of two or more rivers. First, we have to infer the directions of the rivers. It is calculated as follows. If a river acts as sink of the water flow the arc that represents that river is considered as an outgoing arc. On the other hand, if a river delivers water flow to a bifurcation, the corresponding arc is considered as an incoming arc. Then we use altitude values of the arcs to determine if an arc is an incoming or an outgoing arc.
- f. In a bifurcation, if all the weights of the incoming arcs are equal, the weight of the outgoing arc is the arithmetic sum of these weights, otherwise, the greatest weight is the value assigned to the outgoing arc. This condition is resumed in Equation 4.
- g. Process the remaining arcs, until reaching the end terminal point, the weight of this point is the desired depth.

¹ *W* denotes the width in pixels of the object.

$$w(p) = \begin{cases} w(a_1) + 1, & \text{if } w(a_1) = w(a_2) = \dots = w(a_n) \\ \text{Greatest}(w(a_1), w(a_2), \dots, w(a_n)) & \text{otherwise} \end{cases} \quad (4)$$

This algorithm does not require demonstration due the fact that we use the technique proposed by Horton [4].

4 Tests and Results

In this section, we show the results obtained by applying our algorithm. The application was tested with a bank of 100 raster images. In about 97% of all processed images the results have been satisfactory. Figure 5(a) shows the original image and 5(b) the processed image.

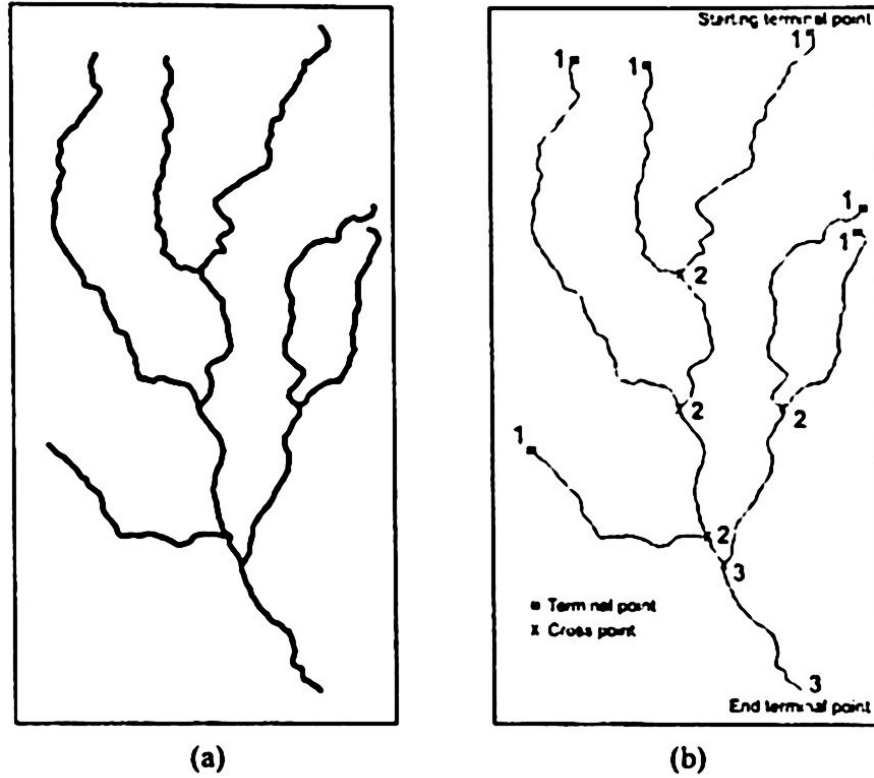


Fig. 5. Algorithm result. (a) Source image (b) Output image

4.1 Pathological cases

We identified at least two pathological cases, which can produce erroneous results. The first case is when two independent rivers touch in a bifurcation; see Figure 6. The cross point fusion processes the skeleton and proceeds to make a fusion as we mentioned before. However in this case, it is not necessary to carry out this fusion that represents clear algorithm disadvantage. We are working on the algorithm improvement to solve this case by another (semantic-based approach).

The second case is when the parasite branches are presented. We solved this case by applying a morphological operation (opening with 3x3-structured element); see Fig. 7. Obviously a river rarely takes this form, but is useful to show the pathologic case.

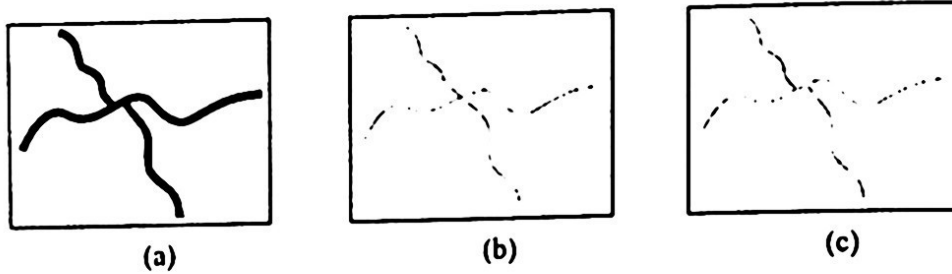


Fig. 6. Pathological case. (a) Source image. (b) 8-connected skeleton. (c) Output of cross point fusion

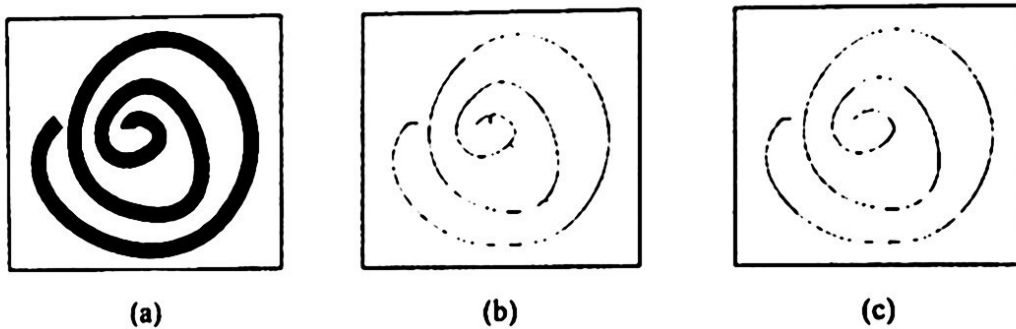


Fig. 7. Second pathological case. (a) Original image. (b) 8-connected skeleton without applying morphological operator. (c) 8-connected skeleton in an original image with morphological operation

5 Conclusions

In this paper, we have presented an algorithm that correctly determines the depth of a basin. The main advantages of the presented algorithm are:

- Due to its low complexity it can be applied to real-time process
- It uses only two variables to determine the depth of a basin.
- Use DEM allows us to take into consideration the elevation characteristics as well as flow direction of the basin.

However, one pathological case (section 4.1) remains unsolved in our approach. We believe that it can be solved by adding another technique based on semantic analysis of "river's independence". Our future research will be concerned with this approach.

Acknowledgments

The authors of this paper wish to thank to the Centre for Computing Research (CIC), the General Coordination of Postgraduate Study and Research (CGEPI) and to the National Polytechnic Institute (IPN) for their support. Additionally, the authors wish to thank the reviewers of this work for their pertinent comments.

References

1. A. A. Amini, T. E. Weymouth, and R. C. Jain, "Using dynamic programming for solving variational problems in vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12(9):855-867, 1990.
2. A. Gruen and H. Li. "Semi – automatic road extraction by dynamic programming". *ISPRS International Archives of Photogrammetry and Remote Sensing*. 30(3/1):324-332, 1994.
3. D. Craig, N. Olaf, and R. Dianne "Semi-Automated Extraction of Rivers from Digital Imagery". *GeoInformática*, Vol. 6, Number 3 (2002).
4. Horton, R. E.: Drainage basin characteristics. *Transactions American Geophysical Union*, Vol. 13 (1932) 350-361.
5. J. L. Díaz, C. Yáñez, G. Guzmán, "Algoritmos para generar esqueletos k-conectados basados en criterios de conectividad". *XII International Congress in Computation*. ISBN: 970-36-0098-0.
6. N. Otsu, "A threshold method from gray-level histograms". *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62-66.
7. F. Jiulun, X. Winxin, "Minium error thresholding: A note". *Pattern Recognition Letters*, 18:705-709, 1997.
8. R. González, E. Woods, "Digital Image Processing". Second Edition, Prentice Hall.
9. <http://www.physicalgeography.net/fundamentals/10ab.html>.
10. G. Guzmán, "Metodología Extendida para el Conteo de Objetos en Imágenes". Tesis de Maestría, CIC – IPN, 2003.

Extracting the Parametric Model of Duffing's Oscillator by Using a High Gain Observer

*Carlos Aguilar-Ibáñez, Juan Mendoza C., Rafael Martínez G., and Rubén Garrido M.

*CIC-IPN, Av. Juan de Dios Bátiz s/n Esq. Manuel Othón de M.

Unidad Profesional Adolfo López Mateos

Col. San Pedro Zacatenco, A.P. 75476

México, D.F. 07700, México

Phone: (52-5) 729-6000 xt. 56568, FAX: (52-5) 586-2936

email: caguilar@cic.ipn.mx

CINVESTAV-IPN, Departamento de Control Automático,

Av. IPN 2508, A.P. 14740, México, D.F. 07360, México

Phone: (52-5) 061-3733, FAX: (52-5) 747-7089

email: rguerra@ctrl.cinvestav.mx

Abstract. In this article we propose a simple high gain observer for extracting the unknown parameters of the Duffing's oscillator. It is shown that this system is observable and identifiable algebraically, with respect to a well-chosen output. Hence, an extended differential parameterization of the output and its time derivatives can be obtained. Based on these facts, we evaluate the obtained parameterization in a finite number of times to build a set of algebraic equations, and then, the parametric model is obtained by an inverse matrix. Although, the time derivatives output are not available, we overcome this difficulty by using an practical High-Gain Observer.

Keywords. Mechanical Oscillator, Chaos Reconstruction and High-Gain Observers.

1 Introduction

An interesting problem in chaos theory and its applications is the reconstruction of the unknown variables and parameters of a chaotic system. This problem is important because any experimental dynamical system possesses only a few variables or parameters that can be measured (see [Parlitz *et al.*, 1994] and [Abarbanel, 1996]). In some cases, it is necessary to estimate or reconstruct the unavailable quantities in order to completely determine the system's state and to achieve a good synchronization or to predict the system's behavior.

Roughly speaking, there are two approaches for reconstructing a chaotic attractor. The first relies on control theory; like the procedure based on system inversion (see [U. Feldmann *et al.*, 1996], [H. Huijberts *et al.*, 2000] and [M.S. Suarez *et al.*, 2003]; and, traditional identification schemas and state observer design (see [G. Cheng, 1995], [A. S. Poznyak *et al.*, 1999], [I. Chavez M. *et al.*, 2002] and [I. Chavez M. *et al.*, 2002]). The second approach is based on time series from a particular chaotic system, which are used in the so-called time delay reconstruction of a phase space

(see [Parlitz *et al.*, 1994], [K.T. Alligood *et al.*, 1997], [I. Makoto *et al.*, 1997], [T. Sauer *et al.*, 1991], [T. Stojanovski *et al.*, 1997] and [F. Takens, 1981]).

In this communication, we deal with the reconstruction of the Duffing's System (DMO) by means of measurements of the position state, which is considered as the system output. The on-line identification procedure is based on algebraic properties that the DMO satisfies ([R. Martinez *et al.*, 2001] and [I. Chavez M *et al.*, 2002]). Those properties allow finding a differential parameterization of the output and its time derivatives. Then, we evaluate the obtained parameterization in a discrete set of time to form a set of linear equations, where the parameters of the DMO are the variables of the obtained linear equations. And finally, we recover the unknown parameter by solving a set of linear equations. We should mention that time evaluation of the differential parameterization requires the unavailable output time derivatives. Such difficulty is overcome by using a high-gain observer (HGO). An HGO does not require an accurate model, and the error can be as small as desired (see [Dabroom and Khalil, 1999]), where the error is the difference between the original system measured signal and the signal estimated by the observer.

The rest of this work is organized as follows. Section 1 gives a brief description of the DMO. Section 2 is devoted to studying some important algebraic properties of the differential equations of the DMO. Also, in Section 2 we present an identification procedure by means of the previously introduced algebraic properties assuming that the time derivatives of the selected output are available. Finally, in the same Section, we present an HGO for computing the unavailable time derivatives of the output. Section 3 contains the results of the simulations while Section 4 is devoted to giving some conclusions. Finally, in the Appendix we provide a proof of Propositions 1 and 2.

2 Duffing's Mechanical Oscillator

Consider the traditional DMO, described by:

$$\begin{aligned}\dot{x} &= v \\ \dot{v} &= -p_1 v - p_3 x^3 - p_2 x + A \cos(\omega t);\end{aligned}\tag{1}$$

where x measures the oscillator position, A is the amplitude of the forcing function, ω is the forcing frequency, p_1 is the damping coefficient, and p_2 and p_3 are fixed constants related to a non-linear stiffness function. When the parameters values are in a neighborhood of $\{p_1 = 0.4, p_2 = -1.1, p_3 = 1, A = 2.1, \omega = 1.8\}$, this system has a chaotic behavior (see [Nayfeh & Mook, 1979], [Alligood *et al.*, 1997]).

Recovering the Set of Parameters

We first establish the main problem to solve: We desire to recover the unknown parameters $\{p_1, p_2, p_3, A\}$ from the measured output or available position x . To solve it, we first discuss two important and useful definitions which will allow us to transform the original system into a set of differential parameterizations of the output. The differential parameterization of the output is obtained based on its successive output time derivatives; where the order of the time derivative of the output is a function of the number of parameters that we want to identify¹. Finally, evaluating the parameterization in a finite number of times, it is possible to recover the unknown parameters by means of the inverse of a matrix.

Some Algebraic Properties

We say that a system is algebraically observable if there exists a suitable output provided that all the system variables can be differentially parameterized solely in terms of the output. Moreover, if we can express the parameter vector as a parametric function of the output and a finite number of its time derivatives, we say that the system is identifiable with respect to this output.

Now, let us consider again the DMO and define the output $y = x$. Evidently, we have:

$$\dot{y} = y, \quad (2)$$

$$\ddot{y} = -y p_1 - y p_2 - y^3 p_3 + A \cos(\omega t).$$

If we continue to differentiate the output with respect to time, we obtain:

$$\dot{y} = -y p_1 - y p_2 - y^3 p_3 + A \cos(\omega t),$$

$$y^{(3)} = -y p_1 - y p_2 - 3y^2 y p_3 - \omega A \sin(\omega t),$$

$$y^{(4)} = -y^{(3)} p_1 - y p_2 + (-3y^2 y - 6y y^2) p_3 - \omega^2 A \cos(\omega t),$$

$$y^{(5)} = -y^{(4)} p_1 - y^{(3)} p_2 - (-18y y y - 3y^2 y^{(3)} - 6y^3) p_3 + \omega^3 A \sin(\omega t).$$

After some manipulations, the last set of equations may be rewritten as:

$$M(y)Q = F(\omega, t) \quad (3)$$

Where Q stands for the vector of parameters defined as:

¹ In this case is necessary to obtain from $y^{(1)}$ to $y^{(5)}$ because we want to estimate four unknown parameters.

$$Q^T \triangleq [q_1, q_2, q_3, q_4] = \left[\frac{1}{A}, \frac{p_1}{A}, \frac{p_2}{A}, \frac{p_3}{A} \right] \quad (4)$$

$M(y)$ is the matrix of the output y and its time derivatives, defined by:

$$M(y) \triangleq \begin{bmatrix} -\ddot{y} & -\dot{y} & -y & -y^3 \\ -y^{(3)} & -\ddot{y} & -\dot{y} & -3y^2\dot{y} \\ -y^{(4)} & -y^{(3)} & -\ddot{y} & (-3y^2\ddot{y} - 6y\dot{y}^2) \\ -y^{(5)} & -y^{(4)} & -y^{(3)} & (-18y\dot{y}\ddot{y} - 3y^2y^{(3)} - 6\dot{y}^3) \end{bmatrix}. \quad (5)$$

$F(w, t)$ is the independent vector of the variable y whose components are given as:

$$F(w, t) \triangleq \begin{bmatrix} -\cos(wt), w\sin(wt), w^2\cos(wt), -w^3\sin(wt) \end{bmatrix} \quad (6)$$

Therefore, vector Q is algebraically identifiable with respect to the variable $y(t)$. The relation (3) is referred as the extended differential parameterization.

Next, let us consider the following assumptions:

A1: The solution $y(t)$ and its time derivatives exhibit a chaotic behavior for the system (1).

A2: The time derivatives of the selected output are always available.

Now, we mention an important proposition that allows us to compute the unknown parameter vector.

Proposition 1: Let us consider system (1) with its respective extended differential parameterization (3), under assumptions A1 and A2. Then, the inverse of matrix (5) exists almost for any time.

Proof: (Refer to Appendix).

Notice that, substituting the vector values Q into relation (4), we can recover the set of unknown parameters $\{p_1, p_2, p_3, A\}$. Finally, in next section, we propose a practical numerical differentiator to estimate the time derivatives of the variable y .

A simple HGO

In order to obtain the time derivatives of the output y , we suggest the following scheme to estimate the time derivatives. Let us define vector $Y^T = [y \dots y^{(5)}]$ and let us propose the following filter given by:

$$\dot{\hat{Y}} = A\hat{Y} + HC(Y - \hat{Y}) \quad (7)$$

Where, matrix A defined the well-known Brunovsky form [Dabroom & Khalil, 1999] and

$$H^T = \left[\frac{\alpha_1}{\varepsilon}, \dots, \frac{\alpha_6}{\varepsilon^6} \right], C = [1, 0, \dots, 0] \quad (8)$$

ε is a small positive parameter and the positive constants α_i are selected such that the polynomial defined as:

$$p(s) = s^6 + \alpha_1 s^5 + \dots + \alpha_6, \quad (9)$$

is Hurwitz (see [Dabroom & Khalil, 1999] for more details).

The following proposition allows us to compute the error $\xi = Y - \hat{Y}$.

Proposition 2: Consider the system (7) under assumptions A1. Then, the HGO proposed in (8) is able to recover Y with bounded error

$$\|\xi\| \leq \beta n \varepsilon / \lambda^* \quad (10)$$

Where λ^* is given by

$$\lambda^* = \min \{ \operatorname{Re} [\operatorname{roots}(p(s))] \}, \quad (11)$$

β is a positive constant which depends on the initial conditions $\xi(0)$, and

$$n = \max_{t \in [0, t]} |y^{(5)}(t)|$$

Notice that, we substitute the estimated time derivative $\hat{y}^{(k)}$ instead of $y^{(k)}$, $k = \{1, \dots, 5\}$ into expression (5).

Numerical Simulations

We first test the efficiency of the HGO by computer simulations. The experiments were implemented by using the 4th-order Runge-Kutta algorithm. The computation was performed with a precision of 8 decimal digit numbers, from $t = 0$ seconds to $t = 10$ seconds. To obtain a good performance, the step size in the numerical method was set to 0.0001. The DMO parameter values were set as $p_1 = 0.3$; $p_2 = -1.2$; $p_3 = 1$, $A = 1.8$, $w = 1.9$. The initial conditions were set as $y(0) = 1$ and $\dot{y}(0) = -1$. The polynomial was chosen to be $p(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)$, with $\zeta = 0.707$ and $\omega_n = 0.9$. The gain of the HGO was selected as $\varepsilon = 0.005$.

For this particular simulation after $t = 0.25$ seconds the derivative estimation errors were around:

$$\xi_1 = 0.1 \times 10^{-6}, \xi_2 = 0.3 \times 10^{-5}, \xi_3 = 0.9 \times 10^{-5}, \xi_4 = 0.25 \times 10^{-4}, \xi_5 = 0.1 \times 10^{-3}.$$

Consequently, we obtain a very good estimation of non available derivatives.

Finally, we probe the effectiveness of the described identification method by numerical simulations. The initial conditions and the physical parameters were taken as in the previous experiment.

Figures 1 and 2 show the estimation of the parameter A , p_1 and p_2 and p_3 , respectively.

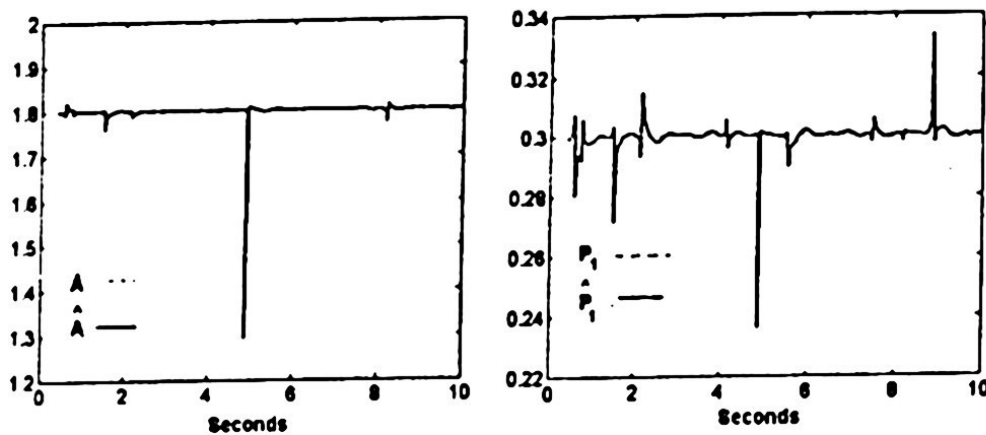


Fig. 1. Identification of parameters A and p_1

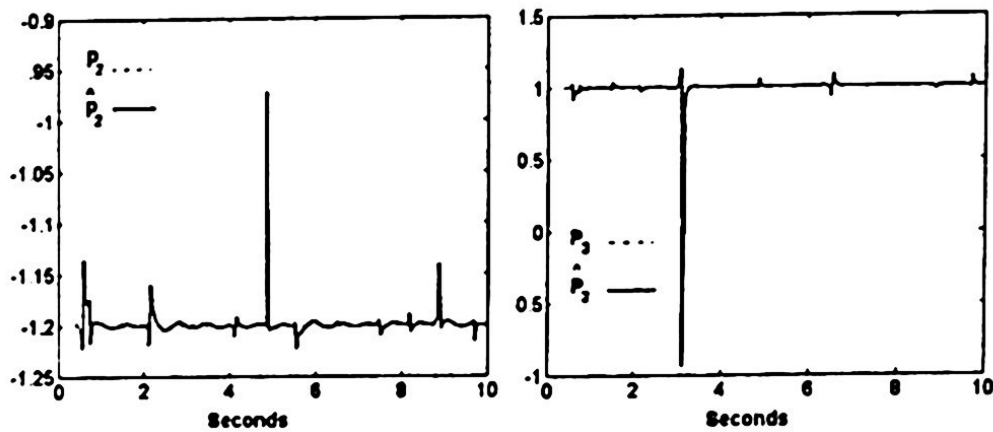


Fig. 2. Identification of parameters p_2 and p_3

3 Conclusions

Based on the algebraic differential approach for the identification problem of the traditionally Duffing's oscillator that has been treated in this paper. The fact that the system is algebraically observable and algebraically identifiable, with respect to a specified variable, allowed us to describe the original system by means of a differential parameterization of the output and its respective time derivatives. This

differential parameterization has all the necessary information to recover the parametric model of the DMO. Then, we proceeded to evaluate the differential parameterization in a finite number of times in order to form a set of algebraic equations, where the unknown parameters of the DMO were obtained by computing an inverse matrix. The lack of the output's time derivatives required in the set of algebraic equations was overcome by the design of an HGO, where the observation errors can be as small as needed by tuning a specific parameter accordingly to Proposition 2.

The obtained identification solution was illustrated in a numerical simulation, where the HGO recovers the output's time derivatives, and then, the corresponding unknown parameters can be revealed.

Acknowledgements : This research was supported by the Coordinación de Posgrado e Investigación (CGPI-IPN), under research Grant 20040877

Appendix

Proof of Proposition 1:

We first suppose that the set of functions $\{\dot{y}(t), y(t), y(t), y^{(3)}(t)\}$ is linearly dependent in a time interval I where $y(t)$ shows a chaotic behavior according to A1. This implies the existence of a set of constants c_1, c_2, c_3 , and c_4 , different from zero such that:

$$c_1 y(t) + c_2 \dot{y}(t) + c_3 y(t) + c_4 y^{(3)}(t) = 0$$

Notice that if $c_1 = 0$ then the last second order differential equation turns into a first order differential equation, therefore, $y(t)$ is a monotonic decreasing or monotonic increasing function (see p. 332 of [Alligood *et al.*, 1997]) This case is not possible, because $y(t)$ has a chaotic behavior. Also, by the Poincaré-Bendixon theorem (see p. 337 of [Alligood *et al.*, 1997]) it is well-known that a second order differential equation that does not depend on time cannot exhibit a chaotic behavior. Thus, $\{\dot{y}(t), y(t), y(t), y^{(3)}(t)\}$ is linearly independent in a time interval I .

Proof of Proposition 2:

Evidently, vector \dot{Y} can be written as:

$$\dot{Y} = AY + \delta_y. \quad (12)$$

With $\delta^T = [0, \dots, y^{(6)}]$ Subtracting (12) from (7), we obtain the following differential equation of the error:

$$\dot{\xi} = [A - HC]\xi + \delta_y. \quad (13)$$

Notice that the characteristic polynomial of $A = A - HC$ is given by $p(s, \varepsilon)$, which is also Hurwitz. That is, the proposed H assigns the eigenvalues of A at $1/\varepsilon$ times the roots of $p(s)$ (9). Hence, the error ε satisfies

$$\varepsilon(t) = e^{A(t-t_0)} \left(\varepsilon(0) + \int_{t_0}^t e^{A(t_0-s)} \delta_y(s) ds \right). \quad (14)$$

Since A is exponentially stable and the signal $y^{(3)}$ is bounded, we also have the following inequality:

$$\xi \leq \beta e^{-\lambda^* t} \xi + \beta \eta \varepsilon (1 - e^{-\lambda^* t}) \lambda^* \rightarrow \beta \eta \varepsilon / \lambda^*. \quad (15)$$

Where the positive constants β, λ^*, η are previously defined in Proposition 2.

References

- [1] K. T. Alligood, T. D. Sauer and J. A. Yorke, *Chaos : An Introduction to Dynamical Systems*, Springer-Verlag, New York, 1997.
- [2] H. D. I. Abarbanel, *Analysis of Observed Chaotic Data*, (Springer Verlag New York, Inc.), 1996
- [3] A. Dabroom and H. Khalil, Numerical differentiation using high-gain observers, *Int. J. of Control* 72(1999) 1523-1537.
- [4] U. Feldmann, M. Hasler and W. Schwarz, Communication by chaotic signals: The inverse system approach, *Int. J. Circuit Theory Appl.* 24 (1996) 551-579.
- [5] H. Huijbert, H. Nijmeijer and R. Willems, System Identification in Communication with Chaotic Systems, *IEEE Transactions on Circuits and Systems -I: Fundamental Theory and Applications* 47 (2000) 800-808.
- [6] I. Makoto, C. W. Wu and L. O. Chua, Communication Systems Via Chaotic Signals From a Reconstruction Viewpoint, *International Journal of Bifurcation and Chaos* 7 (1997) 275-286.
- [7] U. Parlitz, R. Zöllner, J. Holzfuss and W. Lauterborn, Reconstructing Physical Variables and Parameters From Dynamical Systems, *International Journal of Bifurcation and Chaos* 4, (1994) 1715-1719.
- [8] M. S. Suárez-Castañón, C. Aguilar-Ibáñez and R. Barrón-Fernández, On recovering the parameters and velocity state of the Duffing's oscillator, *Physics Letters A* 308 (2003) 47-53.
- [9] T. Sauer, J. Yorke and M. Casdagli, Embedology, *J. Stat. Phys.* 65 (1991) 579-616.

- [10] T. Stojanovski , U. Parlitz , L. Kocarev and R. Harris. Exploiting delay reconstruction for chaos synchronization, *Physics Letters A* 233 (1997) 355-360.
- [11] F. Takens, Detecting strange attractors in turbulence, *Dynamical Systems and Turbulence*, eds. Rand, D.A. & Young, L.-S. Springer-Verlag, Berlin, (1981) 366-381.
- [12] I. Chavez-Manriquez, R. Martínez-Guerra and A. Osorio-Cordero, Parametric identification of linear time varying systems using a proportional reduced order observer, *American Control Conference (ACC2002)*, Anchorage, Alaska, USA, pp. 3236-3241, 2002.
- [13] A. S. Poznyak, W. Yu, and E. N. Sanchez, Identification and Control of Unknown Chaotic Systems via Dynamic Neural Networks, *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications* 46, No. 12 (1999).
- [14] A. S. Poznyak, W. Yu, H. S. Ramirez, and E. N. Sanchez, Robust identification by dynamics neural networks using sliding mode learning, *Appl. Math. Comput. Sci.* 8 (1998) 101-110.
- [15] R. Martínez-Guerra, R. Aguilar and A. Poznyak. Estimation for HIV transmission using a reduced order uncertainty observer, *Proceedings of the American Control Conference*, Arlington, VA June 25-27, 2001.



On an Stabilizable Control Force for the Inverted Pendulum Cart System

Carlos Aguilar I. ¹, Octavio Gutierrez F. ¹

¹ CIC-IPN, Av Juan de Dios Batiz s/n Esq. Miguel Othón de M,
Unidad Profesional Adolfo Lopez Mateos
Col. San Pedro Zacatenco, A. P 75476
México D. F., 07700, México
Fax (52-5): 586- 29-36, caguilar@cic.ipn.mx.

Abstract. A novel control force is presented to stabilize the inverted pendulum mounted on a cart. The control force is based on proposing a partial feedback linearization, in conjunction with Lyapunov's second method. This simple, but efficient strategy guarantees that the closed-loop system is locally asymptotically stable around the unstable equilibrium point. Additionally, the controller has a very large attraction domain and it is robust with respect to damping .

Keywords: Classical Mechanics; Nonlinear Systems, Lyapunov's second method;

1 Introduction

The study of the simple inverted pendulum cart system has been one of the most interesting problems in classical mechanics and modern control theory. The device consists of a pole whose pivot point is mounted on a cart. The pendulum is free to rotate about its pivot point. The cart can move horizontally perpendicular to the axis's pendulum and is actuated by a horizontal force. The mechanical problem is to bring the pendulum from large initial pendulum deviation to the upper unstable equilibrium position by moving the cart on the horizontal plane. This system has attracted the attention of many researchers, as seen by a growing list of articles (for example, see [1], [2], [3], [4], [5], [6] and [7]). The interest is due to the fact that, the device is non-feedback linearizable by means of dynamics state feedback (see [8]), and hence, it is not linearizable by means of dynamic state feedback control either. This obstacle makes it especially difficult to perform some controlled maneuvers; for instance, there is no continuous force which globally stabilizes the upright equilibrium of the pendulum with zero displacement of the cart [9]. Nevertheless, the problem can be solved producing at least one discontinuity in the acceleration cart. However, it is well-known how to construct a linear locally stabilizing controller [10] but, the linear based controls design presents the inconvenience of having a very small domain of attraction.

In this article, we present a control force to locally asymptotically stabilize the inverted pendulum cart system around its unstable equilibrium point¹, for a very large attraction domain.

Also, intuitively the proposed control force allows us to transfer the pendulum from the stable equilibrium point to the unstable equilibrium points, which are the lower and the upper resting angular position, respectively. This task is done by means of switching a suitable parameter. The system stability is demonstrated by using Lyapunov's second method.

This paper is organized as follows. Section 1 introduces the normalized non-linear model of mechanical device. Next, a partial feedback linearization of the non-linear equations is obtained. Section 2 presents a non-linear controller for the stabilization of the device, and some computer simulation results depicting the performance of the closed-loop system are presented. Finally, Section 3 gives the conclusions.

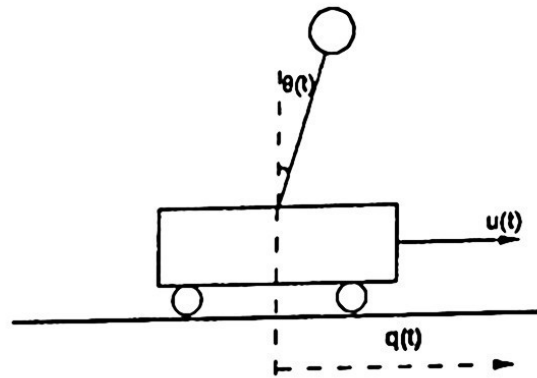


Fig. 1. The inverted pendulum cart system

2 The Inverted Pendulum Cart System

Consider the traditional inverted pendulum mounted on a cart (see Figure 1). The nonlinear model of the system, which can be obtained from either the Newton or the Euler-Lagrange equations, see more detail (Lozano and Fantoni) is given by

$$\begin{aligned} mL \cos \theta \ddot{x} + mL^2 \ddot{\theta} - gmL \sin \theta &= 0 \\ (M + m) \ddot{x} + Lm \cos \theta \ddot{\theta} - mL \dot{\theta}^2 \sin \theta &= f \end{aligned}$$

where x is the cart displacement, θ is the angle that the pendulum forms with the vertical, f is the force applied to the cart, acting as the control input. M and

¹ Which is the upper angular position with zero displacement of the cart

m stand for the cart mass and the pendulum mass concentrated in the bob, L is the length of the pendulum.

To simplify the algebraic manipulation in the forthcoming developments, we normalize the above equations by introducing the following scaling transformations,

$$q = x/L, \quad u = f/(mg), \quad d\tau = dt\sqrt{g/L}, \quad \delta = M/m$$

This normalization leads to the simpler system,

$$\begin{aligned} \cos\theta \ddot{q} + \ddot{\theta} - \sin\theta &= 0 \\ (1 + \delta) \ddot{q} + \cos\theta \ddot{\theta} - \dot{\theta}^2 \sin\theta &= u, \end{aligned}$$

where, with an abuse of notation “.” stands for differentiation with respect to the dimensionless time τ . Then, a convenient partial feedback linearization input is proposed as follow (see Spong),

$$u = \cos\theta \sin\theta - \dot{\theta}^2 \sin\theta + v(\sin^2\theta + \delta)$$

which produces the feedback equivalent system:

$$\begin{aligned} \ddot{\theta} &= \sin\theta - \cos\theta v, \\ \ddot{q} &= v \end{aligned} \tag{1}$$

Notice that for the new input $v=0$ and $\theta \in [0, 2\pi]$ the aforementioned system has two equilibrium points one is an unstable equilibrium point $(\theta, \dot{\theta}, q, \dot{q}) = (0, 0, 0, 0)$ and the other is a stable equilibrium point $(\theta, \dot{\theta}, q, \dot{q}) = (\pi, 0, 0, 0)$. The issue is to stabilize the system around its unstable equilibrium point, *i.e.*, we wish to bring the pendulum to its upper position and the cart displacement to zero simultaneously.

3 A Practical control law

Traditionally, the problem of designing a stabilizable input control law for system (1) is based on the well known Lyapunov method. Roughly speaking, it consists of proposing a positive definite function (or *Lyapunov function*) provided that, its time derivative along the trajectories of system (1) be at least semi-definite. The very hard problem is how to find the *Lyapunov function*. We relax the problem by setting off a

semi-definite instead of a positive definite function, and then the asymptotic stability is assured by a simple linearization of the closed loop-system.

Let us first introduce the following auxiliary variable:

$$\xi(x) = \sin\theta + k_p\theta + k_d\dot{\theta} + (k_p q + k_d \dot{q})\cos\theta + \alpha\dot{q} \quad (2)$$

Where x is the vector state defined as $x = [\theta, \dot{\theta}, q, \dot{q}]$. The design parameters k_p, k_d and α are positive constants that will be selected later.

Next, let us introduce the following semi-definite function,

$$V(x) = \frac{1}{2}\xi^2(x) \quad (3)$$

The time derivative of $V(\cdot)$ along the trajectories of the system (1) is then given, by

$$\dot{V}(x) = \xi(x)(\Omega(x) + \alpha v)$$

where

$$\Omega(x) = k_d \sin\theta + (k_p + \cos\theta - k_p \sin\theta)\dot{\theta} + (k_p \cos\theta - k_d \sin\theta\dot{\theta})\dot{q} \quad (4)$$

proposing v provided that

$$\Omega(x) + \alpha v = -K\xi(x)$$

Clearly, we have.

$$v = -\frac{1}{\alpha}(K\xi(x) + \Omega(x)) \quad (5)$$

Comment: The differential equations set formed when by system (1) and the feedback force v given in (5), is referred to as the non-linear closed-loop system.

Surprisingly, the proposed control law v turned out to be able to asymptotically stabilize the system (1) around the unstable equilibrium point $x = 0$, for a large attraction

domain². $D \subset R^4$. Where the control parameters $\{K_i, K_p, K_d, \alpha\}$ are selected such that the linearization of the closed-loop system around of $x = 0$ is Hurwitz, as we will show in the following proposition, which summarizes the previous discussion as:

Proposition: The nonlinear closed-loop system is locally asymptotically exponentially stable to a desired equilibrium point $x = 0$ if the design parameters K_i, k_p, k_d and α are chosen so that the following polinomial in the complex variable s is Hurwitz:

$$p(s) = s^4 + \left(K_i - \frac{1}{\alpha}\right)s^3 - \left(1 + \frac{k_d}{\alpha} + \frac{K_i}{\alpha}\right)s^2 - \left(K_i + \frac{kdKi}{\alpha} + \frac{k_p}{\alpha}\right)s - \frac{K_i K_p}{\alpha} \quad (6)$$

Proof:

The Jacobian linearization of the closed-loop system (see (1) and (5)) around the desired equilibrium point $x = 0$, is given by: $\dot{x} = Ax$, where

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 + \frac{k_d + K_i(1 + k_p)}{\alpha} & 1 + \frac{k_d K_i + K_p}{\alpha} & \frac{K_i K_p}{\alpha} & \frac{(\alpha + K_d)K_i - k_p}{\alpha} \\ 0 & 0 & \alpha & \alpha \\ -\frac{k_d + K_i(1 + K_p)}{\alpha} & -1 - \frac{K_d K_i - K_p}{\alpha} & -\frac{K_i k_p}{\alpha} & -\frac{(\alpha + k_d)K_i - k_p}{\alpha} \end{bmatrix},$$

computing the characteristic polynomial of matrix A , we have the expression (6). Hence, selecting the design parameters such that $p(s)$ is Hurwitz, we guarantee that the closed-loop system be at least locally asymptotically exponentially stable.

Note that if the parameters K_i, k_p and k_d are positive, it is necessary that α be negative

Remark: Eventually, we will try to rigorously make the estimation of the attraction domain D by means of Lyapunov functions in conjunction with La Salles's Theorem. It is worth mentioning that an efficient estimation of the set D is a difficult problem. It turns out to be a very demanding process in terms of computational resources, because it involves an optimization problem. In other cases, it is necessary to solve a partial

² where $D = \{x(0) \in R^4 / \lim_{t \rightarrow \infty} x(t) = 0\}$

differential equation (see [13] and the references included in this paper). For this reason, we believe that the detection of D is beyond the scope of this work.

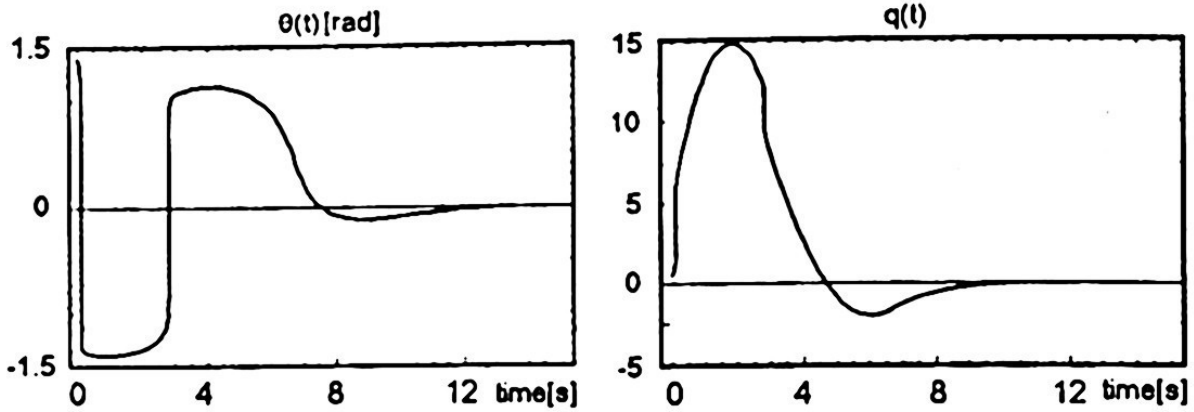


Fig. 2. Parameter α

4 Numerical Simulations

Simulations were carried out to evaluate the efficiency of the proposed feedback controller for three experiments. The experiments were implemented in *Mathematica* by means of a traditionally Runge-Kutta algorithm; the step size of the method was chosen to be equal to 0.001.

In the first experiment, we used the proposed controller (5), when it was applied to the nonlinear model (1). The design parameters were as $K_v = 10, k_p = 0.499, k_d = 1.272$ and $\alpha = -0.1$ and the initial conditions were set as $\theta(0) = 1.45, \dot{\theta}(0) = 0.2, q(0) = 0.5$ and $\dot{q}(0) = 0$. Figures 2 and 3 show the closed loop responses of the feedback equivalent system.

In the second experiment, we considered the design parameters and the initial conditions as before, but introduced a dissipative force in the unactuated direction, i.e. we add the damping $-\gamma\dot{\theta}$ into the first differential equation of the model (1), with $\gamma = 0.3$. Figures 4 and 5 show the robustness of the proposed nonlinear control when damping is considered in the numerical simulations. Notice that it is not generally true that damping makes a feedback-stabilized equilibrium asymptotically stable. That is to say, damping in the unactuated direction θ - direction tends to enhance stability while damping in the actuated direction y - direction tends to destabilize (see [14] and [15]).

Finally, we considered a swinging task to bring the pendulum from the lower resting position $x(0) = (\pi, 0, 0, 0)$ to the upper resting position $x(t) = 0$. To accomplish it, we

first used the input v to produce an instability in order to pull over the pendulum from the downward angular position. Then, when the pendulum was on the upper angular position, input v was switched to make that unstable equilibrium point become a asymptotically stable point. The switch in v was carried out by changing parameter α this was done taking $\alpha = -0.1$ when $\theta \in [-\pi/2, \pi/2]$ and $\alpha = 0.1$ in other cases. The other design parameters were set as before.

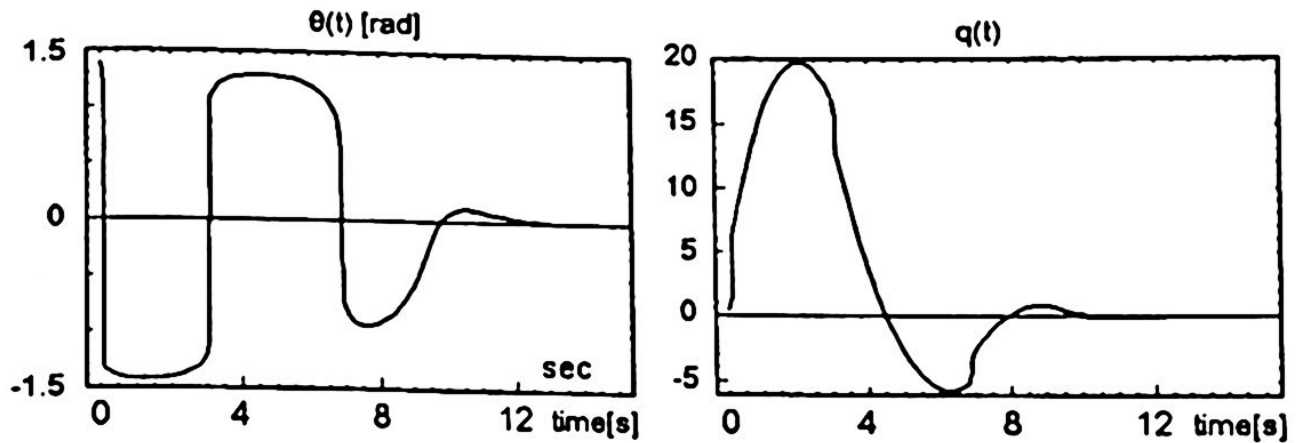


Fig. 3. Closed-loop responses when damping is considered α

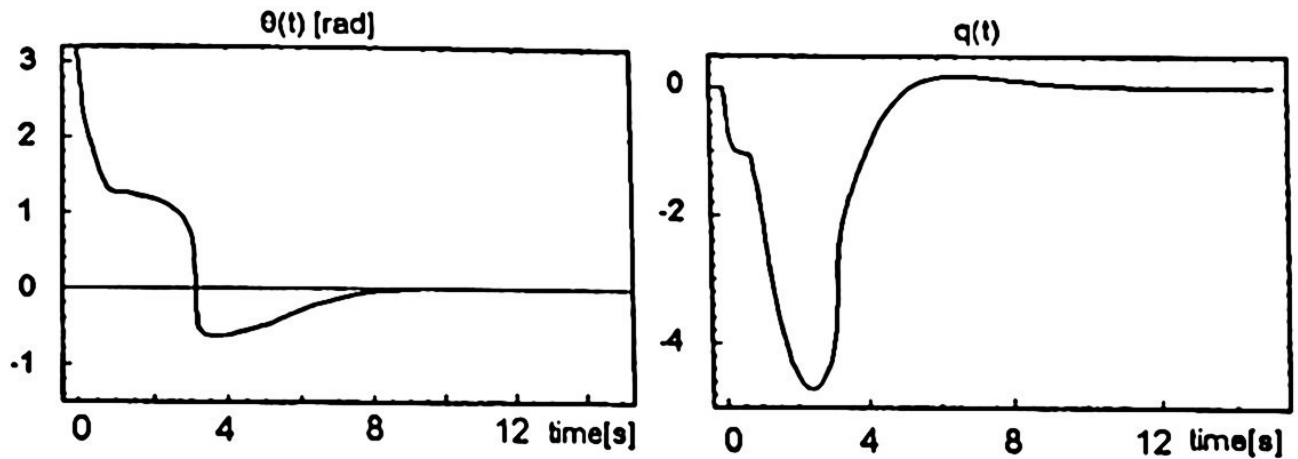


Fig. 4. The transfer from lower resting to upper resting positions

5 Conclusions

We have presented a practical nonlinear control for the inverted pendulum cart system. The goal of the proposed controller is to change the unstable equilibrium point into a locally asymptotically stable equilibrium point. The advantages of the presented controller is that it has a very large attraction domain, as we showed in the numerical simulations. It is robust with respect to dissipative forces. Also, the controller allows us to establish a simple strategy to bring the pendulum from the downward resting position towards the upward resting position, by means of a simple change of sign of

differential equation (see [13] and the references included in this paper). For this reason, we believe that the detection of D is beyond the scope of this work.

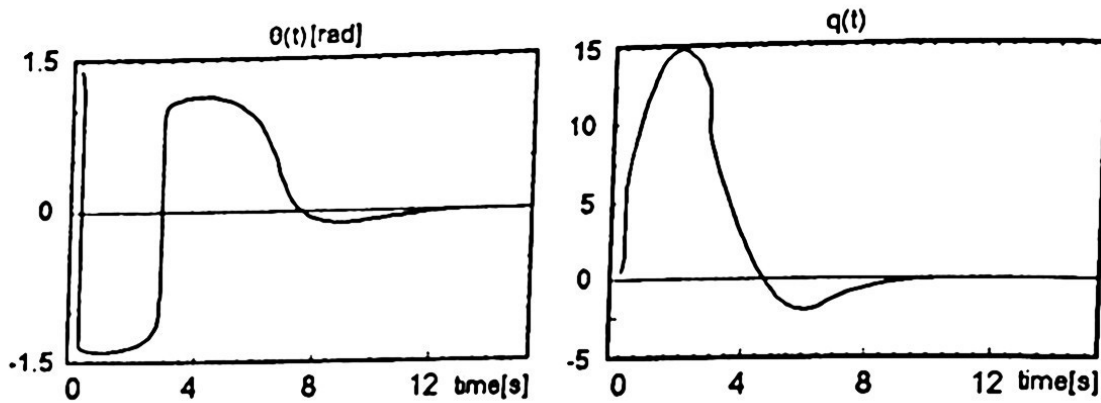


Fig. 2. Parameter α

4 Numerical Simulations

Simulations were carried out to evaluate the efficiency of the proposed feedback controller for three experiments. The experiments were implemented in *Mathematica* by means of a traditionally Runge-Kutta algorithm; the step size of the method was chosen to be equal to 0.001.

In the first experiment, we used the proposed controller (5), when it was applied to the nonlinear model (1). The design parameters were as $K_v = 10$, $k_p = 0.499$, $k_d = 1.272$ and $\alpha = -0.1$ and the initial conditions were set as $\theta(0) = 1.45$, $\dot{\theta}(0) = 0.2$, $q(0) = 0.5$ and $\dot{q}(0) = 0$. Figures 2 and 3 show the closed loop responses of the feedback equivalent system.

In the second experiment, we considered the design parameters and the initial conditions as before, but introduced a dissipative force in the unactuated direction, i.e. we add the damping $-\gamma\dot{\theta}$ into the first differential equation of the model (1), with $\gamma = 0.3$. Figures 4 and 5 show the robustness of the proposed nonlinear control when damping is considered in the numerical simulations. Notice that it is not generally true that damping makes a feedback-stabilized equilibrium asymptotically stable. That is to say, damping in the unactuated direction θ - direction tends to enhance stability while damping in the actuated direction y - direction tends to destabilize (see [14] and [15]).

Finally, we considered a swinging task to bring the pendulum from the lower resting position $x(0) = (\pi, 0, 0, 0)$ to the upper resting position $x(t) = 0$. To accomplish it, we

first used the input v to produce an instability in order to pull over the pendulum from the downward angular position. Then, when the pendulum was on the upper angular position, input v was switched to make that unstable equilibrium point become an asymptotically stable point. The switch in v was carried out by changing parameter α this was done taking $\alpha = -0.1$ when $\theta \in [-\pi/2, \pi/2]$ and $\alpha = 0.1$ in other cases. The other design parameters were set as before.

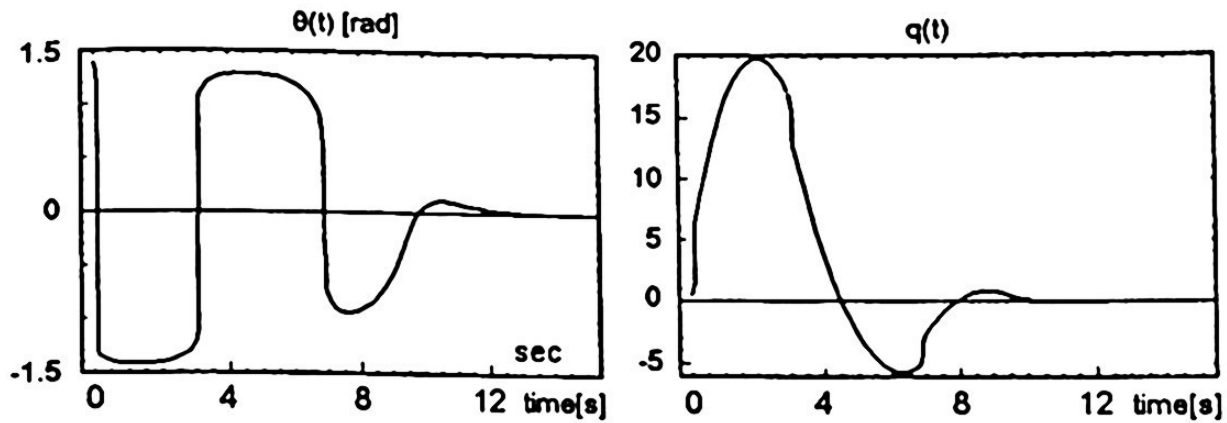


Fig. 3. Closed-loop responses when damping is considered α

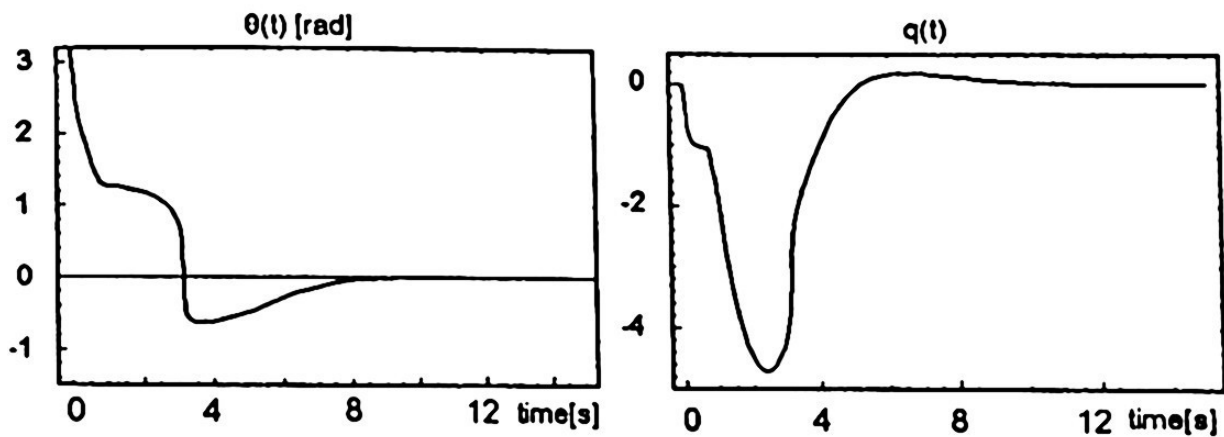


Fig. 4. The transfer from lower resting to upper resting positions

5 Conclusions

We have presented a practical nonlinear control for the inverted pendulum cart system. The goal of the proposed controller is to change the unstable equilibrium point into a locally asymptotically stable equilibrium point. The advantages of the presented controller is that it has a very large attraction domain, as we showed in the numerical simulations. It is robust with respect to dissipative forces. Also, the controller allows us to establish a simple strategy to bring the pendulum from the downward resting position towards the upward resting position, by means of a simple change of sign of

the suitable parameter α as we showed in the third experiment. It is quite interesting to mention that the swing up motion is caused by producing an unstable behavior in the lower angular position that brings the pendulum out of that region, and then, when it is in the upper half, the sign of α is shifted to stabilize it asymptotically in the top resting position. Finally, the closed-loop stability system was shown by means of a simple linearization of it. Even when the domain of attraction of the closed-loop equations has not been computed in this brief article, we assert that this controller produces a large computable domain of attraction, in comparison with traditionally based linear control approaches whose basin attraction are very small, as mentioned in [12] and [16].

Acknowledgments

This work was supported by CIC-IPN, and by the Coordinación General de Posgrado e Investigación (CGPI-IPN) under research Grant 20040877. Also this paper is dedicated in memory of Professor Leopoldo Arustegui (CBTIS 78)

References

1. Bloch, A. M., N. Leonard and J. E. Marsden [1997], Stabilization of mechanical systems using controlled Lagrangians, *Proc. CDC*, 36, 2356-2361.
2. Bloch, A. M., N. E. Leonard, and J. E. Marsden [2000], Controlled lagrangians and the Stabilization of Mechanical Systems I: The First matching theorem, *IEEE Trans.on Sytems and Control*, 45, 2253-2270. (pdf)
3. E. A. Jackson, "Perspectives of nonlinear Dynamics", Cambridge University Press, 1989.
4. K. Furuta, M. Yamakita, S. Kobayashi, "Swing up control of inverted pendulum using pseudo-state feedback", *Journal of System and Control Engineering*, Vol. 206, pp. 263-269, 1992.
5. E. I. Butikov, "On the dynamic stabilization of an inverted pendulum", *Am. J. Phys.* Vol. 69(6), 1-14, 2000.
6. W. T. Thomson, *Theory of Vibrations with Applications*, George, Allen and Unwind, London, 1981.
7. R. Olfati-Saber, "Fixed Point Controllers and Stabilization of the Cart-Pole and Rotating Pendulum", *Proc. of the 38th Conf. on Decision and Control*, pp. 1174-1181, Phoenix, Az. Dec. 1999.
8. B. Jakubczyk, W. Respondek, "On the linearization of control systems", *Bull. Acad. Polon. Sci. Math.*, 28:517-522, 1980.
9. A. S. Shiriaev, A. L. Fradkov, "Stabilization of invariant sets of nonlinear systems with applications to control of oscillations", *Int. J. of Robust and Nonlinear Control*, 11:215-240, 2001.
10. H. Kwakernaak, R. Sivan, *Linear Optimal Control Systems*, Wiley, New York, 19972.
11. I. Fantoni, R. Lozano, "Non-linear Control for Underactuated Mechanical System", Springer Verlag, London, 2002.
12. M. V. Spong, "Energy Based Control of a class of Underactuated Mechanical Systems," *IFAC World Congress*, San Francisco, CA, July, 1996.

13. A. Barreiro, J. Aracil, D. Pagano, "Detection of attraction domains of non-linear systems using bifurcation analysis and Lyapunov functions", *Int. J. of Control*, Vol. 75, no. 5, pp. 314-327, 2002.
14. Woolsey, C., A. M. Bloch, N. E. Leonard, and J. E. Marsden [2001], Physical dissipation and the method of controlled Lagrangians, *Proceedings of the European Control Conference*, Porto, Portugal, September 2001, 2570-2575
- 15 Bloch, A. M., D. Chang, N. Leonard, and J. E. Marsden [2001], Controlled Lagrangians and the stabilization of mechanical systems II: Potential shaping, *Trans IEEE on Auto. Control*, 46, 1556-1571.
16. M. W. Spong, , "The Swing Up Control Problem for the Acrobot", *IEEE Control Systems Magazine*, Vol. 15, No. 1, pp. 49-55, 1995.



Reconstructing Equations of the Rossler's System from the y-variable

Carlos Aguilar-Ibáñez, Jorge Sanchez H, Fortunato Flores, A, **Rafael Martínez G., Rubén Garrido M. and Miguel S. Suarez C.

*CIC-IPN, Av. Juan de Dios Bátiz s/n Esq. Manuel Othón de M.
Unidad Profesional Adolfo López Mateos
Col. San Pedro Zacatenco, A.P. 75476
México, D.F. 07700, México

Phone: (52-5) 729-6000 xt. 56568, FAX: (52-5) 586-2936
email: caguilar@cic.ipn.mx

**CINVESTAV-IPN, Departamento de Control Automático,
Av. IPN 2508, A.P. 14740, México, D.F. 07360, México
Phone: (52-5) 061-3733, FAX: (52-5) 747-7089
email: rguerra@ctrl.cinvestav.mx

Abstract. A simple way to determine the parameters of Rössler's system based on a suitable output (y-variable) is presented in this paper. The fact that the nonlinear system is observable and algebraically identifiable, with respect to the selected output, allows us to propose, in a first stage, a high-gain observer to estimate the output's time derivatives. And then, based on these facts two suitable schemes to recover the parameters are presented.

Keywords. Chaos; Parameter identification , States reconstruction and Genetic Algorithms.

1 Introduction

Reconstruction of chaotic attractors, from one or more available variables, is an interesting and challenging problem, which has attracted the attention of many researchers because of its theoretical and practical importance (see [Baker *et al.*, 2001], [Broomhead & King, 1986], [Gibson *et al.*, 1992], [Poznyak *et al.*, 1999], [Lainscsek & Gorodnitsky, 1996] and [Crutchfield & McNamara, 1987]). The objective is to find an inverse of mathematical or empirical models; that is, extract, from a partial knowledge of these models, the underlying dynamics ([Parlitz, 1995]). This problem is important: first, experimentally, in general it is not possible to measure or observe the complete state of a given system, or in some cases only a few physical parameters are available (see [Stojanovski *et al.*, 1996]); second, it allows us to verify the accuracy of some empirically derived models (see [Lainscsek & Gorodnitsky, 1996]); and third, to verify how secure a communication system is when the encoding system is chaotic (see [Stojanovski *et al.*, 1996], [Parlitz *et al.*, 1994] and [Kocarev *et al.*, 1992]). Roughly speaking, the problem has been solved in two ways. The first approach has been, so far, dominated by the delay embedding methodology founded on the delay reconstruction of states known from non-linear time series analysis (for general background on this fascinating area, the reader is referred to the easily readable books [Alligood *et al.*, 1997] and [Hand & Berthold, 2002]). Also, we

recommend reading the papers ([Sauer *et al.*, 1991], [Takens, 1981], [Tokuda *et al.*, 2002], [Stojanovski *et al.*, 1996], [Parlitz *et al.*, 1994], [Broomhead & King, 1986] and [Hand & Berthold, 2002]). The other approach is based on control theoretical ideas, such as inverse system design and system identification, as devices to recover parameters and unknown or difficult-to-measure states ([Feldmann *et al.*, 1996], [Poznyak *et al.*, 1999], [Poznyak *et al.*, 1998], [Cheng & Dong, 1995] and [Suárez *et al.*, 2003]).

In this paper, we deal with the problem of recovering Rossler's parameters by means of the knowledge of an available output (*which is the y -variable of the well-known Rossler's model*). The on-line identification approach is based on the algebraic properties of observability and identifiability that Rossler's model satisfies. Those properties allow us to find a differential parameterization of the output and a finite number of its time derivatives. This parameterization is used in two identification approaches: in the first we use a traditionally least-square method which is solved by an *ad-hoc* genetic algorithm (GA). In the second approach, we assume that a specific parameter is given, then an algebraic on-line parametric identification can be formulated. Although the two approaches require the non-available output's time derivatives (from 1st to 3rd) this inconvenient is overcome by using a practical high-gain observer (HGO), which works, as known, like an approximate differentiator (see [Parlitz, 1995]). The HGO is not based on the Luenberger observers, since it does not require an accurate model.

The remainder of this paper is organized as follows. Section 2 is devoted to studying some important algebraic properties of the Rössler's system. In Section 3, we introduce a simple HGO for estimating the time output derivatives; next, we develop two identification procedures based on the previously introduced algebraic properties. In the observer design and in the two identification schemes, we present computer simulation results depicting their performance. Section 4 is dedicated to giving the conclusions. Finally, in the Appendix we give a brief survey of GA and also present proofs of Propositions 1 and 2.

2 Rossler's System Properties

We consider the popular nonlinear Rössler's System (RS), which is described by

$$\begin{aligned}\dot{x} &= -(y + z) \\ \dot{y} &= x + ay \\ \dot{z} &= b + z(x - c)\end{aligned}\tag{1}$$

It is well known that in a large neighborhood of $\{a = b = 0.2, c = 5\}$ this system presents a chaotic behavior and is considered for exhibiting the simplest possible strange attractor ([Strogatz, 1994]). Originally, the RS, which is credited to Otto Rossler, arose from work in chemical kinetics ([Rössler, 1976]).

The fundamentals of this work are based on the algebraic properties of observability and identifiability that the RS satisfies (see [Martínez & Diop, 2004] and [Martínez & Mendoza, 2003]). The definitions of these are as follow.

Definition 1: Let us consider an undetermined system of ordinary differential equations

$$G(t, X, \dot{X}, P) = 0, \quad (2)$$

where $x^T = \{x_1, \dots, x_n\} \in R^n$ is a state vector and $P^T = \{p_1, \dots, p_l\} \in R^l$ is a constant parameter vector. Suppose that there exists a smooth, local and one to one correspondence between solution $X(t)$ of system (2) and an arbitrary function $y(t) = h(t, X(t)) \in R$, then, state x_i is said to be algebraically observable with respect to $y(t)$, if it satisfies the following algebraic relationship.

$$x_i = \frac{f_i(y, \dots, y^{(m)}, P)}{g_i(y, \dots, y^{(s)}, P)}; m \leq s \text{ with coefficients in } R \quad (3)$$

where f_i, g_i and h are smooth maps, $y^{(k)}$ is the k^{th} derivative of y and l, m are integers. Variable y is the output. If x_i is observable for every $i = 1, \dots, n$, then we say that the system is completely observable

Definition 2: Consider again system (2) under the same conditions of Definition 1. If we can find a smooth map $W : R^l \rightarrow R^l$ such that

$$0 = W(y, \dot{y}, \dots, y^{(j)}, P) \quad (4)$$

then, the parameter vector P is said to be algebraically identifiable with respect to the selected output

Next, we verify that RS satisfies the previous definitions when the output is selected as y . Evidently, this system is found to be algebraically observable with respect to the defined output. To see it, variables x and z can be expressed in the following

$$x = \dot{y} - ay \quad ; \quad z = -\ddot{y} + a\dot{y} - y \quad (5)$$

From \dot{z} we can obtain $y^{(3)}$

$$y^{(3)} = -b + (\ddot{y} - a\dot{y} + y)(\dot{y} - ay - c) + a\ddot{y} - \dot{y} \quad (6)$$

Therefore, system (1) is identifiable with respect to the output y because the last differential parameterization can be rewritten as,

$$0 = W(y, \dot{y}, \ddot{y}, y^{(3)}, P), \text{ with } P = [a, b, c] \quad (7)$$

3 States Estimation and Parameters Identification

Since RS is observable and identifiable, then a HGO is proposed to estimate the time derivatives, from the 1st to the 3rd, of the output. Moreover, it is possible to implement an identification scheme to recover the unknown vector P see [Martínez & Diop, 2004]). So, this section is devoted to solving both problems. Firstly, a HGO is proposed to tackle the time derivatives estimation problem. Then, using the observer's estimation of the derivatives, an identification process can be carried out. Finally, we suggest two alternatives for identifying vector P i) a traditionally least-square method and, ii) an algebraic on-line approach.

Before establishing the estimation and identification problems, the necessary assumptions are presented.

A1 The states of the nonlinear system (1) oscillate around zero. **A2** The set of variables $V_j = \{y(t_k), \dots, y^{(j)}(t_k)\}$ are available $t_k \in \mathcal{T} = (t_1, t_2, \dots, t_N)$, with a sampling time τ selected such that

$$\tau = t_{j+1} - t_j; \quad j = \{1, 2, \dots, n-1\} \quad (8)$$

Notice that A2 will be relaxed by means of an HGO, that is, we estimate with high accuracy the set V_j . Finally, we mention that although the RS identification has been considered by other authors such as [Baker *et al.*, 1996] and [Lainscsek & Gorodnitsky, 1996], they did not use the differential algebraic approach as we did.

3.1 A simple HGO

Based on the previous work of (see [Dabroom & Khalil, 1999]) and, Bonilla we propose the following HGO.

Let us define vector $Y' [y, \dot{y}, \ddot{y}, y^{(3)}]$ and let us propose the following filter given by:

$$\dot{\hat{Y}} = A\hat{Y} + HC(Y - \hat{Y}) \quad (9)$$

Where, A is the well-known Brunovsky form

$$H' = \begin{bmatrix} \frac{\alpha_1}{\varepsilon} & \frac{\alpha_2}{\varepsilon^2} & \frac{\alpha_3}{\varepsilon^3} & \frac{\alpha_4}{\varepsilon^4} \end{bmatrix}, C = [1 \quad 0 \quad 0 \quad 0] \quad (10)$$

ε is a small positive parameter and the positive constants α_i are selected such that the polynomial defined as:

$$p(s) = s^4 + \alpha_1 s^3 + \alpha_2 s^2 + \alpha_3 s + \alpha_4, \quad (11)$$

is Hurwitz (see [Dabroom & Khalil, 1999] for more details).

The following proposition allows us to compute the error $\xi = Y - \hat{Y}$.

Proposition 1: Consider the system (1) under assumptions A1. Then, the HGO proposed in (9) is able to recover Y with bounded error

$$\|\xi\| \leq \beta m \varepsilon / \lambda^* \quad (12)$$

Where λ^* is given by

$$\lambda^* = \min \{ \operatorname{Re}[\operatorname{roots}(p(s))] \}, \quad (13)$$

β is a positive constant which depends on the initial conditions $\xi(0)$, and $n = \max_{t \in [0, T]} |y^{(n)}(t)|$. **Proof:** The proof is in the Appendix.

Notice that constants ε and λ^* are design parameters which can be chosen in order to minimize the error of observation.

3.1.1 Numerical Simulations The efficiency of the HGO had been tested by computer simulations. The experiments were implemented by using the 4th-order Runge-Kutta algorithm. The computation was performed with a precision of 8 decimal digit numbers, from $t = 0$ second to $t = 10$ seconds. To obtain a good performance, the step size in the numerical method was set to 0.0005. The RS parameter values were set as

$a = 0.25$, $b = 0.2$ and $c = 4.2$, and the initial conditions were set as $x(0) = -1$, $y(0) = 1$ and $z(0) = 0$. The polynomial was chosen to be $p(s) = (s^2 + 2\zeta\omega_n s + \omega_n^2)$, with $\zeta = 0.0707$ and $\omega_n = 0.9$. The gain of the HGO was selected as $\varepsilon = 0.005$.

Figure 1 shows the error evolution of each output's time derivatives. As can be seen a very good estimation of $y^{(k)}$, $k = \{1, 2, 3\}$ is obtained.

3.2 Parameters Identification Based on Least-Squares

Under assumptions A1 and A2 a common quadratic function for estimating vector P from the differential relation (6) is presented as:

$$S(N) = \min_{p \in R^3} \sum_{i=0}^{N-1} (\tilde{y}^{(3)}(t_i, p) - \hat{y}^{(3)}(t_i))^2; \text{ with } t_i \in \mathcal{T} \quad (14)$$

where, symbol " $\tilde{y}^{(3)}(t_i, p)$ " denotes the parametric estimator of the 3rd time derivative of y given by

$$\tilde{y}^{(3)}(t_i, p) = -p_1 + (\ddot{y}(0) - p_0 \dot{y}(0) + y(0))(\dot{x}(0) - p_0 y(0) - p_2) + p_0 \ddot{y}(0) - \dot{y}(0) \quad (15)$$

with $p = [p_0, p_1, p_2] \in R^3$.

Notice that if we try to compute the critical points of relation (14), we need to solve three highly difficult nonlinear parametric equations with respect to parameters $\{a, b, c\}$. That is, obtaining analytically Rössler's parameter values is not possible if solely output y is available. Nevertheless, it is feasible to obtain an algebraic expression for parameters a, b and c as long as the Rössler's states x, y and z can be measured [G.L. Baker *et al.*, 1996]. To overcome these difficulties, we instead use a common GA to compute vector p which minimizes expression (14). This method is *ad-hoc*, because, we avoid the necessity of computing derivatives with respect to parameters and of finding the roots of the nonlinear parametric function, and we avoid the possibility of falling into a local minimum (see [Golberg, 1989], [Mitchel, 1998] and [Back *et al.*, 2000]). Finally, we recommend reading a brief description of the GA given in the Appendix in order to understand and to interpret the following numerical simulations.

3.2.1 GA Numerical Simulations To evaluate the efficiency of the proposed identifier method based on GA, a second computer simulation was carried out. Basically, we estimate numerically vector p such that expression (14) is minimized by means of a GA. The sampling time was selected $\tau = 0.1$, the number of samples was set to $N = 25$, the cost was selected $\alpha = 10^{-9}$. The samples were taken in the time interval from 6.5 to 9 seconds. Components of vector $p = [p_1, p_2, p_3]$ were searched for in a region centred on $\bar{p} = [0, 0, 0]$ with radius 7.

Figure 2 shows the process of error minimization. It can be seen that the error tends to zero when the generation number increases. Therefore, the actual and estimated parameters are very close, as shown in Table 1.

Generation	A	B	c	(16)
1	0.1789	0.0221	2.6051	
3	0.1952	0.3112	4.3115	
10	0.2108	0.1624	3.989	
32	0.2389	0.2005	4.1167	
100	0.2498	0.1999	4.2010	

Table 1. Best individual of some generations

3.3 Identification by solving algebraic linear equations

In this section we relax the identification problem a little in order to obtain an easier solution. Assuming that parameter a is known and the set V_y is available, it is possible to obtain a straightforward solution based on simple linear algebra.

Consider again the differential parameterization (6). This produces, after further time evaluation, the following system of linear equation for the missing parameters, b and c .

$$\begin{bmatrix} -1 & -\omega_1(t_i) \\ -1 & -\omega_1(t_i + \tau) \end{bmatrix} \begin{bmatrix} b \\ c \end{bmatrix} = \begin{bmatrix} \omega_2(t_i) \\ \omega_2(t_i + \tau) \end{bmatrix} \quad (17)$$

where $\omega_1(\cdot)$ and $\omega_2(\cdot)$ are defined as:

$$\begin{aligned} \omega_1(s) &= \ddot{y}(s) - a\dot{y}(s) + y(s) \\ \omega_2 &= y^{(4)}(s) + \dot{y}(s) - a\ddot{y} - \omega_1(s)(\dot{y}(s) - a\dot{y}(s)) \end{aligned} \quad (18)$$

The linear equation (17) allows us to recover the unknown parameters, b and c , after some time $t_i > 0$, that is to say, at $t = t_i + \tau$, $\tau > 0$, for which the matrix in (17) is invertible. As we mention in the following proposition:

Proposition 2: *Under assumptions A1 the matrix (17) becomes invertible for almost $t_i > 0$ and $\tau > 0$. Proof (Refer to Appendix).*

Another possibility to estimate the missing parameters can be done assuming that 4th derivative is available. Now, computing the time derivative of equation (6), this yields:

$$y^{(4)} - a\ddot{y} + \ddot{y} - (\ddot{y} - a\dot{y} + y)(\ddot{y} - a\dot{y}) - (\ddot{y} - a\ddot{y} + \dot{y})(\dot{y} - a\dot{y} - c) = 0 \quad (19)$$

Evidently, from equations (6) and (19) we can obtain directly the parameters b and c , respectively.

3.3.1 Numerical Simulations We illustrate the effectiveness of the previously described identification method by using inverse matrix. The initial conditions and the physical parameters were taken as indicated in the previous experiment, except for an abrupt change in the Rössler's parameter values b and c , from 0.2 to 0.35 and 4.2 to 3.5 when $t \geq 5$, respectively.

Figure 3 shows the estimation of parameters b and c based on inverse matrix. As a result, the identification process is quite robust with respect to abrupt parameter variations.

4 Conclusions

The differential algebraic approach allowed us to solve the identification problem for the well-known Rössler's attractor. In this instance, we exploited the algebraic properties of observability and identifiability that Rössler's model fulfils, with respect to a very particular available state which is the y - variable. Therefore, we could obtain a differential parameterization of the output and its time derivatives. Based on these facts, an HGO was used for estimating the output's time derivatives, and then, two identification approaches were designed based on the previously parameterization.

The identification approaches were tested by means of numerical experiments. In the first one, a traditionally least-square method was used and solved by a suitable GA, as shown in Table 1; in the second one, the missing parameters were computed in a straightforward way as an inverse matrix. The performance of the inverse matrix was validated in the presence of an abrupt variation in the missing parameters, as shown in figures 2 and 3.

Acknowledgements : This research was supported by the Coordinación de Posgrado e Investigación (CGPI-IPN), under research Grant 20040877

5 Appendix

A Brief Look at GA (we recommend reading [Bender & Orzag, 1999])

We introduce a traditional GA to minimize expression (14):

(i) Individuals in the GA are vectors (in R^3) of the form

$$q_i = [q_{0,i}, q_{1,i}, q_{2,i}] \quad (20)$$

It is understood that the GA is a real-coded one (as opposed to a binary-coded one).

(ii) Every population consists of 500 individuals.

(iii) The best individual, q_1 (evidently ranked 1st), in generation P_j is passed on to generation P_{j+1} , with no change.

(iv) Several steps are involved in the creation of generation P_{j+1} . They are as follows: *i*) selection; *ii*) crossover; *iii*) mutation.

i) The upper half of P_j passes on to population P'_{j+1} while the lower half is discarded. Note that P'_{j+1} holds just the best 250 individuals in P_j .

ii) To accomplish crossover couples, parents are generated as follows: each individual in P'_{j+1} is assigned a probability which is calculated linearly according to its ranking. Selection of individuals is made by generating random numbers in $[0,1]$ (say α_i) and comparing them to the accumulated probability, $Ap(q_i)$, of each individual. Individual q_i is selected to be part of P'_{j+1} when $\alpha_i \leq Ap(q_i)$. In a first round a member of each couple is selected while in a second round the other member is selected. The crossover algorithm used in this GA is a slight modification of the *flat crossover* (or *arithmetic crossover*) operator. An "offspring" $h = [h_0, h_1, h_2]$ is generated as

$$h_i = \beta q_{i,1} + (1 + \beta) q_{i,2} \quad (21)$$

from "parents"

$$q1 = [q_{0,1}, q_{1,1}, q_{2,1}] ; q2 = [q_{0,2}, q_{1,2}, q_{2,2}] \quad (22)$$

where q_1 is a better individual than q_2 (i.e. q_1 makes the error function smaller than q_2 does) and β is a random number chosen uniformly from the interval $[0.5, 1]$. This interval is used in order to weigh as more "influential" the information carried by the best of the parents. This process is repeated until a set P'_{j+1} with 250 "offspring" is completed. A new population is then created: $P''_{j+1} = P'_{j+1} \cup P''_{j+1}$.

(iii) The mutation algorithm consists of randomly changing a component of 15% of the individuals of P''_{j+1} . Changes are made within the vicinities specified below. This is the final step in creating generation P_{j+1} .

(v) The *cost* of each individual was computed via $S(N, \tau)$ where N is the number of samples and τ is a sampling time. The algorithm stops when the best individual tags a "cost" named α , where α is fixed as small as needed.

(vi) Components of vector $q = [q_0, q_1, q_2]$ were searched in a previously defined "box"; this means.

$$|q_i - q_i| \leq \varepsilon_i, \quad i = \{1, 2, 3\} \quad (23)$$

where q_i is the selected centre and ε_i is the radius which can be chosen as large as needed.

Proof of Proposition 1:

Evidently, vector Y can be written as:

$$Y = AY + \delta_y, \quad (24)$$

with $\delta_y = [0, 0, 0, y^{(4)}]$. Subtracting (24) from (9), we obtain the following differential equation of the error:

$$\dot{\xi} = [A - HC]\xi + \delta_y. \quad (25)$$

Notice that the characteristic polynomial of $A = A - HC$ is given by $p(s, \varepsilon)$, which is also Hurwitz. That is, the proposed H assigns the eigenvalues of A at $1/\varepsilon$ times the roots of $p(s)$ (11). Hence, the error ε satisfies

$$\varepsilon(t) = e^{A(t-t_0)} \left(\varepsilon(0) + \int_{t_0}^t e^{A(t_0-s)} \delta_y(s) ds \right). \quad (26)$$

Since A is exponentially stable and the signal $y^{(3)}$ is bounded, we also have the following inequality:

$$\xi \leq \beta e^{-\lambda^* t} \xi + \beta \eta \varepsilon (1 - e^{-\lambda^* t}) \lambda^* \rightarrow \beta \eta \varepsilon \lambda^* . \quad (27)$$

where the positives constants β, λ^*, η are previously defined in proposition 1.

Proof of Proposition 2:

Let $\Delta(t, \tau)$ be the determinate of matrix (17) which is given by $\Delta(t, \tau) = -\omega_1(t, \tau) + \omega_1(t)$. From the definition of variable $\omega_1(\cdot)$ given in (18) and taking into account system (17), we obtain after some simple algebra, the following:

$$\Delta(t, \tau) = -z(t, +\tau) + z(t). \quad (28)$$

Since the variable $z(t)$ oscillates around zero, we must have $-z(t, +\tau) + z(t) \neq 0$ for almost $t, > 0$ and $\tau > 0$. As the matter of fact $-z(t, +\tau) + z(t) = 0$ only in a finite set of time. $\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$. Therefore, matrix (17) is invertible for almost $t, > 0$ and $\tau > 0$.

Reference

- [1] K.T. Alligood, T.D. Sauer and J. A. Yorke, *Chaos: An Introduction to Dynamical Systems*, Springer-Verlag, New York, 1997.
- [2] Hand and M. Berthold, *"Intelligent Data Analysis: An Introduction"*, 2nd edition, Springer-Verlag, 2002.
- [3] T. Sauer, J. Yorke and M. Casdagli, *Embedology*, *J.Stat. Phys.* 65 (1991) 579-616.
- [4] F. Takens, *Detecting strange attractors in turbulence*, *Dynamical Systems and Turbulence*, eds. Rand, D.A. & Young, L.-S. Springer-Verlag, Berlin, (1981) 366-381.
- [5] Tokuda, U. Parlitz, L. Illing, M. Kennel and H. Abarbanel, *Parameter estimation for neuron models*, *Proc. of the 7th Experimental Chaos Conference*, San Diego, USA, 2002.
- [6] T. Stojanovski, L. Kocarev and U. Parlitz, *A simple method to reveal the parameters of the Lorenz system*, *International Journal of Bifurcation and Chaos*, Vol. 6, No. 12B, pp. 2645-2652, 1996.
- [7] U. Parlitz, R. Zöllner, J. Holzfuss and W. Lauterborn, *Reconstructing Physical Variables and Parameters From Dynamical Systems*, *International Journal of Bifurcation and Chaos*, vol. 4, (1994) 1715-1719.
- [8] U. Feldmann, M. Hasler and W. Schwarz, *Communication by chaotic signals: The inverse system approach*, *Int. J. Circuit Theory Appl.* 24 (1996) 551-579.
- [9] A. S. Poznyak, W. Yu, and E. N. Sanchez, *Identification and Control of Unknown Chaotic Systems via Dynamic Neural Networks*, *IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 46, No. 12, December 1999.

- [8] S. Poznyak, W. Yu, H. S. Ramirez, and E. N. Sanchez, Robust identification by dynamics neural networks using sliding mode learning, *Appl. Math. Comput. Sci.*, vol. 8, 101-110, 1998.
- [9] G. Cheng and X. Dong, Identification and control of chaotic systems, in *Proc. IEEE Int. Symp. Circuits Systems*, Seattle, WA, 1995.
- [10] M. S. Suárez-Castañon, C. Aguilar-Ibáñez and R. Barrón-Fernandez, On recovering the parameters and velocity state of the Duffing's oscillator, *Physics Letters A* 308 (2003) 47-53.
- [11] R. Martínez-Guerra and J. Mendoza-Camargo, Observers for a class of Liouvillian and nondifferentially flat systems, *IMA Journal Mathematical Control and Information*, 2004, In-press.
- [12] O.E. Rossler, An equation for continuous chaos, *Physics Letter*, Vol. 57A, pp.397-398, 1976.
- [13] S.H. Strogatz, *Nonlinear Dynamics and Chaos*, Perseus Publishing, 1994.
- [14] R. Martínez-Guerra and S. Diop, Diagnosis of nonlinear systems using an unknown-input observer: an algebraic and differential approach, *IEE Proc. Control Theory and Appl.*, Vol. 151, No.1, 2004.
- [15] G.L. Baker, J.P. Gollub and J.A. Blackburn, Inverting chaos: Extracting system parameters from experimental data, *American Institute of Physics, Chaos*, vol. 6, No. 4, 528-5633, 1996.
- [16] Lainscsek and I. Gorodnitsky, Reconstructing Dynamical Systems from Amplitude Measures of Spiky Time-Series, 8th Joint Symposium on Neural Computation at the salk Institute, La Jolla, California, May 19, 2001.
- [17] R. Martínez-Guerra, J. Mendoza-Camargo, Observers for a Class of nondifferentially flat Systems, *Proceedings of the IASTED International Conference, Circuits, Signals and Systems*, Cancun, Mexico, May 19-21, 2003
- [18] Goldberg, *Genetic Algorithms: Search and Optimization Algorithm*, Addison-Wesley Publishing, Massachusetts, 1989.
- [19] M. Mitchel, *An Introduction to Genetic Algorithms*, MIT Press, London, 1998.
- [20] T. Back, D. Fogel, Z. Michalewicz, *Evolutionary Computation 1: Basic Algorithms and Operators*, Institute of Physics Publishing, Bristol and Philadelphia, 2000.
- [21] M. Bender, S. A. Orzag, *Advanced Mathematical Methods for Scientists and Engineers*, Springer-Verlag, New York, 1999.
- A. M. Dabroom, H. K. Khalil, iscrete-time implementation of high-gain observers for numerical differentiation, *Int. J. Control*, Vol. 72, No. 17, 1523-1537, 1999
- [22] L. Kocarev, , K. S. Halle, K. Eckert, L. O. Chua, U. Parlitz, Experimental Demonstration of Secure Comunication via chaotic synchronization, *Int. J. Bifurcation and Chaos* 2(3), 709-713, 1992.
- [23] J. P. Crutchfield, B.S. McNamara, Equations of motions a data series, *Complex Systems*, 1(3), 417-52, 1987.
- [24] S. Broomhead, G. P. King, Extracting qualitative dynamic from experimental data, *Physica D*, 20:217, 1986.
- [25] J. F. Gibson, J. D. Farmer, M. Casdagli, S. Eubank, An analytic approach to practical state space reconstruction, *Physica D*, 57:1, 1992.
- [26] P. King, I. Stewart, Phase space reconstruction for symmetric dynamical systems, *Physica D*, 58:216-228, 1992.
- [27] M. Bonilla, M. Malabre, M. Fonseca, On the approximation of non-proper control laws, *Int. J. Control*, 68(4), 775-796, 1997.
- [28] U. Parlitz, Nonlinear time series analysis, *Procc. NDES*, July 28-29, Ireland, 1995.

List of Figures

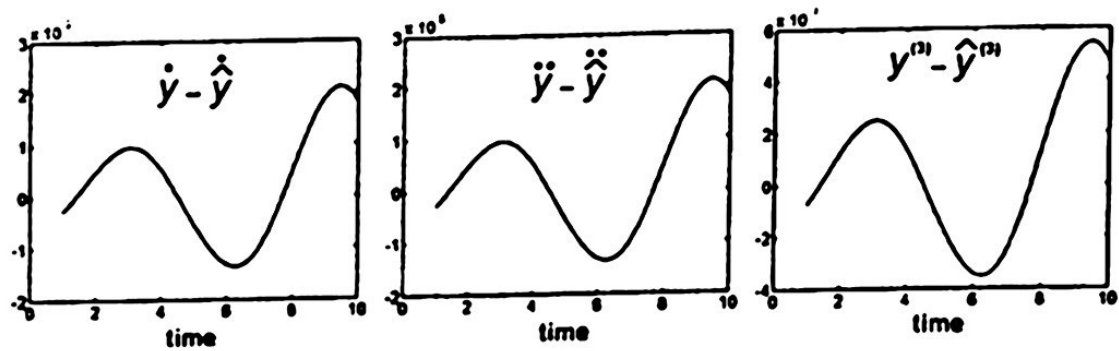


Fig. 1. The error evolution of each output's time derivatives.

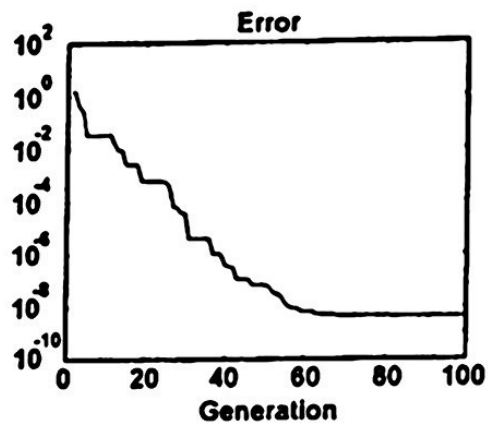


Fig. 2. Process of error minimization.

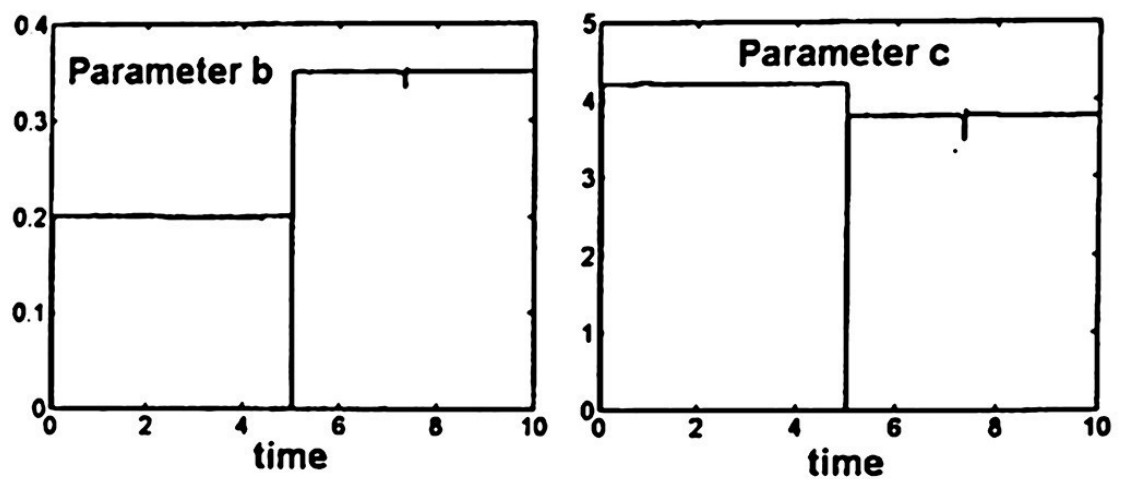


Fig. 3. The estimation of parameters b and c .

Efficient Lossless Data Compression using Advanced Search Operators and Genetic Algorithms

Angel F. Kuri¹, Oscar Herrera²

¹ Instituto Autónomo de México,

Río Hondo No. 1, México D.F.

akuri@itam.mx

Centro de Investigación en Computación,

Av. Juan de Dios Batiz s/n, México D. F.

heoscar@yahoo.com

Abstract. Over the last decades several techniques for lossless data compression have been proposed (such as RLE[1], Huffman Coding[2], Arithmetic Coding[3], LZ[4], BWT[5], and PPM[6]). They all assume or construct a model of the source which generates the message and the compression ratio depends strongly on how accurately the source adheres to the premises. In this paper we propose a new method for efficient lossless data compression in which the original message generated by a non-ergodic source is transformed into another in which it behaves as if it were generated by an ergodic one. This is achieved by identifying a set of groups which display ergodicity when looked upon as independent symbols. The identification of such groups represents an NP Problem[7][8][9] and we, therefore, define a set of Advanced Search Operators which, when combined with a Non Traditional Genetic Algorithm, allow us to find a set of non correlated groups of symbols to achieve the desired transformation and achieve maximum compression.

Keywords. Data Compression, RLE, Huffman, LZ, BWT, PPM, ergodic, NP Problem, Search, Genetic Algorithm.

1 Introduction

RLE is a simple data compression technique restricted to messages with strings of repeated symbols called "runs". When there is no "runs" RLE easily give us negative compression. More sophisticated techniques such as Huffman Coding and its parent the Arithmetic Coding are based on the Shannon Information Theory [10] that presents a study for ergodic sources where entropy represents the theoretical limit for data compression. The best coding of Huffman technique is achieved when the average length of codes reaches the entropy assuming that the receptor knows the dictionary. However, when we consider the structure of the dictionary as an additional header the "best coding" may give us a negative compression. Besides, Huffman Codes does not give us an acceptable compression in case of equiprobable symbols and is unable to compress bits streams.

Adaptive Dictionary techniques such as LZ build a dictionary with groups of adjacent symbols, in consequence, they assume that source is ergodic and the maximum compression ratio is restricted to messages with these characteristics.

PPM techniques calculate the conditional probability of next symbol based on k previous symbols. That implies to correlate the current symbol with the k symbols which precede it.

In our model we transform a message from a non-ergodic source to other which can be view as it were generated by an ergodic source consequently can be optimally coded with some technique that assumes a ergodic source and so we achieve maximum compression.

We intend to correlate as is possible the symbols in a group at the same time we intend to uncorrelated the groups one each other. That groups of symbols have been called metasymbols and they are discussed in section 2.

Since we do not define a priori the number of groups, neither the number of symbols that compose each one and symbols are not necessarily adjacent we have a NP problem[11]. To find metasymbols we have proposed to use:

1. A Non Traditional Genetic Algorithm. The fitness measure not only represents a theoretical limit of compression (as entropy does) but also it considers in an intrinsic way the size of the dictionary (contrary to the entropy). Fitness measure represents the number of bits used to fully encode a message (header plus encoded message), in consequence when the fitness is less than the original size of the message we have a positive compression.
 2. A set of Advanced Operators for Metasymbol Searching (AOMS).
- Both of them are discussed in sections 3 and 4.

2 Metasymbols

To facilitate the description of metasymbol we should conceive a message as an array of symbols indexed from 0 to $L-1$ where L is the length or number of symbols that constitute the message. Before describing a metasymbol is necessary to explain two kind of groups which appear when we find groups of symbols.

2.1 Master Groups and Slave Groups

For simplicity we can think that a metasymbol is a pattern in the message, a group is an instance of that pattern, a master group is the first instance we found in the message (from left to right) and a slave group is a repeated group that previously has appeared as master.

Groups are written as a set of symbols s_z^{ij} where subindex z means the position in the original message, superindex i is a number of group and superindex j indicates a relative index into the group.

Here are some examples:

Master group:

$$\{ x_0^{0,0}, y_1^{0,1}, z_3^{0,3} \}$$

Slave groups:

$$\begin{aligned} & \{x_4^{1,0}, y_5^{1,1}, z_7^{1,3}\} \\ & \{x_8^{2,0}, y_9^{2,1}, z_{11}^{2,3}\} \\ & \{x_{12}^{3,0}, y_{13}^{3,1}, z_{15}^{3,3}\} \end{aligned}$$

Metasymbol:

$$\{x_0, y_1, z_3\}$$

The difference between a master group and a metasymbol is that a metasymbol is a pattern without absolute indices referenced to the message. In a master group the subindices reference the position of the symbol in the original message, whereas, the subindices in a metasymbol represents the offset from the first symbol to any other into the metasymbol, that is why the first subindex always is 0.

Superindices j allows to compare groups to know whether a groups is slave of a master, i.e., all the superindices j of the two groups must match one to one.

The necessary number of metasymbols to codify a message is denoted by $|M|$, where $M = \{\alpha, \beta, \chi, \delta, \epsilon, \dots\}$. In what follows we use Greek letters to represent metasymbols. By definition, a metasymbol is considered different from another one if:

- The constituent symbols differ from those of any other metasymbol.

Example: $\alpha = a_0 b_1 c_2$

$$\beta = a_0 c_1 f_2$$

- They differ in the subindices of the symbols that compose them.

Example:

$$\alpha = a_0 a_1 a_2$$

$$\alpha = a_0 a_1 a_1$$

- They differ in the number of symbols that constitute the metasymbol, called in what follows, length of the metasymbol.

Example:

$$\alpha = a_0 b_1 c_2$$

$$\beta = a_0 b_2 c_1 d_3$$

$$\text{length}(\alpha) = 3$$

$$\text{length}(\beta) = 4$$

3. Non Traditional Genetich Algorithm: Vasconcelos Genetic Algorithm

The Vasconcelos Genetic Algorithm [12] use anular crossover. A variant of Vasconcelos has been proposed using PMX [13] proposed by Goldberg. Vasconcelos Genetic Algorithm is identified essentially because use a crossover between the best and the worst individuals in a population.

3.1 Coding of the genome

The genome is built as a permutation of the indices of the message using weight binary coding. For a message with length L we need $\log_2(L)$ bits for each index and $L \cdot \log_2(L)$ bits for the entire genome.

The groups in the genome are self-delimited because the indices of symbols which belongs to the same group appear ordered of ascending way (from left to right) and if appear an index less than its predecessor means a change of group.

We show an example for the message $x_0y_1A_2z_3x_4y_5B_6z_7x_8y_9C_{10}z_{11}$ where we have identified four metasymbols summarized in table 2 including the master and slaves groups associated to each metasymbol. The corresponding genome is:

1010 1000 1001 1011 0110 0100 0101 0111 0010 0000 0001 0011

Where is possible to identify the groups:

G1: 10

G2: 8,9,11

G3: 6

G4: 4,5,7

G5: 2

G6: 0,1,3

Comparing G2, G4 and G6 we know that G6 is a master group and its slaves are groups G2 and G4.

Groups G1, G3 and G5 are master groups because there is no repetition of them.

Table 1. Metasymbols, Master Groups and Slaves Groups for the message $x_0y_1A_2z_3x_4y_5B_6z_7x_8y_9C_{10}z_{11}$

Metasymbol	Master Groups	Slave Groups
$\alpha = x_0, y_1, z_3$	$x_0^{0,0}, y_1^{0,1}, z_3^{0,3}$	$x_4^{1,0}, y_5^{1,1}, z_7^{1,3}$ $x_8^{2,0}, y_9^{2,1}, z_{11}^{2,3}$
$\beta = A_2$	$A_2^{0,0}$	
$\chi = B_6$	$B_6^{0,0}$	
$\delta = C_{10}$	$C_{10}^{0,0}$	

3.2 Mutation

Note that change a bit from 0 to 1 or from 1 to 0 in the genome can produce an index out of range or duplicated, this is the reason to use other kind of mutation. Alternatively, we propose to permute two indices, so, we will never have indices out of range or duplicated and is no necessary to repair. Mutation is made with probability P_{mut} .

3.3 Crossover

Similarly to the mutation, annular crossover can produce indices out of range or duplicated, then crossover is implemented using PMX. Crossover is made with probability P_{cross} .

4 Advanced Operators for Metasymbol Searching

We define a set of operators which allows to manipulate a message in order to find metasymbols.

Definition. ϵ is a empty symbols used to indicate absence of symbol.

Arr(msg). Take a message and associate consecutive indices starting with 0 y ending with L-1.

|Arr(msg)|. The number of symbols different of ϵ contained in Arr(msg) .

$f_{Arr}(s_i)$. Calculate the frequency of symbol s_i in the array Arr.

$F_{Obj}()$. Calculate the distribution of frequencies of symbols contained in an object such as a matrix, an array, a column of a matrix, etc., and is represented by a set of pair $(s_i, f(s_i))$.

Roulette($F_{Obj}()$). Given a distribution of frequencies of symbols s_i select any s_i using proportional selection.

TH(ϵ). Return a random number between 1 and L inclusive which will act as a threshold for the minimum number of repetitions of a group.

Pos(s_i). Is an array of indices where symbol s_i appears in the array Arr.

Pos(s_i) $_k$. The k-th element of Pos(s_i).

Right(Arr, n_k). An array of size L which first n_k symbols are the last n_k symbols of Arr and the rest is completed with empty symbols ϵ . The subindices starts with 0 and finish with L -1. Each symbol has associated a pair of superindices (n_k, i) with $i=0,1,2,..., L-1$.

Vsi $_{pos(s_i)_k}$ = Right(Arr, n_k) where $n_k = L-pos(s_i)_k$

SMsi = { Vsi $_{pos(s_i)_k}$, $k=0,1,2,...,f(s_i)-1$ }

Sel(Cj^*, s_i^*) $_{SM}$. Because SMs $_i$ can be view as a matrix it means to get from SMs $_i$ the column Cj that contains the symbol s_i with the highest frequency, minimum j and that has not been selected previously. s_i^* means all the instances of symbol s_i contained in column Cj. $Pcol$ is the probability of selecting any other column and symbol which do not comply with the restrictions previously described.

| s_i^* | The number of instances of symbol s_i contained in column Cj

Elim(SMsi $s_{e1}, s_{e2}, ..., s_{en}$) = SMsi replacing s_{e1}^{ij} with a empty symbol ϵ_{e1}^{ij} and whenever s_{e1}^{ij} dont be the j-th element of V $_k$ in SM $_A$

Elim(Arr $s_{e1}, s_{e2}, ..., s_{en}$) = Arr replacing s_{e1}^{ij} with ϵ_{e1}^{ij}

Elim(Arr $pos(s_i)$) = SM $_A$ replacing $s_{pos(s_i)_k}^{ij}$ with ϵ_{e1}^{ij} where $k=0,1,2,...,f(s_i)-1$

Restrict(SM $_A$ $s_{e1}, s_{e2}, ..., s_{en}$) = { V $_k$ of SM $_A$ such that e_i belongs to V $_k$ }

AddColumn(Cj^*). Each time we apply Sel(Cj^*, s_i^*) $_{SM}$ we get a column and its number of column j which are stored in an array of columns.

EmptyArr(ϵ). Reset the array of columns Cj^* .

MatrixofMasterAndSlaves(ϵ). After apply several times Sel(Cj^*, s_i^*) $_{SM}$ we get an array of columns that is possible to order under j in ascendant order and join them to construct a matrix. That matrix contains master and slaves groups as rows.

GetMasterAndSlaves(MatrixofMasterAndSlaves). Given a matrix with groups is possible to identify the master group comparing the first subindex of all of them. The group which have the minimum subindex is the master group.

GetMetasymbol(). Given a master group is possible to get a metasymbol in the next way:

Master group:

$$\{ s_{x0}^{ij0}, s_{x1}^{ij1}, s_{x2}^{ij2}, \dots, s_{xm}^{ijm} \}$$

Metasymbol:

$$\{ s_{j0}, s_{j1}, s_{j2}, \dots, s_{jn} \}$$

Once defined the operators we can describe the Algorithm for searching metasymbols in a compact way.

1. *Arr(msg)*.
2. *EmptyArr()*.
3. *F_{Arr}()*.
4. *DELIM = Roulette(F_{Arr}())*.
5. *Pos(DELIM)*.
6. *SM_{si}*
7. *th = TH()*
8. *AddColumn(0)*
9. *for i=0 to L-1 do F_{Cj}*
10. *Sel(Cj*, s_i*)_{SM_{si}}^{Pcat}*
11. *if |s_i*| < Th go step 17*
12. *AddColumn(Cj*)*.
13. *Elim(SM_{si} s_i*)_j*
14. *Elim(Arr s_i*)*
15. *SM_{si} = Restrict(SM_{si}, s_{e1}, s_{e2}, ..., s_{em}) = {V_k of SM_A such that e_i belongs to V_k}*
16. *repeat step 9 to 14*
17. *MatrixofMasterAndSlaves()*.
18. *GetMasterAndSlaves(MatrixofMasterAndSlaves)*.
19. *GetMetasymbol()*.
20. *Repeat step 2 to step 19 while |Arr(msg)| > 0*

This Algorithm has been summarized in a Operator Called *CatastrophicMutation()* used in the Genetic Algorithm. *CatastrophicMutation* is made with probability *Pcat*.

In the next example we illustrate how is possible to find metasymbols from the message *xyAzxyBzxyCz*.

Given the message *msg = "xyAzxyBzxyCz"*

L=12

Arr=x₀y₁A₂z₃x₄y₅B₆z₇x₈y₉C₁₀z₁₁

F = {(x,3),(y,3),(A,1),(z,3),(B,1),(C,1)}

Pos(x) = {0,4,8}

V_{x0} = Right(Arr,12)

V_{x4} = Right(Arr,8)

V_{x8} = Right(Arr,4)

$$SM_{si} = \left\{ \begin{array}{cccccccccccc} x_0^{0,0} & y_1^{0,1} & A_2^{0,2} & z_3^{0,3} & x_4^{0,4} & y_5^{0,5} & B_6^{0,6} & z_7^{0,7} & x_8^{0,8} & y_9^{0,9} & C_{10}^{0,10} & z_{11}^{0,11} \\ x_4^{1,0} & y_5^{1,1} & B_6^{1,2} & z_7^{1,3} & x_8^{1,4} & y_9^{1,5} & C_{10}^{1,6} & z_{11}^{1,7} & \epsilon^{1,8} & \epsilon^{1,9} & \epsilon^{1,10} & \epsilon^{1,11} \end{array} \right\},$$

$\{x_8^{2,0} y_9^{2,1} C_{10}^{2,2} z_{11}^{2,3} \epsilon^{1,4} \epsilon^{1,5} \epsilon^{1,6} \epsilon^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\}$
 $\}$
 Let Th=3
 AddColumn(0)
 $F_{C\sigma}=\{(x,3)\}$, $F_{C\tau}=\{(y,3)\}$, $F_{C\gamma}=\{(A,1),(B,1),(C,1)\}$, $F_{C\delta}=\{(z,3)\}$, $F_{C\epsilon}=\{(x,2)\}$,
 $F_{C\zeta}=\{(y,2)\}$, $F_{C\eta}=\{(B,1),(C,1)\}$, $F_{C\theta}=\{(z,2)\}$, $F_{C\kappa}=\{(x,1)\}$, $F_{C\lambda}=\{(y,1)\}$, $F_{C\mu}=\{(C,1)\}$,
 $F_{C\nu}=\{(z,1)\}$

$Sel(Cj^*, s_i^*)_{SMsi}^{I_{cnd}=0}$ returns $Cj^*=1, s_i^*=y_{1,5,9}$

AddColumn(1).

Elim(SMsi s_i^*)_i

$SMsi=\{ \{x_0^{0,0} y_1^{0,1} A_2^{0,2} z_3^{0,3} x_4^{0,4} \epsilon_5^{0,5} B_6^{0,6} z_7^{0,7} x_8^{0,8} \epsilon_9^{0,9} C_{10}^{0,10} z_{11}^{0,11}\},$
 $\{x_4^{1,0} y_5^{1,1} B_6^{1,2} z_7^{1,3} x_8^{1,4} \epsilon_9^{1,5} C_{10}^{1,6} z_{11}^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\},$
 $\{x_8^{2,0} y_9^{2,1} C_{10}^{2,2} z_{11}^{2,3} \epsilon^{1,4} \epsilon^{1,5} \epsilon^{1,6} \epsilon^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\}$
 $\}$

Elim(Arr s_i^*). Arr= $x_0 \epsilon_1 A_2 z_3 x_4 \epsilon_5 B_6 z_7 x_8 \epsilon_9 C_{10} z_{11}$

$SM_{si}=Restrict(SM_{si}, s_{c1,c2,...,cm})$

$SMsi=\{ \{x_0^{0,0} y_1^{0,1} A_2^{0,2} z_3^{0,3} x_4^{0,4} \epsilon_5^{0,5} B_6^{0,6} z_7^{0,7} x_8^{0,8} \epsilon_9^{0,9} C_{10}^{0,10} z_{11}^{0,11}\},$
 $\{x_4^{1,0} y_5^{1,1} B_6^{1,2} z_7^{1,3} x_8^{1,4} \epsilon_9^{1,5} C_{10}^{1,6} z_{11}^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\},$
 $\{x_8^{2,0} y_9^{2,1} C_{10}^{2,2} z_{11}^{2,3} \epsilon^{1,4} \epsilon^{1,5} \epsilon^{1,6} \epsilon^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\}$
 $\}$

$F_{C\sigma}=\{(x,3)\}$, $F_{C\tau}=\{(y,3)\}$, $F_{C\gamma}=\{(A,1),(B,1),(C,1)\}$, $F_{C\delta}=\{(z,3)\}$, $F_{C\epsilon}=\{(x,2)\}$,
 $F_{C\zeta}=\{\}$, $F_{C\eta}=\{(B,1),(C,1)\}$, $F_{C\theta}=\{(z,2)\}$, $F_{C\kappa}=\{(x,1)\}$, $F_{C\lambda}=\{\}$, $F_{C\mu}=\{(C,1)\}$,
 $F_{C\nu}=\{(z,1)\}$

$Sel(Cj^*, s_i^*)_{SMsi}^{I_{cnd}=0}$, returns $Cj^*=3, s_i^*=y_{3,7,11}$

AddColumn(3).

Elim(SMsi s_i^*)_j

$SMsi=\{ \{x_0^{0,0} y_1^{0,1} A_2^{0,2} z_3^{0,3} x_4^{0,4} \epsilon_5^{0,5} B_6^{0,6} \epsilon_7^{0,7} x_8^{0,8} \epsilon_9^{0,9} C_{10}^{0,10} \epsilon_{11}^{0,11}\},$
 $\{x_4^{1,0} y_5^{1,1} B_6^{1,2} z_7^{1,3} x_8^{1,4} \epsilon_9^{1,5} C_{10}^{1,6} \epsilon_{11}^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\},$
 $\{x_8^{2,0} y_9^{2,1} C_{10}^{2,2} z_{11}^{2,3} \epsilon^{1,4} \epsilon^{1,5} \epsilon^{1,6} \epsilon^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\}$
 $\}$

Elim(Arr s_i^*). Arr= $x_0 \epsilon_1 A_2 \epsilon_3 x_4 \epsilon_5 B_6 \epsilon_7 x_8 \epsilon_9 C_{10} \epsilon_{11}$

$SM_{si}=Restrict(SM_{si}, s_{c1,c2,...,cm})$

$SMsi=\{ \{x_0^{0,0} y_1^{0,1} A_2^{0,2} z_3^{0,3} x_4^{0,4} \epsilon_5^{0,5} B_6^{0,6} \epsilon_7^{0,7} x_8^{0,8} \epsilon_9^{0,9} C_{10}^{0,10} \epsilon_{11}^{0,11}\},$
 $\{x_4^{1,0} y_5^{1,1} B_6^{1,2} z_7^{1,3} x_8^{1,4} \epsilon_9^{1,5} C_{10}^{1,6} \epsilon_{11}^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\},$
 $\{x_8^{2,0} y_9^{2,1} C_{10}^{2,2} z_{11}^{2,3} \epsilon^{1,4} \epsilon^{1,5} \epsilon^{1,6} \epsilon^{1,7} \epsilon^{1,8} \epsilon^{1,9} \epsilon^{1,10} \epsilon^{1,11}\}$
 $\}$

MatrixofMasterAndSlaves().

$Cols=\begin{bmatrix} x_0^{0,0} \\ x_4^{1,0} \\ x_8^{2,0} \end{bmatrix} \begin{bmatrix} y_1^{0,1} \\ y_5^{1,1} \\ y_9^{2,1} \end{bmatrix} \begin{bmatrix} z_3^{0,3} \\ z_7^{1,3} \\ z_{11}^{2,3} \end{bmatrix}$

GetMasterAndSlaves(MatrixofMasterAndSlaves).

Master group: $\{x_0^{0,0} \ y_1^{0,1} \ z_3^{0,3} \}$

Slave groups: $\{x_4^{1,0} \ y_5^{1,1} \ z_7^{1,3} \}$
 $\{x_8^{2,0} \ y_9^{2,1} \ z_{11}^{2,3} \}$

GetMetasymbol().

$\{x_0 \ y_1 \ z_3\}$

In a similar way we can find the metasymbols $\{A_0\}$, $\{B_0\}$ and $\{C_0\}$.

5 Experiments

Several experiments have been tested to probe the efficiency of the algorithm, here we show the result of a experiment with the message

"AAAAAAAAAABBBBBBBBBBCCCCCCCCCDDDDDDDDDEEEEEEEEEEE
 FFFFFFFFFFGGGGGGGGGGGHHHHHHHHHHHHHHHHHHHHHHJJJJJJJJKKKKKKKKKKK
 LLLLLLLLLLLMMMMMMMMMMMMNNNNNNNNNNNOOOOOOOOOOOPPPPPPPPP
 PQQQQQQQQQRRRRRRRRRRSSSSSSSSSSTTTTTTTTTTTUUUUUUUUUUUV
 VVVVVVVVVVWWWWWWWWWWWXXXXXXXYYZZZZZZ
 ZZZZzaaaaaaaaabbbbbbbbbbccccccccddddddeeeeeeeefffffiigggggggggg
 hhhhhhhhhhhiiiiiiijjjjjjjkkkkkkkkkkllllllllmmmmmmmmnnnnnnnnnnnoooooo
 oooooppppppppppqqqqqqqqqrrrrrrrrssssssssstttttttuuuuuuuuuuuvvvvvvvvvvwww
 wwwwwwwwwxxxxxxxxxyyyyyyyyyyyzzzzzzzzzz"

than can be compressed successfully with RLE whereas Huffman Technique provides negative compression (-1.44 %).

Our method find just one metasymbol α , where

$\alpha = \{A_0, B_{10}, C_{20}, D_{30}, E_{40}, F_{50}, G_{60}, H_{70}, I_{80}, J_{90}, K_{100}, L_{110}, M_{120}, N_{130}, O_{140}, P_{150}, Q_{160}, R_{170}, S_{180}, T_{190}, U_{200}, V_{210},$
 $W_{220}, X_{230}, Y_{240}, Z_{250}, a_{260}, b_{270}, c_{280}, d_{290}, e_{300}, f_{310}, g_{320}, h_{330}, i_{340}, j_{350}, k_{360}, l_{370}, m_{380}, n_{390}, o_{400}, p_{410}, q_{420}, r_{430}, s_{440}, t_{450},$
 $u_{460}, v_{470}, w_{480}, x_{490}, y_{500}, z_{510}\}$

that is repeated 10 times. Now, the message encoded with metasymbols looks like $\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha\alpha$, and we can compress up to 84.28 % using Huffman Coding and including the dictionary. A full description about the implementation of this method can be found in [14].

6 Conclusions

The selecting of the basic symbols (called metasymbols) in a message is important because determines the maximum compression ratio we can achieve. We showed that is possible to find metasymbols and apply a entropy coder to the new message to get better compression ratios.

Advanced Search Operators combined with Vasconcelos Genetic Algorithm have been used successfully to find patterns in an approach to the associated NP problem. Because we do not define a priori the number of metasymbols, neither the number of symbols which constitute them, we think in a kind of unsupervised clustering which can be of interest in data mining.

7 References

- [1] Nelson, M., Jean L., (1995). *The Data Compression Book*, Second Edition, M&T Books Redwood City, CA.
- [2] Huffman, D. A. (1952), *A method for the construction of minimum-redundancy codes*. Proc. Inst. Radio Eng. 40 (9): 1098-1101.
- [3] Witten, I. H., R. Neal, and J. G. Cleary, (1987), *Arithmetic coding for data compression*, Communications of the ACM 30 (6): 520-540.
- [4] Ziv, J., and Lempel, A., (1977), *A Universal Algorithm for Sequential Data Compression*. IEEE Trans. on Inf. Theory IT-23.3, 337-343.
- [5] Burrows M., and Wheeler, D. J., (1994), *A block-sorting lossless data compression algorithm*, Digital Syst. Res. Ctr., Palo Alto, CA, Tech. Rep. SRC 124.
- [6] Cleary, J. G. and Witten, I. H. (1984), *Data compression using adaptive coding and partial string matching*. IEEE Transactions on Communications, Vol. 32, No. 4, 396-402.
- [7] Paturi, R., Rajasekaran, S., and Reif, J.H. (1989), *The Light Bulb Problem*. In Second Workshop on Computational Learning Theory.
- [8] Steven, P. (1996), *A Hybrid Local Search Algorithm for Low Autocorrelation Binary Sequences*. Technical Report, Department of Computer Science, National University of Ireland at Cork.
- [9] Auluck, F.C. and D.S. Kothari, (1946), *Statistical mechanics and the partitions of numbers*, Proceedings of the Cambridge Philosophical Society 42.
- [10] Shannon, C.E., (1948), *A Mathematical Theory of Communication*, Bell Sys. Tech. J. 27, 379-423, 623-656.
- [11] Kuri, A., Herrera O., (2003), *Metrics for Symbol Clustering from a Pseudoergodic Information Source*, IEEE Proceedings of ENC2003.
- [12] Kuri, A., (1999), *A Comprehensive Approach to Genetic Algorithms in Optimization and Learning*. Editorial Politécnico.
- [13] Goldberg, D., (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing. Company.
- [14] Kuri, A., Herrera, O., (2003), *Lossless Scheme for Data Compression using Metasymbols*, Memorias del XII Congreso Internacional de Computación - CIC2003.

Networks and Distributed Systems

.

BPIMS-WS: Service-Oriented Architecture for Business Processes Management

Giner Alor Hernández, José Oscar Olmedo Aguirre

Research and Advanced Studies Center of IPN (CINVESTAV).

Electrical Engineering Department. Computing Section.

Av. Instituto Politécnico Nacional 2508, Col San Pedro Zacatenco. 07360 México, D. F.

e-mail: gineralor@computacion.cs.cinvestav.mx, oolmedo@delta.cs.cinvestav.mx

Abstract. Service-Oriented Architecture (SOA) development paradigm has emerged to improve the critical issues of creating, modifying and extending, solutions for enterprise application integration, process automation and automated exchange of information between organizations. Web services technology follows the SOA's principles for developing and deploying applications. Besides, Web services are emerging as the platform for service-oriented architecture (SOA), for both intra- and inter-enterprise communication. In this work, a business processes integration and monitoring system has been developed to automate, integrate and monitor many of the enterprise business processes described as Web services without recurring to large investments in software development and deployment. The contribution of this work consists in a service-oriented architecture that follows the SOA's principles of improving economical benefits of business collaborations. Our service-oriented architecture is based on a layered design that meet crucial design aspects like abstraction, scalability and interoperability.

Keywords: BPEL4WS, Business Process Integration, Service-Oriented Architecture, Web Services.

1 Introduction

Recently, the Service-Oriented Architecture (SOA) development paradigm has emerged to focus on radically improving the efficiencies of creating, modifying and extending solutions for enterprise application integration and process automation between organizations. SOA redefines the concept of an application from being an opaque procedural implementation mechanism to that of an orchestrated sequence of messaging, routing and processing of events. Web services are becoming the dominant technology for developing and deploying applications following the SOA's principles so that the platform and language independent interfaces of web services allow the integration of heterogenous systems. Also, a growing number of commercial enterprises are redefining their business processes under this technology.

Therefore, business process management based on the SOA paradigm facilitates the design, analysis, optimization of business processes. It achieves this by separating process logic from the applications that run them, managing the relationships among

process participants, integrating internal and external process resources, and monitoring process performance. Having this into account, we have developed a system named BPIMS-WS (Business Processes Integration and Monitoring System based on Web Services) to enable the integration, composition and monitoring of the processes developed by the business partners involved. Furthermore, BPIMS-WS offers additional functionality for the dynamic integration of enterprises, discovery and invocation of business processes accessible as Web services.

The rest of this paper is structured as follows. In the next section we provide the main characteristics of BPIMS-WS. In the following sections we present the architecture proposed and discuss its design principles. Next, we describe a case of study for describing the functionality of BPIMS-WS. Then we review the related work in this area and emphasize the contributions of our work.

2 BPIMS-WS

BPIMS-WS contains a UDDI [1] node where commercial enterprises, services and products are registered. For the classification of business processes, products and services in the repository, BPIMS-WS uses broadly accepted ontologies like NAICS [2], UNSPSC [3] and RosettaNet [4]. In a similar way to the functionality provided by a UDDI [1] node, Web services can be registered, published and discovered in BPIMS-WS. This UDDI node is the main component of our service-oriented architecture. As Web services proliferate inside the enterprise, there is a need to advertise, discover and reuse services, and UDDI is the standards-based mechanism for doing this. Additionally, BPIMS-WS enable the composition of Web services. These types of Web services are created, instantiated and executed dynamically in a BPEL4WS [5] engine. Also, BPIMS-WS comprises enterprise intra-workflow and inter-workflow Web services. Enterprise intra-workflow Web services are structured orchestrations of composite Web services that describe the internal activities developed as the intended behavior of an enterprise. We have considered the design of Enterprise intra-workflow Web services so that they can be created and instantiated in the third layer of our service-oriented architecture and executed by a WS-CDL [6] engine. Enterprise inter-workflow Web services describe the orchestration of the long-running conversation behavior of enterprise intra-workflows. In a similar way, Enterprise inter-workflow Web services have been considered so that they can be created, instantiated and executed in the fourth layer.

The BPIMS-WS service-oriented architecture has a layered design following three SOA's principles: (1) Integration, (2) Composition, and (3) Monitoring. The business processes integration is offered in the first layer of BPIMS-WS architecture. The creation of new Web services through the composition of existing business processes described as Web services is offered in the second layer of BPIMS-WS architecture. The monitoring business processes is provided starting from second layer. In the following sections, we describe with more detail the functionality of each layer of our service-oriented architecture proposed.

According to the emphasis on automation, BPIMS-WS can be accessed in two modes of interaction, either as a proxy server or as an Internet portal. In the first mode, BPIMS-WS can interoperate with other systems or software agents. Like a proxy, BPIMS-WS receives XML-encoded requests that are completed with business partner binding information, and then forwarded to the corresponding enterprise workflows. Eventually the responses to the requests are received back from the business partners' workflows and then replied to the requesters. In the second mode, BPIMS-WS acts as an Internet portal that provides to the users a range of options among the Web services available through the brokering system. In this mode, BPIMS-WS presents to the users diverse GUIs to get access to the Web services provided. The Internet portal is executed on a Tomcat application server.

3 BPIMS-WS Architecture

The BPIMS-WS service-oriented architecture has a layered design. The basic functionality of BPIMS-WS is situated in the bottom layer, while the more complex functionality is situated in the upper layer. Like in other layered architectures, the purpose of each layer is to provide the access to the Web services required by the upper layers, hiding the details of how the Web services are implemented. The layers are abstracted in such a way that each enterprise workflow or software agent communicates with any counterpart of the upper layers. In this context, each layer has a defined function as explained in that follows. Fig. 1 shows the general architecture of BPIMS-WS. In Fig.1 each layer has a well-defined function, as it is briefly described next.

3.1. Web Service Layer

The business processes integration is provided in this layer. For doing this, we have developed an integration brokering service that allows to publish and discover Web services to make business processes integration. For the discovery of Web services, the integrating brokering service uses the UDDI node. A set of simple Web services are contained in the integration brokering service. The set of simple Web services consults the information stored in the UDDI node using an ODBC/JDBC compliant database. The simple Web services consist of the following basic operations:

- 1) **Web Services Registry** comprises operations intended to store information in the UDDI node about: (i) potential businesses partners (businessName, description, discoveryURL, contactName, phone and e-mail), (ii) products (productCode), and (iii) services (serviceName, description, accessSOAP, accessWSDL and serviceCode). Examples of these Web services are save_Business, save_Services and save_Products.
- 2) **Web Services Search** consists of operations deemed to search for product technical information. Examples of these Web services are: get_Quantity, get_Price,

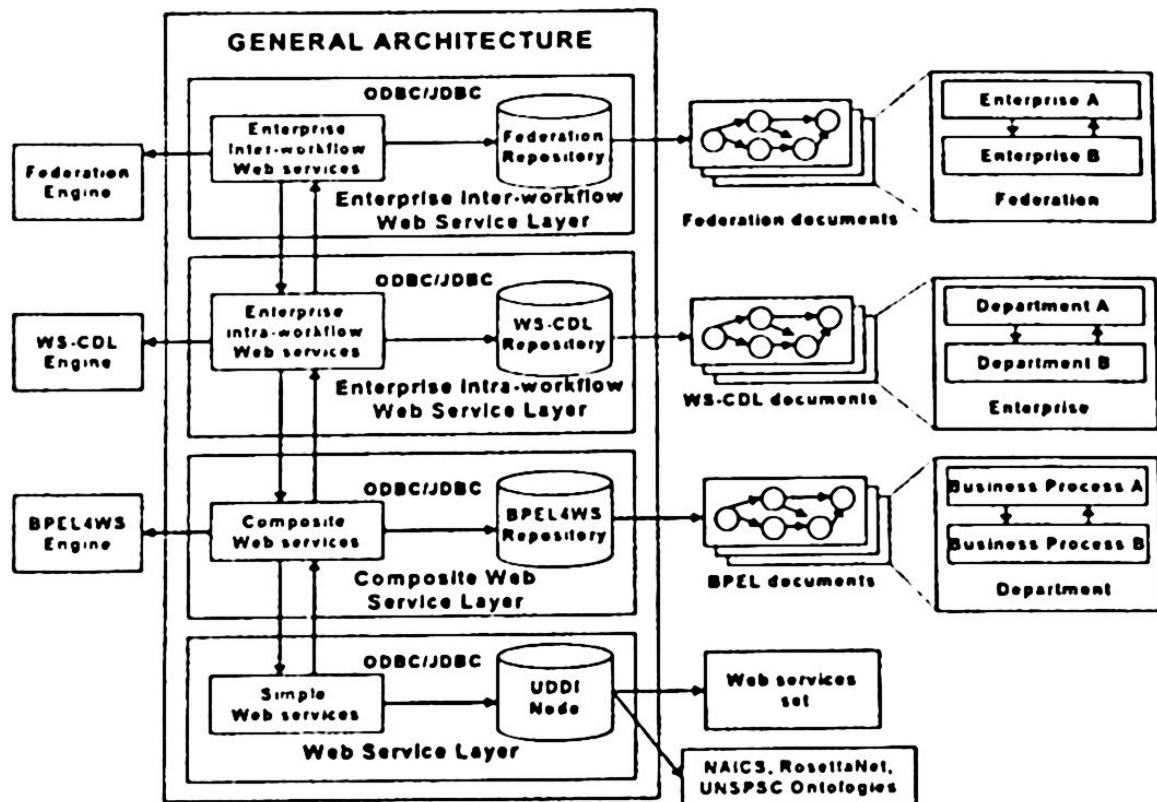


Fig. 1 General Architecture of BPIMS-WS

3) Web Services Meta-Information includes operations intended to retrieve both services information and BPIMS-WS meta-information. Examples of these Web services are `get_BusinessOntology`, `get_ServicesOntology`, `get_ProductsOntology`, `get_ServicesList`, `get_RegisteredBusiness`, `find_Product`, `find_Service`, `find_Business` and `get_PasswordRecovery` [7].

The structure and behavior of this layer can be understood with the following example. Assume that a client is willing to find the technical information available about her preferred product. First, the client must select the type of the product she wants from a range of options offered through the Internet portal (Step 1 in Fig 2). Then, BPIMS-WS obtains the request and formulates a query to the UDDI node. The result to the query is a list of all the suppliers that have the requested product in their stocks (Step 2 in Fig 2). Then, for each one of these suppliers, BPIMS-WS formulates another query to the UDDI node to retrieve the corresponding URL, that contains the Web service specification corresponding to RosettaNet's PIP 2A5 (Query Technical Information). Once located the URL, BPIMS-WS builds requests for the invocation of the associated Web services to the enterprises found. Then, BPIMS-WS sends those requests to the enterprises and to obtain later on the responses (Step 3 in Fig 2). Next BPIMS-WS extracts the required information and builds a XML document. This XML document is presented in HTML using the Extensible Stylesheet Language (XSL) (Step 4 in Fig 2). The answer contains information concerning to the product (according to the invoked Web service) and the electronic address (discoveryURL) of

the enterprise that offers that product. By means of using simple Web services, a client can get the price, the delivery time or the quantity available in stock of any registered enterprise in the UDDI node. Additionally, BPIMS-WS can also work based on more elaborated searching criteria [8]. In summary, software agents and applications can get access to all the Web services that have been registered. The Web service layer of the architecture of BPIMS-WS is shown in Fig. 2.

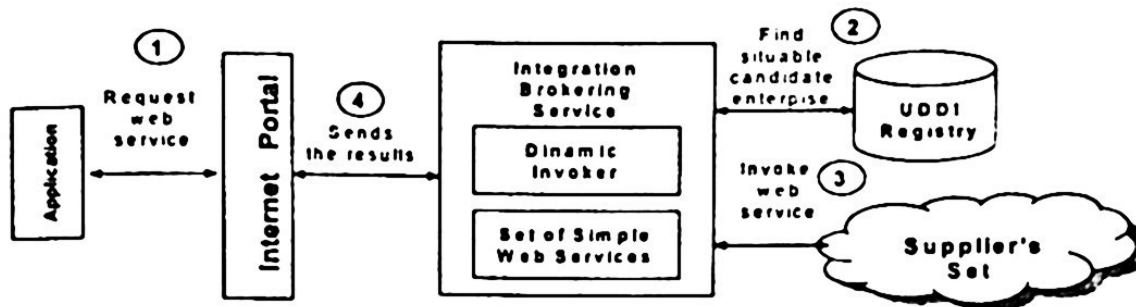


Fig. 2 Architecture of BPIMS-WS in the Web Services layer

3.2. Composite Web Services Layer

The support for composite Web services is provided in this layer. A composite Web service is the orchestration of several simple Web services. Composite Web services can be created at both design and execution time. The architecture of BPIMS-WS in the composite web services layer is shown in Fig 3.

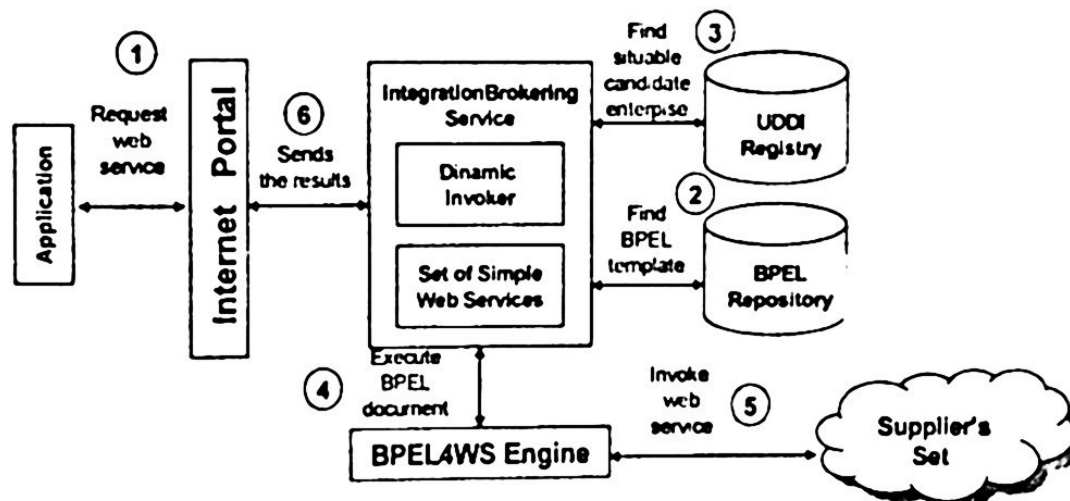


Fig. 3 Architecture of BPIMS-WS in the Composite Web services Layer

For the execution of a composite Web service is firstly necessary to locate a suitable template from the BPEL4WS repository that describes the intended commercial activities [9]. In this schema, the templates are completely determined since commer-

cial partners are known before hand. Next, BPIMS-WS retrieve from a database the location of the BPEL4WS workflow template that uses the commercial activity (Step 2 in Fig 3). Once the template is located, BPIMS-WS uses the WSDL document and all related configuration files in order to instantiate them. BPIMS-WS obtains the templates that can be used to find the suppliers that offer the products required by the commercial collaboration. A query to a database containing the WSDL documents provided by BPIMS-WS can retrieve the appropriate Web services to obtain a number of pieces of commercial information like price, delivery time, quantity, and purchase access point of the products (Step 3 in Fig 3). These services, based on UNSPSC and RosettaNet ontologies, are `get_PriceandDeliveryTime`, `get_ProviderQuantity`, and `get_ProviderURLBuy`, respectively. The related WSDL documents are then analyzed, and all the relevant information is retrieved and used to complete the templates.

The instantiated templates are allocated in a BPEL4WS engine for execution (Step 4 in Fig 3). To communicate with the running workflow, BPIMS-WS builds SOAP messages containing the information provided by the client. Here, the client sends to the running workflow, the information necessary to run the workflow such as product code and the required quantity in a SOAP message. The workflow verifies the constraints and pre-conditions and then it is executed. (Step 5 in Fig 3). Whenever the workflow has been successfully terminated, it sends back to the client the list of suppliers satisfying the conditions (Step 6 in Fig 3). Then, the workflow is de-allocated from the workflow engine. After the client selects the suppliers, a BPWL4WS template for placing a purchase order is now retrieved from the repository, completed and executed as described before. By enacting this workflow, the purchase orders are sent to the suppliers and the corresponding answers from each supplier are eventually received.

A wide variety of other composite Web services involving some optimization criteria have also been developed and tested like minimum delivery time, distributed purchase, etc.

3.3. Enterprise intra-workflow Web service layer

This layer comprises a repository of WS-CDL documents [6] containing a set of enterprise intra-workflow Web services. A WS-CDL document represents an enterprise intra-workflow. An enterprise intra-workflow defines the behavioral aspects and the dependency relationships among the diverse entities that constitute the enterprise. An entity can be a billing department or a marketing department. The enterprise intra-workflow Web services are orchestrations of composite Web services. This orchestration is based on the WS-CDL model. In order to orchestrate the interaction of two or more enterprise workflows, a similar approach to the one described in the Composite Web service layer is followed in this layer that consists on the instantiation of generic process descriptions obtained from the WS-CDL repository and their further execution in the WS-CDL engine.

3.4 Enterprise inter-workflow Web service layer

This layer comprises a repository of inter-workflows descriptions and a set of enterprise inter-workflow Web services. An inter-workflow of an enterprise federation defines the policies and commercial ideologies along some governmental regulations dictated on the enterprises that constitute the federation. A federation can be a group of diverse enterprises that synergistically collaborate in pursuing common or complementary goals. The enterprise inter-workflow Web services integrates enterprise intra-workflow Web services oriented to the satisfaction of society needs such as to reduce the consumption of non-renewable natural resources or to increase food production given a limited budget.

In the next section, we describe how business processes descriptions can be monitored at execution time. This is one of the more relevant aspects of BPIMS-WS in relation to the deployment of business process.

4 Process Activity Monitoring in BPIMS-WS

BPIMS-WS offers facilities for monitoring Web services. This facility is offered in all the layers of the architecture. For the monitoring process, it is necessary to listen to the request/response SOAP messaging of Web service-based business collaboration. SOAP messaging identifies the participants and their communications during the long-running interactions involved in the collaboration. For this end, BPIMS-WS intercepts all SOAP messages to generate a UML sequence diagram from the information about the participants and the order in which the messages are exchanged. For the monitoring of activities, a set of Java classes has been developed to represent a UML diagram in a SVG (Scalable Vector Graphics) representation that can be visualized in an SVG enabled Internet browser. The exchange of SOAP messages during some kinds of business collaboration may be developed very quickly. Therefore, to avoid reducing the performance of the Web services execution, the dynamic generation of UML diagrams uses a buffered mechanism to deal with the fast pacing production of SOAP messages.

To illustrate the functionality of BPIMS-WS, we describe a scenario that integrates several products and services suppliers that it has already been implemented in BPIMS-WS. The right hand side of each one of the screenshots of Fig. 4 shows the UML sequence diagram of the business processes.

5 Case of study: A Supply Chain Management

The case of study describes how BPIMS-WS facilitates the discovery of Web services that are offered by some enterprises that sell electronic components. Suppose the following scenario:

1. The enterprises sell on-line electronic components products. The enterprise has registered their products and their business processes as Web services in BPIMS-WS.
2. A potential client (enterprise) needs to request a purchasing order by using Web services.

In this scenario, how an enterprise can integrate their business processes to locate and invoke the Web services available in BPIMS-WS to buy some products?

BPIMS-WS offers two modalities of interaction, either as a proxy server or as an Internet portal. We will show the integration process in the Internet portal mode. Some screenshots of the internet portal mode are shown in Fig 4.

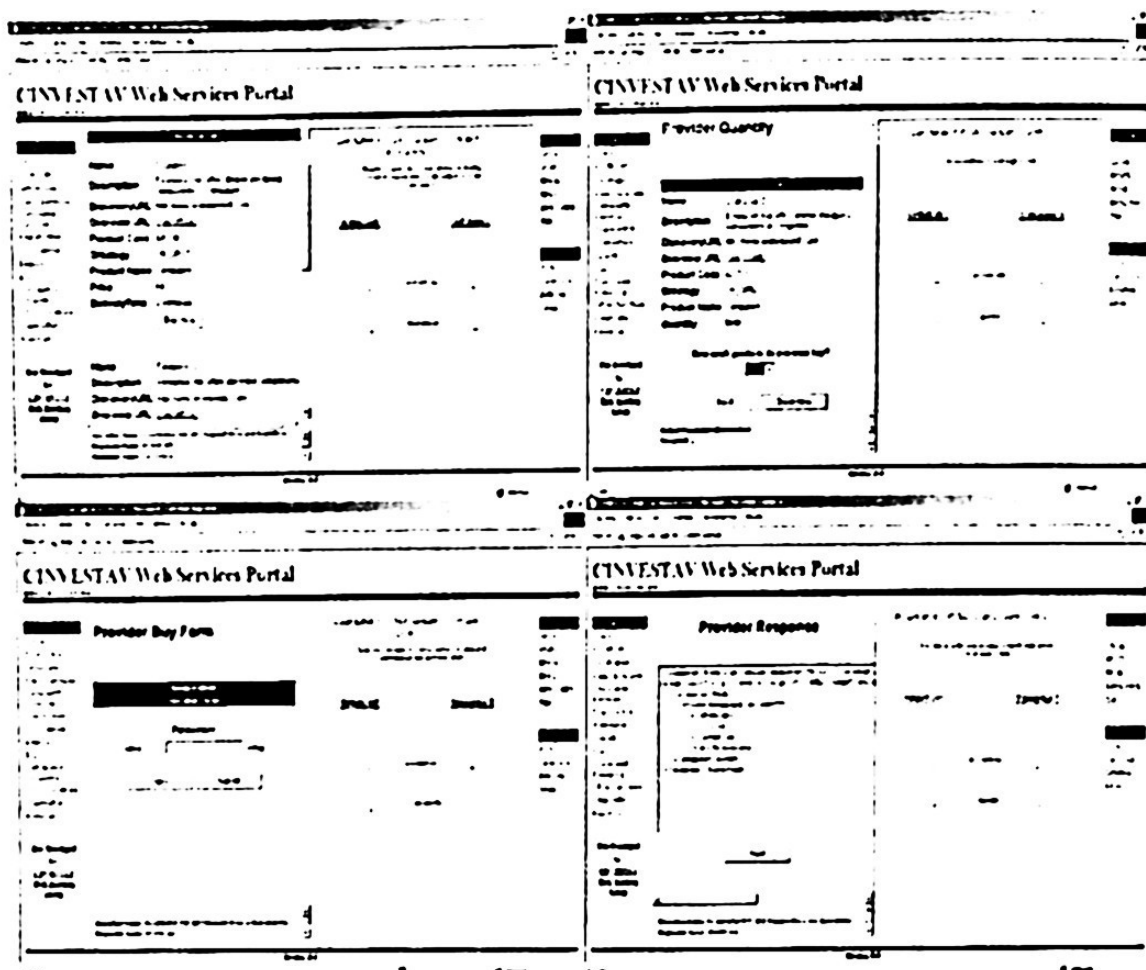


Fig. 4 Screenshots of BPIMS-WS in Internet portal mode

In the Internet portal mode, there is an option in the main menu called "Buy". In this option, BPIMS-WS deploys a graphic interface where the clients can make the search of some product registered. Then, the client can select the purchasing criteria. Suppose that the client has selected the "Show all providers" criteria. Once selected, BPIMS-WS consults the UDDI node to find all the product suppliers [10]. This query returns a list of all the suppliers that have that product. Next, BPIMS-WS retrieves from the UDDI node the URL where the RosettaNet's PIP 3A2 (Request Price and Availability) is described as a Web service for each one of these suppliers (upper-left picture in Fig. 4). Then, BPIMS-WS makes an invocation to each one of these Web services and obtains the corresponding answers. A list of results is obtained as an HTML document [11]. At this point, a list of enterprises may appear as the product suppliers. After the client selects an enterprise from this list, BPIMS-WS formulates a query to the UDDI node to find the URL where the PIP 3A1 (Request Quote) is de-

scribed [10] (upper-right picture in Fig. 4). Next, BPIMS-WS invokes the Web service and obtains the corresponding response. Similarly, the response is shown as an HTML document [11]. Here, the client should select the quantity of products that she wants to buy according to the product availability in stock. Once selected the number of products to buy, BPIMS-WS makes a query to the UDDI node to locate the URL where the PIP 3A4 (Request Purchase Order) is located and analyzes the Web service specification (lower-left picture in Fig. 4). At this point, BPIMS-WS deploys a graphic interface of the Web service specification, so that the client can visualize the activities involved in the purchasing order process. The client is asked to provide the information required to make the invocation of the Web service. Upon completion, BPIMS-WS invokes the Web service. Finally, BPIMS-WS shows the HTML documents containing the answer (lower-right picture in Fig. 4).

6 Related Works

A different approach to the one addressed in this paper is proposed in [12]. A P2P-based orchestration model to support the composition of multi-enterprise Web service is used. Instead in BPIM-WS, a layered architecture to provide support for discovering, creating, composing and deploying Web services was adopted in the P2P model. Our approach follows a client-server model, delegating the task of coordinating elementary activities to a BPEL4WS engine. Besides, monitoring of Web service is a unique feature of our approach. A methodology for business process choreography is proposed in [13]. The methodology focuses on two types of business processes (contract and executable) and provides an interface protocol to represent interoperability patterns between them. Procedures to incorporate existing workflows and generate collaborative processes are defined. BPML [14], BPEL4WS and ebXML [15] are used to define logical and internal processes. This work concerns to level three of our architecture. In comparison, our approach uses WS-CDL for describing any type of Web Service participant regardless of the supporting platform or programming model used by the implementation of the hosting environment.

7 Future Work and Conclusions

So far the design and implementation of the first two layers of BPIMS-WS is almost complete. As future work, we can mention the design and implementation of the WS-CDL engine. It is necessary for developing and deploying enterprise intra-workflow Web services. In enterprise inter-workflow layer, we need to define mechanisms to describe the set of internal behaviors of an enterprise federation. These mechanisms should be able to identify the relationships and interactions among the participants involved. It becomes necessary the design and development of an XML-based language and model that provides these features, along with its inference and execution engine. With this, we will design and develop the set of enterprise inter-workflow Web services corresponding to this level.

In this work, we have proposed a layered architecture for the brokering of business processes for B2B e-commerce. The proposed architecture is based on four layers, where the basic functions are situated at the bottom layer where as the complex func-

tions are situated at the upper layer. Furthermore, we define the functionality of each layer in the architecture. Also, the architecture follows design principles as interoperability, integration, abstraction and scalability. Finally, we developed a system named BPIMS-WS where the concepts and ideas of the architecture are described. BPIMS-WS provides access by means of an Internet portal where a user can appreciate the benefits of intermediation, integration and monitoring in e-commerce B2B.

References

1. UDDI, UDDI Version 3.0, Published Specification, July 19, 2002.
2. North American Industry Classification System, NAICS Homepage, <http://www.naics.com/>.
3. United Nations Standard Products and Services Code, UNSPSC Homepage, <http://www.unspsc.org/>.
4. RosettaNet, RosettaNet Homepage, <http://www.rosettanet.org/>.
5. Business Process Execution Language for Web Services. BPEL4WS 1.1 Specification. IBM May 5 2003.
6. Web Service Choreography Description Language. WS-CDL Specification 1.0. WS-CDL Home, <http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/>.
7. Giner Alor Hdez, José O. Olmedo Aguirre. *Sistema de Intermediación para el Comercio Electrónico basado en servicios Web*. Proceedings CIC 2003.
8. Giner Alor Hdez, José O. Olmedo Aguirre. *Búsqueda, Localización e Invocación Dinámica de Servicios Web utilizando WSIL*. Submitted in CNCIIC-ANIEI 2004.
9. César Sandoval Hdez, Giner Alor Hdez, José O. Olmedo Aguirre. *Dynamic generation of organizational BPEL4WS workflows*. Proceedings ICEEE-CIE 2004.
10. Giner Alor Hdez, César Sandoval Hdez, José O. Olmedo Aguirre. *Descubrimiento Dinámico de Servicios Web en nodos UDDI mediante USML*. Proceedings CORE 2004.
11. César Sandoval Hdez, Giner Alor Hdez, José O. Olmedo Aguirre. *Generación Dinámica de GUI's para la invocación de servicios Web publicados en nodos UDDI*. Proceedings CORE 2004.
12. Boualem Benatallah, Quan Z. Sheng., Marlon Dumas. "The Self-Serv Enviroment for Web Services Composition". IEEE Internet Computing January-February 2003 pages 40-48.
13. Jae-yoon Jung, Wonchang Hur, Suk-Ho Kang, et. al. "Business Process Choreography for B2B Collaboration". IEEE Internet Computing January-February 2004 pages 37-45.
14. Ismael Ghalimi & Jeanne Baker. BPML 101: Implementing the BPML Specification. BPML.org Board Members, March 2002
15. Ebxml, Ebxml Business Process and Business Information Analysis Overview. Ebxml Business Process Team. May 2001.

General Collaboration Architecture to Work with Geographical Information Systems

R. Quintero, M. Torres, M. Moreno, C. Carreto, and G. Guzmán

Geoprocessing Laboratory-Centre for Computing Research-National Polytechnic Institute,
Mexico City, Mexico
{quintero, marcomoreno, mtorres, gguzman}@cic.ipn.mx, ccarreto@ipn.mx
<http://geo.cic.ipn.mx>

Abstract. In the present work, we propose a collaborative-application to the National Center of Disaster Prevention in Mexico (CENAPRED), which is focused on helping in the decision making process and the tasks for preventing natural disasters, related to "Laguna Verde" nuclear plant. This application coordinates the activities of External Plan of Radiological Emergency (PERE) that has been generated for this purpose. In addition, the application is based on a Geographical Information System (GIS) into a collaborative architecture to support the interaction from several entities, which work with special training groups in a virtual reality environment. The architecture consists of a collaboration model and it generates a schema of components to find out the independence and standardization of the system so that it can be implemented in any GIS-platform. Moreover, we present the architecture to convert non-collaborative systems into collaborative ones, without changing the original structure of the application.

Keywords: Collaborative work, Geographical Information System, Collaborative Architecture, Component-based Collaboration.

1 Introduction

Nowadays, Geographical Information Systems (GISs) are powerful and useful tools as means of information, visualization and research or as decision making applications. Recently, intelligent spatial analysis is the main need presented in the Geocomputation trends. Spatial data have an important role in this situation; many times, the information is extended at different places. The problem is greater, because the spatial data present different formats and specifications such as scale, projection, spatial reference, representation type, thematic, DBMS type, and date. For these reasons, the *heterogeneity* of the spatial data formats complicates the spatial analysis.

Other problem related to the heterogeneity in GIS is the collaboration process. This collaboration cannot be accomplished due to several GIS-platforms are totally dependents for any commercial software or environment. Up-to-date, this problem is a barrier to the users that attempt to integrate spatial data and to make spatial analysis in a collaborative way for GIS-Government.

ally, the systems and architectures of collaborative work represent important tools to coordinate group activities. With the mixture of these technologies, it is possible to solve problems related to interoperability and heterogeneity of the spatial applications.

Up-to-date, the appearance of global networks such as Internet and Intranets improve organizational process to allow a better performance of the applications. Recently, some solutions or collaborative tools (independent of the GIS-architecture), are very limited. These solutions attempt to solve particular problems, some examples are shown in [3][4][6][9][10][11]. Other works related to the research integrate workflow solutions and collaborative tools finding group consensus, but it is not a global solution for the participation of multiple users.

In this work, we present an architecture to solve a particular case of study, according to the subject domain specified. Probably, this architecture can be general for any GIS that provides collaboration, cooperation services to multiple users. The case of study of this work is focused on implementing a GIS to support simulation, training and execution processes related to prevention disaster plans (PERE). These plans describe the actions that the specialists must perform in the case of radiological emergency is presented; they consider several actions to guide such as population evacuation, emergency routes, communication and infrastructure available, etc.

PERE is applied to "Laguna Verde" nuclear plant in Mexico. Therefore, the main goal of this work is to provide a solution for this scope and also defines a standard collaboration-architecture to any GIS.

The rest of the paper is organized as follows. In Section 2 we present the case of study. Section 3 describes the problem solution to the case of study and also the problems that represent to implement collaborative aspects. A general architecture for any GIS to provide collaboration capabilities is shown in Section 4. Our conclusions, future works and possible applications are sketched out in Section 5.

2 Case of Study

Nowadays, the activities involved with the External Plan of Radiological Emergency (PERE) [7] are made-up with several efforts of different Mexican Government Institutions (they are denominated as *task-forces*) such as: Army (SEDENA), Interior Ministry (SEGOB), and Navy (SEMAR) among others. Recently, these tasks are performed using a great number of human and economical resources when training exercises and practices on field are carried out. Although certain procedures have been developed and defined in PERE, the actions are very difficult to realize, because the procedures consume very much time and they are complicated. Therefore, it is necessary to count on a computational application that allows helping the process of decision making and optimizing the used resources.

At the moment, exercises of PERE with a degree of acceptable trustworthiness are made periodically. Nevertheless, there are not statistical records that allow comparing the results and progressing obtained at each one of the exercises. For instance, when the training exercise is finalized is not created a digital cartographic registry that describes the trajectory of the *task-forces* (TF) involved, as well as the parameters or variables used in that training exercise.

On the other hand, the dispersion information of radiological pollution agents used during the training is generated in independent program called RASCAL. Therefore, it is necessary to integrate these data by means of a central application so that it can facilitate the manipulation of the information and link it with the spatial data to modify the conditions during the training exercises.

Up-to-date, PERE does not count on simulations that allow observing the displacement of the TF. Also, the procedures are performed by the human experience and the operations are executed in a manual way to solve any task of the training exercise. Additionally, there are not tools that offer a support to provide alternative solutions to obtain maps for evacuation routes and other needs to the users.

The information used in PERE, it is visualized in two dimensions and analogical maps. For this reason, other aspects cannot be contemplated and distinguished. For instance, the topography of the land is an important point of view to carry out the work of each TF.

According to this, it is necessary to find out pedagogical and technological strategies that act with new informatics means together creating an environment of learning with reality. It will allow the interaction of several TF simultaneously in a same virtual scenario. It is important to consider the possibility that a single TF can execute a simulation without the operation of the other TF.

In this proposal, we present the SIGES-PERE application to aid in some tasks related to the External Plan of Radiological Emergency of Laguna Verde Nuclear Plant. The solution is oriented towards the development of a GIS, which will be able to make three fundamental tasks: *Simulation*, *Training* and *Execution* modes.

The *Simulation mode* is the automatic generation of disaster scenarios, with characteristics generated randomly from a spatial database of risk situations. Also, the system when working in this mode, will take the correct actions to solve the contingency based on the corresponding procedures manual.

The *Training mode* as in the *Simulation mode* generates disaster scenarios. Nevertheless, the decisions will be taken by different entities involved in the decision making process. In this way, these entities will practice the previously studied procedures when interacting with the SIGES-PERE.

The *Execution mode* will serve to provide pursuit to the situation in the context of a real emergency. In this operation mode, the disaster scenario will be defined by the user administrator. The decisions will be taken by the commanders of the TF.

In all cases, the information will be visualized in graphical way (spatial data). Dynamic data will appear in real-time over the cartographic information. These data will represent relevant information related to the situation of the scenario such as location of the TF in field, conditions of the population, state of the shelters, and so on.

3 Problem Solution

In this Section we present the solution applied for the problem described in Section 2. Such solution is presented in two parts; first the modeling of the case of study and the architecture proposed to implement the model.

3.1 Operation Modes

SIGES-PERE involves the use of a cartographic base in a digital format; these formats will be raster and vector, which will be used to locate the spatial position (geographic reference) of all the elements involved in PERE.

Therefore, it is necessary to generate the set of rules that must be followed when executing PERE. These rules will allow knowing the behavior of all the operations that will be carried out into the application. In addition, we have to define the conditions that take part in the analysis, such as: climatologic properties, infrastructure, density of population, land use, among others. It implies that the set of rules of PERE and the conditions of the environment will be stored in a *knowledge-base*.

The raised solution is based on three operation modes, which are described as followed.

3.1.1 Simulation mode

In this operation mode random scenarios of disaster will be generated and the system automatically will make the decisions that consider advisable. The scenarios will be generated according to the group of conditions that could be presented. By using the knowledge-base, the catalogue of scenarios defines the values for the variables of risk conditions. The decisions will be carried out considering the procedures defined in the knowledge-base. Next, the general processes that compose this operation mode are described in Fig. 1.

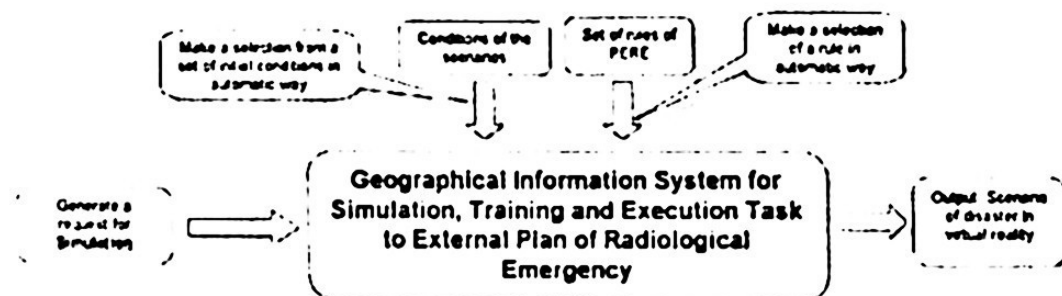


Fig. 1. Processes of the simulation mode into SIGES-PERE

3.1.2 Training mode

In this operation mode the scenarios of disaster will be generated in automatic way, like in the simulation mode. But, in this mode the decisions are taken by the TF. In sense, simulation training exercises of PERE will be able to be made. Nevertheless, the decisions of the operation (activity) to follow could be modified by each TF in a manual way, according to the presented/displayed situation. In addition, the conditions of the scenario could be modified in the moment of executing the maneuvers, which allows providing pursuit to the risk event. This mode will be possible to observe the

displacement and location of the TF, as well as the state of the infrastructure and the displacement and/or location of the population (see Fig. 2).

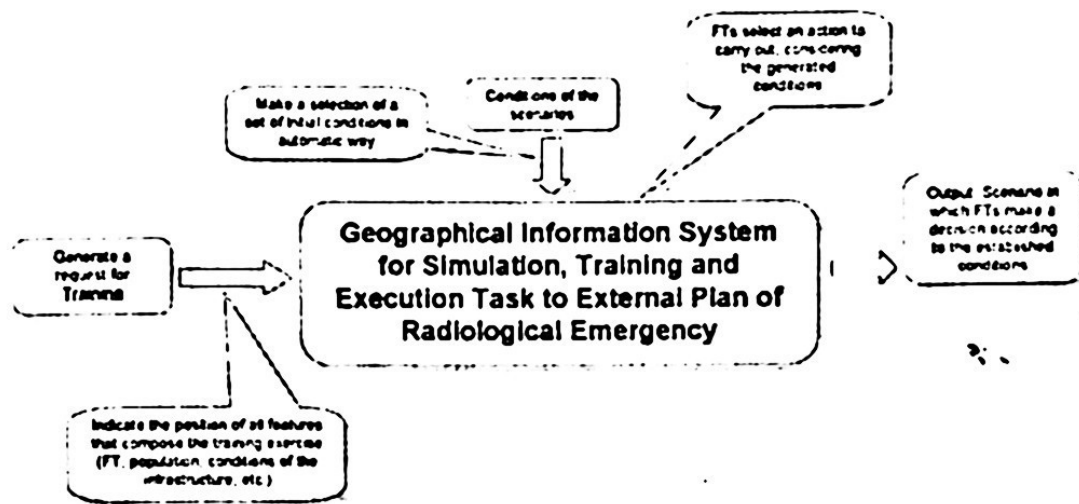


Fig. 2. Processes of the training mode into SIGES-PERE

3.1.3 Execution mode

This operation mode will be used to provide pursuit to the situation in the context of a real emergency. Since the conditions, decisions and pursuit are made-up by each unit of the TF. These elements generate in the system the real scenario of the disaster and it will be performed with base in the field readings (see Fig. 3).

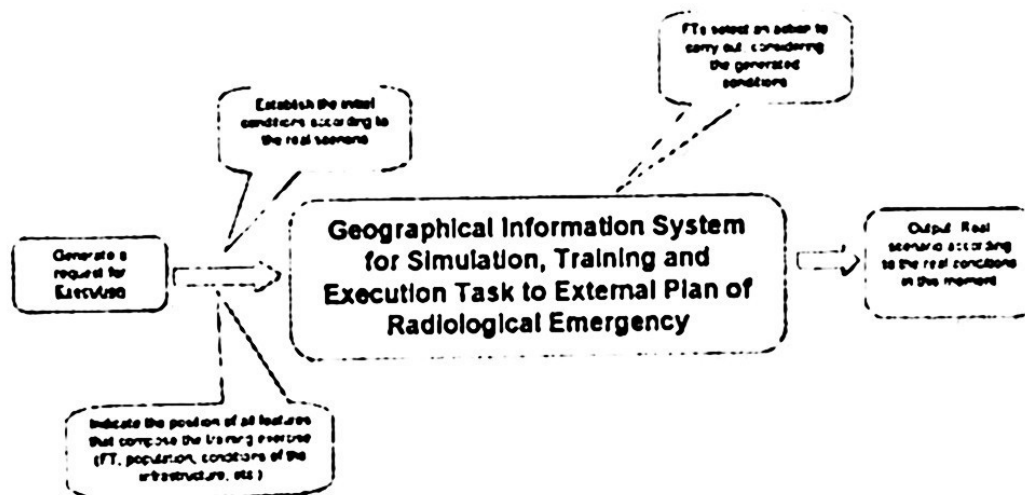


Fig. 3. Processes of the execution mode into SIGES-PERE

3.2 Architecture of SIGES-PERE

The architecture of SIGES-PERE is composed by eight fundamental tiers. This application will be installed within the Primary Emergency Control Center (ECC). On this application, each TF will count on an independent terminal in which they will be able to execute the SIGES-PERE application to interact with the rest of TF, by means of a Local Area Network (LAN). These tiers are grouped in two categories: the category of storage and the functional one.

The *category of storage* is composed by the knowledge-base and the spatial database. The knowledge-base is used to store a catalogue of accidents, which will be defined by the institutions that take part in the prevention of any disaster. Additionally, this knowledge-base will store the environmental conditions that can appear, as well as the set of rules that describe PERE. The spatial database will store the geographic and attributive data of the area, helping in the interaction of the scenarios by means of the development of a base-map.

In the *functional category* are grouped the modules of *Simulation*, *Training*, *Execution* and *Spatial Analysis*. Here, it is possible to find all the processes that belong to the data manipulation of a virtual scenario, as well as the digital cartography.

Similarly, it contains an administration mechanism of input and output processes denominated *GIS Engine*, which will be in charge of handling the requests performed from each entity. The *communication system* sends and receives the data through a communication protocol. The retrieval process of spatial data is performed by ArcIMS, which will be used as a user interface for every TF. In Fig. 4, we depict the architecture of the SIGES-PERE application.

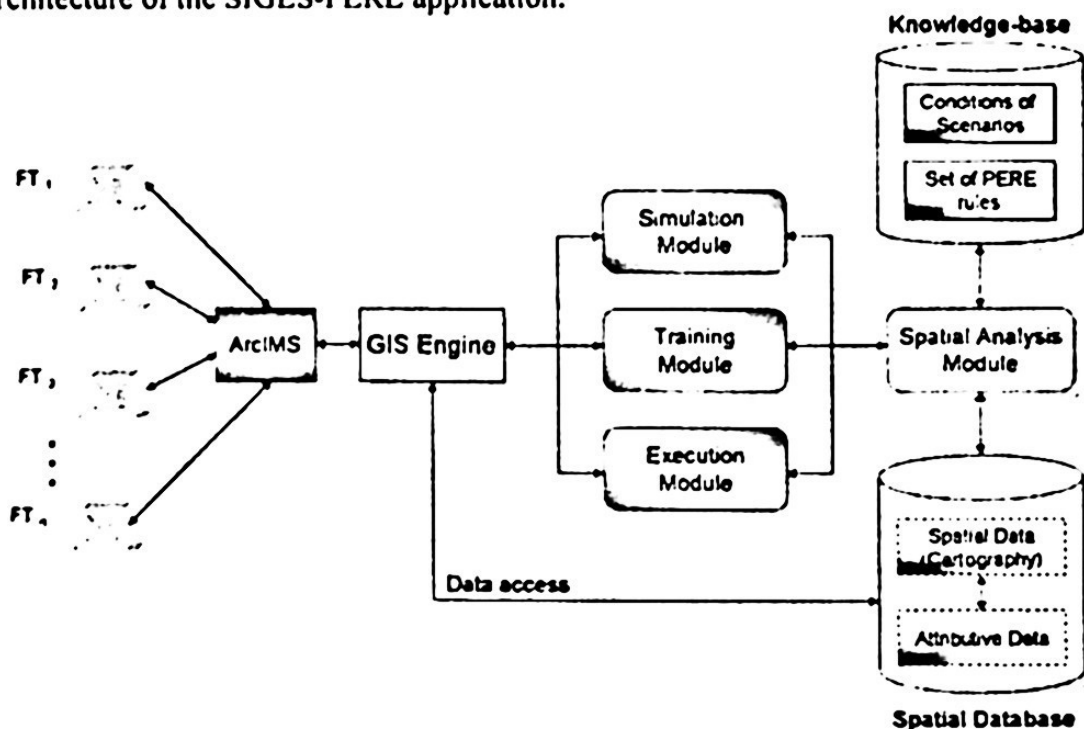


Fig. 4. Architecture of SIGES-PERE

Moreover, in Fig. 4 we appreciate that every TF makes a request to the system by means of ArcIMS. This component sends to GIS Engine the request to access to the spatial database and to retrieve the scenario with the base-map. According to the situation, that has been proposed by the administrator, the scenario is retrieved to the TF and it can operate in any operation mode, which is considered by the administrator when he has determined the training exercise. All the processes are computed by the GIS Engine and the spatial analysis depends on every situation. The values of parameters and conditions are established by the knowledge-base according to the scenario proposed to the practice.

3.3 Collaborative Aspects

In the model presented above there are not modeled any collaborative aspect nor implemented. Nevertheless, it is intuitive that there must be collaborative entities due to the TF are working together in a network environment.

It is necessary to count on a set of elements that allows realizing the collaboration, between users and TF resources, in a natural and intuitive way. Also, such elements must be independents of the TF as well as the operation models of the system. On the other hand, an important technical requirement is that such elements to be also independents of the application to use them directly in other applications.

There are some works made about resolve such problems [1][2]. However, they do not provide a general architecture to collaboration (i.e., it is necessary to implement one for each case of study). Moreover, in these works are not provided *interoperability* between different collaborative applications. Applications that are not necessarily GIS ones. For example, it is not possible to make a GIS interoperates with a Collaborative Virtual World (CVW). Then, we propose an architecture to build collaborative applications that fulfill the constraints outlined above.

4 Architecture of Collaboration-GIS based on Components

As we mentioned on the last section, we propose an architecture for implementing collaboration services into a GIS. We have established several goals to achieve this collaboration:

- To allow performing the collaboration in an intuitive and natural way among different users and entities that form the TF.
- Independent architecture from the task of TF and the operation modes related to SIGES-PERE.
- Independent architecture from the application to be able to use in other applications.
- To provide interoperability aspects to the architecture between different types of applications, which are not necessarily GIS.

4.1 Collaboration Model

The collaboration model is composed of three main features:

- **Task-Forces (TF).** They are defined as the different existing users that interact with the GIS. Each one of them defines the actions, which can take with other TF inside the SIGES-PERE.
- **Collaborative Geographical Information System.** It is defined as any GIS being added; it is composed by a collaboration interface.
- **Resources.** They are defined as the elements of interaction with TF. They can be static or dynamic features of interaction according to the action that represent. These resources are used to access the meaning of other elements like vehicles, shelters, etc.

In our proposed model there are interacting groups of users belonging to different TF, and groups of resource services. We will denote F to the group of all existing TF inside the SIGES-PERE and will denote R to the group of all resources. The capabilities of work and collaboration of any user and the way that they interact is defined by the TF. In other words, when a TF is defined inside SIGES-PERE, its functions must be specified according to the type of TF. Moreover, at this point the set of possible interactions among TF is defined.

Each one of TF partially describes the form that it will interact with others TF defining a set of collaborative interfaces, which provide access to the services.

To establish all possible interactions between the elements of the GIS, we have considered the proposal that is described in [2]. We use a *directed graph* that is shown in equation (1).

$$G = (V, E) \text{ where } V = F \cup R \text{ and } E \subseteq F \cup R \times F \quad (1)$$

Each edge on G defines a particular collaboration; this allows a user to collaborate in a particular way depending on the role of its counterpart. In other words, it has access to a set of different collaboration actions.

For instance, if a CENAPRED user needs to collaborate with a SEDENA user, it is necessary to send a collaboration request to the SEDENA user, which allows both users to exchange their collaboration actions; this collaboration is shown in Fig. 5. Probably, the CENAPRED user can ask information related to the localization of certain group of soldiers, and SEDENA user responds according to the information of the SIGES-PERE.

By using this schema, it is possible that SIGES-PERE defines different types of collaboration among diverse TF. At the same time, it is indispensable to integrate several types of TF, and each one can define the interaction mechanism with other features of SIGES-PERE.

4.2. Architecture of Collaboration

In this section, we describe the proposed architecture for implementing the schema of collaboration in the SIGES-PERE application. The architecture is based on the *Body-Soul* model, which is described in [5]. This model presents a flexible architecture oriented towards to implementing collaboration schemas of different elements into the systems.

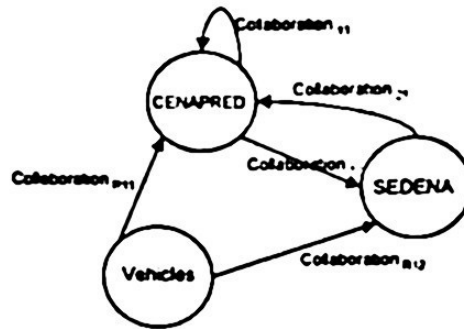


Fig. 5. Collaboration between TF and resources

Therefore, the architecture is composed of three main features as they are described as follows:

1. **Input/Output Information Component (IOIC).** This component is installed over any instances of SIGES-PERE. Its function is to handle and allow that the generated data can be standardized to be distributed for the collaboration interface.
2. **Collaboration Interface.** The interface is in charge of maintaining a repository of references to generate objects that represent the TF. In addition, it administers the collaboration and maintains the state of the collaborating features.
3. **Distributed Objects (TF).** These objects represent to the TF. They maintain the reference to the collaboration interface and allow defining collaboration entities.

As we can observe in Fig. 6, a TF object is generated per each TF, which directly interacts with a reference located on the collaboration interface. It administers the references of all the TF defined. The collaboration relationships between the TF are performed by means of the collaboration interface, it is not necessary to consult the SIGES-PERE to obtain the information requested, it is made-up through the IOIC, then it will return the information requested to the collaboration interface to handle and distribute it to the TF that are collaborating with it.

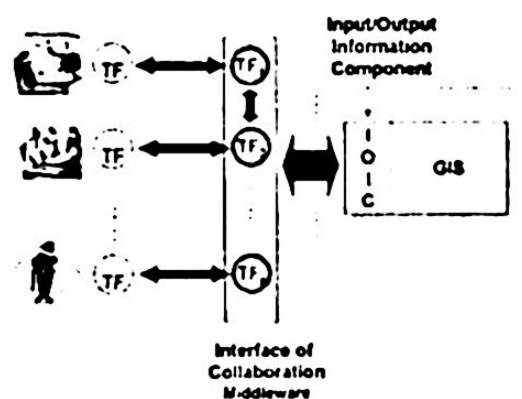


Fig. 6. Collaboration architecture for SIGES-PERE

The architecture provides interconnectivity services for the users. The system provides services of any collaboration type such as collaboration tools (forums, boards, etc), decision making systems, workflows, smart functions, and so on.

Our proposal consists of an architecture that has a component-based schema. Although we applied this architecture to the SIGES-PERE, it is possible to use it for any GIS and also for any information system. The development of architecture components has been made integrating aspects of free software using tools and open source, which are viable option to provide us the opportunity of reducing the cost of the solution.

The SIGES-PERE has been implemented in C++. The collaborative model has been developed in Java and RMI (described in [8]). We made the integration using JNI, but future versions we will use C++/CORBA to implement the collaboration modules.

4.3. Preliminary Results

The purpose of this work is to translate a common GIS-application into a collaborative one. By applying this architecture, we have converted the GIS-application, which is called SIGES-PERE into an embedded-collaborative system. This application solves the problematic outlined in Section 2, and it integrates the collaborative aspects mentioned in Section 3.

SIGES-PERE is a GIS, which contains common functions such as: Spatial Visualization Operations (Pan, Zoom In, Zoom Out), Spatial Overlapping Operations, 2D and 3D Representation Operations, Virtual Flights and an efficient user-friendly interface to the clients. In Fig. 7 is shown the main window of SIGES-PERE.

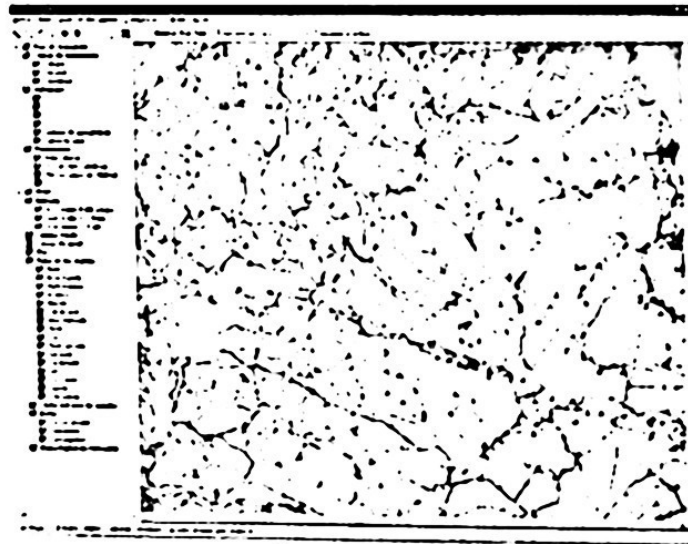


Fig. 7. Main window of SIGES-PERE

The collaboration model is applied over the system; this provides a collaboration interface for users of distinct TF. The implementation is still in the test phase; however, it allows users to collaborate with a restricted number of tools. As we mentioned, one of the goals of the architecture is to maintain the independence of the application. Due to this, it is necessary to implement the collaboration interface for each service provided by SIGES-PERE.

In the implementation, we are using distributed objects to make collaboration. The distributed objects are implemented in Java, using the Java RMI platform. In future versions, we will use C++/CORBA, because all components that we use to make spatial analysis are developed in this language. So, the current version uses JNI to communicate Java with the application wrote in C++.

Moreover, we have made interoperability tests between SIGES-PERE and a Collaborative Virtual World defined in [1]. As a result, users can interact within the virtual world exchanging messages and resources.

5 Conclusions and future work

This work presents a solution to help in the tasks of the PERE, as well as the training of the TF that are in charge of executing these activities in field. The system allows integrating dispersed data within a collaborative GIS-environment in which all the TF can use the data when they need. Also, the TF can interact with each other to optimize resources and avoid duplicate tasks.

Moreover, we present an architecture to convert non-collaborative systems into collaborative ones, without changing the original structure of the application. This is made by adding components to provide the collaborative behavior to the IO system.

With this approach, it is possible to develop the collaborative system independently of the particular system. An advantage of this; is the possibility of changing the collaboration policies without modifying the system ones. Just, it is necessary to change the roles that each user plays in the collaborative environment. Another feature of the architecture is to provide the mechanism to define the collaboration policies. It is made in a formal way by means of a *directed graph* in which all roles and collaborations are defined. With such definition, it is possible to assign permissions and restrictions to specific roles independently of other ones, and without the use strict schemas as user levels or hierarchies.

On the other hand, with the architecture introduced, it is possible to interoperate with other applications using the same architecture. It is performed by means of the implementation of distributed objects that provide us access to the resources in each collaboration system, and that can be accessed by any other distributed object that knows its interface (defined by the architecture). In this line, a future work is related to integrate the architecture a feature of *service-discovery*. It can improve the interoperability between collaborative systems, due to by means of this *service-discovery*, it is not necessary to know neither the precise interface nor the type of application to interoperate with it.

Another work is to define security schemas that allow users being sure about the confidentiality, integrity, service availability, access control and no-repudiation, because some TF will be far from the conflict zone. For example, one of the TF is the Federal Government represented by the President and its cabinet. So, the security system is very important.

In future works, we have considered to perform some changes in the implementation to migrate the collaboration modules from Java RMI to C++/CORBA. With this change, we can guarantee a multi-platform focus.

Acknowledgments

The authors of this paper wish to thank the Centre for Computing Research (CIC-IPN), General Coordination of Postgraduate Study and Research (CGPI), National Polytechnic Institute (IPN) and the Mexican National Council for Science and Technology (CONACYT) for their support.

References

1. Carreto, C.: *Modelo y Arquitectura para la Implementación de Mundos Virtuales Distribuidos Colaborativos*, M.S. Thesis, Centre for Computing Research, Mexico City, (2004).
2. Carreto, C., Menchaca, R. and Quintero, R.: *Modelo de Interacción para Entidades Colaborativas dentro de un Ambiente de Realidad Virtual sobre Internet*. XI International Congress on Computing CIC'2002, Mexico City, (2002).
3. Gardels, K.: *Open GIS and on-line environmental libraries*, ACM SIGMOD Record, Vol. 26 (1), (1997).
4. Manoharan, T., Taylor, H., Gardiner, P.: *A collaborative analysis tool for visualisation and interaction with spatial data*, Proceedings of the seventh international conference on 3D Web technology, (2002).
5. Menchaca R. and Quintero R.: *Distributed Virtual Worlds for Collaborative Work based on Java RMI and VRML*, Proceedings of the IEEE 6th International Workshop on Groupware CRIWG 2000, Madeira, Portugal, (2000).
6. Min-Soo Kim, ODYSSEUS/OpenGIS: *Design and Implementation of an OpenGIS System Based on the ODYSSEUS Object-Relational DBMS*, M.S. Thesis, Database and Multimedia Lab, Computer Science Division, Korea Advanced Institute of Science and Technology, (2002).
7. *Plan de Emergencia Radiológico Externo*, National Center of Disaster Prevention of Mexico, (2001).
8. Quintero, R. and Menchaca, R.: *Implementation Aspects of Synchronous Collaborative Distributed Virtual Worlds*, 5th IASTED International Conference on Computer Graphics and Imaging, Kawaii Hawaii, USA, (2002).
9. Rauschert, I., et al: *User interfaces: Designing a human-centered, multimodal GIS interface to support emergency management*, Proceedings of the tenth ACM international symposium on Advances in geographic information systems, (2002).
10. Rujukan, S.: *Geographical information system: a management perspective*, Ottawa: WDL Publications, (1998).
11. Sergio, P., et al: *Coordination aspects in a spatial group decision support collaborative system*, Proceedings of the 2001 ACM Symposium on Applied Computing, (2001).

Exploiting Features of Collaborative Social Networks in the Design of P2P Applications

José Mitre, Leandro Navarro-Moldes

Polytechnic University of Catalonia, Spain
{jmitre, leandro}@ac.upc.es

Abstract. This paper shows a model of a P2P network to facilitate collaborative activities, which takes advantage of features of a particular social collaborative network –small-world, clustering, community structure, assortative mixing, preferential attachment and small groups–. We present a description of this model based on collaborative groups and limited flooding. Our simulation results show that our model improves scalability and performance of flooding based searches (search results in $O(1)$ hops), message cost, support for node instability, keyword searches, load balancing (number of replicas proportional to demand) in comparison with P2P networks that use DHT or flooding.

1 Introduction

In recent years, many studies have been aimed at finding the properties of social networks, which is a set or group of people, known as the actors, linked together or joined by some pattern of interaction [1]. These efforts arose not only from the interest inherent in patterns of human interaction, but also from the structure of the resulting network. These studies have focused on a number of properties that seem mainly to affect their performance. Among these properties, perhaps those most widely studied are degree distribution [2, 4], the “small world” effect [5-7], clustering [5], community structure [8-10], resilience to the deletion of network nodes [2, 11, 12] and navigability or search ability of networks [13, 14].

Along with other authors [2, 3], we claim that these properties have a great impact on the way that these social networks operate and offer valuable information about the way in which information travels and is routed through the network. Then, this information can be used in the design of networks of computers created by groups of people who work together in order to improve both the performance of the applications and the efficient use of resources.

The model we propose provides the basic mechanisms for modular P2P applications for affiliation collaborative groups of people geographically dispersed, enabling activities such as reading, discussion, writing and modification of documents and diffusion of awareness information, events, news, to be carried out in a distributed way. It is currently being implemented in Java with JXTA [15].

We can define a collaborative social network (CSN) as a social network in which

the actors are people collaborating in some activity and the links joining them together denote their collaboration. Affiliation CSN, are a type of social network having certain natural topologic properties that make them quite well structured.

This paper is organized as follows: in Section 2 we describe some of the features of affiliation CSN that have been exploited in the design of our P2P model. In Section 3 we present an analysis of costs generated by searches in P2P networks using DHT, flooding and collaborative groups and how replication helps to improve the performance and availability of resources. In Section 4 we explain our P2P model as well as the mechanisms for peer communication and management. Sections 5 and 6 details our simulation environment and results. In Section 7 we present some applications focused on collaboration, and Section 8 presents our conclusions.

2 Collaborative Social Networks Features

Community Structure: [8-10] is the property of many social networks for forming communities through the union of people in groups. Grouping occurs for many reasons – shared interest, working for the same company, geographical proximity, etc. In many social networks it is possible for people of a similar type to be drawn together and then to divide up naturally into groups, so that the density of links within the group is greater than the density of the links among them [9].

Assortative Mixing: A social network is said to show assortative mixing [12] if in that network the people wishing to associate with others all have something in common.

Preferential Attachment: In the majority of social networks, the addition of new nodes occurs by preferential attachment [8], in such a way that the new nodes are connected to other nodes by preference; to nodes with a greater degree, for example, or to those that are most popular.

Clustering: is the probability of two people meeting if they share one or more mutual acquaintances. For example, in collaborative groups, people tend to introduce their collaborators to others belonging to other groups, thus fomenting new collaboration and thereby increasing the clustering coefficient [8].

Affiliation Networks: an affiliation network [16] is a network in which the actors are joined together by common membership of groups of some kind. Some studies show groups of academics, actors and business people as affiliation networks.

Degree Distribution: In real social networks the degree distribution follows a power law [13], which indicates a heterogeneous topology in which the majority of nodes have a small degree and a small fraction of highly connected nodes.

Small World: CSN form “Small Worlds”. Typically, participants are separated by short paths [13] of known intermediates. Clearly, news of important findings can circulate more quickly in a network where nodes are more closely connected.

3 Analysis of cost search and replication

As was showed in the previous section, in CSN communities emerge. The effect is that the probability that objects of interest in a community are within the scope (inside

the limits) of that community is very high. In economic terms, we could say that a search without considering communities is more costly, in terms of load (number of messages and peers), and the benefit, chance of and response time to finding an object, may be lower than a search inside a community, where more benefit is obtained at lower cost. Another argument giving additional support to CSN is the fact that people collaborating tend to use the same resources (sharing) during the time span of the collaboration, which is long term.

Let us imagine a P2P computer network used by a collaborative group. Let us suppose that a peer X creates an object A . If all peers, including X , were always connected, the probability that any other peer Y would find the object A is $P(A)=1$, given that we are assuming that A is always available (or X always connected, which is equivalent). But P2P networks, in general, are very instable due to continued node arrivals and departures. Given that peers tend to be end-user machines rather than dedicated servers, there is no guarantee that the peers will not be disconnected from the network at random. This generates low availability of resources and a great overload due to searches for these resources [17]. Thus, the probability of finding the object A decreases with the instability of peers. Now let us suppose we replicate that object A on another random peer. In that case, the availability of A is higher, as the probability of finding the object A or a replica is also higher. Thus, replication helps to cope with peer departures and failures and then to improve resource availability. However it also has a cost [18].

In next section there is an analysis for the cost implied in searching and replicating objects in P2P networks. We have analyzed two search strategies: those that use flooding algorithms (unstructured) and those that use DHT (structured). The two search strategies are studied with and without object replication, and using a structured P2P network based on the CSN topology. The cost figure is defined as a function of the amount data exchanged in messages during the search and the transfer of the target object.

3.1 Search without replication

The cost C_q of a search without replication is defined by the cost c of query messages, routing messages and the transfer of the requested object.

Let us initially assume a P2P network without failures (all nodes connected), where the object of interest is located at only one peer (no replication) which is, in the worst case, located at the far end of the flooding region. The total cost of a query is the sum of all messages generated to find the object, and the actual transfer of the object. This cost depends on the number of messages each peer sends to their neighbors (k), and on the scope of the search in the number of hops (TTL), which is related to the diameter of the network (ϕ).

We use four types of different messages generated by a Gnutella type query (GNet) [19] over a random network: the query message m , the message returned by the peer where the object is located m' , the message requesting the transfer of an object m'' and the message containing the object o . Therefore:

$$C_q = N_m \cdot c(m) + N_{m'} \cdot c(m') + N_{m''} \cdot c(m'') + c(o) \quad (1)$$

N_m corresponds to the number of messages of type m generated during the search. We can see that the number of messages and the number of potential responses increases exponentially with each hop.

Structured P2P networks that use Distributed Hash tables provide an important cost reduction in the search and location of objects. The search cost, related to the number of messages, in most DHT is $N_m \sim \log_k(N)$, where N is the number of nodes, and k the order of the search tree, usually $k=2$.

	Number of messages N_m	Diameter	Topology
Unstructured search (gnutella)	$N_m = k \sum_{i=0}^{h-1} (k-1)^i$	ϕ_{GNet}	Random
Structured search (DHT)	$N_m = \log_k N$	ϕ_{DHT}	K-degree tree
Social network based (CSN)	$N_m = k \sum_{i=0}^{h-1} (k-1)^i$	ϕ_{CSN}	Social network (clusters)

The total cost of a query C_q (equation 1) continues to be valid, with a different value of N_m : the search is more directed, and a much lower number of peers are traversed. After the object is located, the request and transfer process occurs as previously, and therefore the rest of the cost function does not change as in (1).

In CSN, given the properties of CSN (Section 2), we can assume the existence of groups with small diameter. Given that these groups are made up of people with common interests, and since the information relevant to a group can usually be found within their group, our hypothesis is that when searching for an object it is highly probable that the object will be found within the group interested in this object. In terms of the diameter of search (scope): $\phi_{CSN} \ll \phi_{DHT} \ll \phi_{GNet}$

Therefore, the search cost will be dramatically reduced from the diameter ϕ of the global random network to the diameter of a smaller cluster structured by social links.

3.2 Search with replication

Replication has two notable effects: increasing resilience to node failures, and reducing the scope of search. Replication also has an overhead cost, but the influence on the cost for each query is small because the cost of replication must be split among all transfers of a given object. This factor is $r/p(o)$, where r is the number of replicas in the P2P network, and $p(o)$ the popularity of the object o , or the number of requests in a given period. Therefore:

$$C_q = N_m c(m) + N_m c(m') + N_m c(m'') + \left(1 + \frac{r}{p(o)}\right) c(o) \quad (2)$$

Replication strategies vary in time, number and location of replicas[20]. In Gnutella network replicas may be located randomly, in DHT they are located in precise locations dictated by the DHT graph, roughly at the same average distance. In CSN, replicas are located close to the demand: at one hop in terms of interest for most queries since it is based on the social network topology.

In the following analysis of resilience to node failures, we consider that there may

be multiple copies of each data object at distinct peers, but for generality we do not take into account the location of replicas.

The existence of r replicas in a P2P network with peers connected with probability p_c increases the chance of a successful query: finding at least one copy of an object. If P_h is the probability of finding at least one replica during a search, then, $P_h \approx 1 - (1 - p_c^r)'$.

If the P2P network is fault-tolerant: it has multiple routes to reach peers with replicas, then P_h only depends on the probability of disconnection of the peer holding a replica, i.e.: $\alpha = 1$.

In P2P networks with search based on minimum spanning trees, a message between two nodes depends on all previous nodes on the search tree ($\alpha = \phi$) where ϕ is the diameter of the network. This value of P_h depends on the complete path to each replica which, in the worst case, it has ϕ peers. Therefore the total cost C of a successful query with r replicas is given by:

$$C(\phi) = \frac{1}{1 - (1 - p_c^r)'} C_q \quad \text{where } (\alpha = 1 \text{ or } \phi) \quad (3)$$

Compared to the cost without replication ($r = 1$), as r increases, C_q increases only very little (at least for popular objects), but as P_h has a potential growth with r towards 1, $C(\phi)$ decreases: with more replicas, the object tends to be found in one single query.

As one may readily observe, increasing in the number of replicas also increases the probability of reaching an object (accessibility), but it also increases the storage cost with an increasing number of replicas.

3.3 Discussion

While the search cost in DHT algorithms is in the order of the diameter ϕ of the network, search with flooding algorithms on topologies with k neighbors grows in the order of k^ϕ . The use of a CSN topology instead of a typical random network is always beneficial, since the object of interest can almost always be found in the proximity and within the same community, with a much smaller diameter. It can be located in fewer hops, thus enabling flooding-based search algorithms can generate much less traffic.

Replication helps to reduce the impact of node disconnection as shown by the term $(1 - p_c^r)'$ from equation (3), but it also introduces a new cost with the number of replicas r in the network. There is a storage cost when the overall storage space is limited, which has not yet been considered. There is also a transfer cost to create each replica, and the location of the replicas may help to reduce the search cost.

Random replication does not guarantee that a replica is close to (few hops away) from peers who may need it. On-path replication, where an object can be replicated at every peer on the path of a successful query: from the peer with an object to the requesting peer. The disadvantage in this strategy is that the object transfer degrades have to go through several peers instead of a direct connection.

The cost of both replication strategies is significantly reduced if the P2P network used a CSN topology, since using any replication strategy replicas can be located very close to peers with most potential demand. Replication and search can be limited to members of a small community in which the object is located, without the necessity of diffusing the object throughout the entire network. This is because these group members have the greatest interest in the object, since it was generated within their interest group. The maintenance of the CSN topology has an overhead cost, but given that this topology evolves very slowly, we assume the cost is negligible.

Then, combination of replication mechanisms with a CSN topology can assist in appreciably reducing search costs in the network when compared with costs generated by traditional P2P networks, in addition to increasing the probability of access to the required resources. This affects both the performance of the applications and the efficient use of resources. As a result, the search cost is dramatically reduced going from the diameter of the global network to the diameter of a smaller cluster, and replicas are more effective since they are located close to the demand.

4 P2P model for CSN

This section describes our collaborative P2P model. First, we define the components that participate in the model.

Let a collaborative P2P network, which helps the collaborative work between people who might be geographically separated.

A *servent* (server and client) is a computer connected to the collaborative network. Every *servent* holds a list of known groups (GroupId List) and a list of group members identified by their ServentId. These lists may be incomplete, and they are kept consistent using an epidemic consistency algorithm [21] (the details are beyond the scope of this paper). *Servents* provide interfaces by means of which people can exchange messages, share information, carry out searches, compare data, and generally pursue the collaborative work.

A *servent* X is a *neighbor* of Y when X is directly connected to Y by a logical connection using this model, provided that X and Y belong to the same group. This logical connection is created if X and Y are direct collaborators.

A *group* is a sub-network of the network formed by the *servents* associated to people who share interest in common topics. New *servents* joining a group must follow the same rules of behavior as in real life; that is, by affiliation or invitation of *servents* to a given group. Thus the network topology would be similar to the topological structure of the real collaborative social networks (Section 2). Our model assigns a unique GroupId to every group.

A *member* is a *servent* belonging to a given group. All the *servents* must, by default, be *members* of at least one group. Disconnected *servents* will continue to be *members* unless they explicitly withdraw.

Based on the properties set out in Section 2, our model has three fundamental mechanisms for carrying out cooperation functions: 1) connection and join, 2) replication, and 3) search.

4.1 Connection and join mechanism

In many cooperation networks, users connect and disconnect from the network several times a day. It is therefore easy to see that a cooperation network must have mechanisms that manage the connection and disconnection of nodes from the network. First connection to a group is a special case, since then the new node must obtain membership information from the group. From the foregoing and for simplicity, we distinguish two types of connection: joining and connection.

4.1.1 Joining

In order for the topology generated by our model to maintain the same properties as those of a real CSN, e.g. Clustering and small-world, the *servents* must have means of connecting to groups that are similar to those used in real life (affiliation network). Therefore when a new *servent* joins a group, it will be by invitation or by application from the new *servent* to the group. For a *servent* to be connected to the network for the first time, the person must either establish contact with an existing group or create a new group. The *servent* provides a suitable interface to carry out both operations.

When a *servent* creates a new group, he must generate a unique GroupId that identifies the group throughout the CSN. The *servent* must have a ServentId that identifies him, adding to the ServentId list of the group and sending a message to a number of *servents* of the other groups using an epidemic dissemination algorithm, to notify them of his existence. They will feed in turn the initiating *servent* with information about the existing groups in the whole network.

If a *servent* wishes to join to one or more existing groups, he must first receive authorization from any *member* of that group and receive the potentially incomplete group's ServentId list. Once the person has chosen the interest group to which he wishes to be connected, the *servent* must send a message to any *member* of that group to apply to such group. If a member accepts the application to join, the ServentId of the new *servent* is added to the ServentId list of each member of the group, using an epidemic algorithm to spread the new ServentId. Once a *servent* becomes a member of one or more groups in the CSN, he can communicate with other members of the group/s and share information.

If someone no longer wishes to belong to a group, he must send a message with his ServentId, via the *servent*, to other (a few neighbors + epidemic propagation) members of the group or groups to which he belongs in order to cancel membership. The other *servents* must delete the ServentId from the group's ServentId list.

4.1.2 Connection

This operation is used for any further connection after joining a group and after having been disconnected for some time. When a *servent* is connecting he must send a message to all his *neighbors* (eventually by epidemic propagation, it will be known by all the members of the group) informing them that a connection has taken place. Once the *neighbors* have received the message, they must all update their local ServentId list. The connecting *servent* will update his own ServentId list by sending a request to

any neighbor.

In order to know about potential object changes that may have occurred while he was disconnected, the connecting *servent* must launch a search operation (Section 4.3) for events that might have taken place during his absence.

When a *servent* is instructed to disconnect from the CSN, he immediately informs all (a few neighbors + epidemic propagation) connected members that he is about to leave the network, in order to keep the ServentId list up-to-date. In case of connection failure, if a *servent* sending a message receives no reply from another *servent*, the sender will assume that some fault has occurred in the connection with the recipient, and will then proceed to update his ServentId list, indicating that a *servent* is not connected, or informing other members of the change in the ServentId list by epidemic propagation. In this way the list will eventually be up-to-date.

4.2 Replication Mechanism

Given that groups in CSN are made up of people with common interests, and since the information relevant to that group can usually be found within it, we claim that when searching for an object occurs, it is highly probable that the object could be founded within the group interested in this object, in few hops (small-world). Object replication will improve object availability (p2p networks are very dynamic), increase system resilience even during directed attacks to high degree *servents*, and it will improve the performance of search operations without overloading the network. Replication could be carried out solely for the members of the group where the object originates, not necessarily for *servents* outside the interest group or even the entire network. This is because these group members will have the greatest interest in the object, since it was generated within their group (assortative mixing).

We have seen that the frequency with which objects of interest for a particular group are created and modified is in fact low, and the greater part of communication consists of the exchange of ideas via e-mail or chat which do not need to be replicated. This has been confirmed by the analysis of one year event log for the activity performed by a collaborative group of people using BSCW, an application for collaborative work support. It shows that the number of reading events is several magnitude orders higher than the number of writing or modification events [22].

We now present a way of managing replication in our model: When a member creates a new object or modifies one already existing, he must notify that to the group. Immediately after, the *servent* initiates the mechanism to replicate the object to his neighbors in order to reduce the number of replicated objects circulating through the network before arriving at their destinations. For example, let $G = \{A, B, C, D, E, F\}$ where $A, B \dots F$ are *servents* belonging to group G , and B, C, D are the neighbors of *servent* A . When *servent* A creates a new object or when he modifies an object already existing in the network, the given object is replicated only to their neighbors B, C and D , since they have higher need of that object than any other *servent*, given that B, C and D directly collaborate with A . This proximity replication criteria guarantees that the immediate collaborators will have a replica of the object of interest (assortative mixing). Given that the number of replicas is directly related to the *servent* degree,

high degree *servents* will have more replicas, making the network resistant to failures or directed attacks, and balancing load especially for highly connected *servents*.

In the longer term, considering that *servents* have a limited storage capacity, they will have to apply a replacement policy to make room for new objects of higher interest, but they will still keep meta-information on alternative locations of the object, which is roughly equivalent to having the object (one additional hop; the replication mechanism helps *servents* learn the content or at least the location of objects of interest).

Initial simulation results confirm that the number of replicas of an object grows quickly with the number of related search operations, and with the degree (number of *neighbors*) of the originating node which is correlated with popularity.

4.3 Searching Mechanism

With the aim of reducing to a minimum the number of search messages circulating in the network, our search mechanism makes use of both the CSN capacity for forming small-world communities and the replication mechanism.

We have already mentioned that, because of the proposed replication mechanism, the cost of flooding based searches is drastically reduced.

Our model use two types of flooding search – local and external.

Local Search. Search undertaken by a *servent*, to bring his information up to date or looking for an object within the group. A local search can be made for three reasons: 1) When a new *servent* joins a group and needs to know about all the objects shared by the group. So when a new *servent* receives a message on concluding the initial connection process (Section 3.1), he should ask other *servents* for the objects shared by the group. 2) When a *servent* has reconnected after having been disconnected for a certain time; the *servent* must then seek to update information generated in his group during his absence. He will carry out a local search for new or modified objects. And 3) when a *servent* belonging to a group needs a particular object, he will carry out a local search for that object by sending a query to his *neighbors*. The *members* receiving the query message will send information about the object to the requesting *servent* if they have the target object.

External Search. Search carried out outside the group to which the *servent* initiating the search belongs. This situation may arise when a *servent* needs an object that is not available from any of the group members, and must therefore look for it in other groups. The user must explicitly undertake this kind of search when he or she wishes to search for an object throughout the entire CSN.

To make flooding search more efficient, *servents* have information (GroupId) about each existing group within the network. The *servent* initiating the search can locate at least one *servent* from each group and direct the search towards them.

Given our proposal that a number of the *servents* have object replicas of interest to the group, the probability of locating the desired object will be quite high, and search messages may go directly to those *servents* who are most likely in possession of the object. In terms of external searches, our mechanism differs from flooding algorithms in the number of *servents* involved: we select at least one *servent* per group while flooding would contact all nodes up to a maximum number of hops (TTL).

Both in local and external search, objects would be downloaded via a direct connection between the *servent* possessing the object and the *servent* requesting it. Since the object is replicated, it can be done in parallel from various *servents* in order to make download process fault tolerant, faster and more efficient.

As may be easily appreciated, our search strategy is thus less costly $\sim O(1)$ (objects located in a single hop in most cases) than classical flooding $\sim O(N)$ and can be more flexible than DIIT, typically $\sim O(\log N)$.

5 Evaluation

The analysis described in Section 3 provides some insight into the potential benefits of our model. But in order to test the validity of our proposal and to evaluate the effects of features described in Section 2, we developed a simulation infrastructure implementing our model. We simulate our model using the j-sim simulator [23].

We implement the search and replication mechanisms on the top of collaborative networks. The topologies of these networks are based on a Newman's algorithm [9]. Using this algorithm we have generated networks with properties such as: clustering, community structure and small-world. Different randomly generated topologies are formed by interest-based groups of 10, 50 and 100 *servents*, where each node represents a person, and each link models a personal relationship in the collaborative social network, which corresponds to a link in our P2P network.

For each topology we run 1,000 differently seeded simulations, consisting of N requests (one for each *servent*) for a single object created on a random *servent*.

In each simulation cycle, we randomly designate a *servent* to be the object initiating a search, among those without a replica: at the end of the simulation, every *servent* will have done just one search and will hold one replica.

Since that search cost is directly related with the search scope, the goal of our experiments is to measure the cost introduced by our model by measuring the long path necessary to find an object and the load generated by queries.

Our simulation results, see Figure 1, reveal that the greatest long-path length to reach a replica is very small (less than 3, order of $O(1)$). It is also possible to see that approximately after that 50% of *servents* have executed a query, and have got a replica; the long-path to reach a replica is almost 1. This result is comparable with results for DIIT based P2P networks. Based on other studies [24], the characteristic diameter in Gnutella is smaller than 12 hops and over 95% of the nodes are at most 7 hops away from one another. In our case, the diameter is smaller than 6 for each cluster of 500 *servents* and almost 90% of the *servents* are at most 5 hops away. Nevertheless, with less than 3 hops a query can always be resolved and on average 1.2 hops to big groups (4 in Gnutella). Therefore we obtain lower search cost and better performance. In addition, popular objects (high number of searches) are easier to find (more replicas) than non-popular objects (low number of searches).

Figure 2 shows both the number of generated messages until the moment the first answer to a query is received (inferior scope), and the amount of messages generated by queries until the TTL expires (superior scope). We achieve low load using a TTL=5, but we could reduce it without significant effect to the hit search using

TTL=3. This value does not change substantially with the size of a cluster, as it depends on the small-world property of the cluster.

Figure 3 shows the amount of participating *servents* until the first answer to the query is obtained, and the number of participating *servents* until the TTL expires. As can be seen, the amount of participating *servents* decreases with time given that as time passes more copies will exist in the cluster and therefore objects will be located faster (when an object is found, the query does not propagate beyond that peer), thus limiting both the amount of messages generated and the number of participating *servents*. Replication helps to reduce the search cost. A typical search in Gnutella can cover up to 1000 *servents* (more than 2 orders of magnitude).

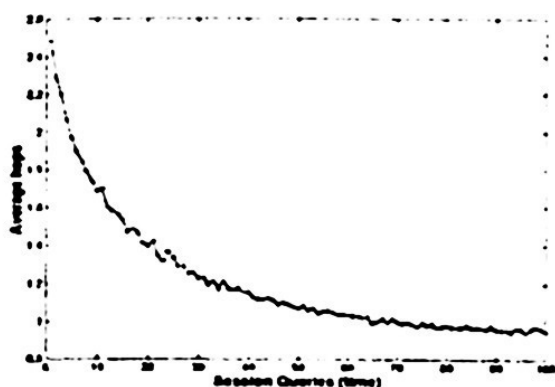


Fig. 1. Average number of hops

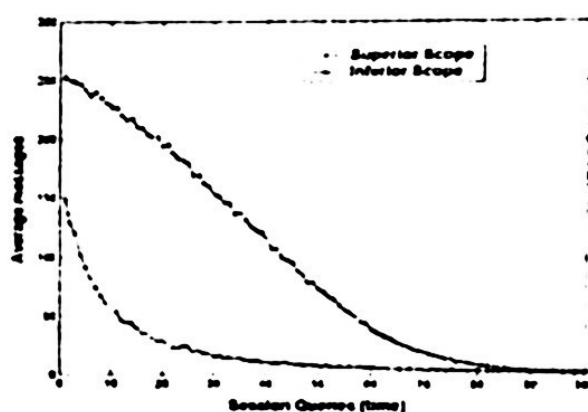


Fig. 2. Average number of messages

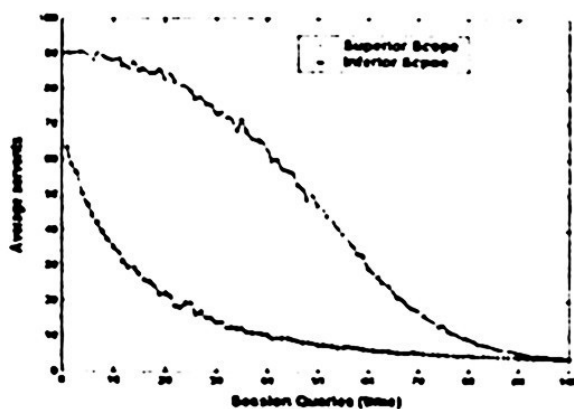


Fig. 3. Average number of Servents

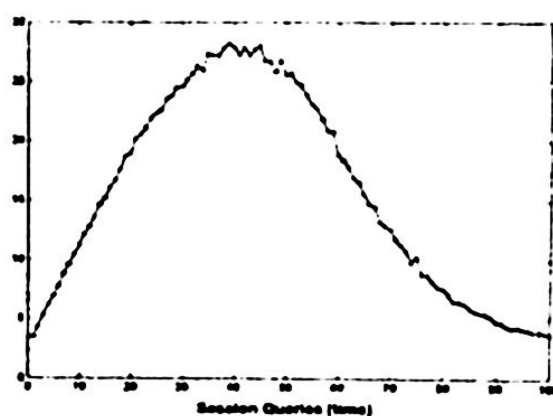


Fig. 4. Average number of Responding Servents

Our experimental results show that there always exists a *servent* that can resolve a query, i.e. search always is successful at locating objects. The minimum average number of *servents* that respond to a query is 3.5 at the beginning of our simulations, when there are only few object replicas (on the order of the average *servent* degree). The top number of *servents* answering a query is obtained in the middle of the experiment: roughly 50% of the nodes can give an answer to a query after 50% of the nodes have obtained a copy of an object. In 99% of the cases there is more than one *servent* to respond to a query. Nevertheless, *servents* rarely need all of the results of a search. By reducing the scope of search, we can greatly reduce the amount of messages that need to be sent and improve the scalability.

Figure 4 shows that in the last queries fewer *servents* respond to the query; this is the result of *servents* with a replica not propagating the query beyond. In Figures 3 and 4 we can see that replication helps to improve scalability given that the more replicas having the cluster, the lower number of participating *servents* will participate to resolve the query. As can be seen in Figure 1, in a cluster with up to 100 *servents*, it is always possible to obtain a query response with TTL=3. With this value, the number of involved *servents* and the amount of messages would decrease drastically. These values do not change significantly for other topologies with a similar or larger number of nodes (tested up to 500 peers), as they depend on the small-world property.

6 Related work

Previous work carried out on collaboration networks [25, 26] has been centred on trying to improve network performance, leaving aside the relevant fact that computer networks give support to social networks with distinctive statistical properties. Unlike previous work, the main idea of our proposal is based on the properties belonging to collaborative social networks.

Iamnitchi [27] put forward ideas about making use of the small-world property and clustering in scientific social networks. In [27], some mechanisms are proposed to facilitate searching, but without suggesting any particular model.

Other [27,28] related work concentrates on identifying clusters of interest to improve the performance of search process so that queries can be steered to peers that are more likely to have an answer. Unlike these works, we do not identify clusters, given that clusters are formed by users through explicit affiliation with groups. Cluster identification algorithms could assign a user node to a cluster with only a subset of files of interest. Letting the user select which groups wants to join guarantees he will be a member of communities of his interest, and have the relevant documents close to hand.

7 Conclusions

A great deal of research work seeks to develop better methods of locating data in P2P networks. These efforts are aimed at improving scalability, greater reliability under dynamic conditions, more efficient searching, and improved performance. The main problem with these systems, however, is that they ignore the fact that computer networks, such as P2P, are made up of people who in turn form social networks with statistical properties which affect the way these networks function.

In this work, we present a proposal for a new P2P model for collaboration networks using the social network topology, and exploiting the inherent characteristics of such networks: small-world, clustering, community structure, assortative mixing, preferential attachment and small and stable groups.

We also show how the combination of interest-based replication mechanisms with CSN properties can assist in appreciably reducing message overload in the network, compared with overload generated by searching in traditional P2P systems, thus improving performance.

We argue that our model use simple search strategy capable of dramatically decreasing the search cost (search results in $O(1)$ hops) compared to a traditional P2P application on the same context. The simulation results show that our model is scalable and has good performance comparable with those that use DHT.

Although this work is focused on collaborative networks, we believe that many of the ideas set out in this paper can also be applied to other types of P2P networks.

The model presented here is an initial approach to making use of CSN topological properties, and as such we are aware that there is still room for improvement. At present, we are engaged in the implementation of a prototype that will enable us to assess improvements in possible extensions of the model.

Acknowledgements

This work has been supported by the Mexican Institute of Petroleum (Instituto Mexicano del Petróleo) and the Spanish MCYT project TIC2002-04258-C03-01.

References

1. S. Wasserman and K. Faust, "Social Networks Analysis". Cambridge University Press, Cambridge. 1994.
2. R. Albert, A.-L. Barabasi, "Statistical mechanics of complex networks". Rev. Mod. Phys., vol. 74, pp. 47-97, 2002.
3. V. Krebs, "The social Life of Routers, Applying Knowledge of Human Networks to the Design of Computer Networks". The Internet Protocol Journal, Vol 3, Num. 4, 14-25, Dec. 2000.
4. S.N. Dorogovtsev and J.F.F. Mendes "Evolution of networks". Advances in Physics 51, 1079-1187. 2002; arXiv:cond-mat/0106144.
5. D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks". Nature 393, 440-442, 1998.
6. L. A. N. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley, "Classes of small-world networks". Proc. Natl. Acad. Sci., Vol. 97, No. 21, 11149-11152, 2000.
7. M. A. Jovanovic, F. S. Annexstein, K. A. Berman. "Scalability Issues in Large Peer-to-Peer Networks - A Case Study of Gnutella". University of Cincinnati, Technical Report 2001.
8. M. E. J. Newman, "Clustering and preferential attachment in growing networks". cond-mat/0104209.
9. M. E. J. Newman and M. Girvan, "Mixing patterns and community structure in networks". Proceedings of the XVIII Sitges Conference on Statistical Mechanics, Springer Verlag, Berlin, 2003.
10. R. Guimera, L. Danon, A. Diaz-Guilera, F. Giralt, A. Arenas, "Self-similar community structure in organisations". arxiv.org/pdf/cond-mat/0211498, Nov 2002.
11. R. Albert, H. Jeong, and A.-L. Barabasi, "Error and attack tolerance of complex networks". Nature 406, 378-382 July 2000.
12. D. S. Callaway, M. E. J. Newman, S. H. Strogatz, and D. J. Watts, "Network robustness and fragility: Percolation on random graphs". Phys. Rev. Lett. 85 5468-5471, 2000.
13. L. A. Adamic, R. M. Lukose, A. R. Puniyani and B. A. Huberman "Search in Power-Law Networks Internet". Physical Review E, 64 46135, 2001.

14. A. Iamnitchi, M. Ripeanu, I. Foster "Locating Data in (Small-World?) Peer-to-Peer Scientific Collaborations". 1st International Workshop on Peer-to-Peer Systems, Springer-Verlag, 2002.
15. JXTA Homepage: www.jxta.org
16. M. Newman, D. Watts, S. Strogatz, "Random graph models of social networks". arXiv:cond-mat.0202208. Proc. Natl. Acad. Sci., to appear.
17. D. Liben-Nowell, H. Balakrishnan, D. Karger. "Observations on the Dynamic Evolution of Peer-to-Peer Networks". In the proceedings of the First International Workshop on Peer-to-Peer Systems (IPTPS '02), March, 2002; Cambridge, MA.
18. K. Aberer, M. Hauswirth, M. Puceva, R. Schmidt, "Improving Data Access in P2P Systems", IEEE Internet Computing, 6(1), January/February 2002.
19. Clip2 Distributed Search Solutions, "Gnutella Protocol Specification v0.4", 2001, www9.limewire.com/developer/gnutella_protocol_0.4.pdf
20. Q. Lv, P. Cao, E. Cohen, K. Li, and Scott Shenker. "Search and Replication in Unstructured Peer-to-Peer Networks". Proc. ACM ICS 2002.
21. R. A. Golding. "Weak-Consistency Group Communication and Membership". PhD thesis, University of California, Santa Cruz, Computer and Information Sciences Technical Report UCSC-CRL-92-52, December 1992.
22. J. M. Marqués, L. Navarro. "WWG: a Distributed Infrastructure to support groups". Proceedings of the ACM Conference: Group 2001 (Group'01).
23. J-sim homepage: www.j-sim.org
24. K. Sripanidkulchai, B. Maggs, and H. Zhang "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems", Infocom 2003.
25. BSCW Homepage (Basic Support for Cooperative Work), <http://bscw.gmd.de/>
26. Groove Homepage: www.groove.net.
27. A. Iamnitchi. "Resource Discovery in Large Resource-Sharing Environments. PhD Thesis. Adriana". Department of Computer Science The University of Chicago Dec 2003.
28. K. Sripanidkulchai, B. Maggs, and H. Zhang. "Enabling efficient content location and retrieval in peer-to-peer systems by exploiting locality in interests", ACM Computer Communication Review, vol. 32, Jan. 2002.

An Approach for Facilitating Development of Web-based Information Systems

Ángel Israel Ortiz-Cornejo, Heriberto Cuayáhuatl y Carlos Pérez-Corona

Universidad Autónoma de Tlaxcala,
Department of Engineering and Technology
Intelligent Systems Research Group
Apartado Postal #140, Apizaco, Tlaxcala, 90300, Mexico
{aortiz, hcuayahu, cperez}@ingenieria.uatx.mx

Abstract. In this paper we present an approach for facilitating development of web-based information systems based on the MVC (Model-View-Controller) design pattern. Our approach consists in the automation of two components: View and Controller. For automating the *view* we propose the use of style sheets, which apply a consistent visual design to the pages that integrate a web-site. For automating the *controller* we propose a markup language (WSML) for specifying the structure and navigation features of web sites, as well as a code generator for automatic coding based on an algorithm using code templates that generates fully functional HTML code by parsing WSML documents. We illustrate our approach with an information system for evaluating basic education. This approach is very useful for building web-based information systems in new domains saving time and effort in the development phase.

1 Introduction

Our information-based society has created a large demand for the development of web-based information systems. Currently, expert programmers perform development once the design is written. However, the development phase may require re-recoding in the following scenarios: because of unsatisfied customers who request changes in the User Interface (UI), because usability testing (usually performed before finishing coding) might require changes in the UI, and because adding functionality after deployed applications might affect part of the coded UI. These scenarios illustrate the fact that a mechanism for coding automatically from a UI document could dramatically reduce time and effort spent in the web-based application development process. In the past, few research efforts have been undertaken in this area. For instance, *WebML* provides a markup language for specifying complex web sites at the conceptual level [1, 2], but it is a general modeling language missing specific primitives of data-entry applications. Current frameworks such as *Wizard* allow developers to specify web-based data entry applications based on XML documents in order to produce fully functional skeletons for the web pages [3], and *VoiceBuilder* allow developers to specify speech applications from either a GUI or web-based application stored in a high-level language in order to automate coding of speech applications [4]. Other tools such as GARP allow developers to generate web reports automatically from a

database schema [5], and the *Ninf Portal* toolkit facilitate the development of Grid portals by automatically generating the portal front-end from an XML document [6]. Despite of the recent work on the rapid prototyping of web-based applications, we must continue developing either automatic or semi-automatic methods for facilitating development of web-based information systems.

In this research we present preliminary investigations towards rapid prototyping web-based systems, based on two markup languages and a code generator. In the remainder of this paper, we first describe the architecture of our approach, WISBuilder (Web-based Information Systems Builder) in section 2. We then describe our markup languages for specifying web content (WSML and WAML) in section 3. In section 4 we describe the code generator. In section 5 we describe a case study using our proposed approach. Finally, in section 6 we provide conclusions and future directions.

2 Architecture of WISBuilder

There are three types of tasks involved in the development of Web-based information systems using the MVC (Model-View-Controller) design pattern: 1) the structural design of a web site, 2) the web-site visual design, and 3) the development of web applications corresponding to the business-logic. A major problem in the design of web sites is that the MVC tasks are highly dependent among them. In figure 1 we illustrate the architecture of WISBuilder, in which we proposes a separation of tasks. In this architecture a web designer specifies web content through a GUI (Graphical User Interface), stored in XML documents (WSML and WAML). Such documents are given as input to the code generator, which uses code templates and style sheets in order to generate fully functional HTML code. Due to the fact that the style sheets as well as the code templates might be generated through any development environment, the tasks of the MVC can be developed in parallel.

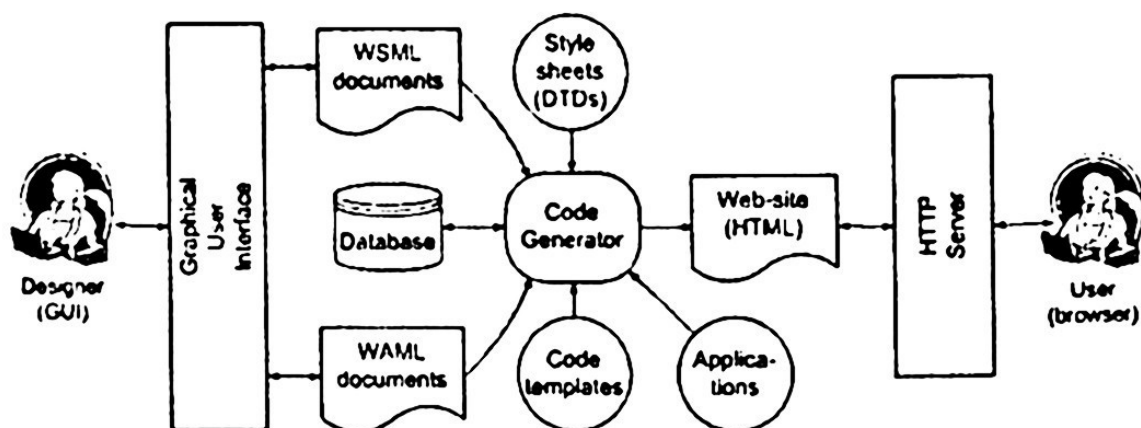


Fig. 1. The architecture of WISBuilder.

The goals of this architecture are threefold: 1) establish a clear separation of tasks for making them independent without losing joint collaboration, 2) build web-based

information systems with a semi-automatic approach, and 3) make the development and maintenance tasks easier. We attempt to apply the MVC design pattern to this architecture through the use of two markup languages: WSML which specifies the structure of a web site, and WAML which specifies web applications. The *controller* is automated through the use of the WSML markup language and code templates. The WAML language is used for embedding applications into the web site and also uses code templates. The *view* is automated through the use of style sheets (stored in a repository) in order to provide a consistent view for the entire web-site. In this way, our approach attempt to facilitate development of web-based information systems.

3 The Markup Languages

We propose two markup languages in order to specify web content in a more declarative language and independent of visual presentation, the resulting documents are given as input to the code generator for generating fully functional code.

3.1 Web Site Modeling Language (WSML)

WSML (Web Site Modeling Language) is an XML language for modeling and designing web sites, which describes the elements contained into a web site. Usually, a web-site is integrated by a set of web pages, which are compounded of elements and navigation features. In the same way, a web-site in WSML is compounded of multiple web pages arranged into sections according to its contents. Each section contains at least one web page, and every page contains different kind of elements like forms, text, links to other pages, etc. The navigation among pages is the feature that allows a web page to access other pages, possibly involving a pre-processing such as a database query. Thus, WSML allows the specification of elements in a structured way, as well as the specification of navigation elements.

It is important to remark that this language allows the design of web sites without taking care about the appearance of web pages and elements inside pages, in such a way that the modeling of the site is independent of its visual style, which suggests a separate design. Another important feature of this language consists in separating the development of web applications from the modeling of the site; in other words, the user can develop applications independently of the web site design. Then, the code generator can embed automatically those applications into their corresponding places. However, the developed applications must be stored in a specific directory, so that the code generator can find them. The web applications embedded into the web pages can be created automatically using WAML (Web Application Modeling Language), a proposed markup language for describing web applications using code templates that can be written in various kinds of code (applets or scripts).

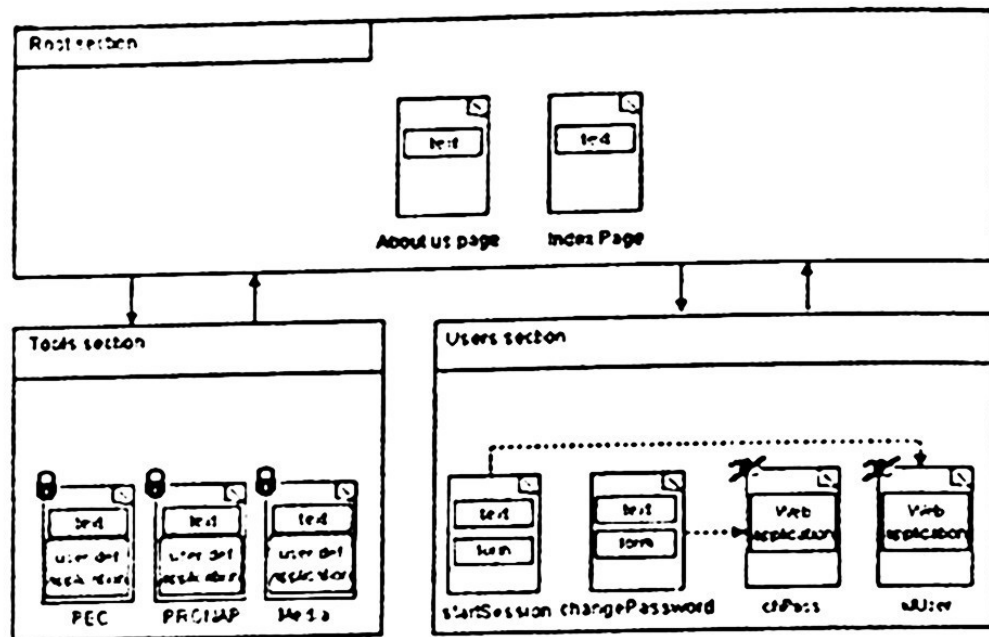


Fig. 1. An example of a web site in the design phase.

In figure 1 we show a web site with three sections: 1) the *Root* section which is the main section of a web-site, 2) the *Tools* section which represent the business logic, and 3) the *Users* section used for user authentication. Every section contains different pages with different content, which can be visible or invisible to the users. A visible page is available to the users; an invisible page can only be accessed when a process (like a form) is triggered. The pages marked with a *lock* are hidden to unlogged users; those pages are only available when a user has started a session. A page containing a form must invoke its corresponding page that contains the application; those pages which receive the requests, must import applications from a directory of applications. Such applications can be written in several kinds of code such as scripts, Java scripts, applets and WAML applications. The arrows linking sections refer to navigability, e.g. the section that points another can access the visible pages in that section.

In WSML, a web site is specified with an element called *website*, the root element (see the following XML document). The root element is divided in *section* elements, which have a unique name and might specify several web pages. The pages of a web site are represented as *webpage* elements, which require the definition of the following attributes: *name* referring to the name of the web page, and *private* referring that this page is only available for logged users. There are two elements used for controlling the navigation: *navList* which define a set of links to external or internal pages, and *navControl* which is used to link the internal sections into the site. Finally, a web page might contain many different elements like text paragraphs, imported files and photos, forms, web applications, and links to other pages, among others. These elements are considered in WSML and further details can be found in the appendixes A and B. Finally, consider the following XML document as a representation of figure 1.

```
<website name="SEPE">
  <section name="ROOT">
    <navControl section="ROOT"/>
    <navControl section="Tools"/>
    <navControl section="Users"/>
    <webpage name="index" title="Principal page.">
      <text>Welcome to the SEPE Web site...</text>
    </webpage>
    <webpage name="aboutUs" title="More about us.">
      <text>We are an institute ...</text>
    </webpage>
  </section>

  <section name="Tools">
    <navControl section="ROOT"/>
    <navControl section="Tools"/>
    <webpage name="PEC" private="yes">
      <text>The PEC is a national program for...</text>
      <appImport name="PecApp" type="JavaScript"/>
    </webpage>
    <webpage name="PRONAP" private="yes">
      <appImport name="PronapApp" type="Applet"/>
      <text>PRONAP (Programa Escuelas de Calidad)</text>
    </webpage>
    <webpage name="Media" private="yes">
      <text>This is the multimedia handler tool for...</text>
      <appImport name="MediaApp" type="Applet"/>
    </webpage>
  </section>

  <section name="Users">
    <navControl section="ROOT"/>
    <navControl section="Tools"/>
    <webpage name="startUserSession">
      <text style="bold">
        To start a user session fill the form and click "Send".
      </text>
      <form action="idUser.php">
        <input type="text" name="username" label="User name:"/>
        <input type="password" name="password" label="Password:"/>
        <submit label="Send"/>
      </form>
      <text>Have a nice day!</text>
    </webpage>
    <webpage name="changePassword">
      <text>
        In this section you can change your password.
      </text>
      <text style="bold">Please fill the form below.</text>
      <form action="chPass.php">
        <input type="text" name="name" label="Your user name:"/>
        <input type="password" name="pass" label="Your password:"/>
        <input type="password" name="newpass"
          label="Write a new password:"/>
        <submit label="submit"/>
      </form>
```

```

    </webpage>
    <webpage name="idUser" visible="no">
      <appImport name="identifyUserApp" type="WAML"/>
    </webpage>
    <webpage name="chPass" visible="no">
      <appImport name="changePassApp" type="WAML"/>
    </webpage>
  </section>
</website>

```

3.2 Web Application Markup Language (WAML)

WAML (Web Application Mark-up Language) is a markup language that specifies applications used for rapid prototyping web pages. Such applications are embedded into WSMML through the element *appImport*. A WAML file defines web applications that can be translated to completely functional applications implemented in different kinds of code; for instance, JSP, PHP, or ASP. The current status of WAML considers three applications: database queries, database reports, and user sessions; and might be extended by frequently used applications. The following fragment of XML document is an example of an application to change password. This application receives the values "name", "pass" and "newpass" as variables; sent from the element *appImport* "changePassApp" (look at the previous XML document).

```

<application type="php" name="changePassApp">
  <sqlDo dbname="SEPE" usr="general" pass="general">
    <query>
      update into users set password=$newpass
      where name=$name and password=$pass
    </query>
    <onsuccess>
      <msg>The password was changed !</msg>
    </onsuccess>
    <onerror>
      <msg>The account doesn't exists or password is wrong.</msg>
    </onerror>
  </sqlDo>
</application>

```

4 The Code Generator

The code generation algorithm, called WAC (Web Application Coder), receives WSMML documents and produces HTML web pages in a structure of directories, established by the web site design. This algorithm performs three main steps: the first step translates WSMML elements for each web page defined in the site into HTML elements. For translating elements a set of code templates are used, which take the values of attributes and elements in order to generate fully functional HTML code. The second step consists in applying style to each web page; in other words, the instructions contained in each web page are graphically organized following the style sheet used.

Finally, in the third step the generated web pages are stored into a set of directories according the web site design.

Following the WAC algorithm (see figure 4.2), two inputs are needed: 1) the WSMML document (well-formed and valid) and 2) the style sheet to apply. First, the root element of the WSMML document is taken, as well as its attribute *name*. Then, a directory with that value is created. If a *webpage* element is found, then it is translated into HTML code using the code templates stored in the *templates* directory. If an *applImport* element is found, a file with the web application is loaded from the *scripts* repository in order to add it to the generated web page. Then, the style sheet is loaded and used to apply the desired visual style to the pages of the web site.

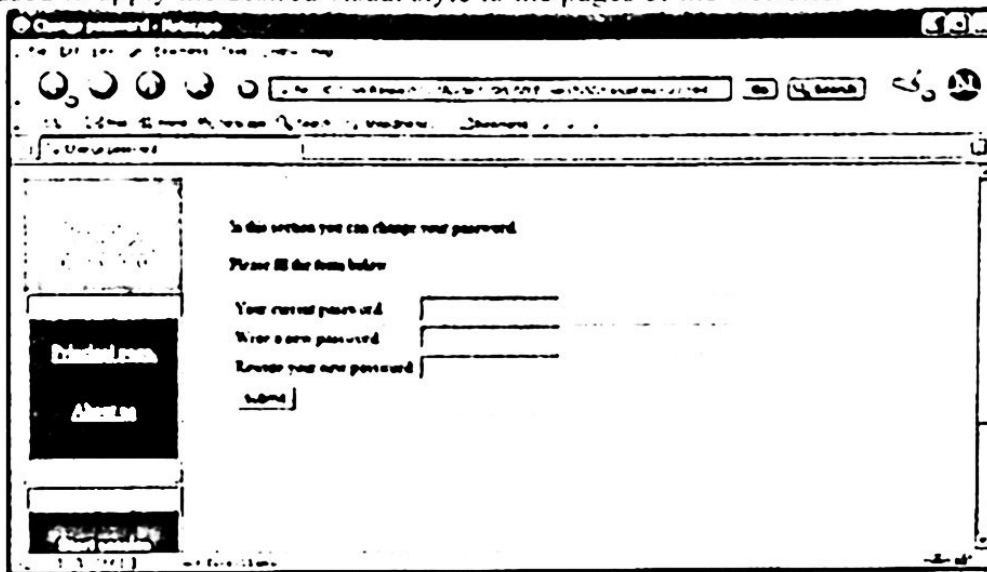


Fig. 4.1. Web page automatically generated from the element *startUserSession*.

The style sheet basically parses the elements contained in each page and generates HTML code. Finally, each page in the site is stored into its corresponding directory inside of the root directory; where each directory corresponds to a section in the site. Figure 4.1 shows the web page automatically generated using the following algorithm.

algorithm WAC(*WSMML document*, *String style*)

input: a WSMML document and an existing style sheet.

output: a set of web pages in HTML code and possibly embedding other kinds of code.

```

root <- root element of the document
sitename <- value of the attribute name in the root element
create a directory called sitename
switch to the recently created directory
for each section element in root do
    sectionName <- value of the attribute name in the current element
    create a directory with the name specified by sectionName
    set the recently created directory as the current directory
    for each navList or navControl element in root do
        createNavList(current navList or navControl element)
    end
    for each webpage element in root do
        createPage(current webpage element)
    end
end
    
```



```

end
function createPage(WSML webpage)
  name <- value of the attribute name of the webpage element
  extension <- the string "html"
  for each element in webpage do
    transformElement(element)
  end
  for each applImport element in webpage do
    importApplication(applImport)
  end
  load the stylesheet from the templates repository specified by the style attribute
  copy the instructions of the webpage element in the stylesheet
  page <- the document obtained from the previous step
  save the content of page, in the current directory, with its name and extension
end

function createNavList(WSML navElement)
  elemName <- the name of the element navElement
  if elemName = "navList" then
    transformElement(navElement)
  else
    navCS <- value of the attribute section of navElement
    pageArray <- all the visible web pages defined in the section navCS
    for each foundpage in pageArray do
      splink <- create a relative link to foundpage
      linkArray <- append splink
    end
    replace the navElement with the elements in linkArray
  end
end

function transformElement(WSML element)
  find a code template for element in the templates repository
  if found then
    apply the code template to element
  else
    keep element unmodified
  end
end

function importApplication(WSML instruction)
  imAppType <- value of the attribute type of the instruction element
  imAppName <- value of the attribute name of the instruction element
  if imAppType = "Script" then
    open the file called imAppName from scripts repository
    replace instruction with the content of the opened file
    extension <- the scripts language type of the imported application
  end
  if imAppType = "WAML" then
    wamIfile <- the root element of the file called imAppName
    in scripts repository
    generateScriptsCode(wamIfile)
    replace the instruction element with the code obtained in the previous step
  end
  if imAppType = "Applet" then
    replace instruction with an Applet invocation of the application called
    imAppName contained in the scripts repository
  end
end

function generateScriptsCode(WAML wamlapp)

```

```

applanguage <- value of the attribute type of the wamlapp element
select just the code templates for the language applanguage in the scripts repository
for each appinstruction element in wamlapp do
    find a code template for appinstruction in the scripts repository
    if found then
        apply the code template to transform this element
    else print an error message
    end
end
end
end
end

```

Fig. 4.2. The code generation algorithm WAC (Web Application Coder).

5 Case Study: A System for Evaluating Basic Education

Our approach has been tested with a web-based information system for evaluating basic education. In this system we implement two evaluation models: PEC and PRONAP [7, 8]. These models are part of the national reform of education in order to improve issues that currently limit the performance in basic education. This system includes three subsystems: optical reading, multimedia, and evaluation. The optical reading sub-system facilitates the data-entry from surveys applied to students, lecturers, and family parents; this is accomplished using the software Remark Office OMR and a scanner with an automatic document feeder. The multimedia sub-system facilitates the transcription of data collected from two devices: audio-recorder and video-recorder. The evaluation sub-system facilitate the analysis, generation, and management of structured and unstructured information of the evaluation models PEC and PRONAP; we use ontologies for the qualitative evaluation and causal relations for the quantitative evaluation. The three subsystems must provide or retrieve contents from an integrated web-site. Thus, using our approach, such subsystems must specify their applications/content using the markup languages WSML and WAML, these sub-systems do not have to worry about visual presentation; besides, they might save programming effort through using WSML and automatically generate code, or they might embed applications/scripts through the use of WAML. Our preliminary investigations suggest the following advantages: consistent visual view, less time and effort spent in programming, less costs in the development face, and less expertise required for designing web sites. We implemented this system using XSLT, JDOM, and Java [9].

6 Conclusions and Future Work

In this paper we have presented an approach for facilitating development of web-based information systems through markup languages and automatic code generation. For such purpose, we described a tool called WISBuilder based on the MVC pattern design in an attempt to automate the development of two components: view and controller. The view is a reusable XSL file that can be implemented in any development environment, with such file we apply a consistent view for the entire web site. The controller is specified through the use of two markup languages, and is intended to use

a GUI environment for easy and fast development. Through the separation of these two components (view and controller), a web-based information system can be developed in parallel in order to save time and separate technical expertise. The markup languages WSMML and WAML were very useful for specifying web contents. Such intermediate languages offer several advantages, for instance, web contents in a more declarative language, several kinds of code can be generated, and dynamic contents for adaptive navigation. The WAC algorithm was useful for automating the process of generating fully functional HTML code given the markup languages and code templates. Therefore, our approach facilitates the development of the *view* through reusability and consistent application of style sheets, and the *controller* through automatic code generation. In this regard, WISBuilder can be very useful for building web sites in new domains with a significant reduction in programming effort. The goals of WISBuilder are twofold: 1) to serve as a testbed for performing research in the area of web-based information systems, and 2) to save time and effort spent in the web development process so that more services might be automated in the world.

As an immediate work we plan to create a Graphical User Interface (GUI) tool for designing web sites based on the markup languages WSMML and WAML, in such a way that WISBuilder might become into a CASE tool for software engineers. Also, we plan to perform more experiments in other domains. After such work, our approach could be extended into a more robust framework in order to include: adaptive navigation [10], validation of data-entry, and automatic testing of the generated web sites. Finally, we plan to offer WISBuilder freely available for educational and research purposes under request.

Acknowledgements

We would like to thank CONACYT and the government of Tlaxcala for the financial support of this research, FOMIX project TLAX-2003-C02-12485. Also, we would like to thank the collaboration of the Direction of Research and Evaluation of Education of the Mexican Ministry of Education - Tlaxcala State.

References

1. Ceri, S., Fraternali, P., Bongio, A.: Web Modelling Language (WebML): a Modelling Language for Designing Web sites. WWW Conference, 2000.
2. Brambilla M., S. Ceri, S. Comai, P. Fraternali: Model-driven development of Web Services and hypertext applications. 2003.
3. Turau, V.: A Framework for Automatic Generation of Web-based Data Entry Applications Based on XML. In Proceedings of Symposium on Applied Computing (SAC 2002), Madrid, Spain, March 2002.
4. Guillen M., Del Rosario M., Sosa V., and Hernandez H.: GARP: A tool for Creating Dynamic Web Reports Using XSL and XML Technologies, In Proceedings of ENC, Tlaxcala, México, September 08 - 12, 2003, 54 - 59.
5. Suzuma, T., Nakada, H., Saito, M., Matsuoka, S., Tanaka, Y., Sekiguchi, S: The Ninf Portal. An automatic Generation Tool for Grid Portals. Seattle, Washington, USA. November 2002.

6. Cuayáhuatl H., Rodriguez M., and Montiel J., VoiceBuilder: A Framework for Automatic Speech Application Development, México, April 2004.
7. PEC: Programa Escuelas de Calidad, <http://www.escuelasdecalidad.net>
8. PRONAP: Programa Nacional para la Actualización Permanente, <http://pronap.ilce.edu.mx>
9. Armstrong, E., Ball, J., Bodoff, S., Bode D., Fisher, M., Fordin, S., Green, D., Haase, K., Jendrock, E: The Java Web Services Tutorial. <http://java.sun.com/webservices>, Feb 2003.
10. Po-Hao Chang, Wooyoung Kim, and Gul Agha: An adaptive Programming Framework for Web Applications. 2003. The IEEE/IPSJ Symposium on Applications and the Internet (SAINT04). Tokyo, Japan January 26 - 30, 2004.

Appendix A: DTD of the WSMML Markup Language

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- ***** -->
<!-- Elements for specifying contents of a web site -->
<!-- ***** -->
<!ELEMENT website (section+)>
<!ATTLIST website name CDATA #REQUIRED>
<!ELEMENT section (webpage+)>
<!ATTLIST section name NMTOKEN "ROOT">
<!ELEMENT webpage ((navControl | navList)*, (text | form | link | include
| appImport | attachXml | publish)*)>
<!ATTLIST webpage
    name NMTOKEN "index"
    section NMTOKEN "ROOT"
    title CDATA #REQUIRED
    private (yes | no) "no"
    visible (yes | no) "yes"
>
<!ELEMENT text (#PCDATA | link)*>
<!ATTLIST text style (title | p | heading | bold | italics | big | small)
"p">
<!ELEMENT link EMPTY>
<!ATTLIST link label CDATA #REQUIRED to CDATA #REQUIRED>
<!ELEMENT navList (link)+>
<!ATTLIST navList title CDATA #REQUIRED id ID #REQUIRED>
<!ELEMENT navControl EMPTY>
<!ATTLIST navControl section NMTOKEN #REQUIRED>
<!ELEMENT form ((input)+, submit, reset?)>
<!ATTLIST form action NMTOKEN #REQUIRED>
<!ELEMENT input EMPTY>
<!ATTLIST input
    type (button | checkbox | file | image | hidden | password |
radio | text) #REQUIRED
    name NMTOKEN #REQUIRED
    label CDATA ""
    maxlength NMTOKEN "20"
>
<!ELEMENT submit EMPTY>
<!ATTLIST submit label CDATA "submit">
<!ELEMENT reset EMPTY>
<!ATTLIST reset label CDATA "reset">
<!ELEMENT attachXml EMPTY>
<!ATTLIST attachXml path CDATA #REQUIRED stylesheet CDATA #REQUIRED>
<!ELEMENT include EMPTY>
<!ATTLIST include
    path CDATA #REQUIRED
    type (text | image | file | xml) #REQUIRED
>
<!ELEMENT publish EMPTY>
<!ATTLIST publish
    FileTypes NMTOKENS #REQUIRED
    dir CDATA #REQUIRED
    orderby (name | ext | size) "name"
    style (table | list) "list"

```

```
>  
<!ELEMENT appImport EMPTY>  
<!ATTLIST appImport  
    name CDATA #REQUIRED  
    type (Script | JS | Applet | WAML) #REQUIRED  
>
```


Appendix B: DTD of the WAML Markup Language

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- ----- -->
<!-- Elements for specifying applications embedded into WAML -->
<!-- ----- -->
<!ELEMENT application (param*, (msg | sqlDo | sqlReport | startSession))>
<!ATTLIST application
    type (jsp | php) #REQUIRED
    name NMTOKEN #REQUIRED
>
<!ELEMENT sqlDo ((query, (onerror? | onsuccess?)*)+)>
<!ATTLIST sqlDo
    dbname NMTOKEN #REQUIRED
    server NMTOKEN "127.0.0.1"
    usr NMTOKEN #REQUIRED
    pass NMTOKEN #REQUIRED
    driver (odbc | mysql | oracle) "odbc"
>
<!ELEMENT sqlReport ((query, (onerror? | onsuccess?)*)+)>
<!ATTLIST sqlReport
    dbname NMTOKEN #REQUIRED
    server NMTOKEN "127.0.0.1"
    usr NMTOKEN #REQUIRED
    pass NMTOKEN #REQUIRED
    driver (odbc | mysql | oracle) "odbc"
>
<!ELEMENT startSession ((requiredField)+,errorMsg)>
<!ATTLIST startSession
    dbname NMTOKEN #REQUIRED
    tablename NMTOKEN #REQUIRED
>
<!ELEMENT requiredField EMPTY>
<!ATTLIST requiredField name NMTOKEN #REQUIRED variable CDATA #REQUIRED>
<!ELEMENT query (#PCDATA)>
<!ELEMENT onerror (msg | sqlDo | sqlReport | (query, (onerror? | onsuccess?)*))>
<!ELEMENT onsuccess (msg | sqlDo | sqlReport | (query, (onerror? | onsuccess?)*))>
<!ELEMENT msg (#PCDATA)>
<!ELEMENT errorMsg (#PCDATA)>
<!ELEMENT param EMPTY>
<!ATTLIST param names NMTOKENS #REQUIRED>

```

A New Swarm Intelligence Approach for Routing Algorithms

Jose L. Aguilar

Universidad de Los Andes, Av. Tulio Febres Cordero, Facultad de Ingeniería, CEMISID, Dpto.
De Computación, Escuela de Ingeniería Sistemas,
5101, Mérida, Estado Mérida, Venezuela,
aguilar@ula.ve

Abstract. In this paper is presented a distributed algorithm based on Ant System concepts, called Combinatorial Ant System, to solve dynamic combinatorial optimization problems. Our approach consists of mapping the solution space of the dynamic combinatorial optimization problem in the space where the ants will walk, and defining the transition probability and the pheromone update formula of the Ant System according to the objective function of the optimization problem. We test our approach on a telecommunication problem.

1 Introduction

Real Ants are capable of finding the shortest path from a food source to their nest without using visual cues by exploiting pheromone information [1]. While walking, ants deposit pheromone trails on the ground and follow pheromone previously deposited by other ants. The above behavior of real ants has inspired the Ants System (AS), an algorithm in which a set of artificial ants cooperate to the solution of a problem by exchanging information via pheromone deposited on a graph. Dorigo [2] proposed the first AS in his Ph.D. thesis. AS has been applied to the traveling salesman problem and quadratic assignment problem, among others combinatorial optimization problems [1-9]. On the other hand, different groups have been working on various extended versions of the AS paradigm (Ant-Q, etc.) [1, 5, 6].

In the AS applied to the Traveling Salesman Problem (TSP), a set of cooperating agents, called ants, cooperate to find good solutions to TSP's using an indirect form of communication through pheromone trails that they deposit on the edges of the TSP graph while building solutions. Informally, each ant constructs a TSP solution in a constructive way: it adds new cities to a partial solution by exploiting information gained from both past experience and a greedy heuristic. Memory takes the form of pheromone trails deposited by ants on TSP edges, while heuristic information is simply given by the edge's weights. There are two reasons to use the AS on the TSP: a) The TSP graph represents the solution space of this problem; b) The AS transition function has goals similar to the TSP objective function.

That is not the case for other combinatorial optimization problems. We have proposed a distributed algorithm based on AS concepts, called the Combinatorial Ant System (CAS), to solve static discrete-state combinatorial optimization problems [8,

9]. The main novel idea introduced by our model is the definition of a general procedure to solve Combinatorial Optimization Problems using AS. In our approach, the graph that describes the solution space of the Combinatorial Optimization Problem is mapped on the AS graph, and the transition function and the pheromone update formula of the AS are built according to the objective function of the Combinatorial Optimization Problem. In this paper we test the CAS on dynamic combinatorial optimization problems, that is, problems changing over time. Particularly, we study a telecommunication problem. This paper is organized as follows: Section 2 presents the AS and the CAS. Section 3 summarizes the experiments. Finally, conclusions of this work are presented in Section 4.

2 Theoretical Aspects

2.1 The Routing Problem like a Dynamic Combinatorial Optimization Problem

A dynamic combinatorial optimization problem is a problem changing over time. That is, it is a distributed time-varying problem which is a current challenger in the combinatorial optimization domain. The dynamic problem that we are going to study is the routing in telecommunication networks. Routing is a mechanism that allows information transmitted over a network to be routed from a source to a destination through a sequence of intermediate switching/buffering stations or nodes. Routing is necessary because in real system not all nodes are directly connected. The problem to be solved by any routing system is to direct traffic from sources to destinations maximizing network performances (e.g., rate of call rejection, throughput, etc.). In real networks traffic, the conditions and the structure of the network are constantly changing, for this reason are necessary dynamic routing algorithms.

2.2 Ant Systems

In general, the behavior of Ant Colonies is impressing to perform their objective of survival. It is derived from a process of *Collective Behavior*. This process is based on the ant communication capacities, which define the inter-relations between them. These inter-relations permit the transmission of information that each ant is processing. The communication among agents (ants) is made through a trace, called *pheromone*. Thus, an ant leaves a certain quantity of pheromone trail when it moves. In addition, the probability that an ant follows a path depends on the number of ants having taken the path (a large quantity of pheromone in a path means a large probability that it will be visited).

AS is the progenitor of all research efforts with ant algorithms and it was first applied to the TSP problem [2, 4]. Algorithms inspired by AS have manifested as heuristic methods that permit resolving combinatorial optimization problems. These algorithms mainly rely on their versatility, robustness and operations based on populations. The procedure is based on the search of agents called "ants", that is, agents with very

simple capabilities that try to simulate the behavior of the ants.

AS utilizes a graph representation (*AS graph*) where each edge (r, s) has a desirability measure γ_{rs} , called *pheromone*, which is updated at run time by artificial ants. Informally, the AS works as follows. Each ant generates a complete tour by choosing the nodes according to a probabilistic state transition rule; ants prefer to move to nodes that are connected by short edges, which have a high pheromone presence. Once all ants have completed their tours, a global pheromone updating rule is applied: a fraction of the pheromone evaporates on all edges, and then each ant deposits an amount of pheromone on edges which belong to its tour in proportion to how short this tour was. Then, we continue with a new iteration of the process.

The state transition rule used by ant system is given by the equation (1), which gives the probability with which ant k in city r chooses to move to the city s while building its t^{th} tour (transition probability from node r to node s for the k^{th} ant) [1-5]:

$$P_{rs}^k(t) = \begin{cases} \frac{[\gamma_{rs}(t)]^\alpha [\eta_{rs}]^\beta}{\sum_{s \in J_k^r} [\gamma_{rs}(t)]^\alpha [\eta_{rs}]^\beta} & \text{If } s \in J_k^r \\ 0 & \text{Otherwise} \end{cases} \quad (1)$$

Where $\gamma_{rs}(t)$ is the pheromone at iteration t , η_{rs} is the inverse of the distance between city r and city s ($d(r,s)$), J_k^r is the set of nodes that remain to be visited by ant k positioned on node r and, β and α are two adjustable parameters which determine the relative importance of trail intensity (γ_{rs}) versus visibility (η_{rs}). In AS, the global updating rule is implemented as follows. Once all ants have built their tours, pheromone (that is, the trail intensity) is updated on all edges according to the equation [1-5]:

$$\gamma_{rs}(t) = (1 - \rho)\gamma_{rs}(t-1) + \sum_{k=1}^m \Delta\gamma_{rs}^k(t) \quad (2)$$

Where ρ is a coefficient such that $(1 - \rho)$ represents the trail evaporation in one iteration (tour), m is the number of ants, and $\Delta\gamma_{rs}^k(t)$ is the quantity per unit of length of trail substance laid on edge (r, s) by the k^{th} ant in that iteration:

$$\Delta\gamma_{rs}^k(t) = \begin{cases} 1/L_k(t) & \text{If edge } (r,s) \in \text{tour completed by ant } k \\ 0 & \text{Otherwise} \end{cases}$$

Where $L_k(t)$ is the length of the tour performed by ant k at iteration t . Pheromone updating is intended to allocate a greater amount of pheromone to shorter tours. The general algorithm is summarized as follows:

Place the m ants randomly on the nodes of the AS graph

Repeat until system convergence

2.1. For $i=1, n$

2.1.1. For $j= 1, m$

2.1.1.1. Choose the node s to move to, according to the transition probability (equation 1)

2.1.1.2. Move the ant m to the node s

2.2 Update the pheromone using the pheromone update formula (equation 2)

2.3 The Combinatorial Ant System

There are two reasons for using AS on the TSP. First, the TSP graph can be directly mapped on the AS graph. Secondly, the transition function has similar goals to the TSP. This is not the case for other combinatorial optimization problems. In [8, 9], we have proposed a distributed algorithm based on AS concepts, called the CAS, to solve Combinatorial Optimization Problems. In our approach, we need to define:

- *The graph that describes the solution space of the Combinatorial Optimization Problem (COP graph).* The solution space is defined by a graph where the nodes represent partial possible solutions to the problem, and the edges the relationship between the partial solutions. This graph will be used to define the *AS graph* (this is the graph where the ants will walk).
- *The transition function and the pheromone update formula of the CAS,* which are built according to the objective function of the Combinatorial Optimization Problem.

In this way, we can solve any Combinatorial Optimization Problem. Each ant builds a solution walking through the AS graph using a transition rule and a pheromone update formula defined according to the objective function of the Combinatorial Optimization Problem. The main steps of CAS are:

- Build the AS graph.
- Define the transition function and pheromone update formula of the CAS.
- Execute the classical AS procedure (or one of the improved versions).

Building the AS graph

The first step is to build the COP graph, then we define the AS graph with the same structure of the COP graph. The AS graph has two weight matrices: the first one is defined according to the COP graph and registers the relationship between the elements of the solution space (COP matrix). The second one registers the pheromone trail accumulated on each edge (pheromone matrix). This weight matrix is calculated/updated according to the pheromone update formula. When the incoming edge weights of the pheromone matrix for a given node become high, this node has a high probability to be visited. On the other hand, if an edge between two nodes of the COP

matrix is low then it means that ideally if one of these nodes belongs to the final solution then the other one must belong too. If the edge is equal to infinite then it means that the nodes are incompatible (they can't be at the same time in a final solution).

We define a data structure to store the solution that every ant k is building. This data structure is a vector (A^k) with a length equal to the length of the solution (number of nodes that an ant must visit). For a given ant, the vector keeps each node of the AS graph that it visits.

Defining the transition function and the pheromone update formula

The state transition rule and the pheromone update formula are built using the objective function of the combinatorial optimization problem. The transition function between nodes is given by:

$$Tf(\gamma_n(t), Cf_{r \rightarrow s}^k(z)) = \gamma_n(t)^\alpha / Cf_{r \rightarrow s}^k(z)^\beta$$

Where $Cf_{r \rightarrow s}^k(z)$ is the cost of the partial solution that is being built by the ant k when it crosses the edge (r, s) if it is in the position r , $z-1$ is the current length of the partial solution (current length of A^k), and, α and β are two adjustable parameters that control the relative weight of trail intensity ($\gamma_n(t)$) and the cost function. In the CAS, the transition probability is as follows: an ant positioned on node r choose the node s to move according to a probability $P_n^k(t)$, which is calculated according to the equation given by:

$$P_n^k(t) = \begin{cases} Tf(\gamma_n(t), Cf_{r \rightarrow s}^k(z)) / \sum_{u \in J_r^k} Tf(\gamma_n(t), Cf_{r \rightarrow u}^k(z)) & \text{If } s \in J_r^k \\ 0 & \text{Otherwise} \end{cases}$$

When $\beta=0$ we exploit previous solutions (only trail intensity is used) and when $\alpha=0$ we explore the solution space (a stochastic greedy algorithm is obtained). A tradeoff between quality of partial solutions and trail intensity is necessary. The pheromone updating rule is defined by the equation (2), where the quantity per unit of length of trail substance laid on edge (r, s) by the k^{th} ant in that iteration ($\Delta\gamma_n^k(t)$) is calculated according to the following formula:

$$\Delta\gamma_n^k(t) = \begin{cases} 1 / C_r^k(t) & \text{If edge } (r, s) \text{ has been crossed by ant } k \\ 0 & \text{Otherwise} \end{cases}$$

Where $C_f^k(t)$ is the value of the cost function (objective function) of the solution proposed by ant k at iteration t . The general procedure of our approach is summarized as follows:

1. Generation of the AS graph.
2. Definition of the state transition rule and the pheromone update formula, according to the Combinatorial Optimization Problem.
3. Repeat until system convergence
 - 3.1. Place the m ants on different nodes of the AS graph.
 - 3.2. For $i=1, n$
 - 3.2.1 For $j=1, m$
 - 3.2.1.1. Choose the node s to move to, according to the transition probability (equation 3).
 - 3.2.1.2. Move the ant m to the node s .
 - 3.3 Update the pheromone using the pheromone update formula (equations 2 and 4).

3 Experiments

3.1 Routing Problem Resolution using the CAS

We can use our approach for point to point or point to multipoint requests. In the case of N nodes, N ants are launched to look for the best path to the destination. For a multipoint request with m destinations, $N.m$ ants are launched. The source node keeps in memory all paths that have been found by ants. Then, it chooses the best one. Finally, the path is reserved and a connection is eventually set up (in the case of a multipoint request, it is spanning trees found by ants to the multiple destination nodes which are compared).

Building the AS graph

For this case we use the pheromone matrix of our AS graph like the routing table of each node of the network. Remember that this matrix is where the pheromone trail is deposited. Particularly, each node i has k_i neighbors, is characterized by a capacity C_i , a spare S_i , and by a routing table $R_i = [r'_{n,d}(t)]_{k_i, N-1}$. Each row of the routing table corresponds to a neighbor node and each column to a destination node. The information of each row of the node i is stored in the respective place of the pheromone matrix (p.e., in the position i, j if k_i neighbor = j). The value $r'_{n,d}(t)$ is used as a probability. That is, the probability that a given ant, where the destination is node d , be routed from node i to neighbor node n . We use the COP matrix of our AS graph to describe the network structure. If there are link or node failures, then the COP graph is modified to show

that. In addition, in each arc of the COP graph is stored the estimation of the trip times from the current node i to its neighbor node j , denoted $\Gamma_i = \{\mu_{i \rightarrow j}, \sigma_{i \rightarrow j}^2\}$, where $\mu_{i \rightarrow j}$ is the average estimated trip times from node i to node j , and $\sigma_{i \rightarrow j}^2$ is its associated variance. Γ_i allows maintenance a local idea of the global network's status at node i . Finally, we define a cost function for every node, called $C_{ij}(t)$, that is the cost associated with this link. It is a dynamic variable that depends on the link's load, and is calculated at time t using Γ_i .

Defining the transition function and the pheromone update formula

In our model (equation 3), $C_j^k(t)$ is the cost of k^{th} ant's route, $\Delta\gamma_{is}^k(t)$ is the amount of pheromone deposited by ant k if edge (i, s) belongs to the k^{th} ant's route (it is used to update the routing table R_i in each node), and $P_{ij}^k(t)$ is the probability that ant k chooses to hop from node i to node j . Ant k updates its route cost each time it traverses a link $C_j^k(t) = C_j^k(t) + C_{ij}(t)$. An ant collects the experience queues and traffic load, which allows it to define information about the state of the network. Once it has reached its destination node d , ant k goes all the way back to its source node through all the nodes visited during the forward path, and updates the routing tables (pheromone concentration) and the set of estimations of trip times of the nodes that belong to its path (COP graph) as follows:

- The times elapsed of the path $i \rightarrow d$ ($T_{i \rightarrow d}$) in the current k^{th} ant's route is used to update the means and variance values of Γ_i . $T_{i \rightarrow d}$ gives an idea about the goodness of the followed route because it is proportional to its length from a point of view and from a traffic congestion point of view.
- The routing table R_i is changed by incrementing the probability $r_{i-1,d}^k(t)$ associated with the neighbor node $i-1$ that belongs to the k^{th} ant's route and the destination node d , and decreasing the probabilities $r_{n,d}^k(t)$ associated with other neighbor nodes n , where $n \neq i-1$, for the same destination (like a pheromone trail). The values stored in Γ_i are used to score the trip times so that they can be transformed in a reinforcement signal $r = f_1(\Gamma_i)$, $r \in [0,1]$. r is used by the current node i as a positive reinforcement for the node $i-1$:

$$r_{i-1,d}^k(t+1) = r_{i-1,d}^k(t) (1-r) + r$$

And the probabilities $r_{n,d}^k(t)$ for destination d of other neighboring nodes n receive a negative reinforcement

$$r_{n,d}^k(t+1) = r_{n,d}^k(t) (1-r) \quad \text{for } n \neq i-1$$

Finally, $C_{ij}(t)$ is updated using Γ_i too

$$C_{ij}(t+1) = \mu_{i \rightarrow j} \sigma_{i \rightarrow j}^2$$

3.2 Result Analysis

We have tested our algorithm on a set of model networks among which is US NSFNET-T1 (composed by 14 nodes and 21 bidirectional links, with a bandwidth of 1.5 Mbits and propagation delay with range from 4 to 20 ms). A number of different traffic patterns, both in term of spatial and temporal characteristics, have been considered. The network performance is expressed in throughput (delivered bits/s) and delivered time from source to destination. We compare our algorithm with the AntNET approach and the Shortest Path First algorithm (SPF) [1, 7]. Due to the space, we present part of the result, see [9] for the rest of experiments. Figures 1 and 2 show some results regarding throughput and packet delay for a Poisson temporal and random spatial distribution of traffic (this is the traffic pattern used) on NSFNET. These results are exemplar of the behavior of our algorithms, results obtained on other traffic pattern and network topology combinations are qualitatively equivalent (see [9] for more details).

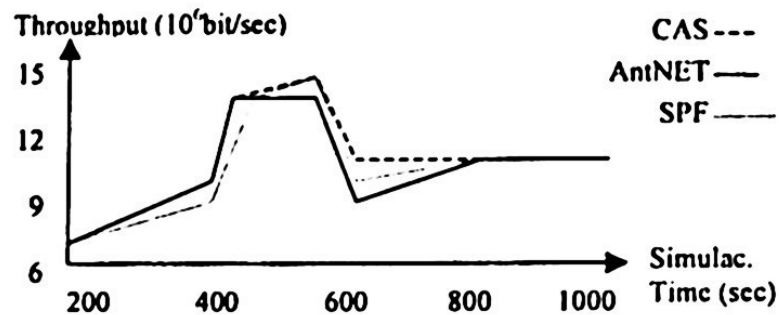


Fig 1. Throughput comparison between the algorithms

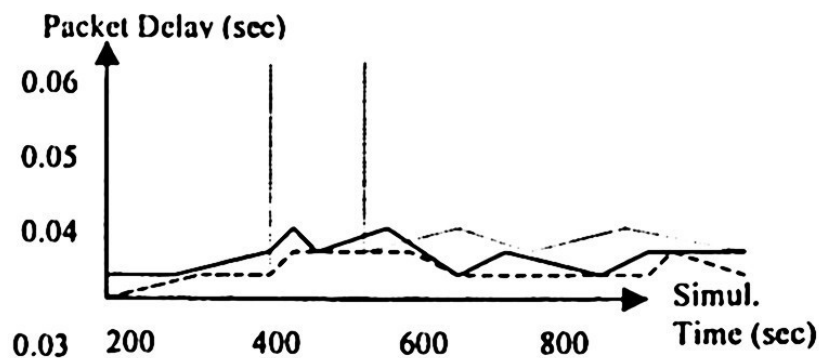


Fig 2. Packet delay comparison between the algorithms

The throughput of our approach is at least as good as that AntNET and the packet delays are much better than that of the others. Particularly, at the beginning our ap-

proach has not the best performance because it has learnt the current network situation, etc. After, it can optimize the route to be chosen in an impressing way.

4 Conclusions

In this work we have presented the versatile of the CAS to solve dynamic combinatorial optimization problems. Our system is suited for both static discrete-state and dynamic combinatorial optimization problems. This versatility has been exemplified by the possibility of using the same model to solve different combinatorial optimization problems (static and dynamic) of various sizes. Our approach can be applied to any combinatorial optimization problems by defining an appropriate graph representation of the solution space of the problem considered, the dynamic procedure to update that representation, and an objective function that guides our heuristic to build feasible solutions. In our approach, the dynamic environment of the combinatorial optimization problem is defined through the COP matrix (it form part of the space where the ants will walk (AS graph)). Ants walk through this space according to a set of probabilities updated by a state transition and a pheromone update rule defined according to the objective function of the combinatorial optimization problem considered.

We have tested our approach on a dynamic optimization problem (the routing problem). The results show that our approach obtains good performances, but we must improve the execution time of a given iteration and reduce the number of iterations. In general, CAS allows making an exhaustive searched, in this way it can obtain better performances than previous heuristic routing algorithms. Furthermore, we will develop a parallel version of our approach, we will test our approach over other dynamic combinatorial optimization problems. In addition, for the routing problem, we will test with a general packet-switching network avoiding the "symmetric path costs", and we will develop a network failure management system based on this approach.

References

1. E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence: from Natural to Artificial Swarm Systems*, Oxford University Press: USA (1999).
2. Dorigo, M. *Optimization, Learning and Natural Algorithms*, Ph.D Thesis, Politecnico de Milano, Italy (1992).
3. Come, D., Dorigo, M., Glove, F. *New ideas in Optimization*, McGraw Hill: Holland, 1999.
4. Dorigo, M., Maniezzo V., Coloni, A. The ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man, Cybem.*, Vol. 26, (1996) 29-41.
5. Dorigo M., Gambardella, L. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Trans. on Evolutionary Computation*, Vol. 1, (1997) 53-66.
6. Hidrobo F., Aguilar, J. Toward a Parallel Genetic Algorithm Approach Based on Collective Intelligence for Combinatorial Optimization Problems, *Proc. IEEE International Conference on Evolutionary Computation*, (1998). 715-720.
7. Schoondervoerd, R., Holland, O., Bruten, P., Rothkrantz, L. Ant-based Load Balancing in Telecommunications Networks, *Adaptive Behavior*, Vol. 5, (1997) 169-207.

8. Aguilar, J., Velásquez, L., Pool, M. The Combinatorial Ant System, *Applied Artificial Intelligence Journal*, Vol. 18, (2004) 427-446.
9. Aguilar, J. La Inteligencia Colectiva en la Resolución de Problemas de Optimización Combinatoria Dinámicos, *Proceeding of the XIII Simposio Internacional de Métodos Matemáticos Aplicados a las Ciencias*, (2004) 21-22.

Sub-Game Perfection in Semantic Web Services

Stephen Kinsella, Juan M. Gomez, Christoph Bussler, and Dieter Fensel

Digital Enterprise Research Institute,
National University of Ireland,
Galway

{stephen.kinsella, juan.gomez, chris.bussler, dieter.fensel} @deri.org

Abstract. We describe the dependence of web services on the pre and post conditions that trigger their use in on-line B2B and B2C transactions. We build a model that highlights these dependencies probabilistically and illustrate the use of the model with an example. We place this model in the context of the development of the nascent Semantic Web. Our result is that, once many web services are chained together, this dependence on different and increasingly complex triggers will produce a decay in the number of successful uses of the web service. This result lays a general theoretical foundation for a game-theoretic examination of web service interaction. We conclude the paper with a program for further work in this area.

1 Introduction

Web Services are usually modeled using process flow descriptions. In this paper we model a rudimentary web service as a series of games, a series of sub games and a series of equilibria. We recognize from the outset that the concept of a subgame-perfect equilibrium is an inherently incomputable one and we are at pains to make the distinction between a backwardly induced equilibrium such as we might find in a game of complete information, and the equilibrium we define inside a game of *incomplete* information. In this paper, we introduce the concept of sub game perfection, and then model a simple web service, buying a book, as depending on pre- and post- conditions, described below. We then generalize this example using the tools of game theory. We take the viewpoint that in order to understand the effectiveness of web services in general, it is necessary to understand the dependency of the web service on its various pre- and post- conditions, whatever they may be. Our lever is the introduction of the concept of subgame perfect Nash equilibria to the literature on web service modeling. To our knowledge, this has not been attempted before.

1.1 What are Web Services?

Complexity of integrating heterogeneous software components can be achieved by means of web services. They provide standard protocols for discovering, invoking, describing and composing services. Current web service technology, based on SOAP, WSDL and UDDI, has a very basic and non-automated interaction model. A web service provider publishes information in an UDDI registry and offers a programmatic

access that potential customers can bind to. For this, interaction via XML based messages on a transport protocol with a web service provider provides the desired functionality. The dynamic binding to the consumed Web services is done using the information from the web service registry. In this process, no real automation is achieved, and the human actor is involved in the loop.

Composition of simple services in complex ones represents a natural evolution of the technology. Mainly in e-Business scenarios, the standardization efforts to integrate them into business processes have evolved into many proposals, from which BPEL4WS [1] seem to be ahead up to now.

However, the ultimate goal is to have an automation process of the aforementioned tasks required for a web service. For this, the new paradigm of Semantic Web Services comes into the arena. To bring web services to its full potential semantics must be used in order to describe web services full capabilities. The Semantic Web and its key enabling technology, ontologies [2] provide a means to do so. They interweave human understanding of symbols with their machine-processability, allowing declaration and description of services. The combination of Semantic Web and Web Services should enable users to locate, select, employ, compose, and monitor Web-based services automatically [3].

There are two main initiatives to describe Semantic Web Services. OWL-S is an upper-ontology for declaring and describing services by employing a basic set of classes and properties. It is composed of three parts, each one describing one aspect of the service: the Service Profile defines what the Service does, the Service Model how it works and the Service Grounding how to access it.

The Web Services Modeling Framework [4] describes a full-fledged framework for describing Semantic Web Services with relies in the principles of loosely-decoupling and strong mediation, plus four main elements, namely: ontologies, goal repositories, web service description and mediators. For the purpose of this paper, we will stress the fact that both approaches count on the notion of pre-conditions and post-conditions when describing actual web services. In a nutshell, preconditions are conditions over the input and post-conditions are conditions over the output. These concepts will be further developed in the following sections.

Why is this approach useful? There are many ways of modeling web services, and few of them incorporate both a process view of the service and a probability-based description of that service process. We believe this approach is both novel and useful in determining the optimal service description. We provide both a general specification and a specific example of the modeling methodology below.

1.2 What is Subgame Perfection?

First some definitions, then an example.

Definition 1 *The normal form representation of an n -player game specifies the players' strategies S_1, \dots, S_n and their payoff functions u_1, \dots, u_n . We denote this game by $G = (S_1, \dots, S_n; u_1, \dots, u_n)$.*

In addition to the normal form representation of the game, which we will use briefly at the end of the paper, there is the *extensive form representation* of the game, which we will use for the majority of the paper. Here the set of all legal paths through the space of positions is laid out in a tree. The informational constraints imposed on the players' choices are represented by grouping various 'nodes' of the tree into information sets; the interpretation being that whenever any node in an information set occurs a player must choose a move without knowing which particular node within the information set has occurred.

Definition 2 *The extensive form representation of a game specifies*

1. *the players in the game,*
2. *when each player has her move,*
3. *what each player can do each one of their opportunities to move, and*
4. *the payoff received by each player for each combination of moves that could be chosen by the players.*

It is important to define all our concepts precisely, because we will be manipulating them in slightly unorthodox ways later on. Thus, a *Nash Equilibrium* is defined in terms of the normal form game as the following:

Definition 3 *In the normal form game $\gamma = (S_1, \dots, S_n; \delta_1, \dots, \delta_n)$ the strategies (s_1^*, \dots, s_n^*) are a Nash Equilibrium if, for each player i , s_i^* is player i 's best response to the strategies specified for the $n-1$ other players.*

Definition 4 *A subgame is an extensive form game that*

1. *begins at a decision node η and is a singleton information set, but not the game's first decision node,*
2. *includes all the decision and terminal nodes following η in the game tree, but no nodes that do not follow η , and*
3. *does not cut any information sets.*

In many textbook examples, in games that assume fully rational players with unbounded computational skills, perfect knowledge of the opponent's skills, with well defined preferences over uncountable sets of lotteries, perfect knowledge of the preferences of the others' preferences, rationality and common knowledge, then the procedure for solving extensive form games is to compute the *Nash Equilibrium* by backwards induction. We will not do so here. Many textbook in game theory (See [5, pp.12-13] for examples) assume a continuum of unbounded integers, for example the reals, \mathbb{R} , as the field onto which all mappings take place. There is no room inside functioning web service architectures for uncountable infinities. For an excellent summary of the seminal contributions to the emerging area of explicitly computable games, see [6], especially pages 89--97.

The procedure for constructing a subgame perfect Nash equilibrium is as follows. First, identify all the subgames that contain terminal nodes in the original game tree. Then replace each such subgame with the payoffs from one of its Nash equilibria.

Now think of the initial nodes in these subgames as the terminal nodes in a truncated version of the original game. Working backwards through the tree in this way yields a subgame perfect Nash equilibrium because the player's strategies constitute a Nash equilibrium in every subgame.

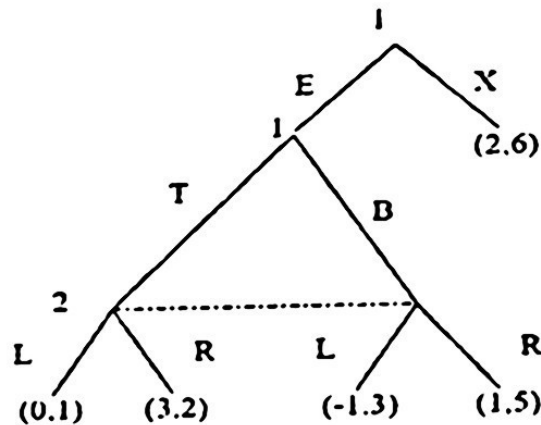


Fig. 1. Subgame Perfection with incomplete information

1.3 An Example

Consider the extensive form game shown in figure 1 above. This game has two subgames: one starts after player 1 plays *E*; the second one is the game itself, which can see from Figure 2 below.

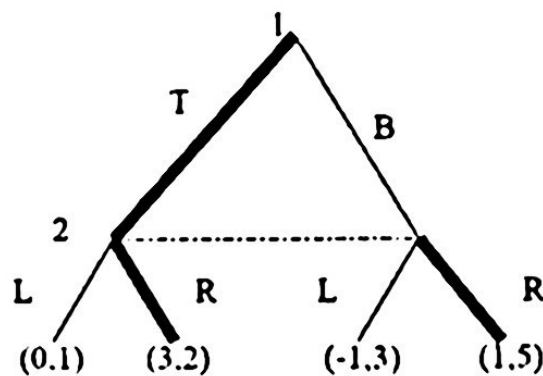


Fig. 2. The Subgames in Relief

We compute the subgame perfect equilibria as follows. We first compute a Nash equilibrium of the subgame, then fixing the equilibrium actions as they are (in this subgame), and taking the equilibrium payoffs in this subgame as the payoffs for entering in the subgame, we compute a Nash equilibrium in the remaining game.

As we can see from figure 3 below, the subgame has only one Nash equilibrium, as *T* dominates *B*: Player 1 plays *T* and player 2 plays *R*, yielding the payoff vector (3, 2).

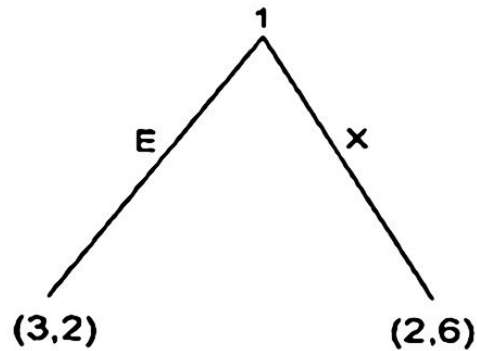


Fig. 3. Final Payoff to the Subgame Perfect Nash Equilibrium

Subgame perfection generalizes the notion of Nash equilibria to general dynamic games via the following definition:

Definition 5 *A Nash equilibrium is said to be subgame perfect iff it is a Nash equilibrium in every subgame of the game.* What is a subgame? In any given game, there may be some smaller games embedded; we call each such embedded game a subgame. Now we are ready to begin our model.

2 The Model

We begin with the description of the general pre—and post-condition schema for web services. Our model takes the ‘players’ of the game to be the pre and post conditions of the semantically enabled web services themselves. That is, they are predicates without which the service cannot function, and returns an annotated null value ϕ for any query sent to it. It will be clear from figure 3 that there is no way to compute the Nash equilibrium without resorting to the use of subgame perfect equilibria first. This is as a result of the inherent incomputability of the subgame schema, which we mentioned above. It should be noted that there are algorithms for computing sequential equilibria on recursively defined games, and that these equilibria will yield a subgame perfect equilibrium [7]. We do not use this formulation because we feel a more general treatment of the concept is less restrictive and, more importantly, more specific from the viewpoint of semantic web services. We begin with the description of the post condition.

2.1 Post Conditions

Post-conditions in WSMF describe what a web service returns in response to its input. OWL-S defines effects, as the results of the successful execution of the service as described by the OWL-s coalition at <http://www.daml.org/services/owl-s/1.0/>. These definitions are quite similar in nature: they define conditions that hold *after* the

correct execution of the service. Hence we define post-conditions as conditions that hold after the execution of the Web Service. Similar to pre-conditions, our definition is not limited to state-of-the-world post-conditions (changes in the state of the world), but it also includes knowledge post-conditions, i.e. knowledge that is made available to the requester after the service execution. In the example of the service presented before, the post-conditions are the list of books in the given country from the given location. Imagine that this service is not a free service, and that the credit card of the requester is charged by a given amount. Then, the charge of the credit card would be a post-condition as well. Pre-conditions and post-conditions fully described the conditions that must be true *before* requesting the service and the ones that are true *after* its execution.

2.2 Post Condition Subgame

The idea. The model takes queries sent to the web service as the 'players' of the game. The web service can handle $n \in (1, \dots, N]$ queries in a given time frame $t \in (1, \dots, T)$. The 'strategies' of the game are simply complete plans of the system, completely deterministic and specified exactly. The payoffs to the queries are their passage through the web service, i.e. completion C , or non-completion, NC , after which the post condition returns a value of 0 at time t to the post conditions via a feedback process Y which is completely specified and depends on $Y = (1 - \alpha)Y_{x,t}^{t-1} + \alpha(\delta_x^{t-1})$, where $0 < \alpha < 1$.

The model assumes that there are discrete queries of the form (x, y) sent to the web service, and that these queries represent the search for a good—in our example, a book. These queries are then processed by the pre-conditions, checked individually and then collectively, and then sent to the service for further processing. Assuming the product is in stock, shipping charges are met, etc, the query is sent on, and fulfills the post condition, returning a payoff of either 1 or 0.

Assumption 1 *There is a closed metric space X and $M(X)$ is the probability distribution on X , where X specifies some field σ , and at any stage of the process, there is a query x, y over σ which assigns some value δ to the post condition, where $\delta \in (0, 1)$.*

Assumption 2 *The eventual payoff to the post condition from the game γ is δ_x^t for some $\delta \in (0, 1)$, defined above, which is the set of all feasible divisions of this post condition.*

It means that the post condition, once satisfied by the result returned by query x at time t , will return a value of 1 (A receipt of success) if the service has worked properly.

Assumption 3 *Each query (x, y) is risk-neutral as regards security, and the future is discounted exponentially.*

It can be seen that D , the set of all feasible divisions of the post-condition's successful or unsuccessful return is

$$D = (x, y) \in [0, 1]^2 \mid x + y \leq 1 \quad (1.1)$$

and the subgame perfect equilibrium will be as follows: query 1 accepts (x, y) with

$$x \geq \frac{\delta(1 - \delta^{2k})}{1 + \delta} \quad (1.2)$$

Thus, at any query time $t = 2n - 2k$ where k is a non-negative integer, the post condition returns a value of 1. And, of course, rejects with

$$x < \frac{\delta(1 - \delta^{2k-1})}{1 + \delta} \quad (1.3)$$

Query y will form a subgame perfect equilibrium around x by definition 4—that the decision node η was a singleton information set— η only considers the situation at the present and previous decision nodes, and not the future nodes. Thus we have query 2 offering

$$(x_{t-1}, y_{t-1}) = \left(1 - \frac{\delta(1 + \delta^{2k+1})}{1 + \delta}, \frac{\delta(1 + \delta^{2k+1})}{1 + \delta} \right) \equiv \left(\frac{1 - \delta^{2k+2}}{1 + \delta}, \frac{\delta(1 + \delta^{2k+1})}{1 + \delta} \right) \quad (1.4)$$

where the subgame perfect equilibrium can be found by backwards induction or solution of the state variable, δ .

Proof Equation 1.4 follows naturally from [8, pp. 111-112, §2] which also includes a proof of incomputable strategies in repeated games.

Consider figure 4 below. Here we see the service description for our example.

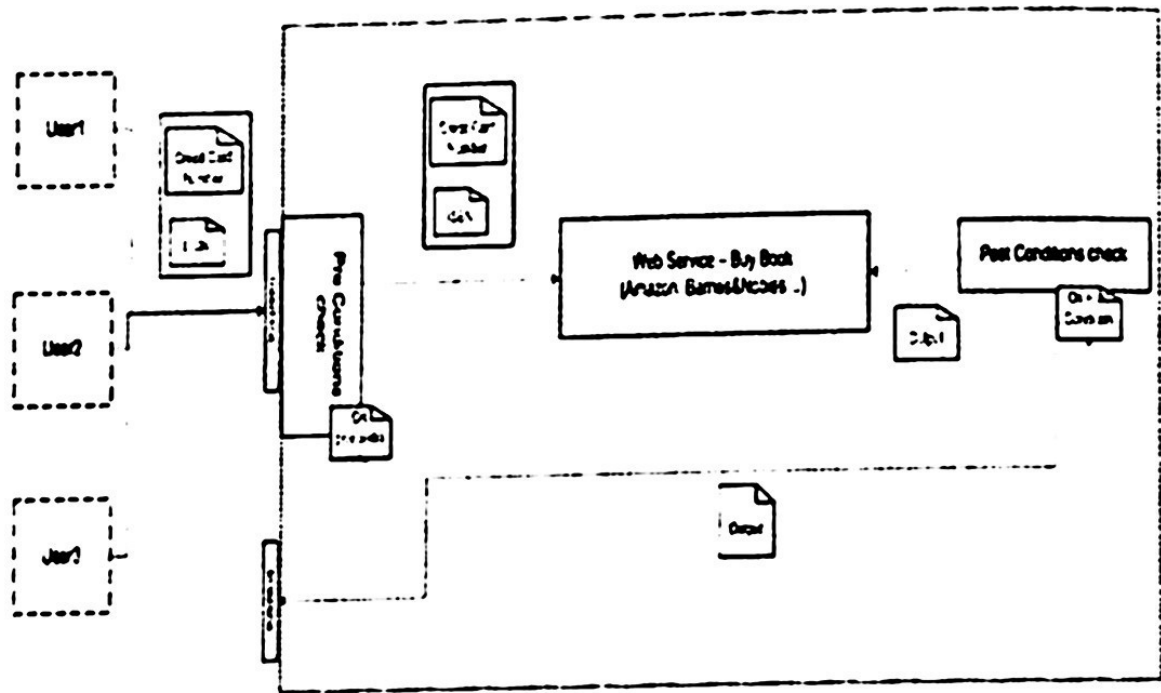


Fig. 4. Service Description 1-Process View

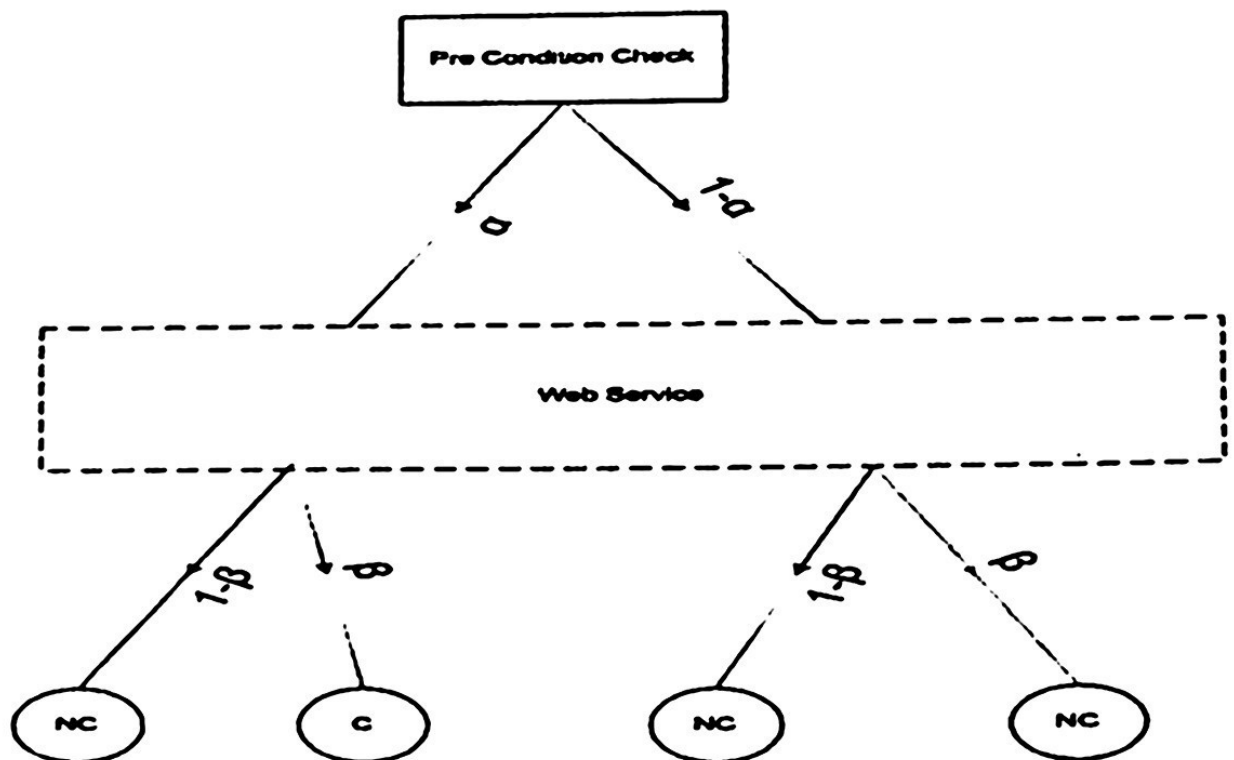


Fig. 5. Service Flow Description 2, extensive form game theoretic view

2.3 Pre Conditions

Pre-conditions in WSMF describe what a web service expects for enabling it to provide its service. OWL-S (<http://www.daml.org/services/owl-s/1.0/>) bases its definition of *Web Service functionality* in concepts coming from the planning community, adapting the definition of a pre-condition to the Web Services domain. In OWL-S, pre-conditions are defined as logical conditions that must be satisfied before requesting the service. These conditions are referred to the state of the world, in contrast with inputs, that are regarded as knowledge pre-conditions. It can be seen that all these definitions reflect a similar concept: conditions that must hold in order to apply an action (similarly, to request a service). Pre-conditions act as pre-requisites to use a service or action, that is, they must hold prior to the use of the service or action. Following this line, we define pre-conditions as conditions that *must* hold previous to the request of the Web Service for enabling it to provide its declared service. As an example, consider a Web Service that provides a list of the books in a given country, from a given location. The pre-conditions of this service are the location and the country. Now imagine this Web Service provides its service only to registered users that paid their monthly subscription rate. In this case, an additional pre-condition would be that the requester has paid the rate.

2.4 Pre Condition Subgame

Let α be the collection of first stage strategies $(\alpha_1, \dots, \alpha_n)$ and β be the collection of second stage strategies $(\beta_1, \dots, \beta_m)$. Then $\beta(\alpha) = (\beta_1(\alpha), \beta_2(\alpha), \dots, \beta_m(\alpha))$ indicates the choices made by the second stage of the web service after returning a positive check on pre-condition α . Again, the expected payoff is a sum over the finite space, X , and recording the 'moves' of (α, β) is $E[\delta_x | \alpha, \beta \geq \delta_x[\bar{\alpha}_x, \alpha_{-x}, \beta], \forall \bar{\alpha}_x \in \alpha]$, and similarly for β_y . This being given, two of the conditions of the subgame must be generated to satisfy definitions 3 and 4.

$$\forall x \in X, E[\delta_x | \alpha, \beta \geq \delta_x[\bar{\alpha}_x, \alpha_{-x}, \beta]] \forall \bar{\alpha}_x \in \alpha \quad (1.5)$$

and

$$\forall y \in X, E\left[\delta_y | \alpha, \bar{\beta}_y(\alpha)\right], \forall \bar{\beta}_y \in \beta \quad (1.6)$$

Condition (1.5) states that every first-stage query will select a best response if that is possible. Condition (1.6) requires that for any realization α in the first stage,

$\beta(\alpha)$ be a Nash equilibrium of the ensuing second-stage game. For any $\alpha, \in \alpha$, let α_x be the set of Nash equilibria of the second-stage game defined after α is realized. Thus, $\beta(\alpha) \in NC(\alpha)$ iff (1.6) holds for $\beta(\alpha)$.

As an example, consider Figure 5. Here we re-enter our example, looking at our 'buy book' example from a different view, reformulating the problem of pre- and post- condition selection in terms of probabilistically defined recursive strategies as discussed in [8]. We have two pre-conditions that must be satisfied before the web service can be initiated. These are: that the prospective customer must have a valid ISBN number and they must have a valid credit card number. Call the probability that the 'ISBN' is valid α and the corresponding probability that the credit card is valid β . Now see the process reformulated as a sequential decision making problem: even if both pre-conditions are being checked in parallel, it is useful to see one's dependency on the other for the initiation of the web service. Now, because of the open-world assumption of most web services, and the incomplete nature of most information sources, a useful way to solve this problem is to search for the subgame equilibrium within the system, in this example the only channel through which the web service will be initiated--- the payoff C , highlighted in red. Only when both pre-conditions are satisfied (i.e. when the extensive form tree structure is at 1 at the decision node, and all other positive probabilities are summed over $M(X)$) will the web service be invoked. It only remains to compute the overall equilibrium. From (1.5) and (1.6) above, we can see that the overall equilibrium for the pre-conditions

will be found at $\hat{\beta}, (\hat{\alpha})$ although the disequilibrium dynamics of a real time functioning system may keep the strict equilibrium set of queries far from this equilibrium. Nevertheless, given our game as constructed, there is an overall ideal solution point, where pre-and post-conditions are at their subgame-perfect equilibria and the feedback mechanism defined above returns a value of 1.

2.5 Implications

The implications of our model for construction of Semantic Web Services could be widely grouped in three main areas.

Execution performance In business processes where the flow of execution reverts into a big workload and fault tolerance problem for the hardware and software systems involved. Our model could be a best-of-breed solution to assign dynamically resources to the web services with a lower probability of being executed. In our further work section, we look to a use case study for just such an eventuality.

Business Intelligence performance In order to avoid the trap of producing huge inconsistent designs or not balanced architectures in modern e-Commerce systems, our model provides a formal and consistent metrics to evaluate the whole performance of the processes and the execution flow in terms of business intelligence.

Graphs and heuristics depending on our model could enhance the re-engineering and come out with a better solution in terms of cost-effectiveness and efficiency.

Non-functional information A detailed description of the service based on metadata or on semantic annotation could be inferred from our model. The Quality of Service (QoS) or policy, as described in [9] could add some information about the performance of the system. Also pay-per-use techniques, such as currently applied in real-world e-Business applications can learn from the estimated use provided by our model.

Amongst others future directions of the model would allow a further analysis of the scalability in high-computing grids or overlay networks.

3 Conclusion and Related Work

This paper has given a new way to characterize web service description as a series of probabilistic dependencies upon several pre- and post- conditions. In our admittedly rudimentary example the web service itself is purposely left a mystery, a 'black box' around which we highlight the ancillary but crucial role of pre- and post conditions in invoking and feeding back to the web service the queries that trigger it. We recognize the fact that, as conceived, this model is uncomputable and therefore of little practical use the web service developer. However, a simple restriction of this more general schema by defining a recursive game would be enough to allow the computation of the subgame perfect equilibrium in a sequential game. As stated above, this model fails only in its generality, not in the execution of the pre- and post- condition schema outlined above. We feel that this amounts to a suggestive failure, and one on which further work can and will be built.

3.1 Related Work

Several approaches investigated the issue of using pre- and post-conditions in the software engineering area. Preconditions, post-conditions and invariants are usually identified during the software development process, especially during the analysis and design phases. The UML specification [10] defines the Object Constraint Language (OCL) [11] that allows a designer to specify constraints of this kind on a particular object or method. Recent trends in the Semantic Web Services area such as the Web Service Modeling Ontology (WSMO) [12] use pre-conditions and post-conditions at the capability level of the web service. Finally, reference implementations such as the Web Services Execution Environment (WSMX) [13] pretend to integrate them in their future conceptual model and used them in runtime, as part of their execution model.

3.2 Future Work

When one introduces one web service and connects it to another and then another in a supply chain, for example, one increases the probability that one failed pre-condition will cause the entire chain of services to fail. Future work will concentrate on three aspects of this problem.

1. What is the specific rate of decay of this chaining of web services?
2. A use-case study based on the Adaptive Services Grid (ASG) project. ASG provides the integration of its sub-projects in the context of an open platform, including tool development by small and medium sized enterprises. Based on semantic specifications of requested services by service customers, ASG discovers appropriate services, composes complex processes and – if required – generates software to create new application services on demand. A case study examining just how long these integrated chains of software services can be will compare the predicted rate of decay to the actual rate observed throughout the project.
3. A tool to simulate the development of complex integration processes based upon the similarity (or lack thereof) of pre- and post- condition architectures.

Other future work will concentrate on refining several sub-classes of game-theoretic models like the rudimentary one above for use in the description of semantic web services in specific cases. The web service itself, modeled as a black box in this paper, can be recast as a sequential decision process, albeit a very complex one. This paper is a small step in that direction.

References

1. Curbera, F., Goland, Y., Klein, J., Leyman, F., Soller, D., Thatte, S., Weerawarana, S.: Business Execution Language for Web Services. BEA Systems and IBM Corporation and Microsoft Corporation (2002)
2. Fensel, D.: Ontologies, A Silver Bullet for Knowledge Management and Electronic Commerce. 2nd edn. Springer (2003)
3. McIlraith, S., Son, T., Zeng, H.: Semantic web services. *IEEE Intelligent Systems*. (Special Issue on the Semantic Web) 16 (2002) 46–53
4. Fensel, D., Bussler, C.: The web service modelling framework wsmf. *Electronic Commerce Research and Applications* 1 (2002)
5. Gibbons, R.: A Primer on Game Theory. Prentice Hall, New York (1992)
6. Velupillai, K.V.: The computable approach to economics. *Taiwan Journal of Political Economy* 1 (1995) 53–109
7. Azhar, A., McLennan, A., Reif, J.: Computation of equilibria in noncooperative games. Technical report, Duke University (1992)
8. Velupillai, K.V.: Computable Economics. Oxford University Press, Oxford (2000)
9. M.Dean, G.Schreiber: OWL Web Ontology Reference. <http://www.w3.org/TR/2003/PR-owl-ref-20031215/> (2004)
10. Booch, G., Rumbaugh, J., Jacobson, I.: The Unified Modelling Language (UML). (1999)

11. Warmer, A., Kleppe, J.: The Object Constraint Language (OCL): Precise modeling with UML. (1985)
12. Roman, D., Lausen, H., Keller, U.: Web Services Modelling Ontology (WSMO). (2004)
13. Cimpian, E., Mocan, A., Moran, M., Oren, E., Zaremba, M.: Web Services Execution Environment (WSMX). (2004)

Modeling and Simulation

Classification of Time Series Based on their Inner Structures

H. Solís-Estrella, E. Bautista-Thompson and J. Figueroa-Nazuno

Centro de Investigación en Computación, Instituto Politécnico Nacional
07738, Mexico D.F
{habacuc, ebautista}@correo.cic.ipn.mx, jfn@cic.ipn.mx

Abstract. There are techniques such as the Singular-Spectrum Analysis (SSA) which jointly with Principal Component Analysis (PCA), help us to analyze the underlying structures of a time series and condense them for their study. Using such information with the Multidimensional Scaling (MDS), we can find a representation that allows locating hidden regularities and thus classify the analyzed data. In this paper we present the study of time series of diverse nature using the three abovementioned techniques and show how they group in a bidimensional plane according to similar patterns within their components disregarding the dynamics of the series.

1 Introduction

Finding similarity among signals in a time series database has drawn plenty of attention in the data mining area in view of the fact that it is a very useful tool for analysis and knowledge extraction of phenomena [1]. A main aspect in similarity search is to define a set of relevant characteristics and find a metric or scheme to classify the data in accordance with defined criteria, i.e. the time series classification.

The time series classification problem is still open because no specific scheme suits all possible required criteria nor takes in account all the essential parameters, thus for particular needs new classification proposals must be developed. Traditionally, for the classification of time series, two approaches are taken, the first one is to classify according to scalar parameters of the data [2], the second is to classify according to the time series' morphology using a transformation [3] or lower dimensionality representation of the data [4,5]. Here we propose a scheme that attempts to uncover hidden relations among time series classifying them according to the underlying structures rather than the external waveform. The paper layout is as follows: in section 2 we review the analysis techniques, SSA and PCA, in section 3 the visualization and classification technique, MDS. Finally in section 4 we discuss the experimental results.

2 Analysis of the Time Series

2.1 Singular-Spectrum Analysis

Singular Spectrum Analysis (SSA) is a technique for time series analysis that incorporates elements of several disciplines such as multivariate statistics, multivariate geometry, dynamical systems and signal processing. It aims at a decomposition of the data into a sum of a small number of interpretable components such as trends, oscillatory patterns and random constituents [6]. The basic algorithm of the SSA technique consists of four steps. Given a time series $F = (f_0, f_1, \dots, f_{N-1})$ of length N and L an integer called "window length"

1. Construct the trajectory matrix X of the time series as follows:

set $K = L + N + 1$ and define the L -lagged vectors $X_j = (f_{j-1}, \dots, f_{j+L-2})^T, j = 1, 2, \dots, K$

$$X = (x_{i+j-2})_{i,j=1}^{L,K} = [X_1, \dots, X_K] \quad (1)$$

2. Obtain the Singular Value Decomposition (SVD) of the matrix X via eigenvalues and eigenvectors of the matrix $S = XX^T$. We thus obtain a representation of X as a sum of rank-one biorthogonal matrices X_i ($i = 1, \dots, d$), where d is the number of nonzero singular values of X
3. Split the set of indices $I = \{1, \dots, d\}$ into several groups I_1, \dots, I_m and sum the matrices X_i within each group. The result of the step is the representation

$$X = \sum_{k=1}^m X_{I_k} \quad \text{where} \quad X_{I_k} = \sum_{i \in I_k} X_i \quad (2)$$

4. Average over the diagonals $i+j=\text{const}$ of the matrices X_{I_k} . This gives us a decomposition of the original series F into a sum of series

$$f_n = \sum_{k=1}^m f_n^{(k)}, \quad n = 0, \dots, N-1 \quad (3)$$

where for each k the series $f_n^{(k)}$ is the result of diagonal averaging of the matrix X_{I_k} .

2.2 Principal Component Analysis

Related to the time series decomposition with the SSA technique, is the Principal Component Analysis (PCA). Once the series has been separated in different component series, PCA is employed as a complementary tool. The PCA consists in the search of lineally independent components which provide the maximum variance of the data set and are not correlated [7]. PCA is generally used to condense the amount of data and to extract important features from it. Colebrook applied a form of SSA to biological oceanography and noted the duality with the principal component analysis [8].

The procedure to compute the principal components is as follows: let X be a $n \times p$ matrix whose rows represent cases and its columns variables, in addition, X must be

mean-centered. Let a be the yet to be determined projection weights column vector, which will result in the greatest variance when X is projected into a . The variance along a is defined as:

$$\sigma_a^2 = (Xa)^T (Xa) = a^T X^T X a = a^T V a \quad (4)$$

where $V = X^T X$ is the covariance matrix. The projected data variance, σ_a^2 is expressed as a function of V and a .

To maximize σ_a^2 we need to apply a normalization restriction over a vectors which is $a^T a = 1$.

Consequently, the optimization problem can be rewritten as the maximization of the quantity:

$$u = a^T V a - \lambda (a^T a - 1) \quad (5)$$

where λ is a Lagrange multiplier. Differentiating with respect to a , we have:

$$\frac{\partial u}{\partial a} = 2Va - 2\lambda a = 0 \quad (6)$$

which is reduced to its eigenvalues form as:

$$(V - \lambda I)a = 0 \quad (7)$$

Therefore, the first principal component is the eigenvector associated with the largest eigenvalue in the covariance matrix V . The second principal component is the eigenvector associated with the second largest eigenvalue of V , and so on.

3 Multidimensional Scaling

Multidimensional Scaling (MDS) is a method that represents similarity metrics between pair of objects as distances between points in a lower dimensional space. This representation allows the expert observe and explore the data structure and find hidden regularities not easily distinguished in the raw numerical data [9].

MDS takes as input a proximity matrix $\Delta \in M_{n,n}$. Each element δ_{ij} of Δ represents the proximities between the parameters where $\delta_{ij} = (c_i - c_j)^2$ being c_k the aforementioned parameters.

The algorithm begins with a random matrix $X \in M_{n,m}$, where m is the desired number of dimensions and n is the number of variables. Each value x_{ij} represents the coordinates of the variable F_i in the j -th dimension.

Parting from such matrix, the distance among any two variables, i and j , can be calculated using the Minkowsky general distance equation:

$$d_{ij} = \left[\sum_{r=1}^m (x_{ir} - x_{jr})^p \right]^{1/p} \quad (8)$$

Thus, a distance matrix $D \in M_{n,n}$ can be obtained from X .

The MDS solution must be such that there is a maximum correspondence between the initial proximities matrix Δ and the distance matrix D , which is accomplished iteratively modifying X according to:

$$X = \frac{BX}{2n} \quad (9)$$

where the elements b_{ij} of B are calculated using the following criteria:

$$b_{ij} = \frac{-2\delta_{ij}}{\delta_{ij}} \quad \text{if } i = j \quad (10)$$

$$b_{ij} = \sum \sum \frac{-2\delta_{ik}}{d_{ik}} \quad \text{if } i = j \quad (11)$$

$$b_{ij} = 0 \quad \text{if } d_{ij} = 0 \quad (12)$$

Then a monotonous increasing relation $\delta_{ij} < \delta_{kl} \Rightarrow d_{ij} < d_{kl}$ is assumed. To determine the precision, a function that relates distances with similarity $f(\delta_{ij})$ is constructed:

$$\begin{aligned} f: \delta_{ij} &\rightarrow d_{ij} \\ f(\delta_{ij}) &= a + b\delta_{ij} \end{aligned} \quad (13)$$

where a and b are constants to establish.

Therefore, we can apply the S-Stress precision measure, defined as:

$$S - Stress = \sqrt{\frac{\sum_{i,j} (f(\delta_{ij})^2 - d_{ij}^2)}{\sum_{i,j} (d_{ij}^2)}} \quad (14)$$

To complete MDS process, if a desired precision is not met, X is modified according to (15) and start over until the required S-Stress value is achieved.

4 Experimental Results

For the results we present, the experimental data set consists of 30 time series of diverse dynamics as shown in Table 1, the reference nominal classification is according to Sprott [10] and Figueroa [11]. All the time series have a length of 1000 data samples.

The methodology is as follows: the series were decomposed using SSA and then de data was used to obtain the principal components using PCA. The matrix of the first five principal components was fed to the MDS algorithm to obtain the time series

representation in a bidimensional plane according to their structure. Only five components were used since nearly 90% of the variance is contained in those components as we can observe in Table 2.

In Figure 1 we show the final clustering of the series and the three main groups that were identified and the dynamic of the time series that are part of the cluster. In the figure we can appreciate that despite having different dynamics, several signals can belong to a group since they share similar inner structures, the local interactions of those basic structures are what make the time series behave in a particular way.

Table 1. Experimental data set

<i>Time Series</i>	<i>Dynamics</i>	<i>Time Series</i>	<i>Dynamics</i>
Sine	periodic	Dow Jones	complex
Vanderpol	periodic	Kobe	complex
Qperiodic2	quasiperiodic	ECG	complex
Qperiodic3	quasiperiodic	EEG	complex
Mackey-Glass	chaotic	ASCII	complex
Logistic	chaotic	El niño	complex
Lorenz	chaotic	HIV DNA	complex
Rosler	chaotic	Human DNA	complex
Ikeda	chaotic	Lovaina	complex
Henon	chaotic	Plasma	complex
Cantor	chaotic	Primes	complex
Tent	chaotic	SP500	complex
A1	complex	Star	complex
D1	complex	Brownian motion	random
Laser	complex	White Noise	random

For illustrative purposes, in Figure 2 we show the same plane, but now showing the waveforms of the time series. Here we can observe how the elements of the principal clusters are related by the intricacy of the time series. In the leftmost cluster there are very intricate, noise-like time series, at the right extreme there are somewhat smooth waveforms and in the middle top are time series that are not as intricate or as regular, i.e. a blend of the first two clusters. From this results we can also speculate that a discrete and qualitative classifications such as "chaotic", "complex" and "random" are not always the best suited as the dynamics of the phenomena can form a continuous space if we use a quantitative approach.

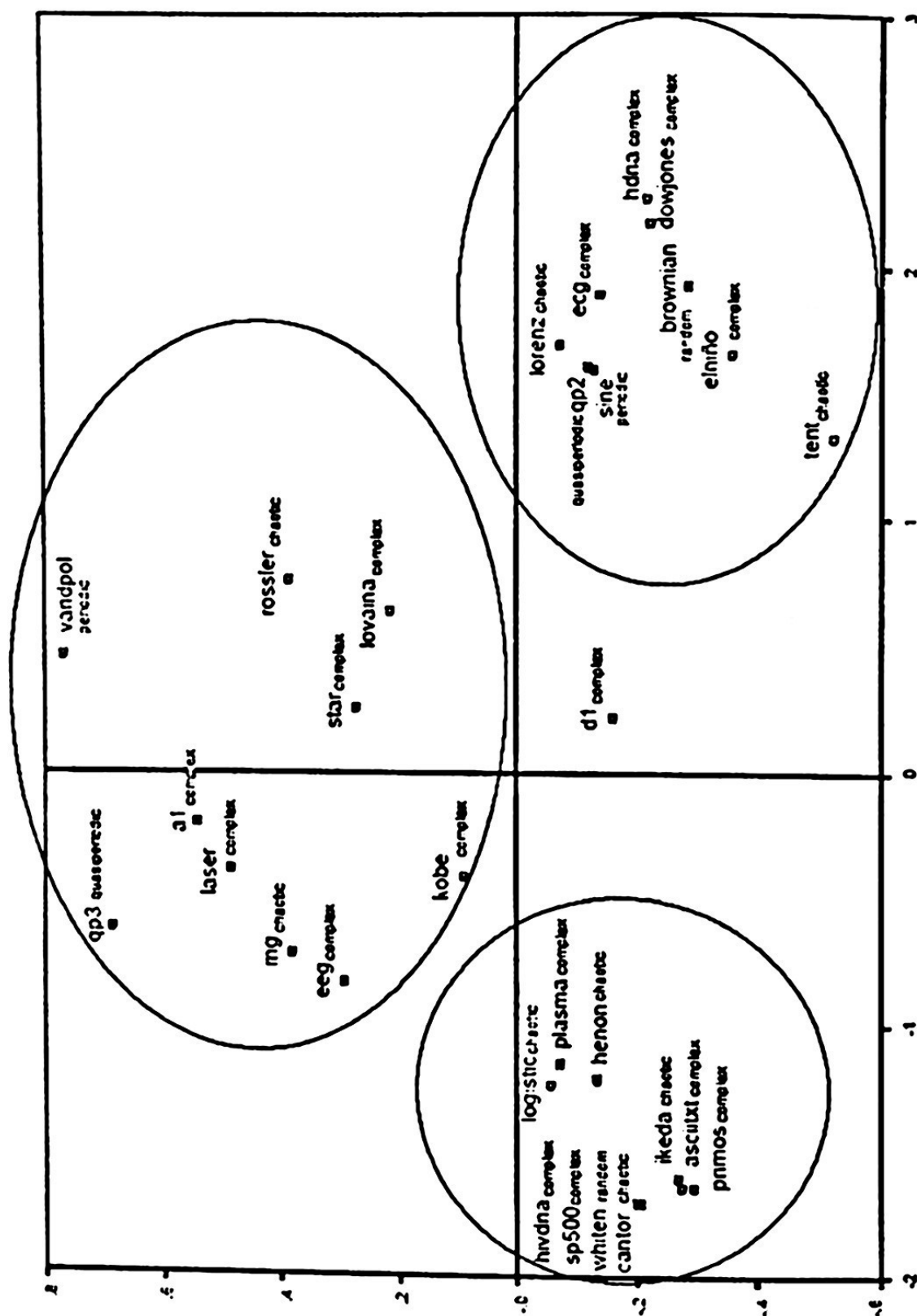


Fig. 1. Clustering of the series in the MDS plane

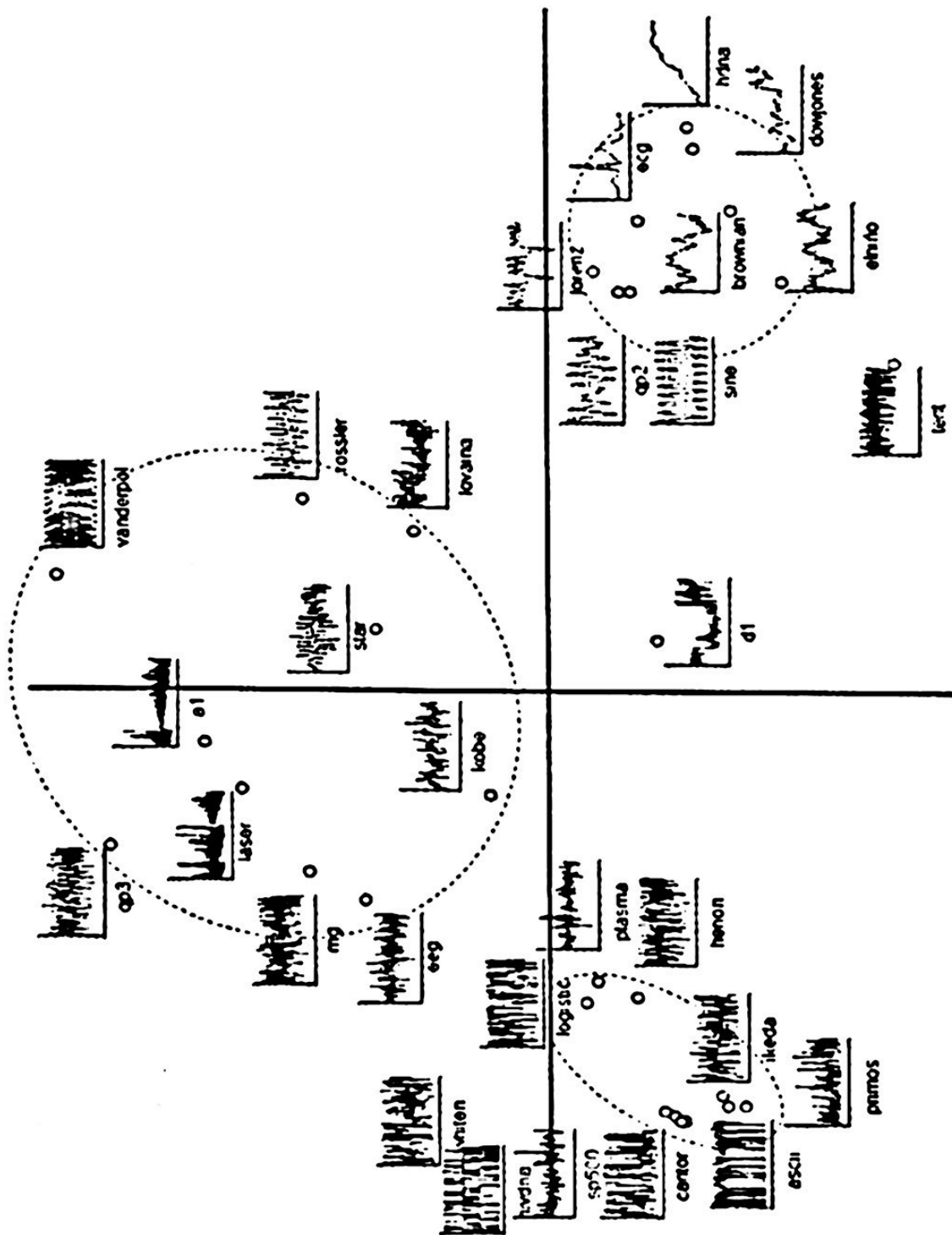


Fig. 2. Waveforms of the experimental data set in the MDS plane

Table 2. Variance percentage of the first five principal components

<i>Time Series</i>	<i>%Var PC1</i>	<i>%Var PC2</i>	<i>%Var PC3</i>	<i>%Var PC4</i>	<i>%Var PC5</i>
Sine	88.158	11.842	0	0	0
Vanderpol	59.752	38.884	0.698	0.609	0.028
Qperiodic2	88.116	11.608	0.247	0.021	0.005
Qperiodic3	33.545	31.809	19.528	13.291	1.156
MackeyGlass	29.777	27.577	16.732	12.074	3.14
Logistic	16.341	15.725	10.029	9.394	6.667
Lorenz	89.332	10.031	0.604	0.031	0.002
Rossler	67.399	31.077	1.113	0.316	0.068
Ikeda	8.037	8.03	7.72	7.589	6.976
Henon	16.398	14.592	10.519	6.5	6.448
Cantor	6.404	6.375	5.757	5.619	5.343
Tent	80.689	2.263	2.23	1.758	1.745
A1	43.903	35.677	6.724	6.425	2.145
D1	65.134	12.674	11.936	4.46	2.148
Laser	38.393	34.912	5.853	5.391	3.606
Dowjones	96.723	2.114	0.431	0.227	0.149
Kobe	23.285	22.457	11.749	10.45	6.366
Ecg	93.729	6.047	0.21	0.012	0.002
Ecg	26.944	24.338	18.28	5.541	3.208
Ascii	7.435	7.297	6.284	6.069	6.047
El niño	87.321	5.593	2.999	1.8	0.908
HIV DNA	6.452	6.332	6.077	5.491	5.297
HumanDNA	99.848	0.094	0.023	0.01	0.006
Lovaina	52.332	38.383	8.265	0.835	0.125
Plasma	18.074	17.677	7.626	6.517	5.984
Primos	7.343	6.482	6.359	6.107	5.526
Sp500	7.407	7.353	6.588	6.528	5.906
Star	56.218	31.681	1.665	1.143	1.07
Brownian m.	94.099	3.936	0.723	0.348	0.247
Whitenoise	6.449	6.178	6.044	5.309	5.296

5 Conclusions

We have analyzed a set of several time series with diverse dynamics by means of two complementary analysis techniques, SSA and PCA and employed the multidimensional scaling for the process of clustering and visualization of the obtained data.

As we can see in the results, the time series did not form clusters based on their dynamic behavior. Instead, the grouping was based in the similarity of basic structures that are common to the series. This means that phenomena with different dynamics can share similar internal patterns although the local interactions of patterns lead to diverse behavior

It is also worth of notice that the three main clusters that were identified are conformed by time series of comparable intricacy, one group was formed from very intricate data another from somewhat smooth waveforms and a third one of a merge of the previous.

Thus, the proposed scheme can be useful to classify data of diverse nature with hidden patterns, disregarding the outward appearance of the time series.

References

1. J. Rodríguez-Elizalde, J. Figueroa-Nazuno, "Quirón: Búsqueda de Similitud en Series de Tiempo por Métodos Espaciales de Acceso", IEEE ROC&C'2003, (2003)
2. J. Friedman, "Regularized Discriminant Analysis" Journal of the American Statistical Association (1989) 165-175
3. Agrawal, R., Lin, K. I., Sawhney, H. S. & Shim, K "Fast similarity search in the presence of noise, scaling, and translation in time-series databases" In proceedings of the 21st Int'l Conference on Very Large Databases (1995). pp 490-50.
4. E. Keogh, and M. Pazzani, "An enhanced representation of time series which allows fast and accurate classification clustering and relevance feedback". In 4th International Conference on Knowledge Discovery and Data Mining, (1998) 239-243.
5. J. Rodríguez-Elizalde, J. Figueroa-Nazuno "Agrupación de Series de Tiempo con Semejanza por Representación Simbólica", IEEE ROC&C'2003, (2003)
6. N Golyandina, V. Nekrutkin, A. Zhigljavsky, "Analysis of Time Series Structure", Chapman & Hall, (2001)
7. D. Hand, H. Mannila, P. Smyth, "Principles of data mining", MIT Press, (2001)
8. J.M. Colebrook, "Continuous plankton records - zooplankton and environment, northeast Atlantic and North-Sea", Oceanol. Acta, 1, (1978) 1948-1975
9. I. Borg, P. Groenen, "Modern Multidimensional Scaling" Springer Verlag, New York, (2001)
10. J. C. Sprott "Chaos and time series analysis" Oxford University Press, (2004)
11. A. Espinosa-Contreras and J. Figueroa-Nazuno, "Análisis del comportamiento de la pérdida de paquetes en la red Internet con técnicas de la dinámica no-lineal", Memorias del Congreso Internacional de Computación CIC2000, (2000), 529-535

7

Evolutionary Algorithm Based on a Markov Graphical Model Selection of Promising Solutions'

Eunice Ponce de León[†], Elva Díaz[†] and Felipe Padilla[†]

[†] Electronic Systems Department, Universidad Autónoma de Aguascalientes,
Ave Universidad #940 Colonia Universidad, CP 20100, Aguascalientes,
Aguascalientes, México.
eponce@correo.uaa.mx
elva.diaz@itesm.mx
fpadilla2000@yahoo.com

Abstract. The goal of this research is to analyze how a new evolutionary algorithm based on Markov graphical model selection of promising solutions finds the optimum for some kind of functions. Probabilistic models have been used for the optimization of deceptive functions, for example, PBIL, MIMIC, BOA, UMDA, BMDA, FDA, EBNA etc. However, all mentioned algorithms are restricted in the complexity of the models used in the search. The algorithm presented in this paper, termed UEMMA (Unrestricted Evolutionary Markov Model Algorithm) was tested with the known deceptive functions. Compared with the results of the others authors the performance of this algorithm in general is not worse and in some cases is better, especially when the functions are more complex.

1 Introduction

Searching for and using a stochastic structure of the space of solutions has been an active line of research within the field of genetic and evolutionary computation. Whereas in genetic algorithms there are crossover and mutation algorithms, in Estimation of Distribution Algorithms (EDA) they have been replaced by learning and sampling of a probability distribution. This new approach for optimization emerged because the genetic algorithm paradigm needed the choice of suitable values for the parameters, and this was converted into a new optimization problem as was shown by Grefenstette [8]. This reason, together with the fact that the prediction of the movement of the population in the search space is extremely difficult, has motivated the birth of a new type of algorithm, EDA. The estimation of the joint distribution associated with the database containing the individuals of the population constitutes the bottleneck of this new heuristic [9]. The previous work began with PBIL (Population Based Incremental Learning) by Baluja [1] and later improved by Baluja and Caruana [2] where a single probability vector of binary independent variables is updated moving to the best vectors of the population, and a memory used as the information from the previous iterations is preserved. In MIMIC, a framework by De Bonnet, Isabell and Viola [4] the dependencies form a chain and the Kullback-Leibler divergence is

[†] Supported by PROMEP/UAAGS-PTC-02-01

used as measure of the fitting of the chain to the population of promising solutions. Baluja and Davies [3] uses optimal dependency trees and the Kullback-Leibler divergence is minimized. One of the most general approaches is the one proposed by Pelikan, Goldberg and Cantu-Paz, known as BOA [13]. This algorithm uses a net where each node can have k successors, allowing variables to be conditionally dependent on sets of variables. The UMDA by Mühlenbein and Pass [11] uses the same type of distribution as does PHIL algorithm, and the main difference is that a memory is used in PHIL and in UMDA the collection of vectors is used instead. The BMDA by Pelikan and Mühlenbein [14] covers second order interactions just as the case of MIMIC, however, the structure used in BMDA is more general than the one in MIMIC. Finally the FDA by Mühlenbein, Mahnig and Rodriguez [12] a factorization of the distribution is given a priori depending of the problem. The estimation of Bayesian Network Algorithm (EBNA) looks for the Bayesian Network whose structure has the maximum posterior likelihood, and whose parameter can be computed directly from the data set [7].

2 Algorithms Description

This paper presents an evolutionary stochastic algorithm that uses as a searching distribution an unrestricted structure in the class of graphic Markov Models (UMMA) as is proposed by Díaz and Ponce de León [5], defined as follows:

- choose an initial population
- determine the fitness of each individual
- perform selection of individuals to be reproduced
- repeat
 - perform crossover
 - perform mutation
 - determine the fitness of each individual
 - perform selection of individuals to be reproduced
- until some stopping criterion applies

The learning model is represented by an hypergraph and is encoded using vectors of binary variables. The graphical model selection algorithm obtains a model that best fit the population of promising solutions using an algorithm for discrete Markov model selection using the convex fitting index (CFI) based in an information measure of divergence, the Kullack-Leibler, and a penalty criteria to accomplish simplicity of

the graph [5]. This index was proposed because of its best performance compared with other two indexes, in a simulated empirical study [Díaz, Ponce de León, 2002]. The population generation uses the Gibbs-Sampler for a Graphic Markov Model defined in Díaz and Ponce de León, 2003 [6].

The UEMMA is defined as follows:

```
Generate a collection of n random vectors

Evaluate the objective function

    if the termination condition is not satisfied    continues

repeat

    order by value

    select  $\tau\%$  from the best vectors

    select and estimate the distribution (UMMA)

    generate a new population with the Gibbs-Sampler

    Evaluate the objective function

until some stopping criterion applies
```

The UEMMA generates a collection of random binary vectors in the first step, second step, evaluate the objective function, third step, if the termination condition is not satisfied continues, fourth step, order the vectors and fifth step, selects the τ percent of the best evaluated vectors. Sixth step, select the structure of the Graphic Markov Model that best accomplish the convex fitting index CFI and estimates the distribution with the UMMA algorithm and seventh step, generates a new population using the Gibbs-Sampler for a Graphic Markov Model [6], eighth step, evaluate the objective function and ninth step, if the termination condition is not satisfied go to the fourth step and repeat.

3 Function Text

3.1 OneMax Problem

This is a well-known simple linear problem used to test the convergence velocity and the scalability. It can be defined as maximizing the function:

$$F_{OneMax}(x) = \sum_{i=1}^n x_i \quad (1)$$

Where x_i is a binary variable for every i . This function has the global optimum at $(1,1,1,\dots,1)$.

3.2 Plateau Problem

This problem was proposed in Mühlenbein and Schlierkamp-Voosen [10]. The individuals of this function consists of a n -dimensional vector, such that $n=mx3$ (the genes are divided into groups of three) The function is defined as:

$$F_{Plateau}(x) = \sum_{i=1}^m g(s_i) \quad (2)$$

Where

$$g(s_i) = g(x_{3i-2}, x_{3i-1}, x_{3i}) = 1 \text{ if } x_{3i-2}=1, x_{3i-1}=1, x_{3i}=1$$

And zero in other case. This function has the global optimum at $(1,1,1,\dots,1)$.

3.3 FC₂ Problem

This function has been proposed in Muhlenbein et al. (1999) [12] and is composed by deceptive decomposable functions as follows:

$$F_{muhl}^s(x) = \begin{cases} 3.0 & \text{for } x = (0,0,0,0,1) \\ 2.0 & \text{for } x = (0,0,0,1,1) \\ 1.0 & \text{for } x = (0,0,1,1,1) \\ 3.5 & \text{for } x = (1,1,1,1,1) \\ 4.0 & \text{for } x = (0,0,0,0,0) \\ 0.0 & \text{otherwise} \end{cases} \quad (3)$$

$$F_{C_2} = \sum_{j=1}^m F_{muhl}^s(s_j) \quad (4)$$

Where $s_j = (x_{5j-4}, x_{5j-3}, x_{5j-2}, x_{5j-1}, x_{5j})$ and $n=5m$.

3.4 FC₄ Problem

This function has been proposed in Muhlenbein et al. (1999) [12] and is composed by deceptive descomposable functions as follows:

$$F^3_{cubani}(x) = \begin{cases} 0.595 & \text{for } x = (0,0,0) \\ 0.200 & \text{for } x = (0,0,1) \\ 0.595 & \text{for } x = (0,1,0) \\ 0.100 & \text{for } x = (0,1,1) \\ 1.000 & \text{for } x = (1,0,0) \\ 0.050 & \text{for } x = (1,0,1) \\ 0.090 & \text{for } x = (1,1,0) \\ 0.150 & \text{for } x = (1,1,1) \end{cases} \quad (5)$$

$$F^5_{cubani}(x) = \begin{cases} 4F^3_{cubani}(x_1, x_2, x_3) & \text{if } x_2 = x_4 \text{ and } x_3 = x_4 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

$$Fc_4(x) = \sum_{j=1}^m F^5_{cubani}(s_j) \quad (7)$$

Where $s_j = (x_{5j-4}, x_{5j-3}, x_{5j-2}, x_{5j-1}, x_{5j})$ and $n=5m$.

3 Experiment, results and discussion

The experiments designed consisted in fixing values of the parameters of the genetic algorithm as follows: for the percent of the best fitted bedividuals in the population t to generate the population $t + 1$ the value is 70, for the probability of selecting individuals at random for mutation the value is .1 , for the weighting coefficient in the CFI fitting index the values is .7. With the combination of factors defined, and 70 percent of the population selected for the new population in the exploration of the searching space, runs of the UEMMA for each type of function were performed as is shown in Table No. 1 .

Table 1. Experimental Results

Problem	OneMax	OneMax	Plateau	Plateau	Fc ₂	Fc ₂	Fc ₄	Fc ₄
Variables	10	10	9	9	10	10	10	10
Mean number of Generations	9	4	5	2	24	18	12	9
Time secs	60.2	47.9	85	45	193.5	116.85	84.05	82.36
Size of Population	100	300	100	300	100	300	100	300
% of Selection	70	70	70	70	70	70	70	70

The preliminary experience with this algorithm shows, as first sight that growing the size of the population diminishes the mean number of generations to obtain the optimum. The optimum was obtained in all functions. The One Max function is, of course, the easiest to optimize, and the Fc₂ is the most difficult. The time of performance is proportional to the number of generations.

4 Conclusions and recommendations

As conclusion, the performance of the algorithm UEMMA in these functions is like as reported in the literature [12]. In general the algorithm has a good performance but more experiments must be done with other functions and problems of different sizes. A comparative study with other algorithms could give new ideas, especially about the treatment of bigger problems.

As a recommendation it will be useful to design a bigger experiment calibrating together the parameters of the model selection algorithm and of the optimization algorithm in order to obtain new ideas of the relation between these two sets of parameters.

References

1. Baluja, S.: Population Based Incremental Learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, (1994).
2. Baluja, S. and Caruana, R.: Removing the genetics from the standard genetic algorithm. Technical Report, Carnegie Mellon University, (1995).
3. Baluja, S. and Davies, S.: Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In D. H. Fisher, editor, Proceedings of the 1997 International Conference on Machine Learning. Morgan Kaufman publishers (1997). Also available as Technical Report CMU-CS-97-

4. De Bonet, J. S. , Isbell, C., and Viola, P.: Mimic: Finding optima by estimating probabilities densities. *Advances in Neural Information Processing*, 9, (1996).
5. Díaz, E., Ponce de Leon, E : Discrete Graphic Markov Models by a Genetic Algorithm. In *Avances en Ciencias de la Computación e Ingeniería de Computo*. Editores: Sossa Azuela, J. H., et al. Ciudad de México (2002) 315-324.
6. Díaz, E., Ponce de León, E.: Generación aleatoria de muestras a partir de un Modelo Gráfico Markoviano. *Memorias del X Simposio de Investigación y Desarrollo Tecnológico*, Aguascalientes, (2003) 77.
7. Etxeberria, R. and Larrañaga, P. : Global optimization with Bayesian networks. In *II Symposium on Artificial Intelligence. CIMA 99. Special Session on Distributions and Evolutionary Optimization* (1999) 332-339.
8. Grefenstette, J. J. : Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1) (1998) 122-128.
9. Larrañaga, P. A Review on Estimation of Distribution Algorithms. :In *Estimation of Distributions Algorithms*. Ed. Pedro Larrañaga and José Lozano. Kluwer Academic Publishers, (2002).
10. Mühlenbein, H. and Schierkamp-Voosen, D. : The science of breeding and its application to the breeder genetic algorithm (BGA). *Evolutionary Computation*, 1 (1993) 335-360.
11. Mühlenbein, H. and Pass, G.: From recombination of genes to the estimation of distributions 1. binary parameters. In A. E. Eiben, T. Bäck, M. Shoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature—PPSN V*, Springer, (1998) 178-189.
12. Mühlenbein, H., Mahnig, T. and Ochoa-Rodriguez, A.: Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5, (1999) 215-247.
13. Pelikan, M., Goldberg, D. E. and Cantu-Paz E.: Boa The Bayesian optimization algorithm. In W. Banzhaf, et al., editors, *Proceedings of the GECCO-1999 Genetic Evolutionary Computation Conference*. Morgan Kaufman Publishers, (1999) 525-532.
14. Pelikan, M. and Mühlenbein, H.: The bivariate marginal distribution algorithm. *Advances in Soft Computing-Engineering Design and Manufacturing* (1999) 521-535.



On Numerical Simulation of Linear Acoustic and Electromagnetic Waves in Unbounded Domains

Denis Filatov

Centre for Computing Research (CIC), National Polytechnic Institute (IPN),
C.P. 07738, Mexico City, Mexico.
denisfilatov@mail.ru

Abstract. We study the problem of constructing local absorbing boundary conditions (ABCs) for the numerical simulation of n -dimensional linear acoustic and electromagnetic waves in unbounded domains. Employing the technique of dimensional splitting, we reduce the construction of the ABCs for the original n -D equation to seeking for an ABC for the one-dimensional wave problem. Then, applying the Laplace transform in time as well as using spline interpolation for the initial data, we obtain an infinite family of functions that approximate the far-field solution with higher and higher accuracy. These functions are used as ABCs. Due to the property of compactness of supports of the splines, the boundary conditions appear to be local both in time and in space, which is extremely important for their numerical implementation. Yet, these ABCs are uncritical to the shape of the artificial boundary and therefore usable for simulating a vast number of practical wave problems in domains of drastically complex geometries. The resulting boundary value problems are well-posed in the sense of existence, uniqueness and stability of solution. Results of the numerical experiments confirm the theoretical study.

1 Introduction

Numerical solution to differential problems in unbounded domains requires introducing a finite computational domain enclosed by an artificial boundary. Indeed, all the matrices and vectors employed to compute the solution must be of finite sizes, and therefore the necessity in truncation of the original domain is dictated by the evident limitations on the computer memory. Hence, it is required to impose adequate boundary conditions for simulating the solution at the boundary points. The adequacy of the boundary conditions implies that 1) the resulting boundary value problem (BVP) must be well-posed in the sense of existence, uniqueness, and stability of solution, and 2) the error between the solutions to the original Cauchy problem and the resulting BVP should be as small as possible in the domain of interest. Boundary conditions satisfying the aforesaid requirements are often called artificial boundary conditions (ABCs) [17].

A special class of ABCs is known as one-way, or absorbing, or non-reflecting boundary conditions (NRBCs) [3, 16]. These boundary conditions appear when studying various wave phenomena in such fields of scientific computing as geophysics [1, 4, 8], theory of elasticity [5, 10, 13], computational fluid dynamics [3, 4, 15], and

others. NRBCs are derived in such a manner that permit waves propagating outward the computational domain only, while no propagation towards the interior is allowed.

For the last three decades there have been developed a whole series of different methods for the construction of NRBCs [2, 4, 5, 8, 10, 11, 13–15] (see also [17]). However, all of them lead to boundary conditions that suffer from some or other disadvantages. For example, many methods provide BCs that can be used with a planar (i.e., rectangular) artificial boundary only; others lead to NRBCs that are non-local either in time or in space, or even both, and hence these BCs are unrealisable from the computational standpoint; third group of methods is oriented to solving a particular class of equations, and therefore fails when applying to problems of other types.

In this paper we present an advanced methodology for the construction of non-reflecting boundary conditions. The key idea consists in the employment of the techniques of operator factorisation and dimensional splitting [12]. Unlike all other NRBCs, ours are geometrically universal and besides appear local both in time and in space. These properties allow to use the approach when numerically solving a wide spectrum of practical wave problems in domains of extremely complex geometries [7]. In addition, the constructed boundary conditions are more accurate than many others, e.g., those derived in [4].

2 Problem Formulation

Consider the n -dimensional wave equation

$$Au \equiv \frac{\partial^2 u}{\partial t^2} - a^2 \Delta u = f, \quad (x, t) \in \mathbb{R}^n \times (0, +\infty) \quad (1)$$

subject to the initial condition

$$u|_{t=0} \equiv g(x), \quad \frac{\partial u}{\partial t}|_{t=0} \equiv 0. \quad (2)$$

Here $u = u(x, t)$ is the function to be sought, $a = a(x, t) \geq 0$ is the wave velocity, Δ is the Laplacian in x , and $f = f(x, t)$ denotes the sources. Let the computational domain be an open region $\Omega \subset \mathbb{R}^n$ with a piecewise smooth boundary Γ . We assume $a = \text{const}$ outside Ω , as well as $\text{supp } f \subseteq \Omega$. In order to solve problem (1)–(2) numerically, it is required to construct a differential operator B for the boundary condition $Bu|_{\Gamma} = 0$; upon this, the operator B must be derived in such a manner that waves propagating from Ω leave the domain without reflections to the inside, that is the boundary condition must be *non-reflecting*.

3 Construction of the ABCs

We shall construct the boundary operator B in two steps. At the first step we employ the method of dimensional splitting [12]: we choose some i from the range $[1, \dots, n]$, freeze the coordinates $\{x_j\}_{j=1, j \neq i}^Y$, and consider equation (1) on Γ with respect to x_i only, i.e.

$$A_i u \equiv \frac{\partial^2 u}{\partial t^2} - a^2 \frac{\partial^2 u}{\partial x_i^2} = 0, \quad (x, t) \in \Gamma \times (0, +\infty), \quad \{x_j\}_{j=1, j \neq i}^Y \text{ are fixed.} \quad (3)$$

At the second step, in order to satisfy the requirement of no reflection of the boundary condition, we factorise the operator A_i onto the incoming and outgoing components— $A_i = A_i^+ A_i^-$ —and rewrite equation (3) the form

$$A_i^+ A_i^- u = \left(\frac{\partial}{\partial t} + a \frac{\partial}{\partial x_i} \right) \left(\frac{\partial}{\partial t} - a \frac{\partial}{\partial x_i} \right) u = 0. \quad (4)$$

It is important to emphasise that we decoupled the incoming and outgoing waves in the *original, physical* space \mathbb{R}^n , without involving any additional techniques like Fourier transform [4] or others. Therefore, we avoided imposing undesired restrictions on the shape of the boundary Γ , and hence, on the applicability of the subsequent NRBCs too.

The derivation of the boundary operator is now obvious: if the incoming wave at the point x_i (all the other x_j 's are fixed) is represented by the positive component A_i^+ , then we prohibit it defining $A_i^+ = E$ (here E is the identity operator) and obtain

$$B_i u \stackrel{\text{NR}}{=} A_i^+ A_i^- u = E A_i^- u = \left(\frac{\partial}{\partial t} - a \frac{\partial}{\partial x_i} \right) u = 0; \quad (5)$$

otherwise, if the incoming wave corresponds to A_i^- , it holds $B_i = \partial_t + a \partial_{x_i}$.

In fact, the construction of the one-dimensional NRBC is complete, and now we can derive the general n -D absorbing boundary condition. However, before doing this let us construct a family of 1D non-reflecting boundary conditions basing on the original BC (5). This family will be derived in such a way that its members will be approximate solutions to equation (5) specially adapted to finite difference implementation. So, involving the technique used in [6], we apply to (5) the Laplace transform in time and then approximate the initial condition g from (2) by a family of splines. Next, assuming the image of u to be bounded in x_i in the dual space, we apply the inverse Laplace transform and thus come to an infinite family of approximate solutions to equation (5) in the physical space [7]:

$$u^{[k+1]} = u^{[k]} + \sum_{p=1}^m a^p \frac{\tau_k^p}{p!} \frac{\partial^p u^{[k]}}{\partial x_i^p}, \quad \tau_k = t_{k+1} - t_k, \quad u^{[0]} = g, \quad k \in \{0\} \cup \mathbb{N}. \quad (6)$$

Here $m \geq 1$ is the spline order. Consequently, the general n -dimensional absorbing boundary condition has the form

$$u^{[k+1]}(x) = u^{[k]}(x) + \sum_{p=1}^m a^p \frac{\tau_k^p}{p!} \sum_{i=1}^n s_i^p \frac{\partial^p u^{[k]}(x)}{\partial x_i^p}, \quad x \in \Gamma^{[0]}, \quad (7)$$

where $s_i = s_i(x)$ is the sign function having the value $+1$ if the incoming wave at the point $x \in \Gamma$ in the direction x_i is given by the operator A_i^- , and -1 otherwise (see formula (5)). For the errors produced by solutions (7) on the boundary we have the estimates

$$|\varepsilon_m| \leq a \left(\xi_1 \frac{(\sigma_k)^m}{m!} \sum_{i=1}^n s_i^{m+1} + \xi_2 \sum_{p=1}^m \frac{(\sigma_k)^p}{p!} \sum_{i=1}^n \sum_{j=1}^n s_i^p s_j^p \right), \quad (8)$$

where $\xi_1 = \max_i \left| \frac{\tau_k^{m+1} u^{[k]}(x)}{\tau_k^{m+1}} \right|$ and $\xi_2 = \max_{p,j} \left| \frac{\tau_k^{p+1} u^{[k]}(x)}{\tau_k^{p+1}} \right|$. The following proposition holds.

Proposition 1. *For every $m \geq 1$ the corresponding wave boundary value problem is well-posed in the sense of existence, uniqueness, and stability of solution.*

We shall omit a formal proof of this statement, and only mention that it can be done by means of functional analysis and the theory of generalised functions. For further studies on well-posedness of the resulting BVPs the interested reader may refer to [7].

Let us make two important remarks. First of them concerns the use of splines when approximating the initial data g in the Laplace-transformed equation (5). Specifically, it is essential that we employed *compactly* supported basis functions rather than some *infinitely* supported ones [15]. Due to this we again avoided restrictions on the shape of the boundary Γ , and so derived a family of *geometrically universal* NRBCs; in addition, these BCs are local and adapted to numerical implementation in time.

Another remark relates to the factorisation of the operator A_i in (4) and the subsequent prohibition of the incoming wave at the boundary point. Namely, it can easily be observed that having defined $A_i^- = E$ (or $A_i^+ = E$) we, properly speaking, disregarded the *second* order of the temporal derivative of the wave equation and thus reduced it from two to one. Therefore, the resulting boundary conditions (7) serve only in case of *zero* initial condition $\partial_t u|_{t=0} = h(x)$, as defined in (2). However, to generalise these NRBCs one may perform a preliminary change of variable (with respect to the function u) and to transfer $h \neq 0$ to the right-hand side of equation (1). This will allow to subsequently take into account the presence of the non-zero initial data and hence to construct a more general version of the operator B [7].

4 Numerical Results

We tested the developed method having performed several experiments on functionality and efficiency of the constructed NRBCs. For clearness of presentation of the results we considered the case $n = 2$.

First of all we compared our boundary conditions with those derived in the classical paper by Engquist and Majda [4]. For this we solved the original Cauchy problem (1)-(2) and a few boundary value problems with various boundary conditions. Specifically, we tested the lowest-order BC (7) corresponding to $m = 1$ and the first two NRBCs from [4] analysed by the authors in detail. The artificial boundary was supposed to be the straight line $\Gamma = \{x_2 = 0\}$, the domain of interest was $\Omega = (-6, 6) \times (0, 6)$, and we chose $g(x) = \sin^2 \pi(x_1 + 0.5) \sin^2 \pi(x_2 - 0.5)$, $\text{supp } g \subset [-0.5, 0.5] \times [0.5, 1.5]$, $a = 1$, $\Delta x_1 = \Delta x_2 = 5\tau = 0.05$. To discretise the equation in Ω we used the standard second-order "wave" finite difference scheme [9, p. 228], while on the lateral boundary we imposed the zero Neumann boundary condition. At each time moment on Γ we computed maximum of the reflected wave related to maximum of the incident wave.

In Fig. 1 there are two wave profiles corresponding to the solutions to the infinite-domain problem and the BVP with BC (7). It can be seen that the second profile is similar to the first one, with a small reflection as well. Expressed numerically, Table 1 summarises L_2 -norms of the relative error for different angles of wave incidence. One may observe that there are considerable benefits in the precision of solution, especially for large values of θ , under substantial geometrical flexibility of our NRBCs.

Table 1. Planar artificial boundary: relative error (in %) for different angles of wave incidence

Angle (θ)	BC (7)	1 st E-M	2 nd E-M
0°	2.55	6.06	2.10
10°	2.50	6.07	2.18
20°	2.56	6.33	2.40
30°	3.55	7.91	3.14
40°	5.65	10.76	4.22
50°	6.50	15.40	6.11
60°	7.28	21.99	9.54
70°	7.53	30.36	16.14

To try the method on complex geometries, we considered the domain Ω shown in Fig. 2 (top) and repeated the experiments computing the relative error between amplitudes of the reflected and incident waves. As in the case of planar artificial boundary, there was a weak reflection from Γ , and the error did not exceed 2.5% for $\theta = 0^\circ$ and 7.9% for $\theta = 70^\circ$. At the bottom of Fig. 2 we show the numerical solution at $T = 0.33$ as well.

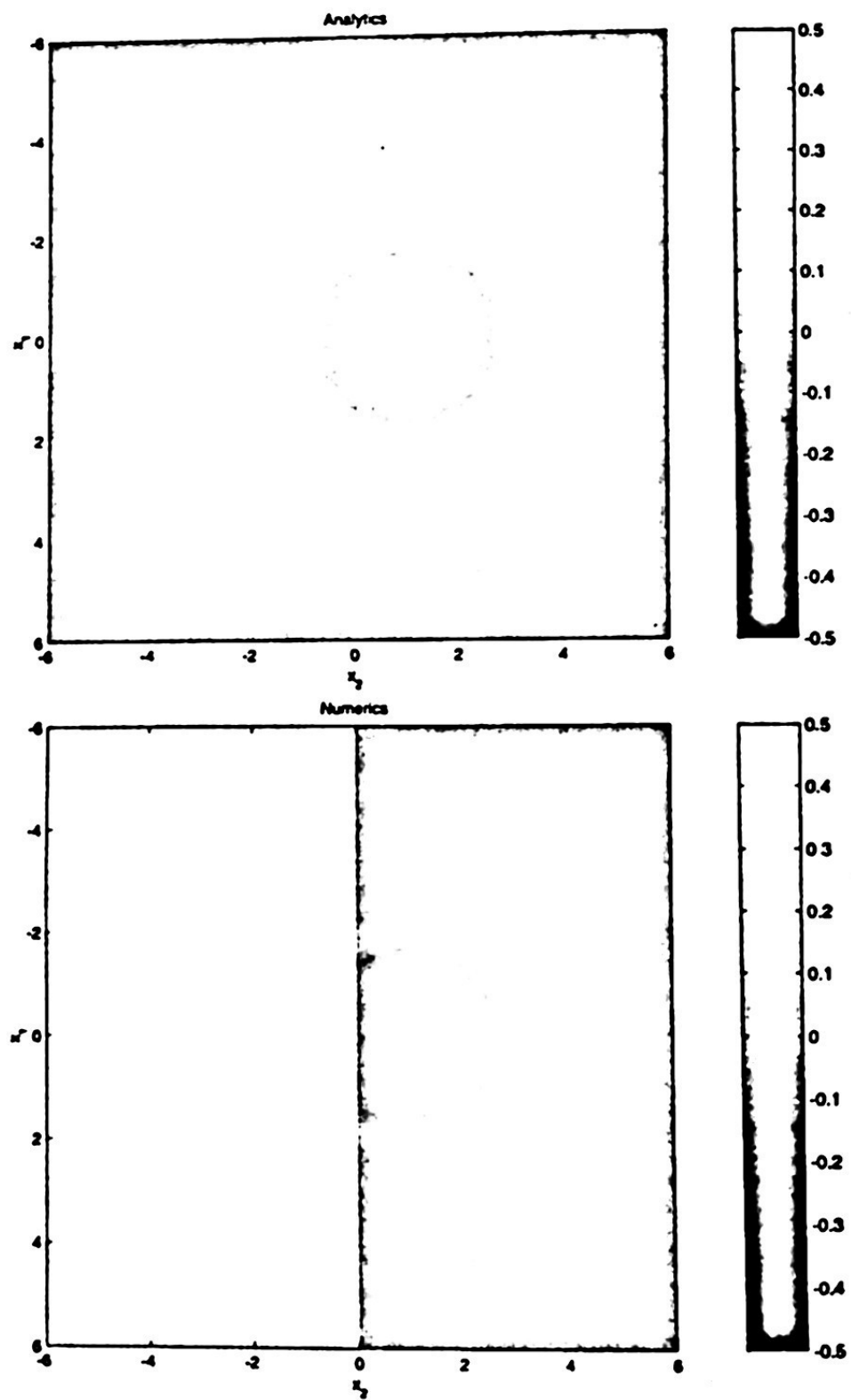


Fig. 1. Planar artificial boundary: solutions to the original Cauchy problem (top) and the resulting BVP (bottom) at $T = 2.0$

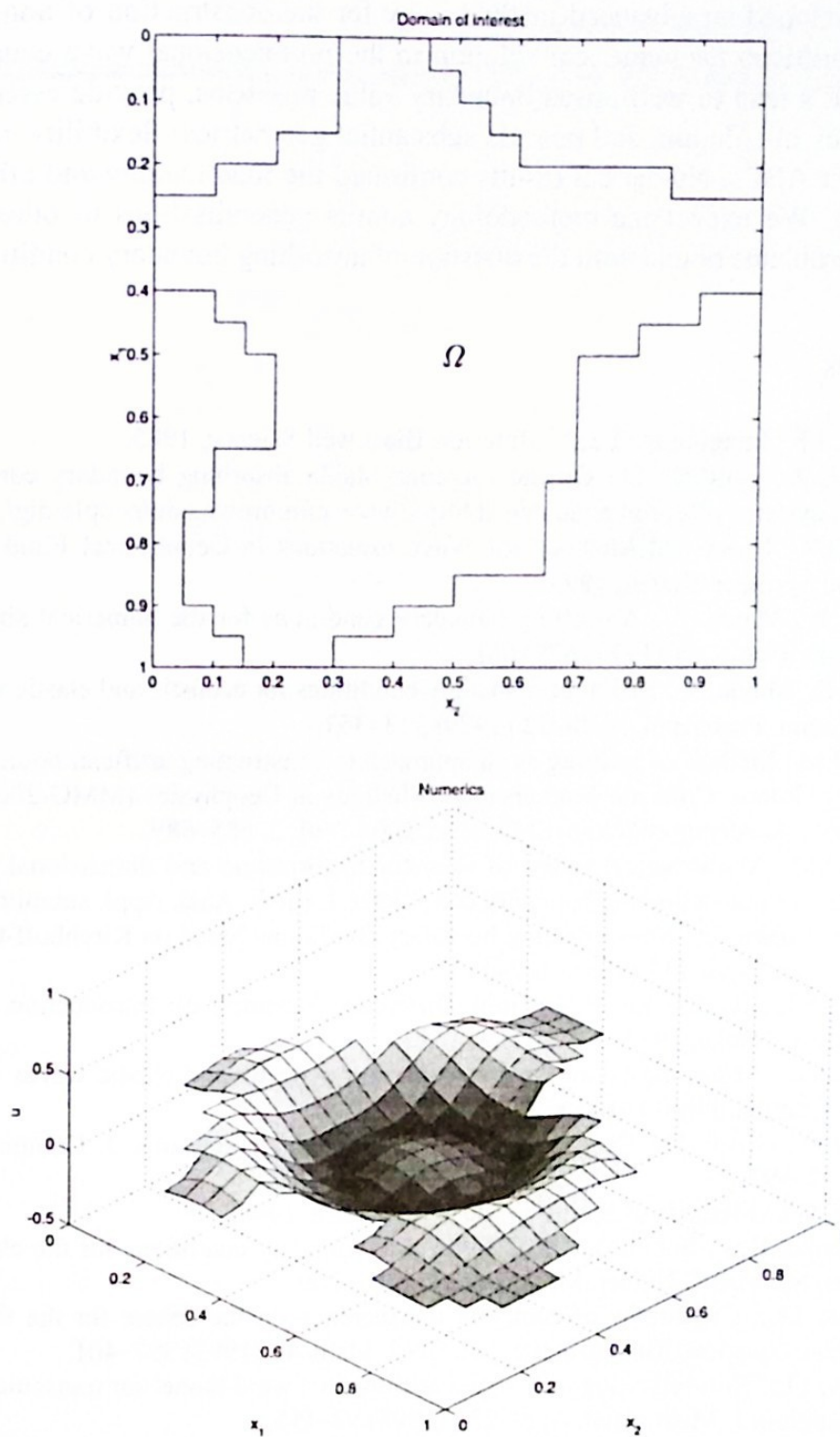


Fig. 2. Experiments with complex geometries: the domain of interest (top) and the BVP solution at $T = 0.33$ (bottom)

5 Conclusions

We have developed an advanced methodology for the construction of non-reflecting boundary conditions for numerical solution to the n -dimensional wave equation. The derived NRBCs lead to well-posed boundary value problems, provide essential gains in the accuracy of solution, and possess substantial geometrical flexibility in comparison with other ABCs. Numerical results confirmed the functionality and efficiency of the approach. We expect the methodology admits generalisations to other types of differential problems bound with the question of absorbing boundary conditions.

References

1. Claerbout, J.F.: *Imaging the Earth's Interior*. Blackwell Science, 1985.
2. Ditkowski, A., Gottlieb, D.: On the Engquist-Majda absorbing boundary conditions for hyperbolic systems. Preprint available at <http://www.cfm.brown.edu/people/dig/>.
3. Durrant, D.R.: *Numerical Methods for Wave Equations in Geophysical Fluid Dynamics*. New York: Springer-Verlag, 1999.
4. Engquist, B., Majda, A.: Absorbing boundary conditions for the numerical simulation of waves. *Math. Comp.* **31** (1977) 629–651.
5. Engquist, B., Majda, A.: Radiation boundary conditions for acoustic and elastic wave calculations. *Comm. Pure Appl. Math.* **32** (1979) 313–357.
6. Filatov, D.M.: Method of splitting as an approach to constructing artificial boundary conditions. *Proc. Intern. Conf. on Mathematical Methods in Geophysics (MMG-2003)*, Russia, Novosibirsk, Akademgorodok, 8–12 October 2003. Vol. 2, 685–689.
7. Filatov, D.M.: Mathematical issues of operator factorisation and dimensional splitting in the problem of non-reflecting boundary conditions. *J. Math. Anal. Appl.* submitted.
8. Givoli, D., Cohen, D.: Non-reflecting boundary conditions based on Kirchhoff-type formulae. *J. Comput. Phys.* **117** (1995) 102–113.
9. Godunov, S.K., Ryaben'kii, V.S.: *Finite Difference Schemes. An Introduction to the Theory*. Moscow: Nauka, 1977.
10. Higdon, R.L.: Absorbing boundary conditions for acoustic and elastic waves in stratified media. *J. Comput. Phys.* **101** (1992) 386–418.
11. Keller, J.B., Givoli, D.: Exact non-reflecting boundary conditions. *J. Comput. Phys.* **82** (1989) 172–192.
12. Marchuk, G.I.: *Methods of Splitting*. Moscow: Nauka, 1988.
13. Mikhailenko, B.G., Soboleva, O.N.: Absorbing boundary conditions for the elastic theory equations. *Siberian J. Numer. Math.* **1** (1998) 261–269.
14. Sofronov, I.L.: Conditions of complete transparency on the sphere for the three-dimensional wave equation. *Russian Acad. Sci. Dokl. Math.* **46** (1993) 397–401.
15. Sofronov, I.L.: Non-reflecting inflow and outflow in a wind tunnel for transonic time-accurate simulation. *J. Math. Anal. Appl.* **221** (1998) 92–115.
16. Trefethen, L.N., Halpern, L.: Well-posedness of one-way wave equations and absorbing boundary conditions. *Math. Comp.* **47** (1986) 421–435.
17. Tsynkov, S.V.: Numerical solution of problems on unbounded domains. A review. *Appl. Numer. Math.* **27** (1998) 465–532.

Digital Systems Design

1. Introduction

In the last decade, the digital systems design has become a very important part of the computer system. The digital systems design is a process of creating a digital system that can perform a specific task. The digital systems design is a process of creating a digital system that can perform a specific task. The digital systems design is a process of creating a digital system that can perform a specific task. The digital systems design is a process of creating a digital system that can perform a specific task.

Adaptive Multichannel Non Parametric Rank M-Type Filter to Remove Impulsive Noise from Color Images

Francisco Gallegos-Funes¹, Volodymyr Ponomaryov², Alberto Rosales-Silva²

¹ National Polytechnic Institute of Mexico, SEPI ESIME-Zacatenco,
Av. IPN s/n, Col. Lindavista, 07738, Mexico D.F., México,
phone/fax (5255)57296000x54622, email: fgallegosf@ipn.mx

² National Polytechnic Institute of Mexico, SEPI ESIME-Culhuacan,
Av. Santa Ana 1000, Col. San Fco. Culhuacan, 04430, Mexico D.F.

Abstract. In this paper, we propose a new class of multichannel filters, the vector median M-type K-nearest neighbor (VMMKNNF) filter to achieve three objectives: noise suppression, chromaticity retention, and edge and detail preservation. The design of VMMKNNF filter is mainly based on the combined RM-estimator with different influence functions. Also, we improve the performance of the proposed filter by using an adaptive non parametric approach in its filtering scheme. The new filter is called adaptive multichannel non parametric VMMKNNF (AMN-VMMKNNF) filter. Simulation results illustrate that the VMMKNNF and AMN-VMMKNNF filters can successfully satisfy to these three objectives, and also demonstrate that their performance surpasses other existing filtering techniques. Finally, we also present the real-time implementation of proposed filters on the DSP TMS320C6711 demonstrating that they potentially provide a real-time solution to quality video transmission.

1 Introduction

In the last decade, many useful techniques of multichannel signal processing based on vector processing were investigated due to the inherent correlation existing between the image channels compared against the traditional component-wise approaches [1, 2]. The most common processing problems are the noise filtering and image enhancement, since these ones are essential functions of any image processing system, regardless of whether the processed image is utilized for visual interpretation or automatic analysis.

Different filtering techniques have been proposed for color imaging. Particularly, nonlinear filters applied to color images have been designed to preserve edges and fine details, and remove an impulsive noise [3-5]. A number of different vector processing

filters based on order statistics (OS) have been introduced recently in color image analysis and restoration [1-5].

In this paper, we introduce the Vector Median M-Type K-Nearest Neighbor filter (VMMKNNF). This filter provides fine detail preservation employing the KNN algorithm [5] and the combined RM-estimator [6-7] to obtain the sufficient impulsive noise suppression for each color channel. The combined RM-estimator is described as redescending M -estimator with different influence functions [5, 8] combined with the median R -estimator [5, 8] to provide better noise suppression. To improve the restoration performance of VMMKNNF we use an adaptive non parametric approach determining the functional form of density probability of noise from data into the sliding filtering window [4]. This filter is called adaptive multichannel non parametric VMMKNNF (AMN-VMMKNNF). Simulation results have demonstrated that the proposed filters consistently outperform other color image filters by balancing the tradeoff between noise suppression and detail preservation. The implementation of the filters was realized on the Texas Instruments DSP TMS320C6711 [9] to demonstrate that they can provide a real-time solution to quality video transmission.

2 RM-Estimators

The R -estimators form a class of nonparametric robust estimators based on rank calculations [5, 8]. The median estimator is the best estimator when any *a priori* information about data Y_j distribution shape and its moments is unavailable [5, 8]:

$$\hat{\sigma}_{\text{med}} = \begin{cases} \frac{1}{2} (Y_{(n/2)} + Y_{(n/2+1)}) & \text{for even } n \\ Y_{((n+1)/2)} & \text{for odd } n \end{cases} \quad (1)$$

where $Y_{(j)}$ is the element with rank j in the sample of size n .

Huber proposed the M -estimators as a generalization of maximum likelihood estimators (MLE) [5, 8, 10]. The standard technique for M -estimate calculation consists of using of Newton's iterative method introducing the influence function $\psi(X, \theta) = \frac{\partial}{\partial \theta} \rho(X, \theta)$ and the

function $w(u) = \begin{cases} \psi(u)/u, & u \neq 0 \\ c, & u = 0 \end{cases}$ as [5, 8, 10]:

$$\hat{\theta}^{(q)} = \frac{\sum_{i=1}^N w_i [Y_i - \hat{\theta}^{(q-1)}] / S_0}{\sum_{i=1}^N w_i [Y_i - \hat{\theta}^{(q-1)}] / S_0} \equiv \frac{\sum_{i=1}^N Y_i \psi(Y_i - \text{MED}\{Y_N\})}{\sum_{i=1}^N 1_{[-r, r]} \psi'(Y_i - \text{MED}\{Y_N\})} \quad (2)$$

where $\hat{\theta}^{(q)}$ is the M -estimate of the sample location parameter θ on a step q and S_0 is a scale estimate; Y_i is a data sample, ψ is the normalized influence function $\psi : \psi(Y) = Y\varphi(Y)$, and Y_N is the primary data sample. Usually $\hat{\theta}^{(0)} = \text{MED}\{Y_N\}$ is the median of primary data and $S_0 = \text{MED}\{Y_i - \hat{\theta}^{(0)}\} = \text{MAD}(Y_N)$ is the median of the absolute deviations from the median. It is evident that eq. (2) represents the arithmetic average of $\sum_{i=1}^N \psi(Y_i - \text{MED}\{Y_N\})$, which is evaluated on the interval $[-r, r]$ [7].

Another way to derive the function $\psi(Y)$ is to cut the outliers off the primary sample. This leads to the so-called lowered M -estimates. Hampel proved in [8] that the skipped median is the most robust lowered M -estimate. Below we also use the simple cut (skipped mean) influence function [5-8, 11].

To enhance the robust properties of M -estimators by using the rank estimate consists of the procedure similar to the median average instead of arithmetic one. The proposed estimator is called combined RM-estimator (Median M-type estimator) [7]:

$$\hat{\theta}^{(q)}_{\text{MM}} = \text{MED} \left\{ Y_i \psi(Y_i - \hat{\theta}^{(q-1)}) \mid i = 1, \dots, N \right\} \quad (3)$$

where Y_i is a data sample; ψ is the normalized influence function $\psi : \psi(Y) = Y\varphi(Y)$; initial estimate is $\hat{\theta}^{(0)} = \text{MED}\{Y_N\}$; and Y_N is the primary data sample.

3 Proposed Multichannel RM-Filters

To increase the quality of filtration via the preservation of the edges and small-size details in the image consists of the use of K elements of the sample whose values are closest to the central pixel value of a sliding filter window. This leads to the widely known KNN (K -nearest neighbor pixels) image-filtering algorithm [5].

The proposed VMMKNNF employs the KNN algorithm [5]. The Vector KNN filter (VKNNF) is written below as [1]:

$$\theta_{\text{KNN}} = \frac{1}{K_c} \sum_{m=1}^N \psi(y_m) y_m \quad (4)$$

where y_m are the noisy image vectors in sliding filter window, which includes $m = 1, \dots, N$ (N is odd) vectors y_1, y_2, \dots, y_N located at spatial coordinates in the filter window, and $\psi(y_m)$ is the influence function that is defined as

$$\psi(y_m) = \begin{cases} 1, & \text{if } y_m \text{ are } K_c \text{ samples whose values are closest} \\ & \text{to the value of the central sample } y_{(N+1)/2} \\ 0, & \text{otherwise} \end{cases} \quad \text{For improving the robustness of}$$

the VKNNF we proposed to use the iterative RM-estimator (3). So, the Vector Median M-type K-Nearest Neighbor filter (VMMKNNF) can be written as:

$$\theta_{\text{VMMKNN}}^{(q)} = \text{MED} \{g^{(q)}\} \quad (5)$$

where $g^{(q)}$ is a set of K_c values of vectors y_m which are weighted by value in accordance with the used influence function $\tilde{\psi}(y_m)$ to the estimate obtained at previous step $\theta_{\text{VMMKNN}}^{(q-1)}$ in a sliding filter window; y_m are the noisy image vectors in sliding filter window, which include vectors y_1, y_2, \dots, y_N located at spatial coordinates (i, j) in the filter window; $\theta_{\text{VMMKNN}}^{(0)} = y_{(N+1)/2}$ is the initial estimate; q is the index of the current iteration; K_c is the number of the nearest neighbor vectors calculated in such a form [7, 12]:

$$K_c = \left[K_{\min} + a \cdot D_i(y_{(N+1)/2}) \right] \leq K_{\max} \quad (6)$$

where a controls the detail preservation; K_{\min} is the minimal number of the neighbors for noise removal; K_{\max} is the maximal number of the neighbors for edge restriction and detail smoothing; and $D_i(y_{(N+1)/2})$ is the impulsive detector [7, 12, 13]:

$$D_i(y_{(N+1)/2}) = \left[\frac{\text{MED} \{ |y_{(N+1)/2} - y_m| \}}{\text{MAD}} \right] + \left[\frac{1}{2} \cdot \frac{\text{MAD}}{\text{MED} \{ y_m \}} \right] \quad (7)$$

where $\text{MED} \{ y_m \}$ is the median of data set in sliding window, and MAD is the median of absolute deviations from median in the same window [5, 7].

Numerical simulations show that, when $K_c > 7$ and $K_c > 350$, the VMMKNNF filter may be replaced with 3x3 median filter and 5x5 median filter, respectively. The algorithm finishes when $\theta_{\text{VMMKNN}}^{(q)} = \theta_{\text{VMMKNN}}^{(q-1)}$. From simulations we found that the iterations con-

verge after one or two iterations.

To improve the impulsive noise suppression and detail preservation performances of VMMKNNF we have introduced the AMN-VMMKNNF that is based on adaptive non parametric approach and determines the functional form of density probability of noise from data into the sliding filtering window [4]. So, AMN-VMMKNNF is presented by combining the adaptive multichannel non parametric and the VMMKNNF according with the reference [4].

The proposed AMN-VMMKNNF can be written as:

$$\hat{x}(y)_{\text{AMN-VMMKNNF}} = \sum_{i=1}^n x_i^{\text{VMMKNNF}} \left(\frac{h_i^{-M} K((y - y_i) h_i)}{\sum_{i=1}^n h_i^{-M} K((y - y_i) h_i)} \right) \quad (8)$$

where x_i^{VMMKNNF} values represent the VMMKNNF providing the reference vector according with [4], y is the current noisy observation to be estimated from given set y_N , y_i are the noisy vector measurements, h_i is the smooth parameter:

$$h_i = n^{-p/M} \left(\sum_{j=1}^n \|y_j - y_i\|_{L_1} \right) \quad (9)$$

where $y_j \neq y_i$ for $\forall y_j, j=1,2,\dots,N$, $\|y_j - y_i\|_{L_1}$ is the absolute distance (L_1 metric) between two vectors, p is a parameter to be determined between $0.5 > p > 0$, M is the dimensionality of the measurement space ($M=3$ for color images), and the function $K(y) = \exp(-|y|)$ is the kernel function in the case of impulsive noise [4].

4 Simulation Results

We evaluate the VMMKNNF and AMN-VMMKNNF and compare their performances with other color filtering techniques proposed in the literature [1-5]. The criteria used to compare the restoration performance of various filters were the *peak signal-to-noise ratio* (PSNR) for evaluation of noise suppression, the *mean absolute error* (MAE) for quantification of edges and detail preservation, the *mean chromaticity error* (MCRE) for evaluation of chromaticity retention, and the *normalized color difference* (NCD) for quantification of the perceptual error [1-5]:

$$\text{PSNR} = 10 \cdot \log \left[\frac{(255)^2}{\text{MSE}} \right], \text{dB} \quad (10)$$

$$\text{MAE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \|y_{i,j} - d_{i,j}\|_{L_1} \quad (11)$$

where $\text{MSE} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N \|y_{i,j} - d_{i,j}\|_{L_2}^2$ is the *mean square error*, M, N are the image dimensions, $y_{i,j}$ is the vector value of pixel (i, j) , of the filtered image, $d_{i,j}$ is the corresponding pixel in the original noise free image, and $\|\cdot\|_{L_1}$, $\|\cdot\|_{L_2}$ are the L_1 - and L_2 -vector norm, respectively,

$$\text{MCRE} = \frac{\sum_{i=1}^M \sum_{j=1}^N \|p_{i,j} - \hat{p}_{i,j}\|_{L_2}^2}{MN} \quad (12)$$

where $p_{i,j}$ and $\hat{p}_{i,j}$ are the intersection points of $y_{i,j}$ and $d_{i,j}$ with Maxwell triangle plan, respectively,

$$\text{NCD} = \frac{\sum_{i=1}^M \sum_{j=1}^N \|\Delta E_{i,j}\|_{L_2}}{\sum_{i=1}^M \sum_{j=1}^N \|E_{i,j}^*\|_{L_2}} \quad (13)$$

where $\|\Delta E_{i,j}\|_{L_2} = \left[(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2 \right]^{1/2}$ is the color error, ΔL^* , Δu^* , and Δv^* are the difference in the L^* , u^* , and v^* components, respectively, between the two color vectors under consideration, and $\|E_{i,j}^*\|_{L_2} = \left[(L^*)^2 + (u^*)^2 + (v^*)^2 \right]^{1/2}$ is the norm or magnitude of the uncorrupted original image pixel vector in the $L^*u^*v^*$ space.

To determine the restoration properties and compare the qualitative characteristics of various color filters, the 3x3 VMMKNNF with simple, Hampel's three part redescending, and Andrew's sine influence functions, the 3x3 AMN-VMMKNNF filter with simple influence function, and the 3x3 vector median (VMF), 3x3 α -trimmed mean (α -TMF), 3x3 generalized vector directional (GVDF), 3x3 adaptive GVDF (AGVDF), 5x5 double window GVDF (GVDF_DW), 3x3 multiple non-parametric (MAMNFE), 3x3 adaptive

multichannel non parametric (AMNF), and 3x3 adaptive multichannel non parametric vector median (AMN-VMF) filters were simulated. The presented filters were computed and used according with their references [1-5] to compare them with the proposed filtering approach. The reason of these filters choosing to compare with the proposed ones is that their performances have been compared with various known color filters and their advantages have been demonstrated.

The 320x320 color test (24 bits per pixel with 16,777,216 colors) widely used images "Lena" and "Peppers" were corrupted by impulsive noise with 20% of spike occurrence in each a channel. Table 1 presents the comparative restoration results for the mentioned images applying the criteria mentioned above. We found from Table 1 that the restoration performance can be analyzing in two cases.

In the first case, when we employ the proposed VMMKNNF and the comparative VMF, α -TMF, GVDF, AGVDF, GVDF_DW filters that use the vector approach in their filtering scheme. One can see from the Table 1 that the PSNR performance of the VMMKNNF filters are changed in favor of the proposed filtering scheme in comparison with cases when other filters are used from 0.56dB (test image "Lena") to about 1.36dB (test image "Peppers"). The MAE performance is changed in favor of the proposed filter from 0.09 value (test image "Lena") and 0.15 value (test image "Peppers"). In the case of use the MCRE and NCD color criteria, the VMMKNNF filters have almost the same performances that the comparative filters but in some cases the performances of VMMKNNF filters are better.

In the second case, when we use the AMN-VMMKNNF filter and the comparative MAMNFE, AMNF, and AMN-VMF filters that employ the adaptive multichannel non parametric approach. Table 1 shows that the PSNR, MAE, MCRE, and NCD performances of the AMN-VMMKNNF filters are changed in favor of the proposed filtering scheme in comparison with cases when other filters are used. So, all calculated values of restoration criteria have shown better performances of the presented AMN-VMMKNNF filter.

Analyzing these two cases one can observe from Table 1 that the restoration performance of VMMKNNF and AMN-VMMKNNF are often better in comparison when other filters are used.

Figure 1 exhibits the processed images for test image "Peppers" explaining the impulsive noise suppression and detail preservation. From Figure 1 one can see that the proposed VMMKNNF and AMN-VMMKNNF appear to have a better subjective quality in comparison with MAMNFE and AMN-VMF.

Table 1. Comparative restoration results for color images corrupted by 20% impulsive noise

Algorithm	Lena				Peppers			
	PSNR	MAE	MCRE	NCD	PSNR	MAE	MCRE	NCD
VMF	21.15	10.73	0.035	0.038	19.92	11.37	0.055	0.043
α -TMF	20.86	14.97	0.046	0.049	19.68	16.76	0.084	0.067
GVDF	20.67	11.18	0.038	0.040	18.65	12.91	0.061	0.049
AGVDF	22.01	11.18	0.028	0.036	20.57	10.93	0.058	0.044
GVDF DW	22.59	10.09	0.028	0.039	20.50	11.60	0.049	0.047
MAMNFE	22.67	9.64	0.027	0.035	21.39	13.61	0.067	0.058
AMNF	20.36	16.51	0.047	0.053	19.21	18.49	0.086	0.073
AMN-VMF	24.14	9.64	0.027	0.035	22.93	10.41	0.052	0.044
AMN-VMMKNNF Simple	24.61	9.05	0.026	0.032	23.48	9.68	0.050	0.041
VMMKNNF Simple	23.15	10.00	0.033	0.034	21.86	10.79	0.058	0.042
VMMKNNF Andrew	23.07	10.01	0.033	0.035	21.88	10.86	0.058	0.044
VMMKNNF Hampel	23.05	10.04	0.033	0.035	21.93	10.78	0.057	0.043

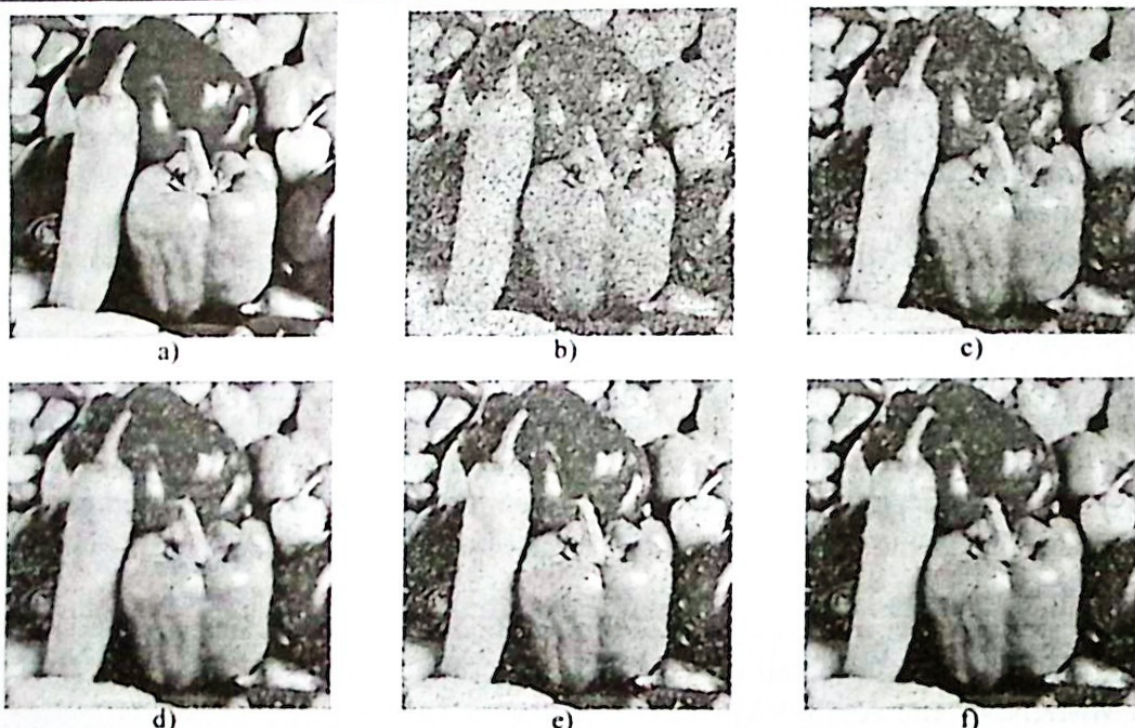


Fig. 1. Subjective visual qualities of restored color image "Peppers", (a) Original test image "Peppers", (b) Input noisy image (corrupted by 20% impulsive noise in each a channel), (c) MAMNFE filtering image, (d) AMN-VMF filtering image, (e) Proposed VMMKNNF filtered image (Simple), and (f) Proposed AMN-VMMKNNF filtered image (Simple)

The values of parameters for AMN-VMMKNNF and VMMKNNF filters and influence functions were found during numerous simulations. The optimal values of the parameters of the proposed filters were: $a=0.6$ and $K_{min}=5$. The parameters of the influ-

ence functions were: $r \leq 81$ for Andrews sine, and $\alpha = 10$, $\beta \leq 90$ and $r = 300$ for Hampel three part redescending. Therefore, there can be existed some variations of about $\pm 5\%$ of PSNR, MAE, MCRE, and NCD performances with the use of other parameter values which are different ones than before presented. Finally, in this paper we have standardized these parameters as the constants to realize the implementation of the proposed algorithms for real-time processing [7, 12].

The runtime analysis of various filters was realized using the Texas Instruments DSP TMS320C6711. This DSP has a performance of up to 900 MFLOPS at a clock rate of 150 MHz [9]. The filtering algorithms were implemented in C language using the BORLANDC 3.1 for all routines, data structure processing and low level I/O operations. Then, we compiled and executed these programs in the DSP TMS320C6711 applying the Code Composer Studio 2.0 [9].

According to the restoration performance results obtained in the Table 1 their processing time values are depicted in the Table 2. The processing time in seconds includes the time of acquisition, processing, and storing of data. Analyzing the Table 2 we found the following results: the processing time of proposed VMMKNNF with different influence functions has values about 0.3s or less and these times are less than for other filters with exception of VMF, α -TMF, and AMNF. The processing time values of AMN-VMMKNNF are larger than for any another filter, but as it has been proven such a filter presents the better performance.

Table 2. Processing time for different filters on color images degraded by 20% impulsive noise

Algorithm	Processing Time	
	Lena	Peppers
VMF	0.039	0.039
α -TMF	0.087	0.087
GVDF	0.564	0.559
AGVDF	0.620	0.620
GVDF DW	0.721	0.720
MAMNFE	0.832	0.832
AMNF	0.095	0.095
AMN-VMF	0.648	0.648
AMN-VMMKNNF Simple	3.687	3.643
VMMKNNF Simple	0.296	0.296
VMMKNNF Andrew	0.199	0.261
VMMKNNF Hampel	0.199	0.231

We can also conclude that the proposed VMMKNNF can process up to 5 images of 320x320 pixels per second depending on the influence function applied. The processing time performance of VMMKNNF depends on the image to process and do not almost vary for different noise level. These values depend on the complex calculation of influence functions and parameters of the proposed filters.

We also applied the proposed filters to process video signals. Since most video sequences have high correlation between consecutive frames, it is clear that the 3-D filtering can be more efficient than the 2-D filtering in terms on PSNR [7]. However, the 3-D filtering requires relatively large amount of computations and processing time [7]. In some applications such as computer vision systems or medical imaging the consecutive frames of a video sequence have no any correlation. For these reasons we only investigated 2-D image filtering in the case of the video sequences. It has been processed the QCIF (Quarter Common Intermediate Format) video color sequences "Miss America" and "Flowers" to demonstrate that the proposed methods potentially can provide a real-time filtering solution. This picture format uses 176x144 (24 bits per pixel with 16,777,216 colors) luminance pixels per frame. The test video color sequences were contaminated by impulsive noise with 20% of spike occurrence in each a channel. The restoration performance (PSNR, MAE, MCRE, and NCD) for one frame of the mentioned sequences are presented in Table 3. Table 4 presents minimum and maximum processing time of the filters in the case of use of 150 and 120 frames of video sequences "Miss America" and "Flowers", respectively. Figure 2 illustrates the filtered frame showing subjective visual criterion for the sequence "Flowers" according to the Table 3.

Table 3. Comparative restoration results for the first frame of the different video sequences corrupted by 20% impulsive noise

Algorithm	Miss America				Flowers			
	PSNR	MAE	MCRE	NCD	PSNR	MAE	MCRE	NCD
VMF	20.08	9.99	0.065	0.046	15.94	24.38	0.065	0.072
α -TMF	19.48	17.40	0.097	0.097	16.32	27.68	0.068	0.067
GVDF	17.61	14.09	0.083	0.069	15.95	24.24	0.063	0.060
AGVDF	20.72	9.93	0.068	0.052	16.42	23.88	0.065	0.058
GVDF-DW	20.82	9.96	0.062	0.049	16.38	23.97	0.054	0.061
MAMNF	21.81	13.18	0.077	0.073	16.89	25.82	0.056	0.064
AMNF	18.99	19.25	0.092	0.107	15.80	30.04	0.068	0.073
AMN-VMF	24.10	8.92	0.061	0.047	17.40	23.16	0.053	0.058
AMN-VMMKNF Simple	25.05	7.92	0.055	0.042	17.68	22.40	0.054	0.056
VMMKNF Simple	21.98	9.58	0.067	0.048	16.88	23.48	0.067	0.057
VMMKNF Andrews	22.15	9.53	0.064	0.050	16.84	23.51	0.065	0.058
VMMKNF Hampel	22.18	9.47	0.063	0.050	16.82	23.56	0.065	0.059

Applying the same criteria that were used in the Table 1, the Table 3 demonstrates that the better restoration performances were obtained by VMMKNF and AMN-VMMKNF (see Figure 2). The Table 3 only presents the restoration performance for the first frame of the video sequences, because for all other frames we obtained similar results demonstrating the better performances of proposed filters in comparison with other ones.

One can see from the Table 4 that the processing times of proposed AMN-VMMKNNF technique have larger values in comparison with other filters usage. Therefore, the proposed VMMKNNF can process from 7 up to 16 frames per second employing any influence function. It is clearly that in case of an image that has 3 or 4 times less than 320x320 pixels the proposed VMMKNNF filtering technique can preserve the edges and small-size details, and remove impulsive noise better than other filters practically with standard film velocity for computer vision applications.

Table 4. Minimum and maximum processing time in the case of the different filters for all frames of the video color sequences

Algorithm	Processing Time			
	Flowers		Miss America	
	Min	Max	Min	Max
VMF	0.0153	0.0153	0.0153	0.0153
α -TMF	0.0213	0.0213	0.0213	0.0213
GVDF	0.2026	0.2189	0.1869	0.2187
AGVDF	0.2263	0.2426	0.2106	0.2424
GVDF DW	0.7092	0.7591	0.7205	0.7574
MAMNFE	0.3219	0.3219	0.3219	0.3219
AMNF	0.0371	0.0371	0.0371	0.0371
AMN-VMF	0.2506	0.2506	0.2506	0.2506
AMN-VMMKNNF Simple	1.4426	1.4503	1.3417	1.4454
VMMKNNF Simple	0.1218	0.1266	0.1109	0.1251
VMMKNNF Andrew	0.0765	0.0837	0.0898	0.1213
VMMKNNF Hampel	0.0595	0.0702	0.0917	0.0983

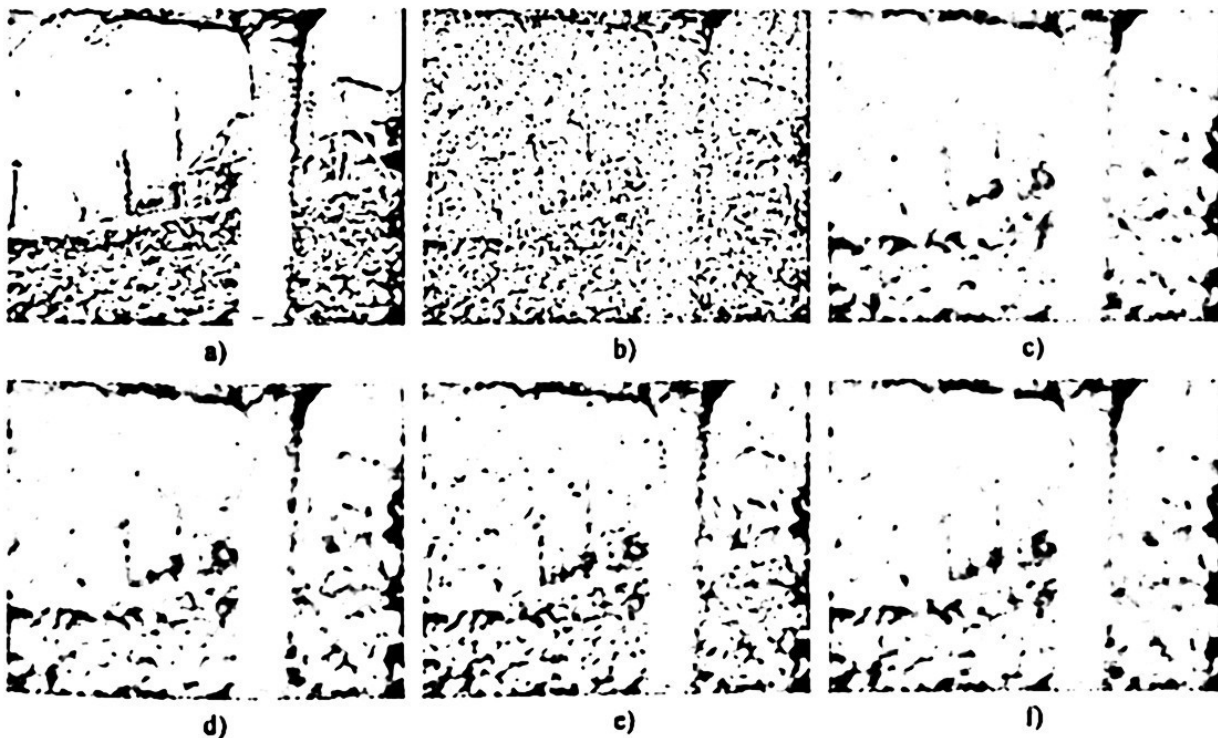


Fig. 2. Subjective visual qualities of restored color frame of video sequence "Flowers". (a) Original test frame "Flowers", (b) Input noisy frame (corrupted by 20% impulsive noise in each a channel), (c) MAMNFE filtering frame, (d) AMN-VMF filtering frame, (e) Proposed VMMKNNF filtered frame (Simple), and (f) Proposed AMN-VMMKNNF filtered frame (Simple)

Finally, the simulation results show two important criteria to choose the multichannel RM-filter type: restoration performance and processing time. We propose to use the VMMKNNF when it is necessary to realize on-line processing, for example for video color sequences, because such the filters have the minimum processing time. In this case, the Hampel influence function into the VMMKNNF is more convenient for application because it provides less processing times in the proposed filtering technique. For other applications where the processing time does not important we recommend the use of AMN-VMMKNNF due that it provides the better performance in noise suppression and detail performance in comparison with other filters.

5. Conclusions

The designed VMMKNNF and AMN-VMMKNNF are able to remove impulsive noise and preserve the edges and details in the color imaging. These filters use the RM-estimator with different influence functions, which is applicable to color image processing. The proposed AMN-VMMKNNF use an adaptive non parametric approach to provide better impulsive noise suppression.

The proposed filters have demonstrated better quality of image processing in visual and analytical sense in comparison with different known color imaging algorithms.

The proposed VMMKNNF filters potentially provide a real-time solution to quality video transmission. The processing time can be reduced if we utilize a DSP with better performance than that used here, for example the TMS320C8X Multiprocessor DSP.

Acknowledgements. The authors thank the National Polytechnic Institute of Mexico and CONACyT (project 42790) for its support.

References

1. Plataniotis, K. N., Lacroix, A., Venetsanopoulos, A. N.: *Color Image Processing and Applications*, Springer Verlag Wien (2000)
2. Bovik, A.: *Handbook of Image and Video Processing*, Academic Press, San Diego CA (2000)
3. Trahanias, P. E., Karakos, D. G., Venetsanopoulos, A. N.: Directional processing of color images: Theory and experimental results, *IEEE Trans. Image Process.*, vol. 5, 868-880 (1996)
4. Plataniotis, K. N., Androutsos, A., Vinayagamoorthy, S., Venetsanopoulos, A. N.: Color image processing using adaptive multichannel filters, *IEEE Trans. Image Process.*, Vol. 6-7, 933-949 (1997)
5. Astola, J., Kuosmanen, P.: *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Boca Raton-New York (1997)

6. Gallegos-Funes, F. J., Ponomaryov, V., Sadovnychiy, S., Nino-de-Rivera, L.: Median M-type K-nearest neighbour (MM-KNN) filter to remove impulse noise from corrupted images, *IEE Electronics Letters*, 38(15), 786-787, (2002)
7. Gallegos-Funes, F. J., Ponomaryov, V. I.: Real-time image filtering scheme based on robust estimators in presence of impulsive noise, *Real Time Imaging*, 10(2), 69-80 (2004)
8. Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., Stahel, W. A.: *Robust Statistics. The approach based on influence function*, Wiley, New York (1986)
9. Texas Instruments Inc., TMS320C6711, TMS320C6711B, TMS320C6711C Floating-Point Digital Signal Processors, SPRS088H, Texas Instruments Incorporated (1998), Revised (2003)
10. Zervakis, M. E., Venetsanopoulos, A. N.: M-estimators in robust nonlinear image restoration, *Optical Engineering*, 31(10) (1992)
11. Peltonen, S., Kuosmanen, P., Astola, J.: Output distributional influence function, *Proc. IEEE EURASIP Workshop on Nonlinear Signal and Image Processing*, 33-37, Antalya, Turkey (1999)
12. Ponomaryov, V. I., Rosales, A. J., Gallegos-Funes, F.: Real-time color imaging using the vectorial order statistics filters, *Proc. SPIE Real-Time Imaging VIII*, vol. 5297, 35-44, San Jose, CA, USA (2004)
13. Aizenberg, I., Astola, J., Bregin, T., Butakoff, C., Eriazarian, K., Paliy D.: Detectors of the impulsive noise and new effective filters for the impulsive noise reduction, *Proc. SPIE Image Processing: Algorithms and Systems II*, vol. 5014, 419-428 (2003)

Lifting Based DWT for Lossy Image/Signal Compression

Oleksiy Pogrebnyak, Pablo Manrique Ramírez, Enrique Guzmán Ramírez

Instituto Politécnico Nacional, CIC-IPN, Av. Juan de Dios Bátiz s/n,
Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F.
E-mail: olek_pmanriq@pollux.cic.ipn.mx

Abstract. A new wavelet transform algorithm based on a modified lifting scheme is presented. The algorithm uses the wavelet filters derived from a generalized lifting scheme at the decomposition stage. The proposed generalized framework for the lifting scheme permits to obtain easily different wavelet filter coefficients in the case of the $(-N, N)$ lifting. The restoration stage of the lifting scheme was modified in a standard wavelet transform fashion to obtain good rate/distortion performance. The algorithm performs decomposition/restoration on the first level whereas at the next level classic wavelet filters are used. For this purpose, the restoration high-pass wavelet filter was derived from the generalized lifting scheme. The obtained restoration high-pass and low-pass wavelet filters operate as a known classic wavelet filters using data upsampling. The low-pass filter for the restoration stage uses the same coefficients of the lifting decomposition predictor. The proposed transform possesses the perfect restoration of the processed data when the quantification is not used and possesses a good energy compactation. The designed algorithm was tested on different test images and the obtained results are comparable with the JPEG2000 9/7 wavelet filtering. The proposed algorithm performs better on the images having high energy of the low-scale details.

Keywords: image processing, wavelets, lifting scheme, lossy data compression

1 Introduction

In the past decade, the wavelet transform has become a popular, powerful tool for different image and signal processing applications such as noise cancellation, data compression, feature detection, etc. Meanwhile, the aspects of fast wavelet decomposition/reconstruction implementation and high rate data compression now continue to be under consideration.

The first algorithm of the fast discrete wavelet transform (DWT) was proposed by S.G.Mallat [1]. This algorithm is based on the fundamental work of Vetterli [2] on signal/image decomposition by 1-D quadrature-mirror filters (QMF), and orthogonal wavelet functions. The next breakthrough concerning a fast DWT algorithm implementation was made by W.Sweldens [3] who first proposed the lifting scheme.

The lifting scheme is known to perform well in lossless data compression. Unfortunately, it has worse rate/distortion characteristics than the classic wavelets and for this reason is not employed for lossy data compression. In this paper, we try to overcome such lifting scheme inefficiency.

We present a DWT algorithm, which is based on the proposed generalized lifting scheme framework. According to this framework, different wavelet filters can be easily obtained. With the proposed method, varying some coefficients, one can obtain the wavelet filter of a different order and frequency response to adjust the wavelet decomposition properties. This way, the decompositions can be optimized to achieve a minimum of the entropy in the wavelet domain.

2 Lifting Scheme Generalization

The lifting scheme is widely used in the wavelet based image analysis. Its main advantages are: the reduced number of calculations; less memory requirements; the possibility of the operation with integer numbers. The lifting scheme consists of the following basic operations: splitting, prediction and update.

Splitting is sometimes referred to as the lazy wavelet. This operation splits the original signal $\{x\}$ into odd and even samples:

$$s_i = x_{2i}, \quad d_i = x_{2i+1}. \quad (1)$$

Prediction, or the dual lifting. This operation at the level k calculates the wavelet coefficients, or the details $\{d^{(k)}\}$ as the error in predicting $\{d^{(k-1)}\}$ from $\{s^{(k-1)}\}$:

$$d_i^{(k)} = d_i^{(k-1)} + \sum_{j=-N/2}^{N/2} p_j s_{i+j}^{(k-1)}, \quad (2)$$

where $\{p\}$ are coefficients of the wavelet-based high-pass FIR filter and \tilde{N} is the prediction filter order.

Update, or the primal lifting. This operation combines $\{s^{(k-1)}\}$ and $\{d^{(k)}\}$, and consists of low-pass FIR filtering to obtain a coarse approximation of the original signal $\{x\}$:

$$s_i^{(k)} = s_i^{(k-1)} + \sum_{j=-N/2}^{N/2} u_j d_{i+j}^{(k)}, \quad (3)$$

where $\{u\}$ are coefficients of the wavelet-based low-pass FIR filter and N is the prediction filter order.

Sometimes, the normalization factors can be applied to the wavelet details and approximations to produce the transformed data values more similar to the classic DWT algorithm.

The inverse transform is straightforward: first, the signs of FIR filter coefficients $\{u\}$ and $\{p\}$ are switched; the inverse update followed by inverse prediction is calculated. Finally, the odd and even data samples are merged.

A fresh look at the lifting scheme first was done in [4], where the FIR filters that participate in the prediction and update operation are described in the domain of Z-transform. According to this approach, the transfer function of the prediction FIR filter can be formulated as follows [5]

$$H_p(z) = 1 + p_0(z + z^{-1}) + p_1(z^3 + z^{-3}) + \dots + p_{\frac{\tilde{N}}{2}-1}(z^{\tilde{N}-1} + z^{-\tilde{N}+1}) \quad (4)$$

The $H_p(z)$ must has zero at $\omega = 0$, i.e., at $z = 1$. It can be easily found [4] that this condition is satisfied when

$$\sum_{i=0}^{\frac{\tilde{N}}{2}-1} p_i = -\frac{1}{2}. \quad (5)$$

When the condition (5) is satisfied, $H_p(-1) = 2$ and $H_p(0) = 1$ that means the prediction filter has gain 2 at $\omega = \pi$ and unit gain at $\omega = \frac{\pi}{2}$. Thus, if one want to obtain the normalized detail coefficients, the normalization factor must be equal to $\frac{1}{2}$.

Following this approach, the transfer function for update filter can be obtained. We prefer to formulate this transfer function in the terms of $H_p(z)$ [5]:

$$H_u(z) = 1 + H_p(z) \left\{ u_0 [z] + [z^{-1}] + u_1 [z^3] + [z^{-3}] + \dots + u_{\frac{N}{2}-1} [z^{N-1}] + [z^{-N+1}] \right\}. \quad (6)$$

Similarly, $H_u(z)$ must has zero at $\omega = \pi$, i.e., at $z = -1$. It can be easily found [4] that this condition is satisfied when

$$\sum_{i=0}^{\frac{N}{2}-1} u_i = \frac{1}{4}. \quad (7)$$

When the condition (7) is satisfied, $H_u(0) = 1$ and that means the prediction filter has gain 1 at $\omega = 0$.

An elegant conversion of the formulas (5), (7) in the case of (4,4) lifting scheme was proposed in [4] to reduce the degree of freedom in the predictor and update coefficients. With some modifications, the formulas for the wavelet filters coefficients are as follows:

$$p_0 = -\frac{128+a}{256}, \quad p_1 = \frac{a}{256}, \quad (8)$$

$$u_0 = \frac{64+b}{256}, \quad u_1 = -\frac{b}{256}, \quad (9)$$

where a and b are the parameters that control the DWT properties. Also, in [4] it was found the correspondence between these control parameters and the conventional (non-lifted) wavelet filters.

The behavior of the lifting scheme with the respect of a and b values can be evaluated easily in the frequency domain using transfer functions (4), (6) with $z = e^{-j\omega}$. Figures 1 and 2 represent the magnitude of the frequency responses of high-pass and low-pass (2,2) and (4,4) filters with $a=0$, $b=0$ and $a=16$, $b=16$, respectively. These figures were obtained with MATHCAD200i software.

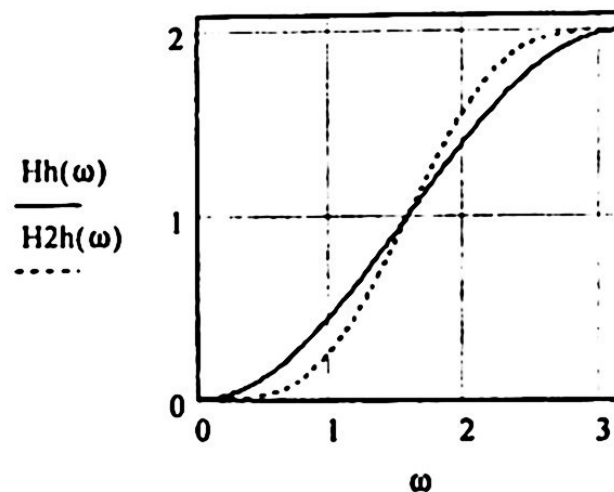


Fig. 1 Frequency response of lifting predictor with $\tilde{N} = 2$ and $a = 0$ ($H2h(\omega)$, dotted line) and $\tilde{N} = 4$ and $a = 16$ ($Hh(\omega)$, solid line)

By simulations with MATHCAD2000i software, we found that better frequency response of (4,4) lifting filter can be obtained with the different a and b parameters. In particular, when $a = 20$, the transition band of the high-pass filter is the narrowest. However, increasing more the coefficient a value leads to the appearance of the false side lobe in the low-frequency band.

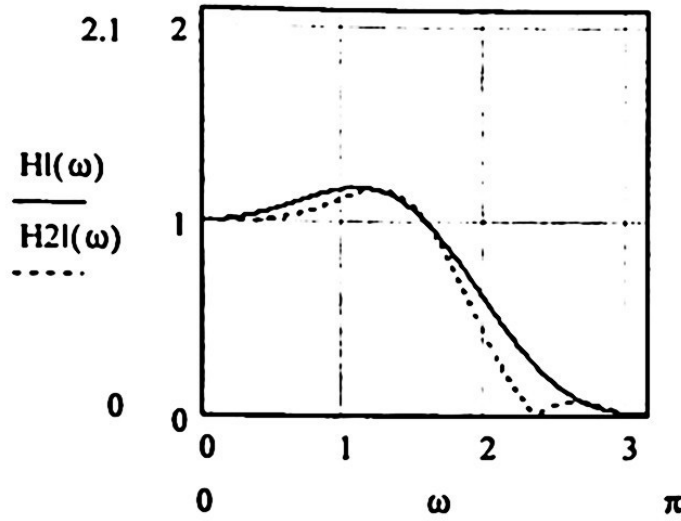


Fig. 2 Frequency response of lifting update filter with $\tilde{N} = 2, N = 2$ and $a = 0, b = 0$ ($H2l(\omega)$, dotted line) and $\tilde{N} = 4, N = 4$ and $a = 16, b = 16$ ($Hl(\omega)$, solid line)

The behavior of update filter is more complicated. The frequency response of this low-pass filter depends not only on b parameter value, but inherently on the a value. This fact is explained by the nature of the lifting scheme. In practice, the influence of the coefficient a value (which is relatively small) can be avoided choosing this value to produce the predictor with the desired frequency response and then, choosing the appropriate value of the coefficient b . For example, if we choose $a = 20$, the narrowest transition band in the update filter is obtained with $b = 9$ meanwhile with the greater b a side lobe appears in the high frequency band.

Using the generalization of the lifting scheme (4)-(7), we found by simulations that the coefficients of the lifting filters of an arbitrary order higher than 4 can be found according to the recursive formulas [5]:

$$p_0 = -\frac{128+a}{256}, p_1 = -\frac{a}{256}, p_2 = -\frac{p_1}{c}, p_1 = p_1 - p_2, \dots, p_{\tilde{N}} = -\frac{p_{\tilde{N}-1}}{c}, p_{\tilde{N}-1} = p_{\tilde{N}-1} - p_{\tilde{N}} \quad (10)$$

$$u_0 = \frac{64+b}{256}, u_1 = \frac{-b}{256}, u_2 = \frac{u_1}{d}, u_1 = u_1 - u_2, \dots, u_N = \frac{u_{N-1}}{d}, u_{N-1} = u_{N-1} - u_N \quad (11)$$

where the parameters c, d controls the filter characteristics. This way, the wavelet filters of an arbitrary order can be derived. The formulas (10), (11) are the extension of the equations (8), (9) and use the same parameters a, b to control the characteristics of the lifting filters. The parameters a, b control the width of the transition bands and the new

control parameters c, d control the smoothness of the pass and stop bands to prevent the appearance of the lateral lobes: with greater values of a, b the values of c, d tend to be greater. The behavior of the coefficient d is of a rapid increasing, and in practice it has large value sufficiently high to say that the influence of the terms in $H_u(z)$ of the order higher of 3 may be neglected. Thus, in practice, one can use lifting update filter of the order $N \leq 6$ without a significant widening of the update filter transition band: the width of this transition band, mainly, is determined by the lifting predictor frequency characteristics. Fig. 3 shows the frequency responses of (10,4) lifting wavelet filters with $a = 28, b = 8, c = 3, d = 3$.

3 Design of Lifting Scheme Based Algorithm for Lossy Data Compression

Generally, the performance of lifting scheme in data compression is worse than of the classic DWT based on subband decomposition, especially in case of lossy compression. This fact can be explained by the following properties of the lifting wavelets.

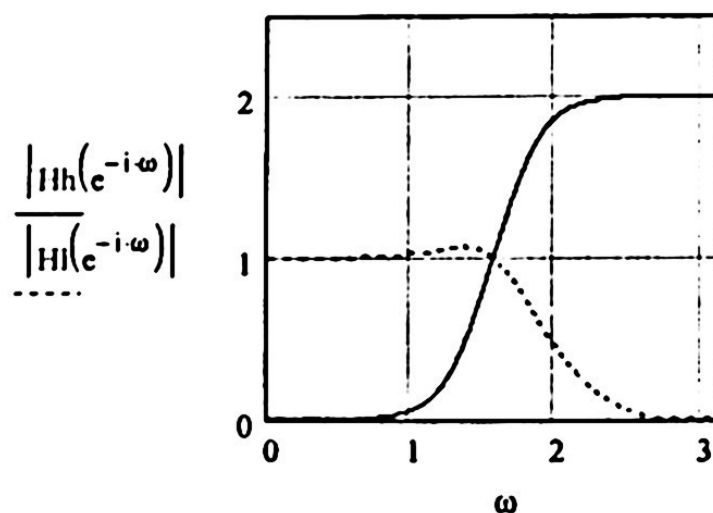


Fig. 3 Frequency responses of (10, 4) lifting filters with $a = 28, b = 8, c = 3, d = 3$

First, the pass band gain of the classic DWT filters frequently is $\sqrt{2}$ meanwhile the lifting filters have gain 1 and 2. The normalization of the gain values of lifting decompositions to $\sqrt{2}$ permits the use of the same quantization technique and facilitates the comparison of the data reconstructed by the different algorithms.

Second, the center of the transition band of high-pass lifting predictor filters is always at $\frac{\pi}{2}$ and the center of the transition band of the low-pass filter is far from the $\frac{\pi}{2}$ tending

to the high frequencies. Besides, the transition centers of the classic wavelet filters are optimized to give the aliasing both in high-pass and low-pass filters resulting in better rate/distortion performance for the majority of the natural images. From the other hand, the frequency response of the high-pass lifting filters produces lower values of the wavelet coefficients and, in some situations, the rate/distortion performance of the lifting scheme can be better when the filter characteristics are fit to the spectral characteristics of the image. Such situations are common in case of the images with high energy of the low-scale details. For this reason, the lifting filters usually perform better in the rate/distortion sense at the first level of the wavelet decomposition (when the coefficients a, b, c, d are adjusted properly) meanwhile the classic wavelet filters perform better at higher levels.

Another way to solve the problem of the fixed at $\frac{\pi}{2}$ high-pass filter transition band is introduce the even powers of z in Eqs. (4), (6), but such a technique leads to a classic-like wavelet filter design.

The problem of the gain normalization is easy to resolve by introducing the normalization factor in the lifting filters:

$$H_p(z) = \frac{1}{\sqrt{2}} + p_0(z + z^{-1}) + p_1(z^3 + z^{-3}) + \dots + p_{\frac{\tilde{N}}{2}-1}(z^{\tilde{N}-1} + z^{-\tilde{N}+1}) \quad (12)$$

$$H_u(z) = \sqrt{2} + H_p(z) \left\{ u_0 [z + z^{-1}] + \dots + u_{\frac{N}{2}-1} [z^{N-1} + z^{-N+1}] \right\} \quad (13)$$

$$p_0 = -\frac{128+a}{256\sqrt{2}}, \quad p_1 = -\frac{a}{256\sqrt{2}}, \quad p_2 = -\frac{p_1}{c}, \quad p_1 = p_1 - p_2, \dots, \\ p_{\tilde{N}} = -\frac{p_{\tilde{N}-1}}{c}, \quad p_{\tilde{N}-1} = p_{\tilde{N}-1} - p_{\tilde{N}} \quad (14)$$

$$u_0 = \frac{64+b}{128}, \quad u_1 = \frac{-b}{128}, \quad u_2 = \frac{u_1}{d}, \quad u_1 = u_1 - u_2, \dots, u_N = \frac{u_{N-1}}{d}, \quad u_{N-1} = u_{N-1} - u_{\tilde{N}} \quad (15)$$

The second problem is more complicated. To solve it we propose to employ a hybrid algorithm that uses classic DWT filters at high levels of the decomposition, and the same subband coding technique is used at the first level but with the FIR filters derived from the lifting scheme. To this end, the classic-like wavelet filters need to be found. One can use the inverted predictor (12), (14) for low-pass restoration filtering. At the same time, the high-pass filter for the reconstruction can be derived multiplying directly the polynomials (12), (13). For example, in case of (10,4) lifting the transfer functions of a pair of the reconstruction wavelet filters has the following representation:

$$H_{low-pass}(z) = \frac{1}{\sqrt{2}} + p_0(z + z^{-1}) + \dots + p_4(z^9 + z^{-9}) \quad (16)$$

$$\begin{aligned} H_{high-pass}(z) = & \sqrt{2} + H_{low-pass}(z) \left[u_0(z + z^{-1}) + u_1(z^3 + z^{-3}) \right] = \sqrt{2} + 2(p_0 u_0 + p_1 u_1) \\ & + u_0 \sqrt{2}(z + z^{-1}) + (p_0 u_0 + p_1 u_0 + p_0 u_1 + p_2 u_1)(z^2 + z^{-2}) + u_1 \sqrt{2}(z^3 + z^{-3}) \\ & + (p_1 u_0 + p_0 u_1 + p_2 u_0 + p_3 u_1)(z^4 + z^{-4}) + (p_2 u_0 + p_1 u_1 + p_3 u_0 + p_4 u_1)(z^6 + z^{-6}) \\ & + (p_3 u_0 + p_2 u_1 + p_4 u_0)(z^8 + z^{-8}) + (p_3 u_1 + p_4 u_0)(z^{10} + z^{-10}) + p_4 u_1(z^{12} + z^{-12}) \end{aligned} \quad (17)$$

The correspondent frequency responses of the filters (16), (17) with $a = 30, b = 8, c = 3, d = 3$ are shown in Fig. 4.

4 Experimental Results

The described in the previous section algorithm were tested on a 128x128 artificial image and a set of 512x512 images ("Lena", "Baboon", "Barbara", "Boat", "Peppers") shown in Fig. 5.

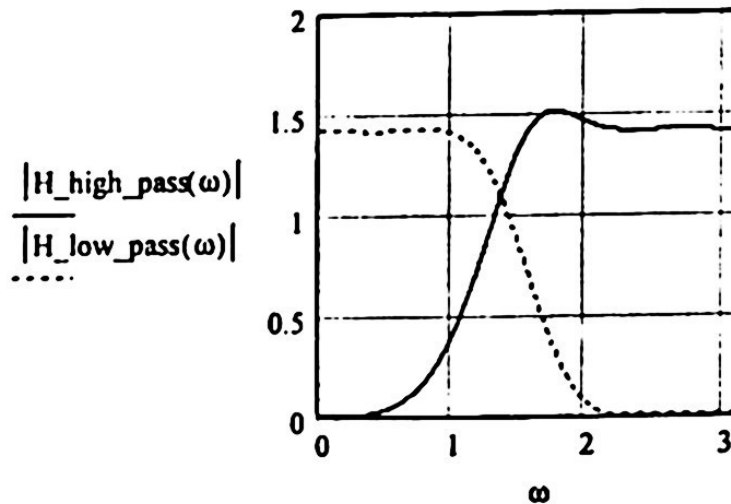


Fig. 4 Frequency responses of the derived from lifting scheme restoration wavelet filters (16),(17)

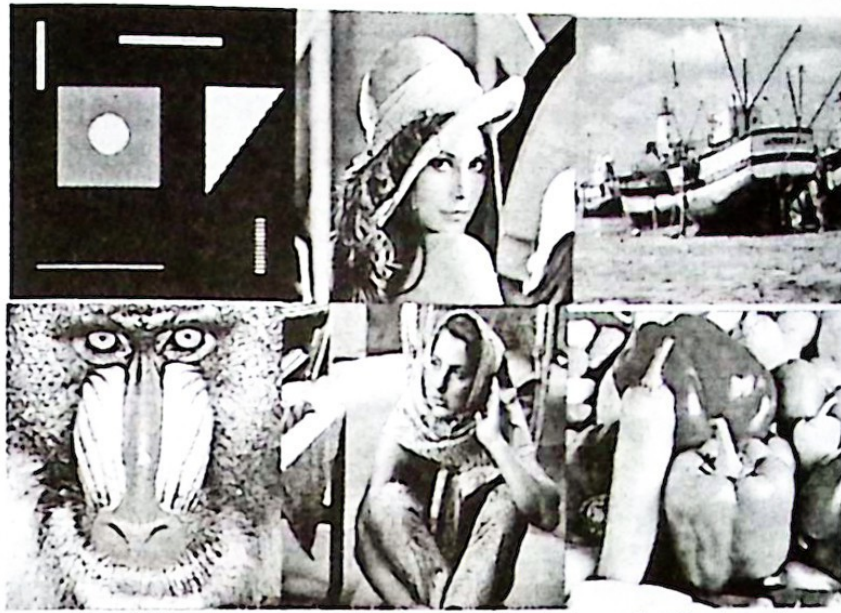


Fig. 5 Test images: artificial, "Lena", "Baboon", "Barbara"

Fig.6 presents the simulation results for the compression of the test images decomposed on one level by the JPEG2000 9/7 wavelet filters and the proposed algorithm. From the analysis of the curves presented in this Figure, it follows that the proposed lifting algorithm performs better than 9/7 wavelets for all test images, especially at high compression rates.

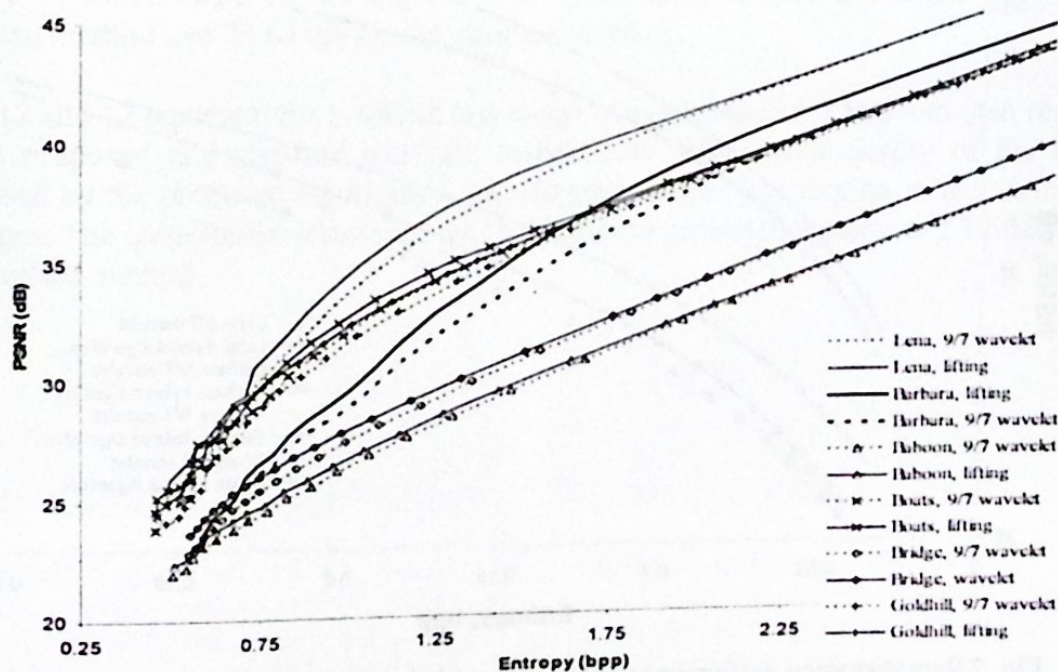


Fig. 6 Rate/distortion performance of the proposed algorithm vs. JPEG2000 9/7 DWT for one level of decomposition

Fig. 7 and 8 show the resulted rate/distortion dependencies obtained with the proposed hybrid algorithm and with JPEG2000 9/7 DWT. In simulations for data compression, the parameters a, b, c, d were varied to obtain the better PSNR y lower entropy per pixel for each test image. The best results were obtained with $a=20+30$, $b=8+12$, $c=2+7$ and $d=2+7$. The order \tilde{N} also were varied and in all cases the best results were obtained for $\tilde{N}=10$. The better value of order N in all cases was $N=6$.

It follows from the results presented in the Fig. 7, 8 that the designed algorithm possesses good energy compaction ability comparing to the JPEG2000 9/7 DWT performance. For the test images "Lena" and "Mandrill" the proposed algorithm performs slightly worse. For the artificial test image the proposed algorithm performs significantly better. For the test image "Barbara" the developed algorithm performs slightly better in energy compaction than the JPEG2000 9/7 DWT.

The obtained simulation results show that the energy compaction characteristics of the proposed algorithms depend on the predictor and update filter orders. Generally, for better data compression the higher orders are required. However, in the simulations, the predictor filter order was limited to $\tilde{N}=10$ and the update filter order was limited to $N=3$ due to the necessity of analytic calculation of the high-pass restoration filter transfer function for different \tilde{N} and N . These limitations were done because in this paper we only wished to demonstrate that the proposed method is competitive with the standard one.

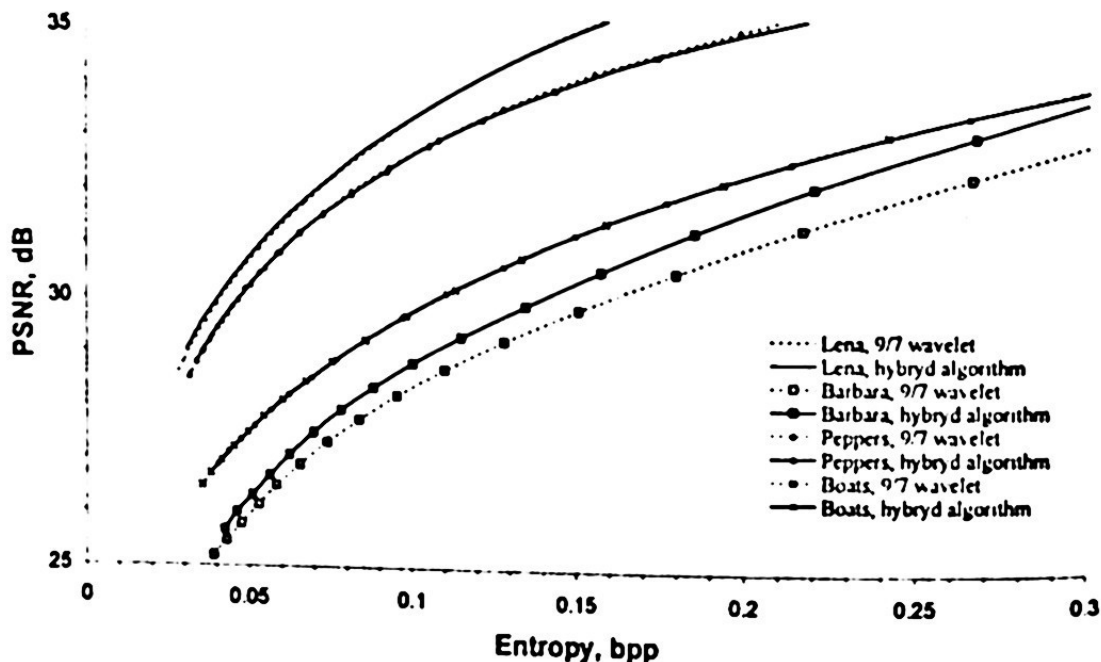


Fig. 7 Rate/distortion performance of the proposed algorithm vs. JPEG2000 9/7 DWT

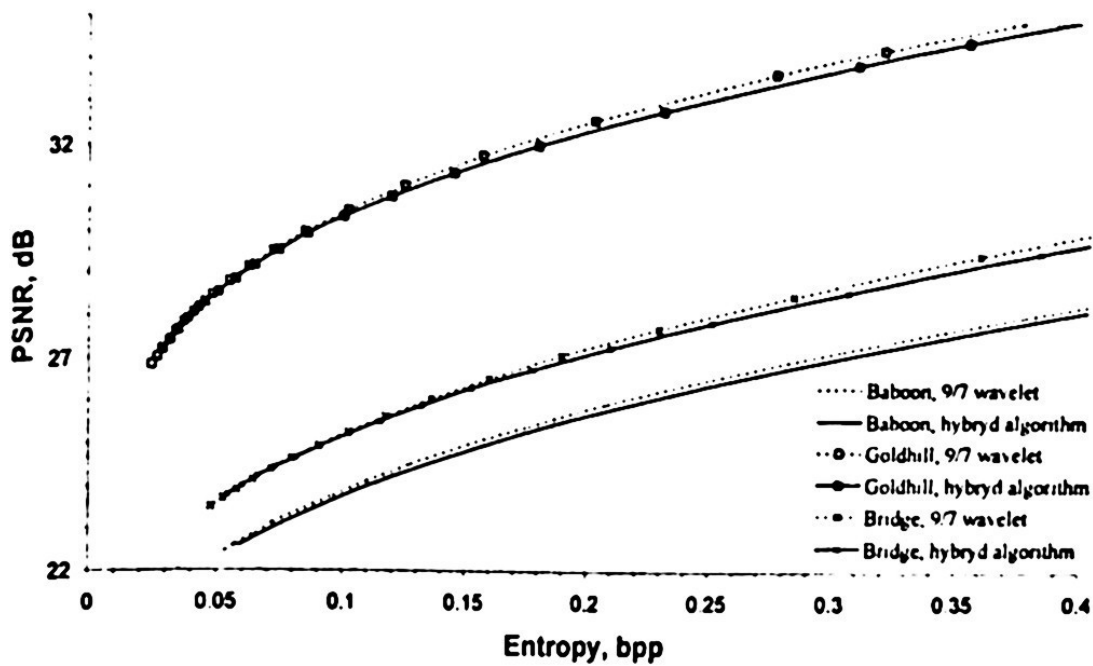


Fig. 8 Rate/distortion performance of the proposed algorithm vs. JPEG2000 9/7 DWT

Fig. 9 and 10 represent the test image "Barbara" compressed to 0.15 bpp and then restored by the proposed and standard methods, respectively. The visual quality of the both images is almost the same but the proposed algorithm produces less ringing in the vicinity of the borders of the objects. The quantitative results are 30.06 dB for the proposed method and 24.61 dB for the standard method.

Fig. 12 and 13 represent the artificial test image compressed to 0.3 bpp and then restored by the proposed and standard methods, respectively. The visual quality of the image processed by the proposed algorithm is significantly better: less ringing in the vicinity of the edges. The quantitative results are 37.15 dB for the proposed method and 35.88 dB for the standard method.



Fig. 8 Test image "Barbara" compressed to 0.15 bpp (normalized entropy value) and restored using the proposed modified lifting algorithm with $a=28$, $b=15$, $c=3$, $d=2$, $\tilde{N}=10$, $N=6$; PSNR=30.06 dB.



Fig. 9 Test image "Barbara" compressed to 0.15 bpp (normalized entropy value) and restored using JPEG2000 9/7 DWT, PSNR=24.61dB.

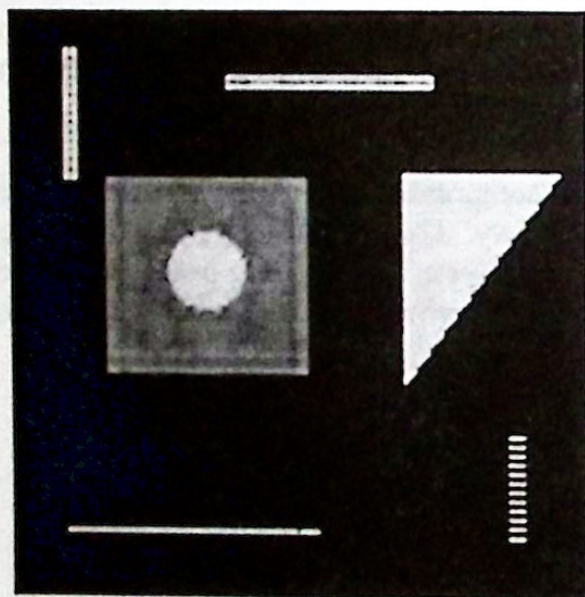


Fig. 10 Artificial test image compressed to 0.3 bpp (normalized entropy value) and restored using the proposed modified lifting algorithm with $a=4$, $b=2$, $c=7$, $d=6$, $\tilde{N}=10$, $N=6$; PSNR=37.15 dB.

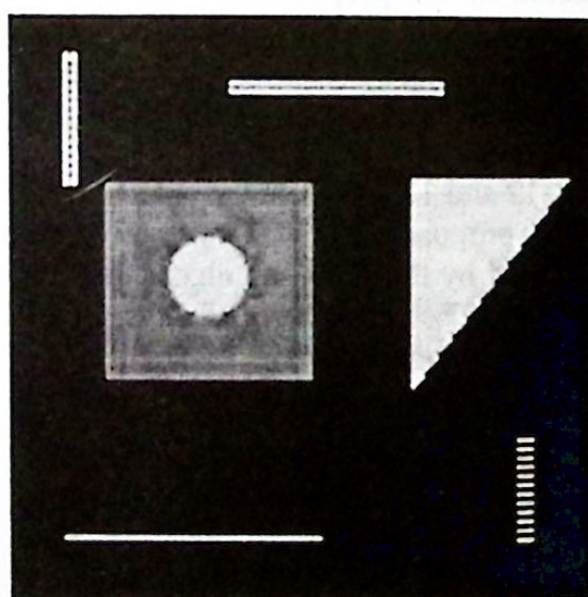


Fig. 11 Artificial test image compressed to 0.3 bpp (normalized entropy value) and restored using JPEG2000 9/7 DWT, PSNR=35.88dB.

5 Conclusion

A novel algorithm of DWT based on the developed generalized lifting scheme is presented. The energy compaction properties of the presented technique are better or comparable in comparison to the standard JPEG2000 wavelet filters. The rate/distortion performance of the proposed algorithm can be enhanced adjusting a set of parameters that gives the possibility to maximize the compression according to the statistical properties of the processed data.

6 Acknowledgements

This work was supported by the Consejo Nacional de Ciencia y Tecnología (CONACYT) as a part of the research project MARINA #11055 and by Instituto Politécnico Nacional as a part of the research project #2004285.

References

1. S.G.Mallat, A theory for multiresolution signal decomposition: The wavelet representation, *IEEE Trans. Patt. Anal. Mach Intell.*, Vol. 11, No. 7, p.p.674-693, 1989.
2. Martin Vetterli, Multi-dimensional sub-band coding: some theory and algorithms, *Signal Processing*, Vol. 6, pp.97-112, 1984.
3. W.Sweldens, The lifting scheme: A new philosophy in biorthogonal wavelet constructions, *Wavelet Applications in Signal and Image Processing III*, A.F.Laine and M.Unser, editors, *Proc. SPIE 2569*, p.p. 68-79, 1995.
4. Hoon Yoo and Jechang Jeong, A Unified Framework for Wavelet Transform Based on The Lifting Scheme, *Proc. of IEEE International Conference on Image Processing ICIP2001*, Tessaloniki, Greece, October 7-10, p.p.793-795, 2001.
5. Oleksiy Pogrebnyak, Pablo Manrique Ramirez "Adaptive wavelet transform for image compression applications" *Proc. SPIE #5203*, Applications of Digital Image Processing XXVI, Andrew G. Tescher Chair/Editor, 5- 8 August 2003, San Diego, USA. ISSN: 0277-786X, ISBN: 0-8194-5076-6.

Interpolation with Splines and FFT in Wave Signals

¹ Luis Pastor Sánchez Fernández

¹ Center for Computing Research. Mexico City, Mexico. lsanchez@cic.ipn.mx

Abstract. In the digital measurement systems, the sampling frequency is essential by their influence in the measurement's accuracy. Their effect is decisive when it is necessary to obtain the signal pick values. The methods to improve the quality of the acquired data are related with system operation in real time or if it is admitted to carry out samples taking and later on the data analysis.

In this paper, the relationship between the sampling frequency and measurement's maximum error is obtained to sinusoidal continuous signals. This relationship can be extended for signals where one or several fundamental harmonic are outstanding.

By means of an analytic procedure, the mathematical expressions are obtained and maximum error is determined for he picks values measurement. The data are processed by means of the Fast Fourier Transform (FFT) and it is also used, a cubic correction (interpolation of cubic splines).

1 Introduction

At present, the wide use of computerized means in measuring, processing and control systems requires the analysis of the main factors affecting the quality of the information acquired.

The methods used for reducing the effect of a relatively low sampling frequency, without the need to increase it in the data acquisition system, will depend on whether the signal samples require processing while being acquired, or if it is permitted to take some samples, and then process them.

This paper intends to be useful from the academic as well as from the investigating point of view. Its practical value is based on the following reasons:

1. It enables you to calculate, by means of mathematical expressions, the maximum error in computerized measurements of a signal harmonics due to the sampling frequency which influences in the total maximum error of measurements. Such error is also influenced by the sensor's accuracy, which you cannot act upon, and the quantification error of the analogical/digital converter. The latter may be reduced to negligible values with 12- and 16-bit A/D converters.
2. It is possible to choose a sampling frequency that does not represent a heavy burden for the data acquisition system, that is, just the one that is strictly necessary, since it can be programmed on an operative system that is not necessarily a real-time one like Windows or LINUX. This often occurs on using personal computers, which is a tendency for computerized measuring systems in laboratories [5]. This implies an efficient as well as a rational use of the PC's central processing unit. At this point, the main thing is to make little use of the PC's central processing unit; otherwise, it would

be necessary to use an additional hardware, such as an intelligent data acquisition board (with its own processor and operative system) or devices for setting up a distributed system, which makes it more expensive and complex. It is important to properly plan and program such system, for instance, for every variable to be measured, or for choosing a correct sampling frequency that is neither low nor extremely high. In many applications it is useful to "suitably" reduce the sampling frequency in order to acquire, save, and transmit the information and then interpolate in the receiver. The latter would mean that an equivalent and higher sampling frequency has been used [7].

Shannon's sampling theorem [1], [9] establishes the minimum angular sampling frequency (W_s) to reconstruct a continuous $X(t)$ signal based on the samples taken in T time periods, denoting $X^*(t)$ as the $X(t)$ digitalized signal.

Considering that:

$$W_s = 2\pi / T$$

Where:

W_s : sampling frequency in radians per second.

T : sampling period in seconds.

For a sinusoidal signal:

$$X(t) = A \sin(w t)$$

being w the angular frequency of the $X(t)$ signal. If W_s is greater than or equal as $2w$, then the $X(t)$ continuous signal in time can be reconstructed on the basis of the samples taken by a digital device, just as a data acquisition system using an analogical/digital converter and a PC, as shown in Fig.1.

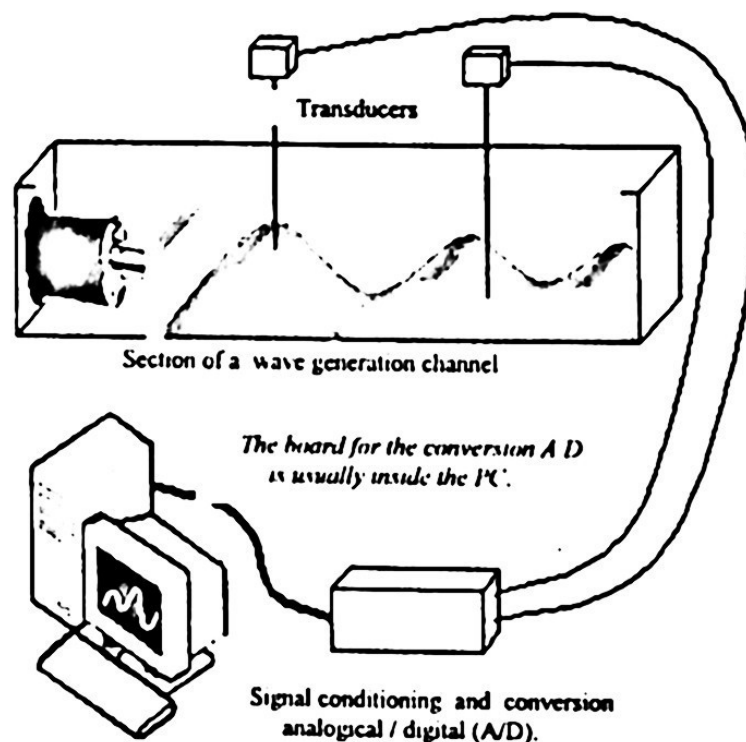


Fig.1. Diagram for measuring waves

For a non-sinusoidal signal, made up of many harmonics, as in the case of an irregular wave signal, the sampling frequency would have to be higher than or the same as twice the highest interest frequency in the spectrum of the signal being measured through its samples, acquired in T periods of time. Being able to reconstruct the continuous signal through its samples does not guarantee the accuracy of this reconstruction. Practical criteria related with the types of applications have been used. For instance, for control systems [1] the sampling frequency should be based on the knowledge of its influence on its own operation. Thus it is reasonable to consider that the highest interest frequency must be closely related to the bandwidth of the closed-loop control system. Therefore, the choice of such sampling frequency must be based on the bandwidth or the rising time of the closed-loop control system. It is adequate to take it between 10 and 30 times greater than the bandwidth or choose the sampling period between 4 and 10 times smaller than the rising time, all of which can be small in relation with the criteria to be followed in the typical applications of signal processing [8], [10]. A relatively low sampling frequency in control systems is caused by the fact that its dynamics has low-pass filter characteristics and its typical-time constants are much greater than the closed-loop response time.

Taking into account the primary processing operations set up to improve the accuracy and the destination of the information obtained, there are other practical criteria, such as considering the peak frequency (F_p) of the signal spectrum and taking the sampling frequency (F_s) in accordance with the following relationship:

$F_s \geq 8F_p$ this criterion is recommended for measuring and irregular-wave generation systems in research labs [2], or for taking the sampling period between one tenth or one twentieth of the significant-wave period [4].

Here, F_s has been used to denote the sampling frequency in Hertz (Hz), equivalent to cycles per second or samples per second.

All cases have been based on practical criteria guaranteeing a suitable accuracy depending on the applications, but no mathematical relationships liable to accurately establish the maximum error that might be produced due to the sampling frequency has been stated. This is important when it is necessary to calculate the error of a measuring system which is influenced by all the elements involved, from the continuous-variable sensor up to the analogical/digital converter and the frequency with which the samples are being taken, as shown in Fig.1.

2 Relationship between Sampling Frequency and Error

In digital measuring systems, apart from the quantification error due to the analogical/digital conversion, an error occurs due to the sampling frequency. This error can become extremely serious and should be considered when obtaining the maximum error presented in the measuring.

Next, it is presented an analysis in which a sinusoidal signal is used as an entry, just as it might happen for the generation of regular (sinusoidal) waves, or when it can be considered that there is a fundamental harmonic in the spectrum of the irregular waves. The maximum error of the readings in relation to the sampling frequency when

using a zero order hold is determined. This is equivalent to the fact that a sample of the signal keeps its validity right up to the time when the next sample is taken. This maximum error is given by the following expression [1]:

$$E_{\max abs} = \text{Max} |x(k+1) - x(k)| \quad (1)$$

Being:

$E_{\max abs}$: absolute maximum error of readings.

$x(k)$, $x(k+1)$: values of the signal in $t = kT$ and $t = (k+1)T$, respectively.

T : sampling period (time in-between the taking of the samples of the continuous signal).

k : integer value (1,2,3,...).

From expression (1) it can be interpreted that value $X(k)$ will be the representative value of the signal until the next sample is taken $X(k+1)$. That is why the $\text{Max}|x(k+1) - x(k)|$ can be considered as the absolute maximum error in real time (for which there only is information at the moment of the current sampling and at previous moments) and it is caused by the sampling frequency of the signal when using a zero order hold.

The maximum error is located symmetrically to the origin of the axis of the coordinates where the highest speed of signal change takes place, as shown in Fig. 2, being $E_{\max abs}$ the absolute maximum error.

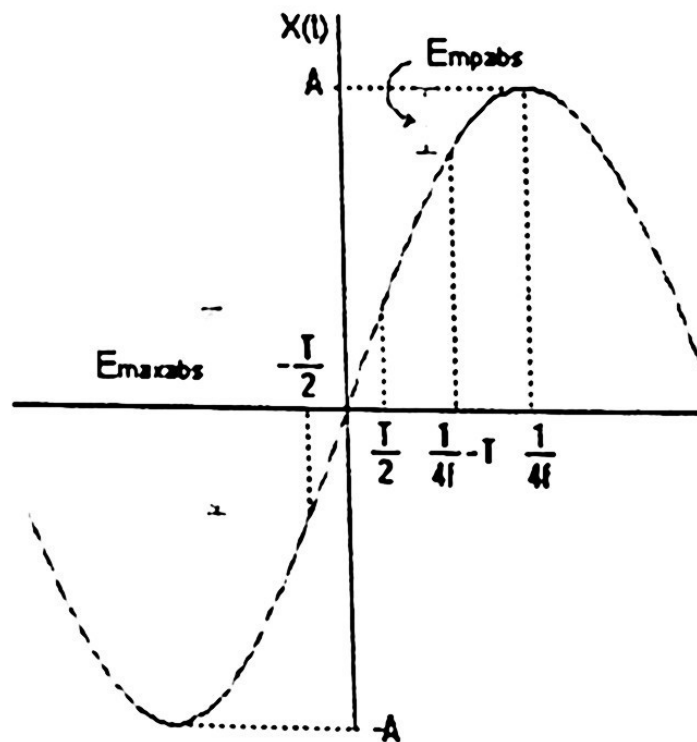


Fig. 2. Absolute maximum error in real time (causal processing) for a sinusoidal signal

E_{mpabs} is the absolute maximum error produced around the peak values in real time or causal processing. From Fig. 2 it can be stated that:

$$X(k+1) = A \sin(2\pi f T/2) \quad (2)$$

$$X(k) = A \sin[2\pi f(-T/2)] \quad (3)$$

Hence:

$$E_{maxabs} = |A \sin(2\pi f T/2) - A \sin[2\pi f(-T/2)]|$$

$$E_{maxabs} = |2A \sin(2\pi f T/2)|$$

$$E_{maxabs} = 2A |\sin(\pi f T)| \quad (4)$$

Being: $T = 1/F_s$

F_s : sampling frequency. On taking $F_s = nf$, where f is the signal frequency and n an integer greater or equal to two, it is obtained that:

$$E_{maxabs} = 2A \sin(\pi/n) \quad (5)$$

The maximum relative error, regarding the peak-to-peak value of the signal, expressed in % will be:

$$E_{maxrel} = \frac{2A \sin(\pi/n)}{2A} \times 100 \quad (6)$$

$$E_{maxrel} = \sin(\pi/n) \times 100 \quad (7)$$

Next, an analysis is presented for formulating the expressions of the maximum error produced around the peak values in causal processing (E_{mpabs}).

The following equation is obtained:

$$X(k+1) = A \sin(2\pi f 1/4f) = A \sin(\pi/2) = A \quad (8)$$

$$X(k) = A \sin[2\pi f(1/4f - T)] \quad (9)$$

$$X(k) = A \sin \left[2\pi f \left(\frac{1}{4f} - \frac{1}{nf} \right) \right]$$

$$X(k) = A \sin \left[\frac{\pi}{2} \left(1 - \frac{4}{n} \right) \right]$$

The sampling period can be written as:

$$T = \frac{1}{nf} \quad (10)$$

It is replaced in the previous expression, thus obtaining:

$$X(k) = A \sin \left[\frac{\pi}{2} \left(1 - \frac{4}{n} \right) \right] \quad (11)$$

$$X(k) = A \sin \left(\frac{\pi}{2} - \frac{2\pi}{n} \right) = A \cos \left(\frac{2\pi}{n} \right) \quad (12)$$

Finally:

$$E_{mpabs} = A - A \cos \frac{\pi p \delta}{n \delta} \quad (13)$$

and the relative maximum error, regarding the peak-to-peak value of the signal, expressed in % will be:

$$E_{mprel} = \frac{A - A \cos \frac{\pi p \delta}{n \delta}}{2A} \times 100 \quad (14)$$

$$E_{mprel} = 0.5 \left[1 - \cos \left(\frac{2\pi}{n} \right) \right] \times 100 \quad (15)$$

In many applications, as in waves analysis, for instance, it is not necessary to make most calculations in real time. Instead, they are to be made on the basis of data previously acquired and determine the maximum and minimum values of the signal, which could be considered as sinusoidal or formed by a main harmonic and other secondary harmonics of small amplitudes. Therefore, in that case, the maximum error that can occur in determining such values is presented in Figure 3 and is called E_{mpabs} as the absolute static maximum error on determining peak values. This is an error in *non-causal processing*, since there is a register of previously acquired information, and then, there is information in previous and subsequent sampling moments for a sampling moment considered as current (k).

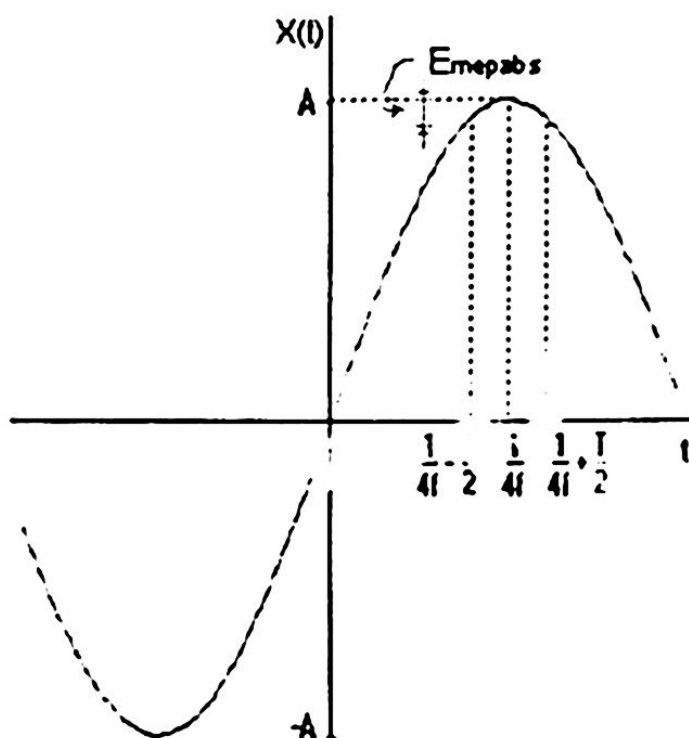


Fig. 3. Absolute static maximum error on determining peak values (non-causal processing)

This error occurs when none of the values of the samples taken coincides with the value of the signal peak. This error will be the maximum when the arrangement of the samples is the one presented in Figure 3; both samples being equidistant from the peak value.

$$E_{\text{max}} = |X(k+0.5) - X(k)| \quad (16)$$

Where:

$$X(k+0.5) = A \sin(2\pi f l / 4f) \quad (17)$$

$$X(k+0.5) = A \sin(\pi/2) = A \quad (18)$$

$$X(k) = A \sin[2\pi f l / 4f - T/2] \quad (19)$$

$X(k) = A \sin\left[\frac{2\pi f}{f_s} \left(\frac{1}{2} - \frac{2fT}{4f} \frac{\delta}{\Delta t}\right)\right] = A \sin\left[\frac{\pi}{2} \left(1 - 2fT\right) \frac{\delta}{\Delta t}\right]$ Replacing $T = \frac{1}{F_s} = \frac{1}{nf}$; then, the expression is:

$$X(k) = A \sin\left[\frac{\pi}{2} \left(1 - 2f \frac{1}{nf} \frac{\delta}{\Delta t}\right)\right] \quad (20)$$

$$X(k) = A \sin\left[\frac{\pi}{2} - \frac{\pi \delta}{n \Delta t}\right] = A \cos\left[\frac{\pi \delta}{n \Delta t}\right] \quad (21)$$

finally, the expression is obtained as:

$$E_{\text{max}} = A - A \cos\left[\frac{\pi \delta}{n \Delta t}\right] \quad (22)$$

and the relative maximum error, regarding the peak-to-peak value of the signal, expressed in % will be:

$$E_{\text{maxrel}} = \frac{A - A \cos\left[\frac{\pi \delta}{n \Delta t}\right]}{2A} \times 100 \quad (23)$$

$$E_{\text{maxrel}} = 0.5 \left[1 - \cos\left[\frac{\pi \delta}{n \Delta t}\right]\right] \times 100 \quad (24)$$

By evaluating expressions (7), (15), and (24), for sampling frequencies, being n times the frequency of the sinusoidal signal, the results are presented in Table 1.

Table 1. Comparison of relative errors according to the sampling frequency

n	E_{maxrel} (%)	E_{mprel} (%)	E_{meprel} (%)
2	100	100	50
4	70.71	50	14.64
8	38.26	14.64	3.8
10	30.9	9.54	2.44
20	15.64	2.44	0.61
50	6.27	0.39	0.098
100	3.14	0.098	0.024
200	1.57	0.024	0.0061
300	1.04	0.0109	0.0027

3 Reducing Error Due to a Low Sampling Frequency.

Several methods can be applied in order to recover the continuous signal from the samples, thus improving the accuracy of their reconstruction without needing to raise its sampling frequency considerably. There are various methods, such as:

1. For non-causal processing, in the series of sampled values, the method can interpolate points that are compatible with the complex vector of such series for which the FFT method [13] is used. For instance, when the peak values of the waves are calculated in maritime hydraulics analysis [12].
2. Interpolation with cubic splines [3], [5], [13], for non-causal processing as well.
3. One order hold or higher, for causal processing, which is frequent in real time applications [1].

3.1 Effect of Interpolation using the FFT Method.

The analysis of the effect of the interpolation of points in the temporary series is carried out. These points are compatible with their complex vector in the domain of the frequency. With this method, points are added to a waves register, thus obtaining an effect close to the one that would occur provided it had been sampled at a higher frequency. The number of points to be added can be chosen, calling it filling factor (F_f) and defining it as follows:

$$F_f = \frac{\text{NPFS}}{\text{NPOS}}$$

Where:

NPFS: number of points of the filled series; NPOS: number of points of the original series. F_f : It must be base power 2 in order to apply Fourier transform algorithm.

Fig. 4 shows part of the water level graph against time for the wave signal. It shows the original series sampled at 3 Hz and the series filled with factor $F_f = 8$.

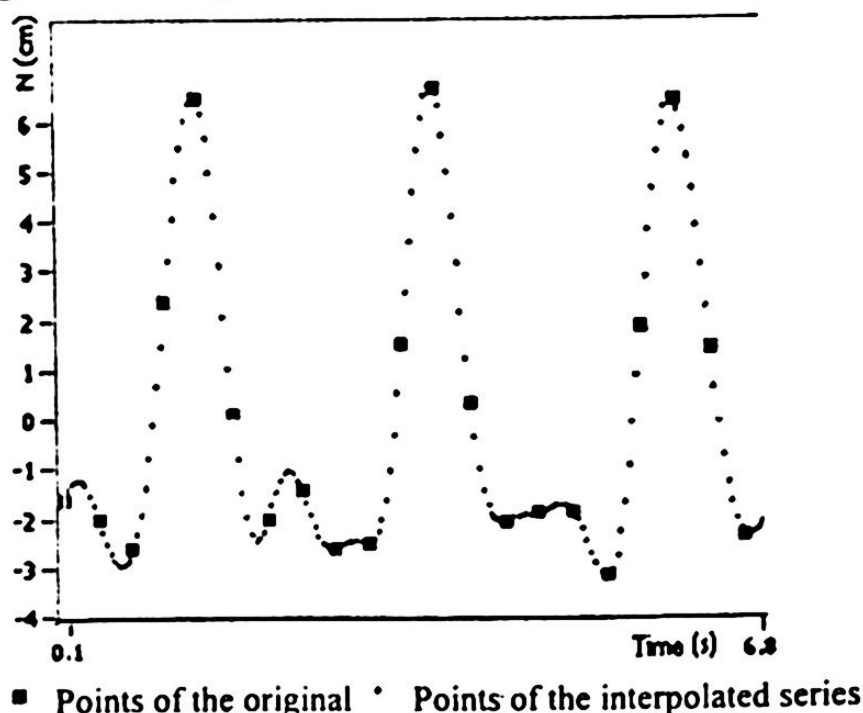


Fig. 4. Example of interpolation with FFT

The original series has been reconstructed with the application of this method, thus counteracting the distortion caused by a low sampling frequency to a great extent. It can be noticed that adding points between the sampled values has made up the signal peaks. Such points keep the compatibility with the complex vector of the original series (in this case, 7 points are added between every two samples).

The waves statistical values are calculated with the interpolated series and the mean and variance of the relative errors are found in respect to the signal sampled at 48 Hz. The summary is shown in Table 2.

It was given the name of interpolated series 1 to those in which 2, 4, 8 and 16 filling factors (F_f) were used for signals sampled with 24, 12, 6 and 3 Hz respectively. For the four signals, the total number of points of the interpolated signals amounted to 4096, which means that the original series sampled at 24 Hz had 2048 points and that the interpolation has only added an intermediate point (between two samples). With this, the series cannot be remade properly. For this signal, the interpolation did not cause a reduction of errors. Errors decrease 4.4, 2.3 and 1.4 times for signals sampled at 12, 6 and 3 Hz, respectively.

Table 2. Summary of the mean and variance of relative errors in %

	24 Hz	12 Hz	6 Hz	3 Hz
Original series	0.48 ± 0.08	1.64 ± 1.52	2.61 ± 3.63	3.1 ± 9.02
Interpolated Series 1	$(F_f = 2)$ 0.48 ± 0.08 <i>There is not improvement</i>	$(F_f = 4)$ 0.37 ± 0.13 <i>It improve 4.4 times</i>	$(F_f = 8)$ 1.16 ± 0.52 <i>It improve 2.3 times</i>	$(F_f = 16)$ 2.28 ± 2.49 <i>It improve 1.4 times</i>
Interpolated series 2	$(F_f = 4)$ 0.04 ± 0.002 <i>It improve 11 times</i>	$(F_f = 8)$ 0.03 ± 0.001 <i>It improve 55 times</i>	$(F_f = 16)$ 0.97 ± 0.76 <i>It improve 2.8 times</i>	$(F_f = 32)$ 1.92 ± 3.09 <i>It improve 1.6 times</i>

It was given the name of interpolated series 2 to those using 4, 8, 16 and 32 (F_f) factors for the signals sampled at 24, 12, 6 and 3 Hz, respectively. The total number of points of the interpolated series amounted to 8192. For 24 Hz, the interpolation added three intermediate points. 7, 15 and 31 intermediate points were added at 12, 6 and 3 Hz, respectively. In the first two cases (sampling at 24 and 12 Hz, respectively) the improvements are significant, with a decrease of errors between 11 and 55 times and more modest results for the last two signals, with very low sampling frequencies of 6 and 3 Hz, respectively.

It can be noticed that once the series sampled at 24 Hz and 12 Hz are interpolated, the mean and the variance of relative errors are similar in both cases, and that, for this example, the interpolation makes it unnecessary to sample the continuous signal at 24 Hz, since it is enough to do it at 12 Hz and interpolate with a $F_f = 8$ filling factor. Slightly smaller mean-and-variance values for 12 Hz can be observed. It is considered that such behavior is determined by fortuitous factors that mainly depend on the way in which the sampled points are arranged with respect to the continuous signal in time, which, as it is irregular, makes such arrangement unpredictable. In

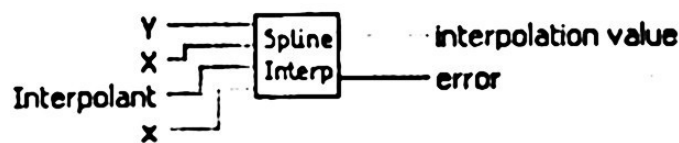
tests made elsewhere and not shown in this paper, when adding the same number of intermediate points in both series, the results obtained keep their characteristics.

In all the tests made, the number of waves registered (35 waves) has been kept constant so that only the sampling frequency exerts its influence on the errors of statistical calculations.

3.2 Interpolation with cubic splines (spline interpolation)

For making this interpolation, the mathematics functions of the LabVIEW are used [5]. They are also available in the MATLAB.

According to the notation in the LabVIEW, the following diagram is used:



Spline Interpolation.vi

Where:

Y: arrangement of values to be interpolated.

X: arrangement of the values of axis x ; in this case it will be time, which will increase during the sampling period in which the data of arrangement Y were acquired.

x: (small case) value of axis x for which an interpolated point is desired. This value will increase in the new sampling period resulting from interpolation, will be smaller than the original sampling period and will be within the range of values of arrangement X .

Interpolant: second derivative of the cubic spline interpolation function which is calculated by means of another virtual instrument (VI), called *spline interpolant.vi*.

error: It returns a value that indicates whether the execution of the function has been successful or not.

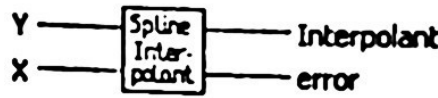
The output value of interpolation z (interpolation value) in interval $[x_i, x_i + 1]$ is given by:

$$z = Ay_i + By_i + 1 + Cy_i^2 + Dy_i^3 + 1;$$

Where:

$$A = \frac{x_{i+1} - x}{x_{i+1} - x_i}, B = 1 - A; C = \frac{1}{6}(A^3 - A)(x_{i+1} - x_i)^2; D = \frac{1}{6}(B^3 - B)(x_{i+1} - x_i)^2$$

The interpolant value can be obtained with another function (virtual instrument: Spline Interpolant.vi), having registers Z , Y as inputs.



Spline Interpolant.vi

Interpolation function $g(x)$ passes through all points: (x_i, y_i) .

$$y_i = g(x_i); \text{ where } i = 0, 1, \dots, n-1.$$

The spline Interpolant.vi virtual instrument obtains the interpolation function $g(x)$ by interpolating at every $[X_i, X_{i+1}]$ interval with a $P_i(x)$ cubic polynomic function having the following conditions:

1. $p_i(x_i) = y_i$
2. $p_i(x_{i+1}) = y_{i+1}$
3. $g(x)$ has the first and second derivative in interval $[X_0, X_{n-1}]$ continuous. Thus:
 - a) $p_i'(x_i) = p_{i+1}'(x_i)$
 - b) $p_i''(x_i) = p_{i+1}''(x_i)$

For $i = 0, 1, \dots, n-2$.

The following equations are derived from the last condition:

$$\frac{x_i - x_{i-1}}{6} g''(x_{i-1}) + \frac{x_{i+1} - x_{i-1}}{3} g''(x_i) + \frac{x_{i+1} - x_i}{6} g''(x_{i+1}) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}$$

Where $i = 1, 2, \dots, n-2$ and $n-2$ equations are obtained with unknown n $g''(x_i)$, for $i = 0, 1, \dots, n-1$. This virtual instrument (spline Interpolant.vi) calculates $g''(x_0)$, $g''(x_{n-1})$ using the following formula:

$$g'(x) = \frac{y_{i+1} - y_i}{x_{i+1} - x_i} + \frac{3A^2 - 1}{6} (x_{i+1} - x_i) g''(x_i) + \frac{3B^2 - 1}{6} (x_{i+1} - x_i) g''(x_{i+1})$$

Where:

$$A = \frac{x_{i+1} - x}{x_{i+1} - x_i}; B = 1 - A = \frac{x - x_i}{x_{i+1} - x_i}$$

This VI uses $g''(x_0), g''(x_{n-1})$ to solve all the $g''(x_i)$, for $i = 1, \dots, n-2$; $g'(x_i)$ is the Interpolant output, which is used as an input in VI *spline Interpolation.vi*. Fig. 5 and 6 show two sections of a series interpolated with cubic splines. It also shows the original series and the series interpolated by means of the FFT (complex interpolation) with the goal of facilitating the comparison. Table 3 shows the correlation coefficients. The second column (24Hz and 24Hz_FFT) is the correlation

coefficient between the signal sampled at 24 Hz and the interpolated one by using Fourier Fast Transform (FFT) in order to obtain a 24 Hz-equivalent sampling frequency, similarly for Columns 3 and 4. In both interpolations, the original series had been sampled at 3 Hz. Due to that, 7 points were inserted between every two samples to obtain a 24 Hz-equivalent sampling frequency. In the figures presented, you can see that the complex interpolation (FFT) achieves a relatively better reconstruction for this type of wave signal, which has harmonics with small amplitudes and higher frequencies.

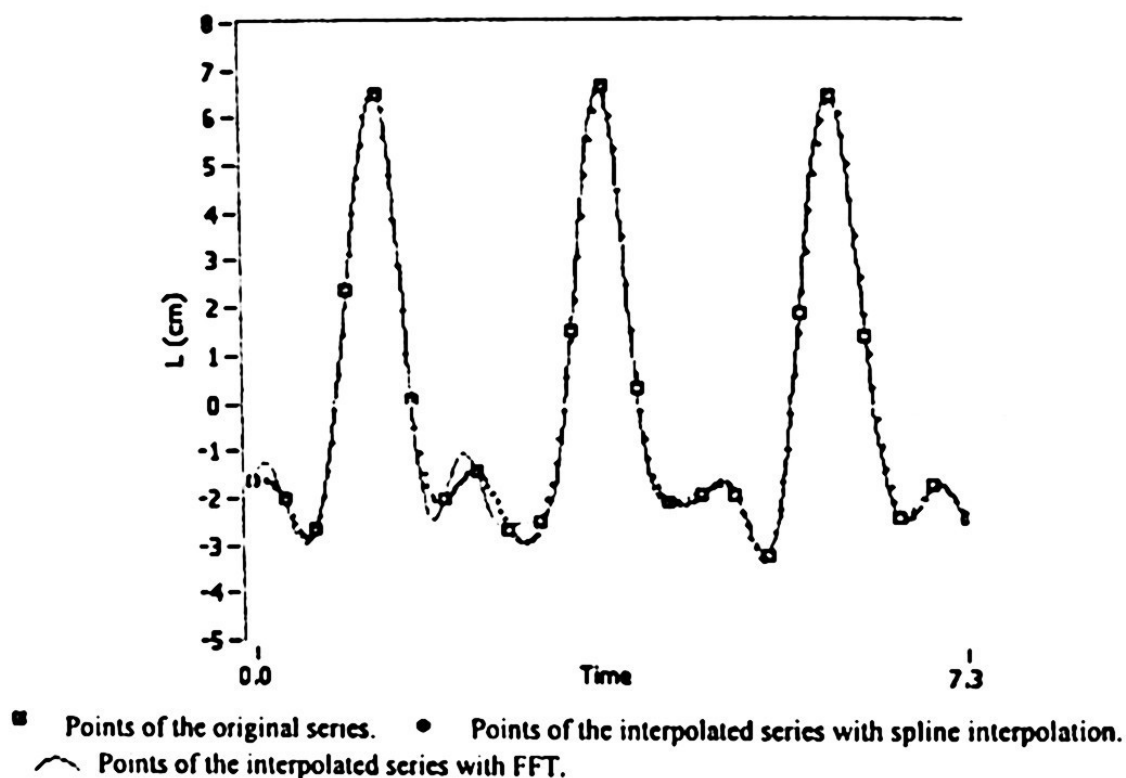


Fig. 5. Interpolation with cubic splines (a section of the series)

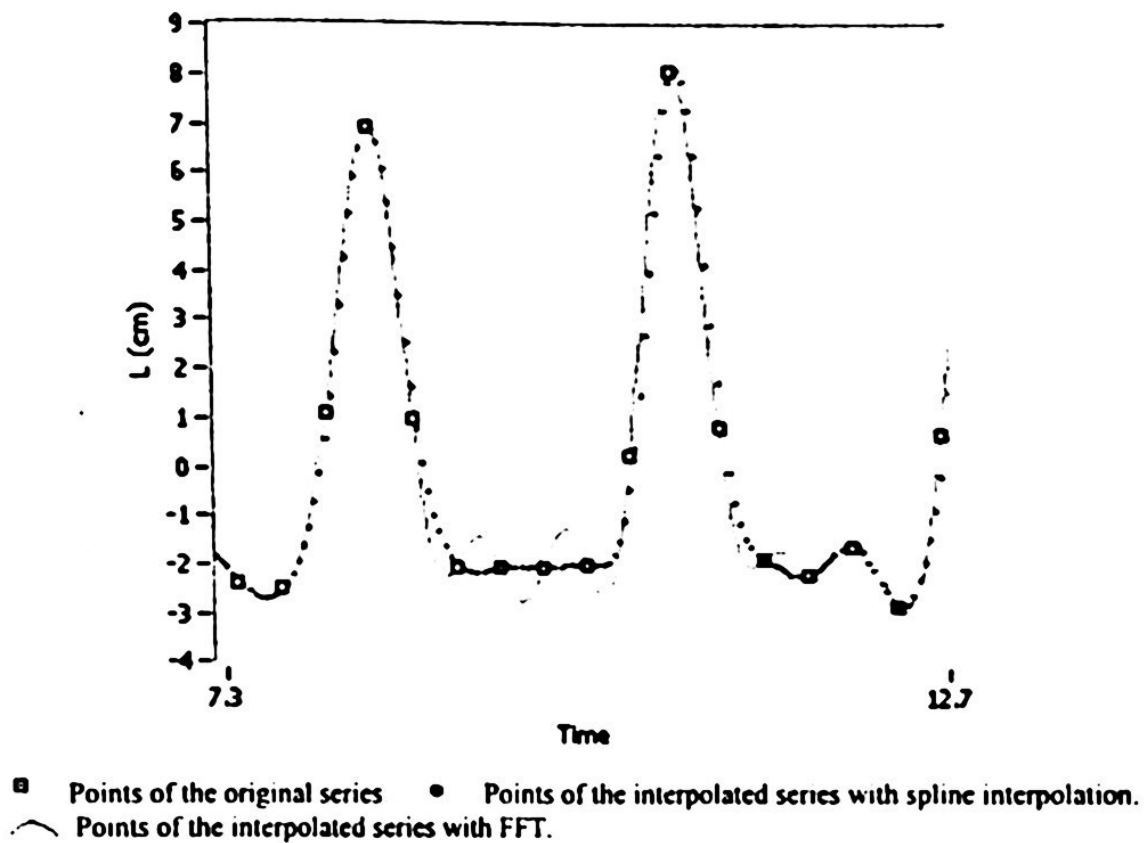


Fig. 6. Interpolation with cubic splines (a second section of the series)

Table 3. Correlation coefficients

	24Hz and 24Hz_FFT	24Hz and 24Hz_Spline	24Hz_FFT and 24Hz_Spline
Coeff. of correlation	0.9621	0.6899	0.8011

4 Conclusions

Expressions (7), (15) and (24) are obtained. They make it possible to calculate the relative maximum error due to the sampling frequency (F_s) for a continuous and sinusoidal signal. This can be applied to signals where one or more fundamental harmonics are relevant; for instance, for a regular-wave signal or for the fundamental harmonic of the irregular waves. This error can be considered an indicator of accuracy due to the F_s , when it has to do with the complete reconstruction of the continuous signal in time $X(t)$ from the samples taken, using a ladder-type reconstruction or a zero order hold. E_{maxrel} is the accuracy in causal processing (ordinary in real time applications) and can be interpreted as the maximum delay in perceiving the real value of the signal. Likewise, E_{mprel} would be the maximum delay, but in values close to the signal peaks. However, E_{meprel} represents the relative maximum error that can occur, in non-causal processing, when you want to calculate the peak values of the signal.

For waves-generation channels with similar characteristics to the one studied, the highest interest frequency in its spectrum is lower than 2 Hz. Therefore, a minimum F_s ,

between 18 and 20 Hz (samples per second) can be taken, which is equivalent to about 10 samples for each cycle (or wavelength) of the highest-frequency harmonic. Interpolation with cubic splines, widely mentioned and well accepted by many authors and available in signal-processing commercial software, reconstructs the series in the same way as when the Fourier Fast Transform is used. However, with the latter you can achieve a more accurate reconstruction for signals of irregular waves or similar characteristics. This is compared graphically and using the correlation coefficients.

The number of waves registered has been kept constant so that only the F_s exerts its influence on the errors from statistical calculations. The greater the number of waves (many authors recommend about 120), the higher the probability of reducing error in statistical calculations, since the latter are based on the use of the largest waves, which can be detected with a greater degree of probability when a high number of samples is taken.

The results obtained are useful in that they make it possible to assess the contribution of the F_s to the total error of the measuring system (from the transducer to the PC) and to compare two methods widely used in non-causal processing and that can easily be used with any commercial software. Their limitations are related to the use of a sinusoidal signal for making the analysis.

References

1. Åström, K.J., & Wittenmark, B. (1990). *Computer Controlled Systems, Theory and Design. Second Edition*, Englewood Cliffs, Prentice-Hall Inc., New Jersey, USA, 1990, pp. 544.
2. Beresford P. J. (1994). *WAVEGEN-Wave Generator Control Software Program, User Manual*, Howbery Park, Wallingford, Oxfordshire, England, 1994, pp. 130.
3. Emery, J., & Thomson, R. (2001). *Data Analysis Methods in Physical Oceanography*. Elsevier, Amsterdam, Netherlands, 2001, pp. 638
4. Goda, Y. (2000). *Random Seas And Design of Maritime Structures. Advanced Series on Ocean Engineering-Volume 15*. World Scientific Publishing Co. Pte. Ltd. Singapore. 2000, pp. 443.
5. National Instruments (2002). *LabVIEW, a Graphical Programming Language to Create Applications, version 6.1*, 2002.
6. www.ni.com/labview.
7. Oppenheim, A.V., et al. (1996). *Signals and Systems*, 2nd Edition, Prentice-Hall, NJ, 1996, pp. 957.
8. Oppenheim, A.V., Schaffer, R.W., & Buck, J.R. (1999). *Discrete-Time Signal Processing. 2nd Edition*, Prentice-Hall International Editions, 1999, 870 pages.
9. Ogata, K. (2001). *Modern Control Engineering, 4th Edition*, Prentice Hall, NY, 2001, pp. 970.
10. Proakis, J.G., & Manolakis, D.G. (1995). *Digital Signal Processing: Principles, Algorithms and Applications, 3rd Edition*, Prentice-Hall, NJ, 1995, pp. 1016.
11. Rosengaus, M.M. (1988). *Experimental study of the generation of constant surge and its attenuation, Thesis of Doctor*, Massachusetts Institute of Technology, USA, 1988.
12. Stansell, P., Wolfram, J. & Linfoot, B. (2002). "Effect of Sampling Rate on Wave Height Statistics", *Ocean Engineering*, vol. 29, núm. 9, Oxford, 2002, UK, pp. 1023-1047.
13. MathWorks, (2003). *One-dimensional interpolation using the FFT method*, july, 2003.
14. <http://www.mathworks.es/access/helpdesk/help/techdoc/ref/ref.shtml>

Digital Systems Design

An Efficient FPGA Architecture for Block Ciphering in Third Generation Cellular Networks

Tomás Balderas-Contreras and René A. Cumplido-Parra

Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1. Postal Code: 72840
Tonantzintla, Puebla, MEXICO
{balderas,rcumplido}@inaoep.mx

Abstract. Third generation cellular networks allow users to transmit information at high data rates, have access to IP-based networks deployed around the world, and implement sophisticated services. Not only is it necessary to develop new radio interface technologies and improve existing core networks to reach success, but guaranteeing confidentiality and integrity during transmission is a must. The KASUMI block cipher lies at the core of the integrity and confidentiality techniques designed for Universal Mobile Telecommunications System (UMTS) third generation networks. In turn, KASUMI implementations must reach high performance and have low power consumption in order to be adequate for network components. This paper describes a hardware architecture designed to perform the KASUMI algorithm efficiently, shows the convenience of taking advantage of the features present in FPGAs, and highlights a technique that might be used to design small reuse-based architectures implementing Feistel-like ciphering algorithms.

Keywords: 3G cellular networks, 3GPP, KASUMI, FPGA.

Resumen: Este artículo describe una implantación en hardware del algoritmo de cifrado de bloques KASUMI, el cual es la piedra angular de las operaciones de confidencialidad e integridad en redes celulares de tercera generación de tipo UMTS (Universal Mobile Telecommunications System). El diseño propuesto explota el principio de reutilización de componentes y las características presentes en FPGAs modernos. Además, la arquitectura resulta ser un buen balance entre alto desempeño y bajo consumo de recursos de hardware.

1 Introduction

The efforts towards the standardization of a block ciphering algorithm to be used in UMTS networks produced the specification of the KASUMI algorithm by the Third Generation Partnership Program (3GPP) [1]. KASUMI has a Feistel structure and operates on 64-bit data blocks under control of a 128-bit encryption key K [2]. KASUMI has the following features, as a consequence of its Feistel structure:

- Input plaintext is the input to the first round.
- Ciphertext is the last round's output.
- K is used to generate a set of round keys $\{KL_i, KO_i, KI_i\}$ for each round i .

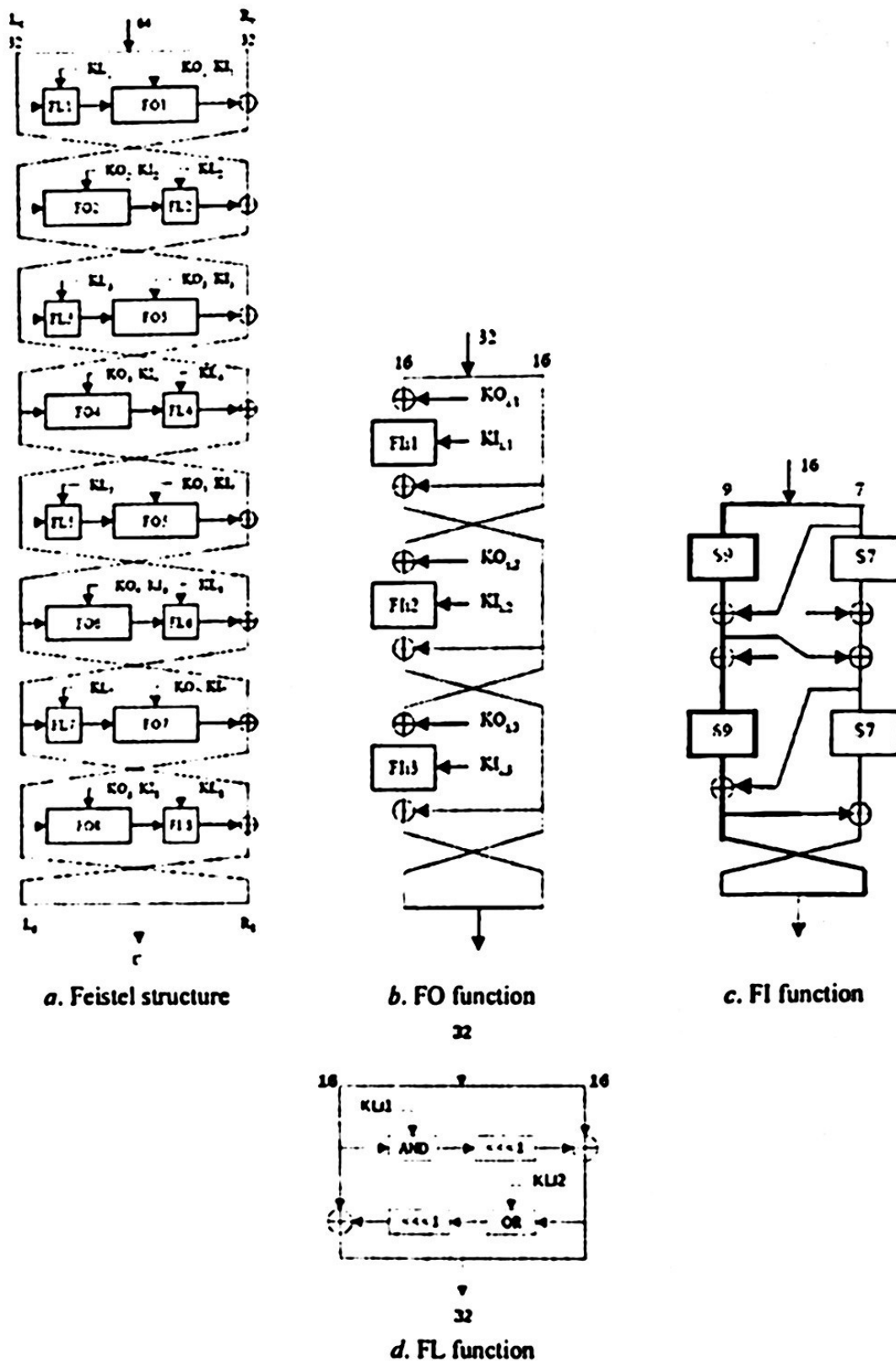


Fig. 1. The KASUMI block cipher

- Each round computes a different function, as long as the round keys are different.
- The same algorithm is used both for encryption and decryption.

KASUMI is based on a previous block cipher called MISTY1 [5]. MISTY1 was chosen as the foundation for the 3GPP ciphering algorithm due to its proven security against the most advanced methods to break block ciphers, namely cryptanalysis techniques. In addition, MISTY1 was heavily optimized for hardware implementation.

As can be seen in Fig. 1, the KASUMI block cipher has a different setup for the functions within rounds. For odd rounds the round-function is computed by applying the FL function followed by the FO function. For even rounds FO is applied before FL. FL, shown in Fig. 1d, is a 32-bit function made up of simple AND, OR, XOR and left rotation operations. FO, depicted in Fig. 1b, is also a 32-bit function having a three-round Feistel organization which contains one FI block per round. FI, see Fig. 1c, is a non-linear 16-bit function having itself a four-round Feistel structure; it is made up of two nine-bit substitution boxes (S-boxes) and two seven-bit S-boxes. Fig. 1c shows that data in the FI function flow along two different paths: a nine-bit long path (thick lines) and a seven-bit path (thin lines). Notice that in Feistel structures, such as the ones used in this algorithm, each round's output is twisted before being applied as input to the following round. After completing eight rounds KASUMI produces a 64-bit long ciphertext block corresponding to the input plaintext block.

Several proposals have been published previously that use different approaches to implement KASUMI in hardware, ranging from reuse techniques, addition of internal registers to reduce critical path and pipelined designs. After analyzing these proposals one can confirm the fact that there is an important and unavoidable tradeoff between performance and complexity in terms of area. The goals of this paper are twofold. First, to reach a good balance between high performance and low complexity during the implementation of KASUMI by taking advantage of the resources present in modern FPGA and the software tools used to implement designs for these devices. Second, to present the methodology used to design a compact datapath that reuses its components again and again to carry out the encryption process. Similar techniques can be applied to other Feistel-like algorithms.

The rest of this document is organized as follows: section 2 describes the design principles of the proposed architecture; section 3 concerns the implementation phase and describes the platform used for this project, provides results from the synthesis process and compares this information with the information provided by other works; finally, section 4 concludes.

2 Design principles

2.1 Datapath for the FO function

The main principle to design for reuse is to specify an architecture consisting of some or all of the components that are needed to perform a round. This design is used every clock cycle in such a way that the output at the end of one cycle is the input for the next cycle. The fewer the components, the larger the number of cycles needed to carry

out the whole ciphering process for one block. Also, the fewer cycles the architecture requires to perform the process, the more complex the architecture is in terms of area.

The approach used in this paper is considered to be a good tradeoff between number of cycles and area complexity in the platform of choice. Instead of simplifying the algorithm at a lower level, at the FI level as in [6], the manipulation is carried out at a higher level, at the FO-function level. Fig. 2 depicts the process followed to design a datapath that reuses components within FO function. Fig. 2a illustrates an alternative parallel view of the FO function depicted in Fig. 1.

- In Fig. 2b two XOR gates are added to FO in order to make upper and lower sections more similar without modifying the behavior of the function. If these two parts were structurally the same, it would be possible to reduce the architecture to only one section, which carries out the whole FO function after two cycles.
- Lower section in Fig. 2b needs a right FI function block in order to be structurally identical to upper section in this modified FO block. In Fig. 2c an additional FI block is added to the lower section. The multiplexors in each section allow selecting the appropriate data flow.
- The whole datapath in Fig. 2c is now ready to be simplified. Fig. 2d shows the final design, which takes two cycles to complete the FO function. Notice that multiplexors are necessary in order to supply the correct values both to XOR gates and FI module depending on if the datapath is in the first or second cycle.

Also notice that the datapath depicted in Fig. 2d contains only one FI block, called dpFI, instead of two as in the previous diagrams. This situation is explained in more detail next because it constitutes another special feature of the design proposed in this document.

The control for this module is implemented as a finite state machine that sets the multiplexors' selector input properly each cycle. Since the design takes two cycles to complete its processing, it requires a two-state control.

2.2 Datapath for the FI function

Fig. 2c shows that the FO module requires two FI blocks to work properly. Since FI contains two seven-bit S-boxes and two nine-bit S-boxes, the simplified datapath that takes two cycles to complete the FO function requires a total number of eight S-boxes. Implementing S-boxes using four 128×7 ROMs and four 512×9 ROMs is such an expensive choice in terms of area to be considered as viable. The solution proposed is to map the S-boxes to dual-port ROMs, which decreases by two the number of ROMs required. The use of this technique exploits the principle of reuse even further because the same S-box is now able to meet two requests at the same time.

Consider two instances of the FI block depicted in Fig. 1c; next replace each pair of S9 S-boxes located in the same position in both FI blocks by a single dual-port S-box, and repeat this procedure with the rest of the pairs of S9 S-boxes and the pairs of S7 S-boxes. The result is the datapath illustrated in Fig. 3, which only contains two dual-port S9 S-boxes and two dual-port S7 S-boxes, and combines two FI functions into one. As before, the thick lines highlight the flow of nine-bit long signals and the thin lines highlight the flow of seven-bit long signals.

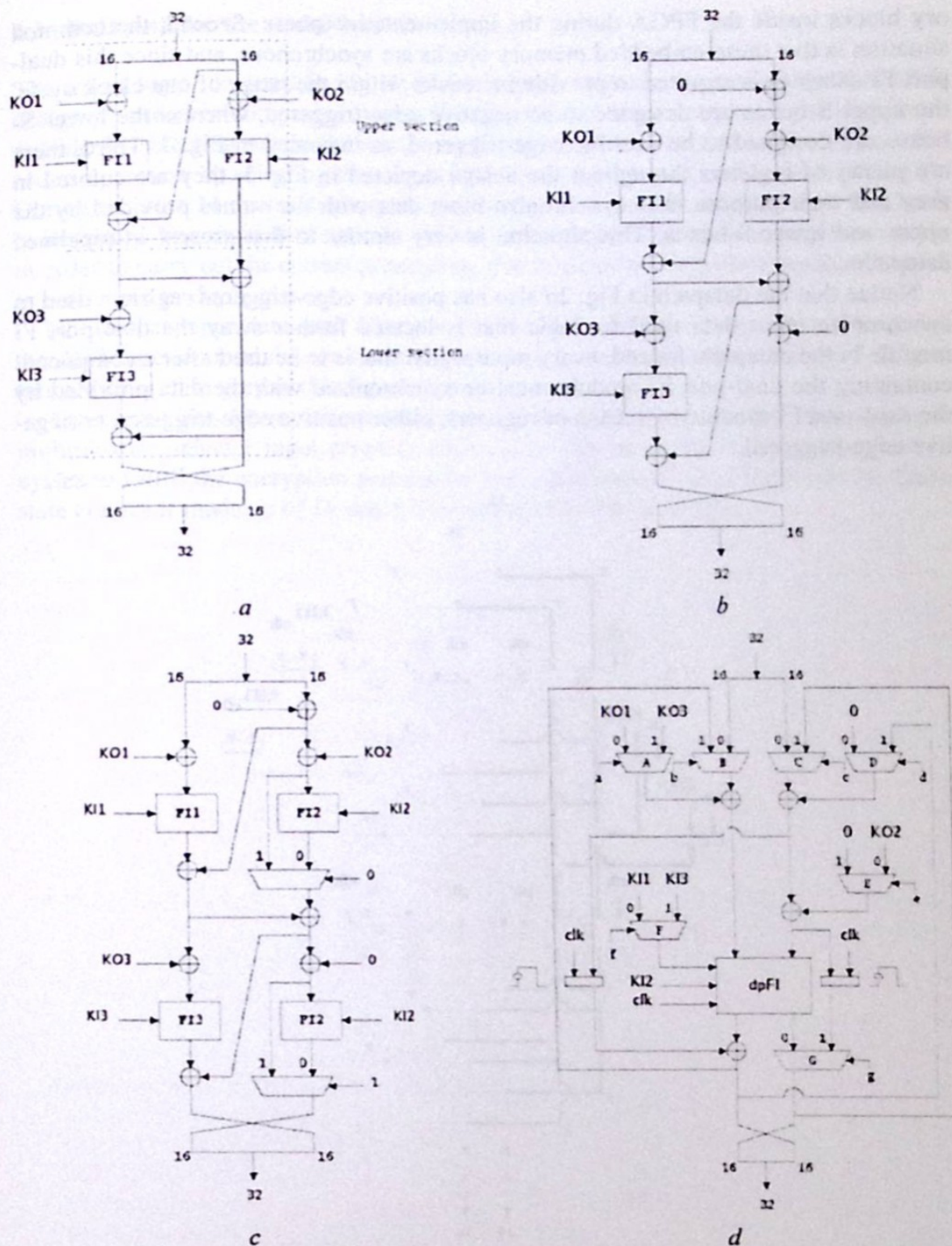


Fig. 2. Sequence of steps to design the datapath that reuses components in FO function

There are several notes to point out concerning this design. First, the four dual-port ROMs used to implement the S-boxes are intended to be mapped to embedded mem-

ory blocks inside the FPGA during the implementation phase. Second, the common situation is that these embedded memory blocks are synchronous, and since this dual-port FI datapath is required to provide its results within the range of one clock cycle, the upper S-boxes are designed to be negative edge-triggered, whereas the lower S-boxes are designed to be positive edge-triggered, as indicated in Fig. 3. Third, there are plenty of registers throughout the design depicted in Fig. 3, they are colored in grey and their purpose is to synchronize input data with the values provided by the upper and lower S-boxes. This situation is very similar to that present in pipelined datapaths.

Notice that the datapath in Fig. 2d also has positive edge-triggered registers used to synchronize input data used by logic that is located further away the dual-port FI module in the datapath. Indeed, every input signal that is to be used after a component containing the dual-port FI module must be synchronized with the data provided by the dual-port FI module by means of registers, either positive edge-triggered or negative edge-triggered.

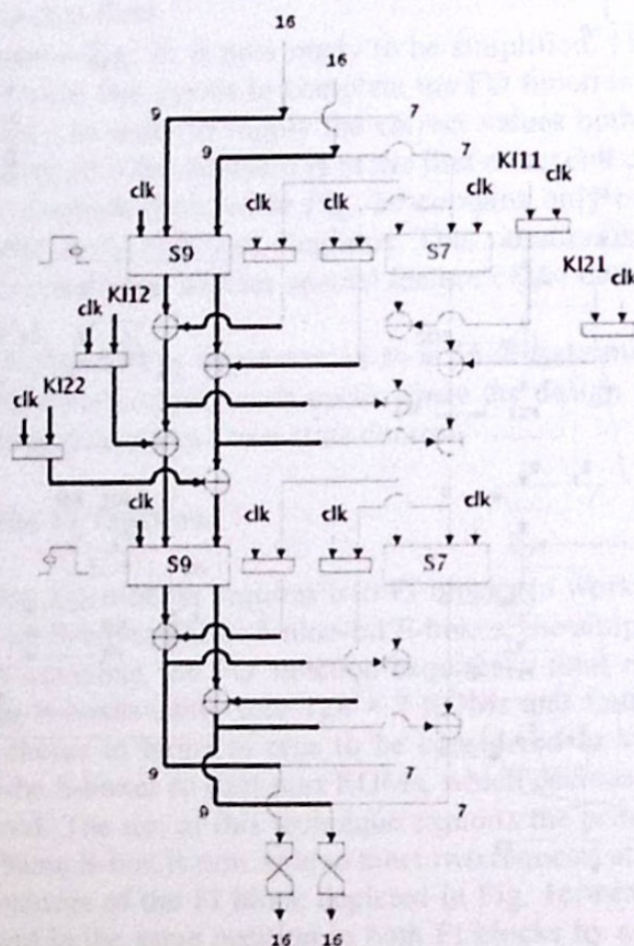


Fig. 3. Datapath implementing two merged FI function blocks

2.3 Datapath for the round logic

The round logic, depicted in Fig. 4, is the highest-level component of the KASUMI design proposed in this paper. During the first two cycles it receives input data from outside by selecting the zero input in multiplexors A and B, and performs an odd-round operation by selecting zero input both in multiplexor C and in multiplexor D. During the next 14 cycles the outputs yielded by the datapath each cycle are fed back to its inputs. The same data must be present at the datapath's inputs during two cycles in order to carry out the correct processing; that is why there is a third registered input in both input multiplexors.

Notice that input data used after the FO function module, which in turn contains the dpFI function module, are synchronized using registers as shown in Fig. 4. Registers are colored in grey in this figure as well.

The control for this module is implemented as a finite state machine that sets each multiplexor's selector input properly each cycle. The datapath in Fig. 4 requires 16 cycles to fulfill the encryption process for every plaintext block. Therefore, the finite state control is made up of 16 states and controls the four selectors.

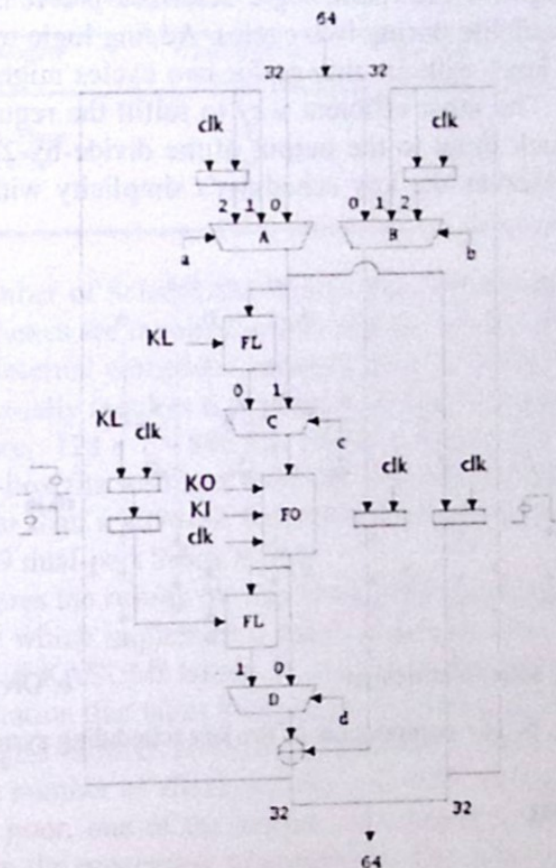


Fig. 4. Datapath for the round logic

2.4 The key scheduler

The key scheduler receives the 128-bit initial input key K and generates the round keys KL (32-bit long), KO (48-bit long) and KI (48-bit long) for each of the eight rounds. All of the previous figures depict how these round keys are used within each of the function modules. Each round key is split into two or three 16-bit parts, and these parts are the ones directly computed by the key scheduler. The input K key is split into eight 16-bit parts K_i , $1 \leq i \leq 8$, and then the scheduler performs left rotation operations (" \ll ") and computes the K_i' values, which are defined as follows:

$$K_i' = K_i \oplus C_i \quad 1 \leq i \leq 8; \quad (1)$$

where C_i is a constant specified in [2].

Fig. 5a illustrates the key scheduler component developed for this project, which adapts easily to different implementation schemes. For this design, the outputs are fed back to the inputs. The inputs are the initial key as an array of 16-bit values and the array of 16-bit constants. In addition to yielding the round keys in a combinational way, this component outputs its input arrays rotated to the left one position.

Notice that the design for the round logic described previously requires that each set of round keys is available during two cycles. Adding logic to the key scheduler to keep its output round keys without change for two cycles might result in a complex and expensive circuit. The most efficient way to fulfill the requirement is to connect the key scheduler's clock input to the output of the divide-by-2 clock divider in Fig. 5b. This technique preserves the key scheduler's simplicity without affecting the ciphering process.

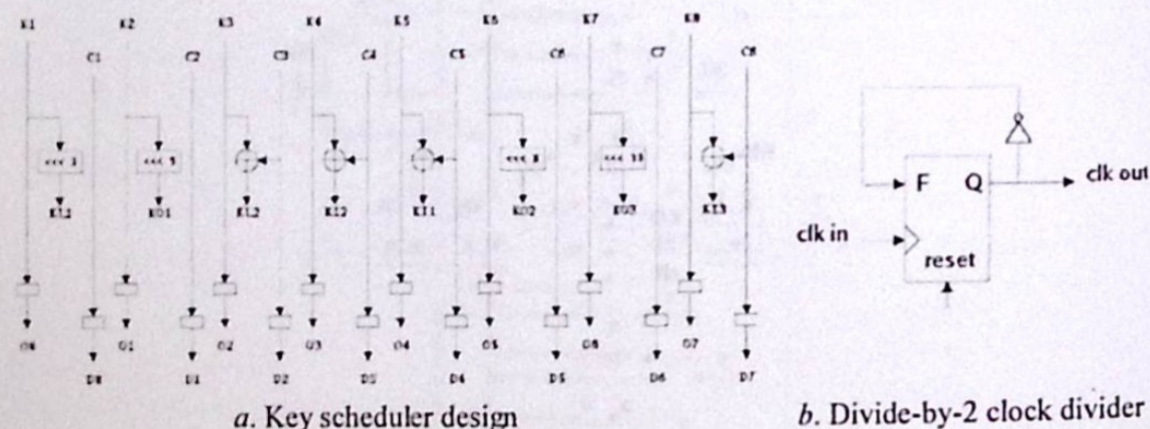


Fig. 5. The components of the key scheduling system

3 Implementation

The architecture is designed in VHDL and synthesized using the Xilinx Synthesis Technology (XST) software synthesis tools. The device of choice is the XCV300E-8-BG432 belonging to the Virtex-E family of devices [7]. The main reason for choosing this family is that the designs reported in the related papers are implemented on

Virtex-E devices, so in order to make a fair comparison the design reported here is implemented on a Virtex-E device as well.

Virtex-E devices contain large blocks of embedded SelectRAM memories. Each of these blocks is a fully synchronous dual-port (True Dual Port) 4096-bit RAM with independent control signals for each port. The data widths of the two ports can be configured in an independent fashion. The specific device considered in this project has 32 blocks of embedded SelectRAM, which provides a total number of 131072 bits of embedded memory.

3.1 Synthesis and results

Table 1 shows the results of the synthesis process concerning utilization of FPGA resources. It can be noticed that the percentage of resources used by the design in each category does not surpass 20%. By analyzing this information it is possible to conclude that the design is quite compact and fits well in any device without occupying a great number of resources.

Table 1. Synthesis results concerning area complexity

Category	Number of elements used	Total number of elements available	Percentage of use
Slices	488	3072	15%
Slice Flip Flops	566	6144	9%
4-input LUTs	898	6144	14%
BRAMs	6	32	18%
GCLKs	1	4	25%

Notice the number of SelectRAM blocks used by the design. Subsection 2.2 states that only four S-boxes are required in this design. Intuitively, these S-boxes would be mapped to four internal embedded memory blocks (SelectRAM), but table 1 shows that the design actually requires 6 of them. Consider the number of bits needed to implement each S-box: $128 \times 7 = 896$ bits for S7 S-boxes and $512 \times 9 = 4608$ bits for S9 S-boxes. A S7 S-box fits well in a 4096-bit SelectRAM block whereas a 4608-bit S9 S-box is far larger than a 4096-bit SelectRAM block, so two of them are required to implement one S9 dual-port S-box ROM.

Table 2 compares the results for this work with the results for other proposals published previously which implement a reuse approach. The work in [6] describes two implementations of KASUMI taking 56 and 32 cycles to cipher one block, results for a third implementation that takes 8 cycles are provided as well. Notice that the design that takes the largest number of cycles to process a block (56 cycles) is the one that requires the least number of slices and the one with the highest frequency. However, its throughput is poor, one of the lowest, as a consequence of the number of cycles needed to perform the processing of one block. The work reported in [4] takes 40 cycles to cipher one block, is the second most expensive in terms of area, due to it reuses two complete rounds, its throughput is comparable to the throughput of the slowest design in [6] and is reported to require 24 SelectRAM blocks to store the S-boxes. The work in [3] is a combination of two approaches: pipeline and reuse, it is the most expensive design in terms of area but one of the fastest implementations. It is able to

process four plaintext blocks at the same time, which increases performance dramatically.

Table 2. Comparison with other implementations

Proposal	Latency	Area (slices)	Frequency (MHz)	Throughput (Mbps)	HW efficiency (kbits/slice)	Number of S-boxes	Number of block RAMs
Work in [6]	56	368	68.13	77.86	211.58	2	N/A
	32	370	58.06	116.12	313.84	4	N/A
	8	588	33.14	265.12	450.88	12	N/A
Work in [4]	40	749	35.35	70.70	94.39	24	24
Work in [3]	pipeline	1100	33	234	212.73	12	N/A
This work	16	488	41.14	164.54	337.17	4	6

N/A = Not applicable

The design proposed in this paper features a novel strategy to deal with the reuse principle, is far cheaper than the two fastest options in terms of area, has one of the highest clock frequencies, and one of the highest throughputs. The main contribution of the design just described is that it achieves such a high performance with fewer hardware resources. The keys to achieve this good balance between speed and area complexity are the sharing of a high-level component (FO in this case), the use of dual-port SelectRAM blocks and a simplified design of the key scheduler.

4 Conclusions

This document presents a novel hardware implementation of the KASUMI block cipher designed using the principle of reuse of components. The architecture developed turns out to be a good balance between high performance and low complexity in area as a result of taking advantage of certain features present in modern FPGAs and some designs strategies. The main features of the architecture proposed are: reuse of higher-level components of the block cipher, which reduces the number of total cycles needed to carry out the process, mapping of the S-boxes to embedded dual-port memory blocks, and the design of a simple key scheduler that takes advantage of a clock-division technique. A similar technique to the one used in this work to design a shared datapath might be used to implement other Feistel-like block cipher algorithms. The design presented here can be implemented in other platforms such as Application Specific Integrated Circuit (ASIC) by carrying out the corresponding implementation process. The design proposed in this document might be incorporated in UMTS network components such as mobile equipment or Radio Network Controllers (RNCs), whether as a dedicated coprocessor or as a functional unit within a larger processor.

Acknowledgments

This work was sponsored by the scholarship number 171498 granted by CONACyT.

References

1. 3rd Generation Partnership Program. 3GPP Home Page. <http://www.3gpp.org>
2. 3rd Generation Partnership Program. Document 2: KASUMI Specification. Technical Specification 35.202. Release 5. Version 5.0.0.
3. Kim, H., et al.: Hardware Implementation of the 3GPP KASUMI Crypto Algorithm. Proc. of the 2002 International Technical Conference on Circuits/Systems, Computers and Communications ITC-CSCC-2002 (2002) 317-320.
4. Marinis, K., et al.: On the Hardware Implementation of the 3GPP Confidentiality and Integrity Algorithms. Proc. of the 4th International Conference on Information Security ISC 2001. LNCS 2200/2001. Springer-Verlag (2001) 248-265.
5. Matsui, M.: New Block Encryption Algorithm MISTY. Proc. of the 4th International Fast Software Encryption Workshop FSE'97. LNCS 1267/1997. Springer-Verlag (1997) 54-68.
6. Satoh, A., Morioka, S.: Small and High-Speed Hardware Architectures for the 3GPP Standard Cipher KASUMI. Proc. of the 5th International Conference on Information Security ISC 2002. LNCS 2433/2002. Springer-Verlag (2002) 48-62.
7. Xilinx, Inc.: Virtex-E 1.8 V Field Programmable Gate Arrays. v2.6. Product Specification (2002).

Predictive Array Access Cache-PAAC71

Alejandro Villar Briones ¹, Oscar Camacho Nieto ¹,
Luis Alfonso Villa Vargas ²

¹ Centro de Investigación en Computación-IPN, Laboratorio de Sistemas Digitales,
Av. Juan de Dios Batiz s/n, Col Nueva Industrial Vallejo, 07738, México DF.
albriones77@hotmail.com, osarc@cic.ipn.mx
<http://www.cic.ipn.mx>

² Instituto Mexicano del Petróleo IMP, Programa de Matemáticas Aplicadas y Computación,
Eje Central Lázaro Cárdenas 152, Col. San Bartolo Atepehuacan, 07738, México DF.
lvilla@imp.mx
<http://www.imp.mx>

Abstract. In this article a memory model appears cache associative of two routes with sequential access and a predicting one on the basis of stride. The development of this model is based on the observation of behavior of several models of memories cache of data, and removing the greater benefit to this behavior. The proposed model improves the behavior of the memory system (memory cache of data) diminishing the latency of access at levels similar to those of a memory of direct access or mapped, and the percentage of error (miss_rate) at levels similar to those of an associative memory. This is significant from the point of view of yield, since several machine cycles for an operation of access to memory are diminished, predicting its effective direction, with sufficient anticipation to its calculation, besides to reduce to the consumption of energy when avoiding unfruitful accesses to other blocks of the memory.

Keywords: memory cache, direct access, associative memory, prediction, stride, latency, number of cycles, hit_rate, miss_rate.

1 Introduction

The innovations in micro architectures have been conceived generally with the observation of the behavior of the programs. The designers frequently introduce new characteristics to micro architecture to operate the patterns of behavior of the program with the purpose of improving the yield of the processor.

Nevertheless the advance in the improvement of the yield of the memories so is not accelerated as in the case of the processor. Reason why it is necessary to explode characteristic of behavior of the memories with the purpose of reducing to the access time to them and the number of errors as far as data none found.

One of the main stages of the process of execution of a program is the compilation of values of the memory and its writing (instructions of load and storage load/store). In this process, it is possible to be taken advantage of the characteristic the space locality and the

temporary locality [1]. Considering these concepts he is as the memory hierarchy is developed, which consists of different levels from memory with different sizes and speeds. Next to the processor it is the denominated cache memory.

At the moment several models of this cache exist that try to diminish the percentage of failure rate (*miss_rate*) by means of having one better policy available [2], handling of replaced data (sweepings or remainder of breaks it), and policy search of data compares or sequential. If to this we added to him that the percentage of failure rate by means of dividing the cache in several blocks can be diminished (asociatividad level), diminishes the time of delay of the processor by a data that it has to be extracted from the following levels of memory when diminishing the latency (cycles by instruction). The problem that presents/displays these types of schemes, when dividing the cache memory in several blocks, is that the time search of the data is increased, since is duplicated in comparison with a memory of a single block or direct mapped.

When an associative memory is referenced, the blocks that conform it is examined in parallel form. When the labels of the blocks have been read, a label comparator selects the block that contains the wished data, and passes the data of the block to the processor. The selector of labels introduces an extra retard between the rating of the block that contains the wished data and the moment in that the data by the processor can be used. This among other factors increases the time search of a data in this type of memories. The figure 1 shows the graph in which it is compared as the memory of direct access has smaller latency of access in almost 45% with respect to the memory of associative type level two.

But on the other hand the associative memory always has demonstrated to have *miss_rate* lower than the one of a memory of direct access, in figure 2 is the difference of *miss_rate* between these two models.

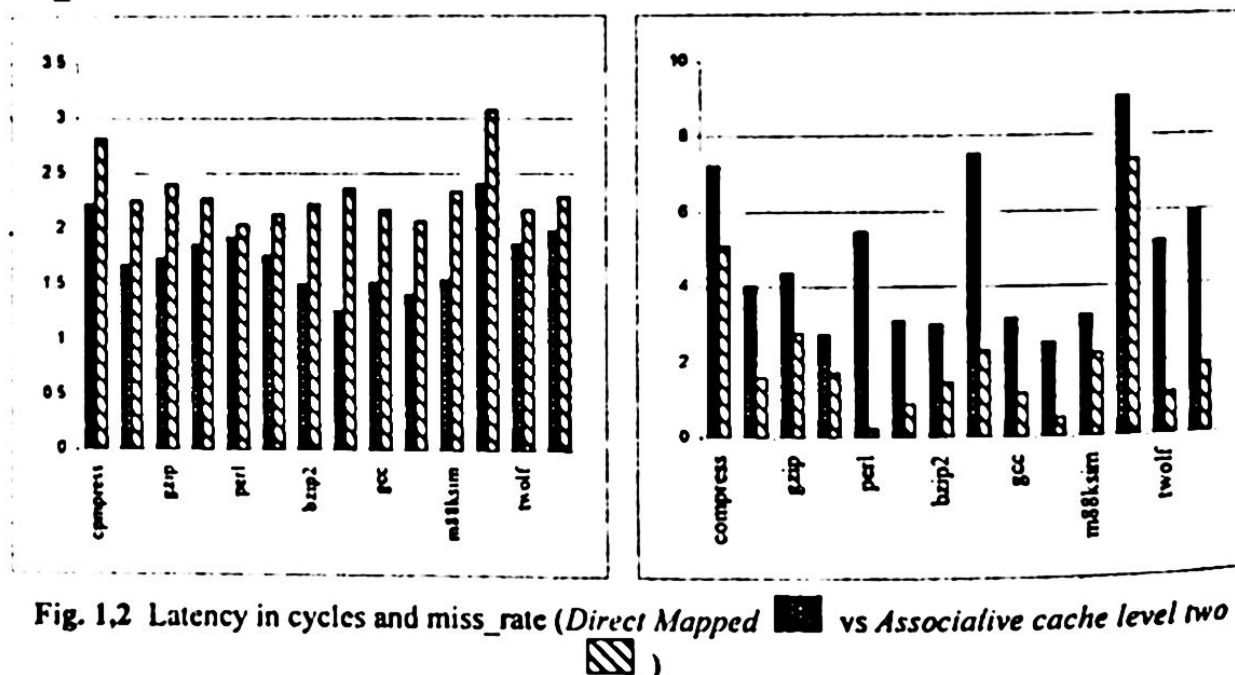


Fig. 1.2 Latency in cycles and miss_rate (Direct Mapped  vs Associative cache level two )

The design of a memory that combines the advantages of both models, associative direct mapped and, has been the objective of many investigators. This memory must have so much *miss_rate* as a latency low, is to say not to try to obtain efficiency to sacrifice the speed of operation of the memory. In order to obtain this objective different models from access have considered, policies of error handling and replaced information (sweepings) and models of prediction of location of the data.

In this articulate, we propose a data cache memory of associative type level two, with sequential access to its blocks, but with a predictor of data's location in some of the blocks. The model which we propose uses a predictor on the basis of stride, with a saturated accountant of two bits (4 states) who allows a greater percentage of guessed right predictions, implementing itself in an associative memory of level two with sequential access. This technique offers *miss_rate* of an associative memory of level two, but with an access time even smaller than a memory of direct access.

2 Antecedents

The implementations of cache memories of sequential type or serial they can be divided in two categories. First, they can be divided in the form in which they are examined at the time of making a data search, since this can be in a static and rigid order or of a dynamic form, making the first search not necessarily in the first block, is to say not always begins the search in the first block. Second, this type of memories can be divided by the type of policy that uses to write their data in each one of the blocks, to read them, or to retire them of such.

The cache memory *Hash-Rehash* type (HR-Cache) propose by Agarwal [3] reduces *miss_rate* of a memory of direct mapped, and although this model can implement the associative level that the designer proposes, available has a performance below an associative memory of level two with policy LRU as far as *miss_rate*. This model consists of the use of a memory of direct access that is acceded by means of two Hash functions that determine a first block of access and subsequent blocks according to the number of Hash functions that are implemented.

The cache memory model *Column-Associative* type (CA-Cache) of Agarwal and Pudar [4], improves *miss_rate* and the access time of HR-Cache; being based on an operation similar to HR-Cache. CA-Cache, it makes use of two blocks, if when making a search in the first block does not locate goes to the second block, and later changes the entrance of this second block towards first, so that later searches locate to the data in the first block diminishing the access time.

Nevertheless the use of this technique implies an extra consumption of energy to exchange the input data of both or more blocks than are implemented. As far as *miss_rate* it continues being higher than the one of an associative memory of level two.

Model that combines the advantages of the techniques before mentioned is proposed by Calder and Elmer [5], Predictive Sequential Associative Cache (PSA-Cache), which

instead of handling a policy of relocation of data it leans of a bit that the possible location of the data identifies, which diminishes the consumption of energy of the CA-Cache' memory, and in which use is made available of a policy MRU and a table of a denominated bit table of steering or direction, with which it determines the dynamic block to which the first search must accede when making.

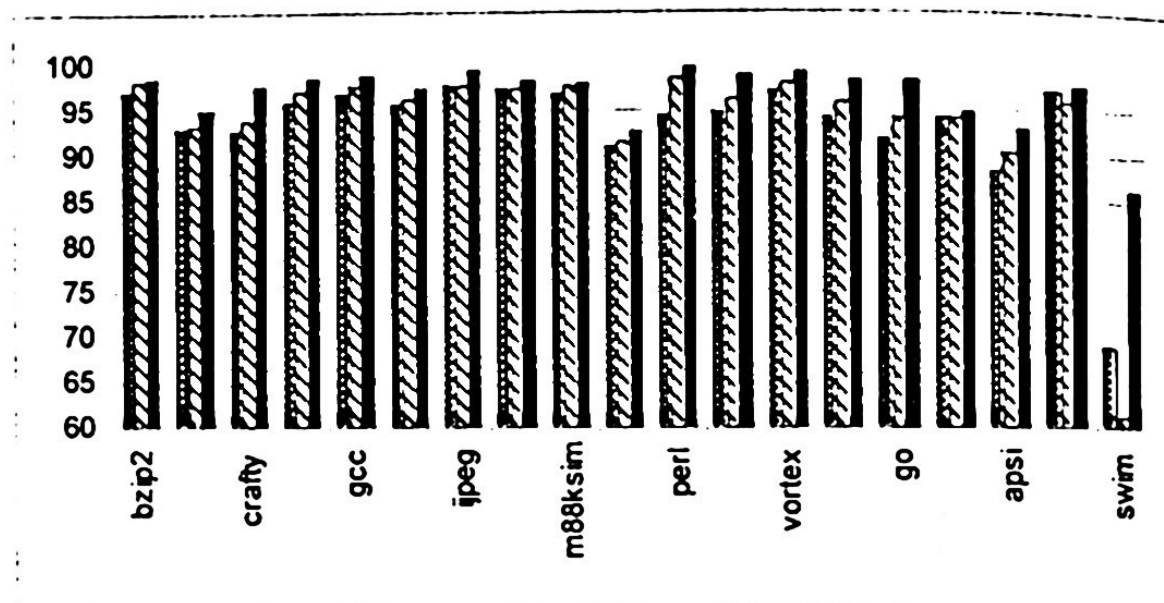


Fig. 3 Hit_rate % (Direct Mapped ■, PSA-Cache ▨ and Associative Cache Level two ▩)

3 Predictive accesses to arrays Cache (PAAC)

Following the tendency to conform the operation of the hardware of the processor to the behavior of the programs, is necessary to detect the behavior of the service loads on the memory models cache that there is in the market. In figure 3 in the graph one of the most recent models of associative memories of sequential access is observed, an associative memory of level two and the traditional memory of direct mapped. The last has a better behavior.

The idea to develop a memory of sequential access arises to determine the internal behavior of the associative memory, since a great number of successes is located in single one of the blocks that conforms it. Figure 4 shows as more of 90% of the successes they are located in block one of the associative memory.

Considering that the access time of an associative memory of level two is two cycles, the one considers that the search of the data in the memory is made simultaneously in both blocks, which consumes a first cycle of clock and later to compare the values that throw each one of the blocks and selecting valid data by a multiplexer a second cycle of clock.

If all the accesses went to first a block one more than the ninety percent of the accesses it would be obtained in a cycle of clock instead of two, in case of not finding the data in

this block the access to the second clear block would become this with a penalty of an additional cycle, nevertheless a percentage of error still below the one of a memory of direct mapped or a single block is obtained.

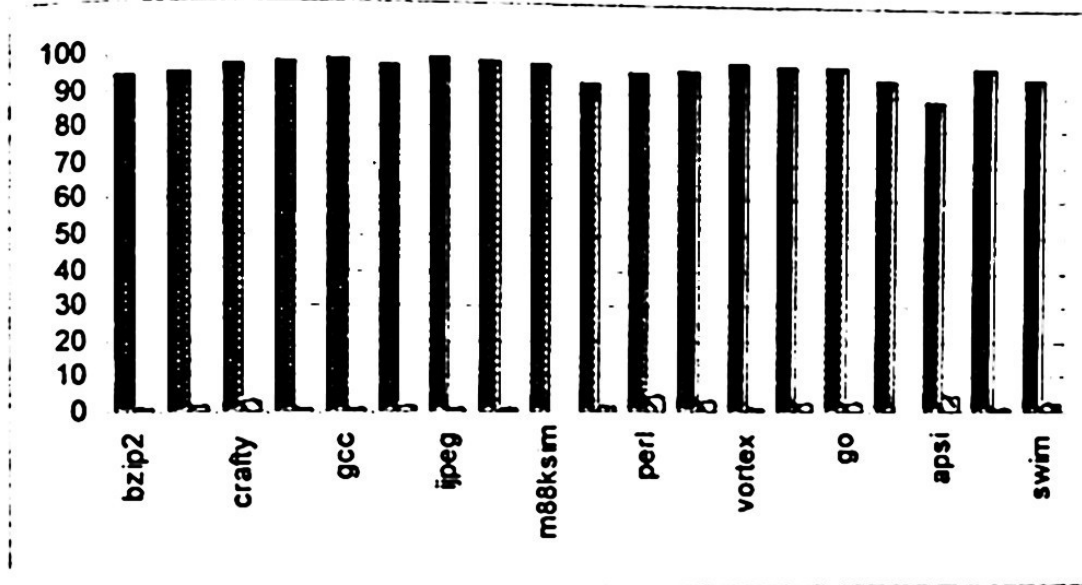



Fig. 4 Hit_rate (%) for each block in an associative cache memory of level two (AS-N2, block one , block two )

As this behavior is easily predictable thought about the use of a predictor on the basis of stride. The predicting one on the basis of stride, in its simpler form predicts the next value by means of determining the behavior of the accountant of the program. The basic idea is to predict future references to the memory on the basis of the passed history for the same instructions of access to memory.

The more intuitive scheme of prediction is the one than it has several iterations of the same search with previous events. This is, when the Program Counter (PC) decoded an instruction of load/store, reviews the entrances of a reference table to see if a file of that instruction exists. If it does not exist, an entrance of this table is used to store the file of this instruction. In case of existing the entrance and the reference for the following iteration he is predictable, is made a pre-search. This basic scheme surrounds the use of the PC and has a table to store the file of instructions of load/store that have acceded to the cache memory of data. Nevertheless for our case this table will be used to determine to that block was made the search of the asked for data.

3.1 Table of prediction reference

The table of reference of prediction, organized like a memory Cache of instructions, maintains the file of previous references or accesses to the memory of data and associate

stride to them (distance between instructions according to the *PC*) for each instruction of load or storage. The entrances of the table are conformed by the following elements:

⌘ *Tag*.- corresponds to the direction of the instruction of load/store.

⌘ *Last direction (operation)*.- that was referenced when the PC looked for the instruction

⌘ *Stride*.- the difference between the two you complete directions that were generated

⌘ *State*.- Two bits that serve like accountant to establish a file of three states of the accesses of the same instruction. To the beginning, the first access accountant in one indicates an initial condition, the second time the accountant is increased and situation is considered in which it is possible to make prediction of the location (block) of the following asked for value, and in the third state a condition of a prediction with high percentage of certainty in which occurs if the prediction is made.

⌘ *Block*.- in this field the location of the data is stored, being first or the second block of the memory. This field is updated after collecting the data of the memory and to corroborate its location

The size of the prediction table originally considered of a size of 32 entrances, nevertheless, when increasing the size of this table has an elevated percentage of prediction more.

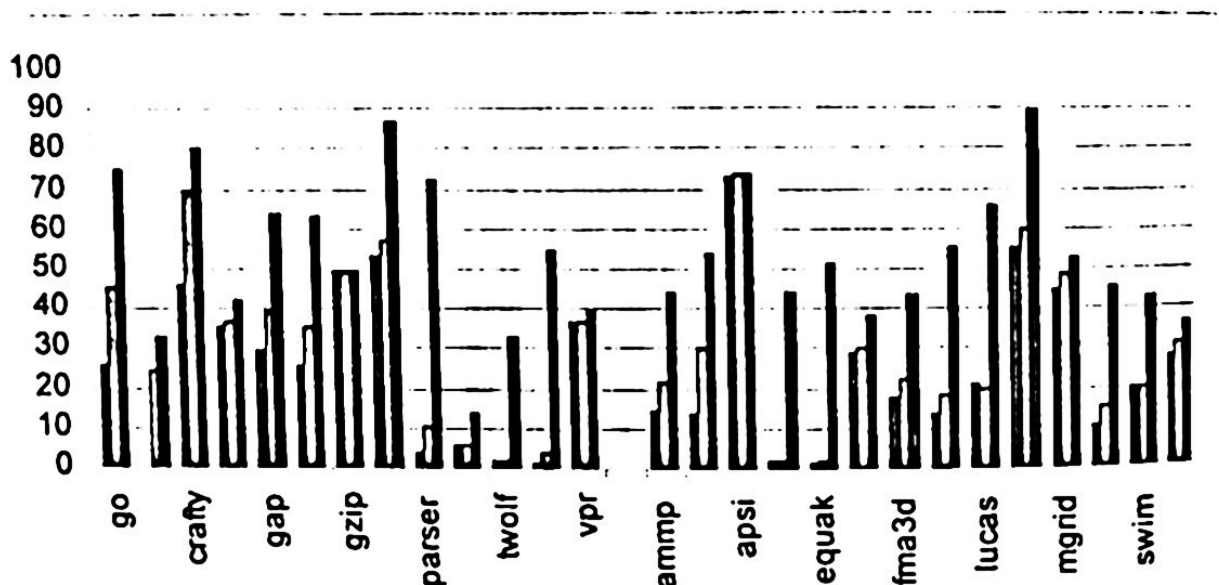


Fig. 5 Hit_rate of prediction (32 entries , 128 entries , 1024 entries , SPEC2000)

Obviously the percentage of erroneous prediction increases, but it is not in the same proportion that the guessed right prediction does. In figure 5 the behavior of the predicting one for different sizes is appraised from the table of prediction with the SPEC2000.

When handling themselves a robot of prediction based on a behavior of stride, and having a so great table, as is it of 1024, no longer takes single the accesses with a very repetitive behavior, but which it also considers other accesses with no longer so repetitive behaviors, which gives foot to have prediction errors, therefore was decided to handle a table of a size of 128 entrances, with which according to the obtained results it is possible to be covered most of the accesses with repetitive behavior

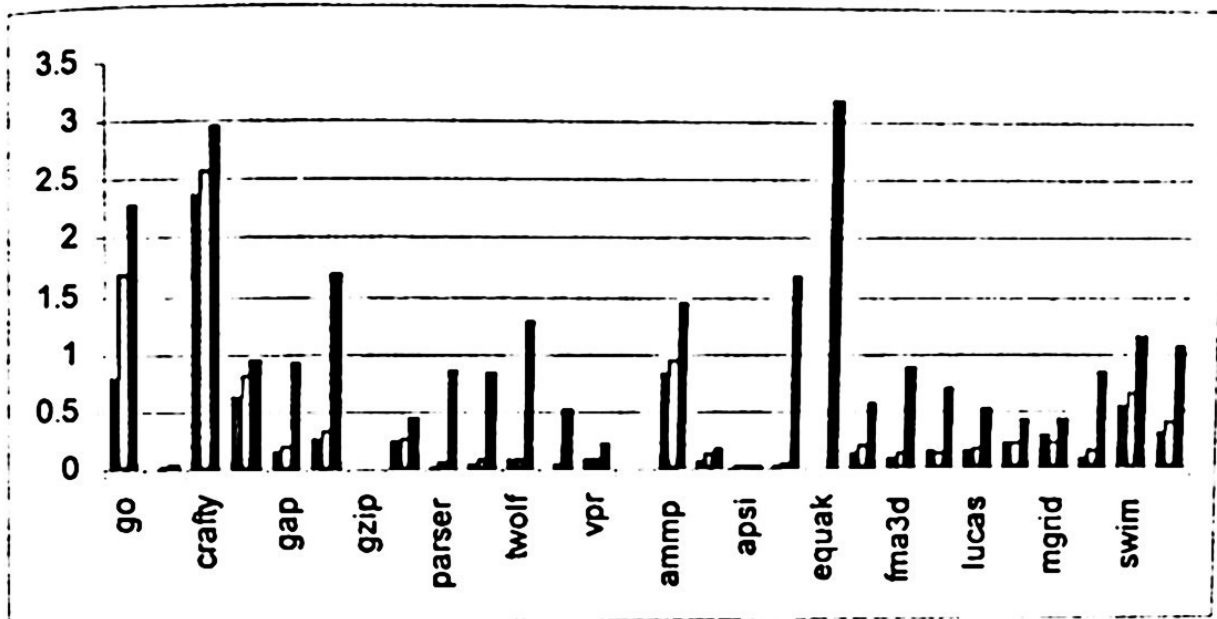


Fig. 6 Miss_rate of prediction (32 entries , 128 entries , 1024 entries , SPECINT)

3.2 Predictor on the basis of stride

The basic mechanism in the prediction on the basis of stride is to keep the effective direction from the operands of memory, it calculates stride for the access, and activates the field of State to verify if the following reference by means of comparing previous stride can be predicted kept with which it finishes calculating. Stride is obtained by means of taking the difference between the effective directions from the two but recent accesses by the same instruction.

The prediction says that she is correct when the direction that comes by the normal flow is equal to the sum of stride calculated but previous stored address, of to be certain this condition is not predicted and the condition of the field is modified state. When appearing a true condition is increased the accountant of the field state, but if the condition is false decrements this accountant.

On the basis of the value of this accountant the predicted value is used to accede to the correct block of the memory breaks in where is the stored data.

In accountant who handles itself for the *State* field, the first state corresponds with a condition of boot of the prediction table activating an entrance of the table. The

accountant changes to the following state whenever the result of stride is similar to the kept one in the first state; in this single state the values are updated and the value of the prediction table is not taken into account. The third state occurs at the time of repeating stride again and now if the value of the field is taken into account block to accede to the predicted block. The fourth state considers a state of very high probability of guessing right to the block that contains the data, since the behavior pattern has repeated at least three times previously (three previous states).

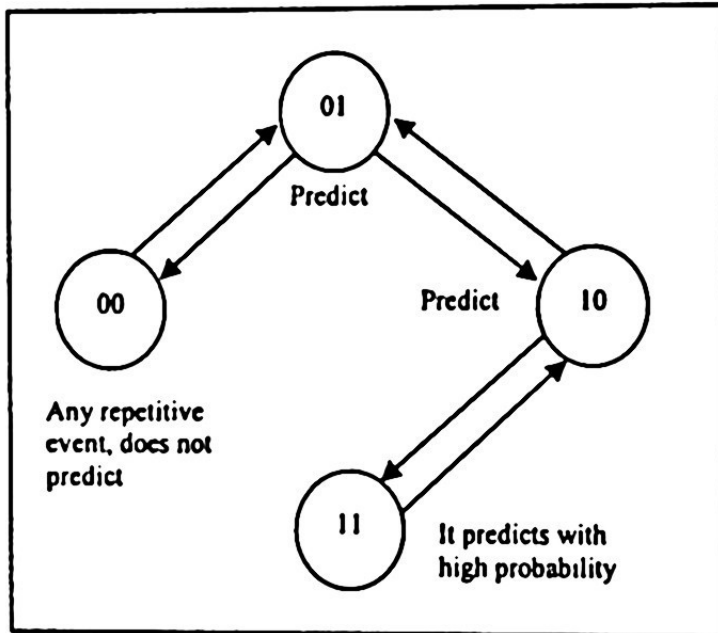


Fig. 7 Counter for state

Nevertheless when making tests and comparisons between simulations have been that when diminishing the value of the accountant of this robot is increased the percentage of prediction in each one of the simulations. Instead of which the prediction is made in state three of the accountant now is made in the second state. Figure 7 shows the operation of the accountant.

SPEC	Pred. 2° state. (%)	Pred 3° state. (%)
Go	45.55	18.22
Bzip2	24.6	14.76
crafty	69.44	34.72
Eon	36.76	29.408
Gap	39.32	23.592
Gcc	35.51	10.653
gzip	49.24	44.316
mcf	57.46	34.476
parser	10.69	5.8795
vortex	3.89	1.167
Vpr	37.12	14.848
ommp	21.57	6.471
applu	30	15
apsi	73.59	51.513
facere	30.38	18.228
fma3d	22.11	13.266
galgel	18.54	10.197
lucas	19.88	11.928
mesa	59.67	38.7855
mgrid	48.65	14.595
sixtrack	15.31	9.186
swim	19.65	11.79
wupwise	30.95	18.57

Table 1. - Percentage of hit_rate of prediction

The reason of using the value of the field block in the second state of the accountant is because many of the accesses made to a same direction of memory are repeated of constant form, if the robot allowed to predict single as of the third time that appears the same pattern of behavior was let go an event that can very possibly be predicted with a high level of certainty. If the prediction is allowed from the second state of the accountant, this characteristic takes advantage of the behavior of the hardware and an advantage of 40% in possible correct predictions over the previous model is obtained. Table 1 presents the data in percentage of *hit_rate* of prediction as of the second and third state of the accountant.

The PAAC altogether this conformed by a table of prediction, a form of sequential access and a politic of replace LRU (Fig. 8). The access to the PAAC is made

of the following way: if the prediction cannot be always made access to the first block of the memory, since it demonstrated (Fig. 4) that most of the data they lodge in this; if the data is found in this block had a latency of a cycle of access, which is similar to a memory of direct mapped, but of not finding the data is acceded to the second block, which brings like consequence a latency similar to the one of an associative memory.

The way that is followed to predict the block in which the looked for data is located initiates with the use of the *Program Counter* (PC) of the instruction of access to the memory, while by a side the effective direction of the memory by means of a sum calculates (it consumes at least a cycle), we indexed our table of prediction of 128 entrances, of which the value of the accountant for that entrance is verified. If the state of the accountant allows the prediction block in that same entrance of the table accedes to the block indicated by the field. In case of success an access in a time of a cycle is obtained, similar to the one of a memory of direct mapped, without mattering if the data were in the primary or secondary block. In case of a prediction error it is acceded to the block in opposition to the one of the prediction, which tares like consequence a penalty of an

additional cycle to which already uses to review the predicted block. The filling of the prediction table is made by means of comparing stride present with stride stored in the prediction table.

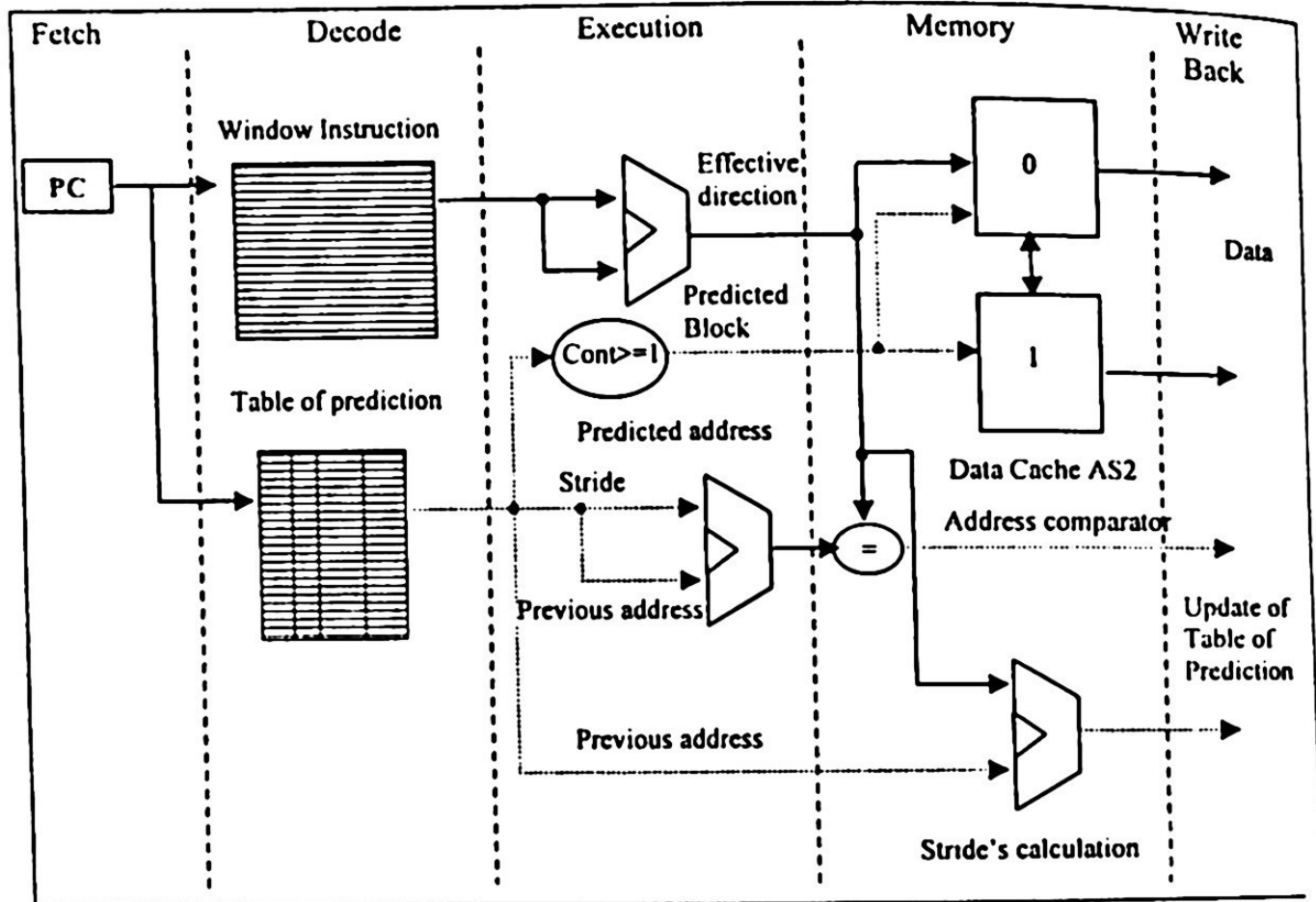


Fig. 8 Prediction's data_path of a block, that it is part from an associative memory of level two AS2 (normal flow \longrightarrow , prediction's flow \dashrightarrow).

Stride is obtained from the difference of the previous direction with the present direction, operation that can be conducted single until the calculation of the effective direction finishes.

In addition a comparison between the predicted direction (extreme of previous direction with stride stored) and the effective direction is made. If both comparisons guessed right, when comparing both stride is increased the accountant of the entrance the table, besides to verify in that block was the data within the cache memory, to updat the field block of the table.

In the next access to the memory the prediction table is verified and on the basis of accountant of events of stride and of the field block it is possible to predict in which from the memory is stored our data.

If the direction of the data is predictable accedes to the block indicated by the prediction table, if the data is located in this block, conserves the latency of direct mapped, but if not finding the data it is acceded to the other block in search of the data. If the prediction were successful and the data was located in block two represent that is eliminated the additional cycle to have to look for the data in the first block of sequential form and soon to accede to block two. When we have a huge prediction's percentage it can be used to reduce the access time to the memory.

4 Methodology of Evaluation

We compared the behavior of different models from cache memories using simulations of service loads, for it was used the Simplescalar [6] We obtained measures from 20 programs, from SPEC95 and SPEC2000. The models simulated memories were: memory of direct mapped (MP), memory Column-Associative type provided with a predictor on the basis of stride (CA-Pred), an associative memory level (two AS-2) and our model (PAAC); all the models with sizes of 8K, 16K, 32K and 64K. To determine performance of memory it is used the formula presented.

$$\text{Performance} = \frac{K + L}{M}$$

K = (hits in block one) + (hits in block two) + (hits by prediction)

L = (miss in any block) + (miss in prediction)

M = Total number of cycles for execution

The latency of success for a direct mapped memory is 1 cycle, for an associative memory level two is of 2 cycles, for our model it is one cycle for block one and predicted hits, two cycles for block two (sequential access). The error latency is 18 cycles for the three models. At last all the access (hits and misses) are added and divided by total number of cycles of benchmark execution.

5 Results

The index of errors for our model (PAAC) is similar to the one of an associative memory of type, since it leaves from the same principle of positioning of data, but the advantage that presents/displays is the access time, table 3 shows the obtained access times for each one of the models. In table 3 the smaller index of latency for the PAAC, even lower is emphasized than the one of a memory of direct mapped of the same size. Even making comparisons between the number of cycles of latency between PAAC and one of direct mapped, a PAAC of 16K has a smaller latency that a memory of direct mapped of 32K,

this point is excellent considering that like the size of our storage device is increased the index of miss rate is reduced.

Spec	MP	CA Pred	AS2	PAAC
applu	5.74	5.77	5.39	5.39
apsi	11.88	9.6	7.11	7.11
bzip2	2.99	1.79	1.46	1.46
compress	7.197	6.7	5.1	5.1
crafty	7.497	6.05	2.29	2.29
gap	4.02	2.80	1.60	1.60
gcc	3.13	2.26	1.14	1.14
go	8.11	5.74	1.67	1.67
gzip	4.348	3.76	2.76	2.76
jpeg	2.49	2.20	0.49	0.49
li	2.72	2.66	1.72	1.72
M88ksim	3.20	2.49	2.20	2.20
mcf	9.00	8.64	7.34	7.34
mgrid	2.96	3.97	2.36	2.36
perl	5.45	1.38	0.23	0.23
swim	31.13	38.91	13.73	13.73
twolf	5.15	3.76	1.11	1.11
vortex	3.07	1.93	0.87	0.87
vpr	5.9	4.09	1.84	1.84

Table 2.- Percentages of miss_rate (%)

SPEC	MP	CA Pred	AS2	PAAC
applu	2.16	2.34	3.12	2.01
apsi	2.28	2.62	3.6	2.12
bzip2	1.51	1.59	2.22	1.25
compress	2.22	2.42	2.81	1.88
crafty	2.26	2.25	2.35	1.43
gap	1.67	1.77	2.25	1.29
gcc	1.52	1.68	2.17	1.21
go	1.78	2.01	2.5	1.5
gzip	1.73	1.83	2.43	1.48
jpeg	1.41	1.66	2.07	1.15
li	1.86	1.69	2.27	1.3
M88ksim	1.54	1.72	2.34	1.42
mcf	2.41	2.87	3.07	2.16
mgrid	1.38	1.72	2.11	1.2
perl	1.92	1.7	2.03	1.1
swim	2.9	3.28	4.02	2.9
twolf	1.87	1.78	2.17	1.23
vortex	1.65	1.76	2.13	1.16
vpr	1.99	2.02	2.29	1.35

Table 3. - Index of latency (cycle)

6 Conclusions and futures works

On the basis of the predicted results and to the obtained ones, we can be made comparisons of the degree of effectiveness of proposed model. In the case of the area predictions of location of data it is possible to be continued implementing improvements as far as possible in the prediction, since in spite of having a percentage of 94% effectiveness, we diminish the *miss_rate* of prediction with the propose of increasing the performance of our model. The application of this model in associative memories with greater number of blocks is the next point of study and to evaluate its yield, also possible implementation in language VHDL for its simulation at level layout to determine its power consumption.

References

1. J.-k. Peir, Y. Lee, and W. Hsu, "Capturing Dynamic Memory Reference Behaviour with Adaptive Cache Topology," Proc. 8th ASPLOS, 1998.
2. J.W. Fu, J.H. Patel and B.L. Janssens, "Stride Directed Prefetching in Scalar Processors", Proc of the 25th Int. Symp. On Microarchitecture (MICRO-25), pp. 102110, 1992
3. Anant Agarwal, John Hennesy, and Mark Horowitz. Cache performance of operating systems and multiprogramming, ACM Transactions on Computer Systems, 6:393-431, November 1988.
4. Anant Agarwal and Steven D. Pudar. Column-Associative caches: A technique for reducing the miss_rate of direct mapped caches. In 20th Annual International Symposium on Computer Architecture, SIGARCH Newsletter, pages 179-190, IEEE, 1993
5. B. Calder, and D. Grunwald, J. Elmer, "Predictive Sequential Associative Cache", Proc. 2nd HPCA, 1996.
6. Todd Austin, Eric Larson, and Dan Ernst, "SimpleScalar: An Infrastructure for Computer System Modeling," IEEE Computer, February 2002.
7. J.-L. Baer and T.-F. Chen, "An Effective On-Chip Preloading Scheme To Reduce Data Access Penalty", ACM, ICS 1991
8. Alan Jay Smith, "Cache Memories", Computing Surveys, 14(4): 473-530, September 1982.
9. M. Hill, A. Smith, "Evaluating Associativity in CPU Caches", IEEE Trans Computers, Vol. 22(12), Dec. 1989
10. M. Jonsn, Sperscalar Microprocessor Design. Englewood Cliffs, New Jersey; Prentice Hall, 1990.
11. David A. Paterson, John L. Hennessey. Computer Organization and Design: The Hardware/Software Interface. Second Edition, Morgan Kaufmann Publishers, 1997.
12. Deszö Sima, Terrence Fountain, Peter Kacsuk, Advanced Computer Architectures: A Design Space Approach. Addison Wesley, 1997
13. Norm Jouppi. Cache writes policies and performance. In 20th Annual International Symposium on Computer Architecture, SIGARCH Newsletter, pages 191-201, IEEE, May 1993.

Instructions-Wake-Up Mechanisms: Power and Timing Evaluation

Marco A. Ramírez^{1,4}, Adrian Cristal¹, Alexander V. Veidenbaum², Luis Villa,³ Mateo Valero¹

¹ Computer Architecture Department U.P.C. Spain
+34-93-401-69-79, Fax: +34-93-401-70-55

e-mails: {mramirez, adrian, mateo}@ac.upc.es,

² University of California Irvine CA
alexv@ics.uci.edu

³ Mexican Petroleum Institute, Mexico.
lvilla@imp.mx

⁴ National Polytechnic Institute, Mexico

Abstract. Power dissipation is a major constraint in the design of new micro-architectures for state-of-the-art microprocessors. One of units that have received considerable attention as a major consumer of power is the instruction queue. This research studies the power consumption and timing of two different designs of the instruction queue and wakeup logic used in modern microprocessors. One design is a standard CAM-based mechanism; the other is a new proposal using a RAM-based direct wakeup mechanism [7]. The energy and timing of CAM and RAM structures used in the instruction queue were evaluated using Spice3 tools for the 70nm technology.

Keywords: Processors Architecture, Issue Queue, CAM Wakeup, Matrix Dependency Wakeup, Direct-Wakeup, Low-Power.

1 Introduction

This work evaluates power and timing of structures which implement the instruction wakeup mechanisms used in modern superscalar processors. It compares traditional designs with a Direct Wakeup scheme proposed in [7]. We pay close attention to the amount of hardware that must be added in order to achieve a power-efficient solution. Section 2 describe how the structures work. Section 3 presents the technology considerations taken into account for evaluations. Section 4 shows a transistor level design of structures used, section 5 presents a new design we propose and its evaluation. Finally, section 6 presents the results and conclusions.

2 Traditional Issue Queues

Instruction queues are structures that permit Out-Of-Order issue and execution, necessary to achieve high Instruction Level Parallelism ILP required in modern microprocessors. One operation that has to be performed in a single cycle by the instruction queue logic is Wakeup-Select.

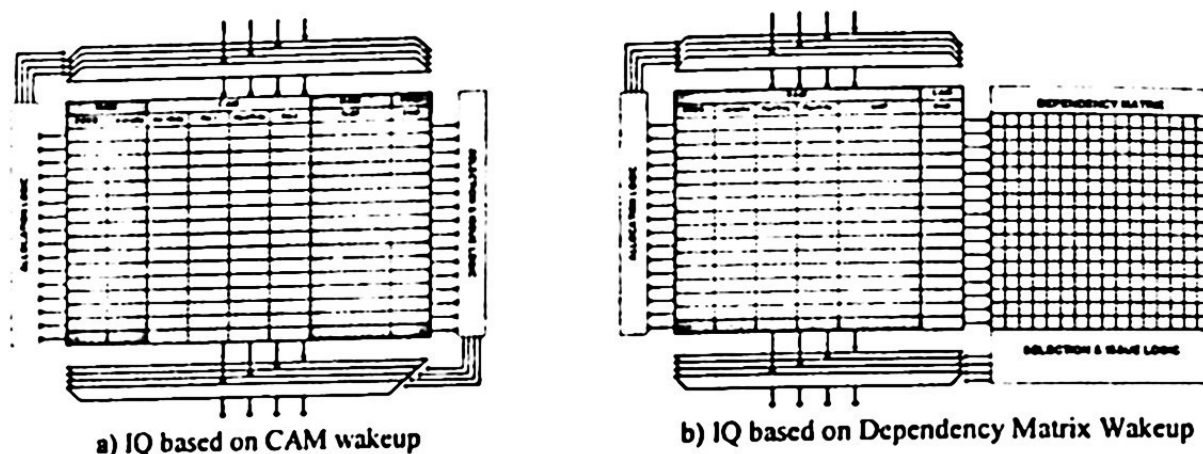


Fig. 1. Issue queue Models

The wakeup logic is responsible for waking up instructions waiting in the issue queue for their source operands to become available. This wakeup action is performed by a Content Addressable Memory. A traditional instruction queue dissipates between 18 and 25% of the total processor power dissipation [5]. A major part of this energy dissipation is due to comparisons performed in the CAM's by the wakeup logic.

There are two ways to wake up instructions (CAM wakeup and Dependency Matrix Wakeup). The first one is by comparing the result tag of a completing instruction with the tags of all source operands of instructions waiting in the issue queue. On a match a respective OpRdy bit is set. Instructions with all its operands available are ready to issue in next cycle. The second way is to perform the comparisons in allocation stage [4]. In this case the source operands are compared with all destination register waiting in the issue queue; matches are used for set the corresponding bits in a dependency matrix. Bits set in a row of the matrix correspond to the instructions to wakeup. Dependency matrix uses column bits counters to evaluate ready instruction, but not make comparisons after the execution. In both two cases, a CAM is required. But the number of comparison ports in CAM-Wakeup model is proportional to the execution width (EW), while in Matrix-Dependency-Wakeup model it is proportional to the decode width (DW).

Selection logic determines which instruction has its operands ready; instructions with all operand available are ready to being issue.

3 Technology Considerations

Traditionally, fanout-of-four (FO4) delay is a metric used to estimate CMOS-circuit speed because it is independent of technology process. The FO4 delay is the time for an inverter to drive 4 copies of same inverter. For static logic, an approximation of the FO4 delay in picoseconds is given by $360 \times L_{\text{drive}}(\mu\text{m})$ [2], where $L_{\text{drive}}(\mu\text{m})$ is the minimal length of transistor gate in micrometers. The FO4 metric is a useful measure to estimate the processor clock speed across technology generations [3]. In the literature the access delay of some structures in the critical path are used as a lower bound of the microprocessors clock cycle time. [2] suggests using the computation delay of very optimized 64-bit adder (5.5FO4 for static logic), under the assumption that to execute two dependent instruction in consecutive cycles clock, the first instruction must compute its result in a single cycle. Considering the pipeline-latches overhead and time to bypass the adder result back to the input for the next instruction, clock period is goes down to 8 FO4 delays in aggressive estimates and 16 FO4 delays in conservative estimates. With these considerations, the FO4 delay decreases from 64ps in an 180nm technology to, while frequency clock increases from 1.09GHz in an 180nm to 5GHz in a 70nm Technology.

3.1 Wire considerations

The source of technology parameters used in this work is Predictive Technology Model (PMT) provided by the device group at UC Berkeley [1]. The model uses the same approach as the International Technology Roadmap for Semiconductors by subdividing the wiring layers into three categories 1) Local lines, for connections within a cell, 2) Intermediate lines for connections between modules and 3) Global lines for chip communications. In addition, it use two structure types a) Global Layer lines are coupled above one metal ground and b) Local and Intermediate Layers lines are coupled between two metal ground.

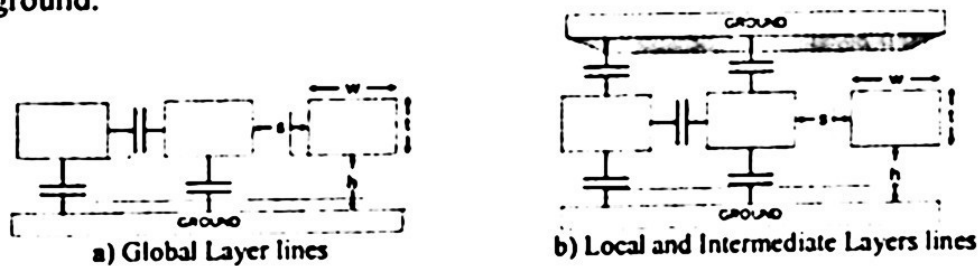


Fig. 2. Wire Structures used in PMT

Table I shows the wire parameters for different technology generations. R_{wire} , C_{couple} , C_{ground} and C_{total} values are based on the analytical model of PMT [1]. Here we can see that while the transistor's speeds are scaling more or less linearly, wires are getting slower

with each new technology generation. This is because the resistance of wires increases with each new technology generation but the capacitances stay more or less constant.

Table 1. Typical Wire Dimension and its parasitic effects

	Width (μm)	Space (μm)	Thickness (μm)	Height _{tp} (μm)	K _{fld} (μm)	R _{wire} $\Omega/\mu\text{m}$	C _c fF/ μm	C _g fF/ μm	C _t fF/ μm
0.180 μm Technology generation @ 1.9 GHz									
Local	0.28	0.28	0.45	0.65	3.5	0.174	0.075	0.030	0.212
Intermediate	0.35	0.35	0.65	0.65	3.5	0.096	0.078	0.038	0.233
Global	0.80	0.80	1.25	0.65	3.5	0.021	0.081	0.080	0.243
0.130 μm Technology generation @ 2.7 GHz									
Local	0.20	0.20	0.45	0.45	3.2	0.244	0.089	0.029	0.236
Intermediate	0.28	0.28	0.45	0.45	3.2	0.174	0.060	0.040	0.201
Global	0.60	0.60	1.20	0.45	3.2	0.030	0.089	0.079	0.258
0.100 μm Technology generation @ 3.47GHz									
Local	0.15	0.15	0.30	0.30	2.8	0.488	0.068	0.028	0.193
Intermediate	0.20	0.20	0.45	0.30	2.8	0.244	0.068	0.037	0.212
Global	0.50	0.50	1.20	0.30	2.8	0.036	0.088	0.082	0.256
0.070 μm Technology generation @ 5.0 GHz									
Local	0.10	0.10	0.20	0.20	2.2	1	0.053	0.022	0.152
Intermediate	0.14	0.14	0.35	0.20	2.2	0.448	0.057	0.031	0.177
Global	0.45	0.45	1.20	0.20	2.2	0.040	0.073	0.082	0.230

3.2 Wire delay

Since the delay time is proportional to the product of resistance and capacitance, evaluated wire delays for various clock rates using Spice3 for different technologies: 180nm @1GHz, 130nm @2GHz, 100nm @3GHz and 70nm @5Ghz. A distributed RC model with 1 μm wire RC basic segment was used. Figure 3 shows the delay for local and intermediate wires expressed in FO4 delays, while Figure 4 shows Global wires delay and distributed RC model.

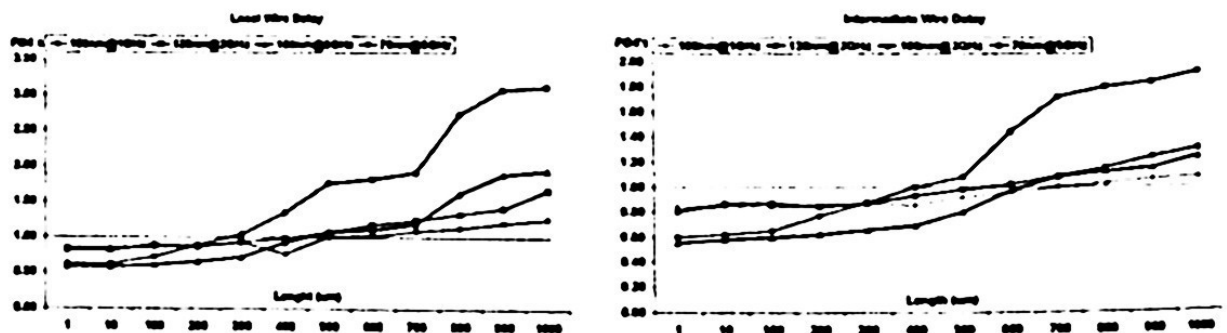


Fig. 3. Wire Delay evaluation using Spice3

For the load at the far end of each segment evaluated, we used a buffer with high speed characteristics: under the assumption that an optimal segment length makes the wire delay equal to or less than one FO4 delay. Wires with delay under one FO4 line in Figures 3 and 4 are considered optimal segment length without repeaters.

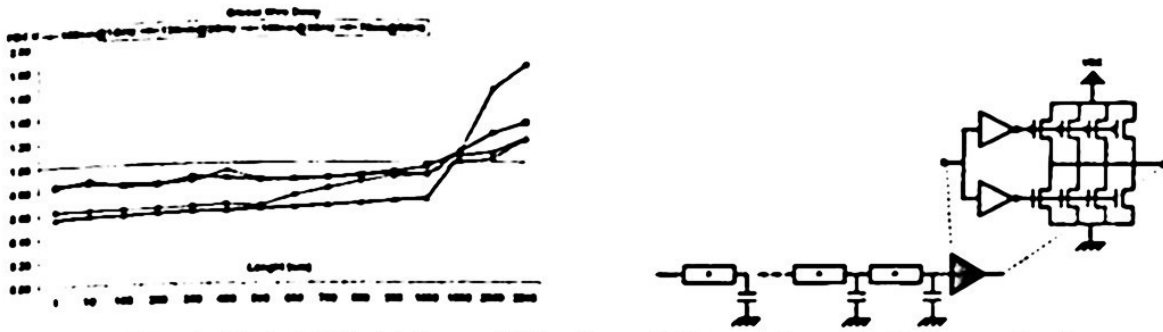


Fig. 4. Global Wire Delay and Distributed RC model used in Delay evaluation

3.3 Reducing Wire Delays

In long wires the insertion of repeaters periodically along the wire brakes the quadratic function of the delay on the wire length and makes it a linear function of the total wire length. The general expression for delay in a wire of length L with distributive RC is given by $T_p = 0.38 \times R \times C \times L^2$, while in a wire with $M-1$ repeaters the expression for delay is $T_p = 0.38 \times R \times C \times \left[\frac{L}{M} \right]^2 \times M + (M-1) t_{\text{buffer}}$ where t_{buffer} is the buffer delay time and M the number of wire segments. The optimal number of buffers on total wire length is obtained taking the first derivative with respect to segments number M , $\partial T_p / \partial M = 0$ then the number of repeaters is $M = L(0.38 \times R \times C / t_{\text{buffer}})^{1/2}$, taking into consideration $M=1$ for a wire with length $2l$, the maximum wire length tolerable without have need of repeaters theoretically is $l = 0.5 \sqrt{(t_{\text{buffer}} / 0.38RC)}$.

With these considerations we designed the CAM and RAM memories for 70nm technology, as explained in the following sections.

4 Microarchitecture of Circuits

In order to evaluate timing and power of the issue queue, we considered 70nm technology parameters. The size of a memory cell is basically determined by the wires. For bit-line and word-line size we used intermediate wire parameters, interconnections between transistors are made with local wires. The length of wires in our designs is below the optimal segment length. Note that we evaluated only the access time of structures and did not consider the latencies of allocation logic and selection logic.

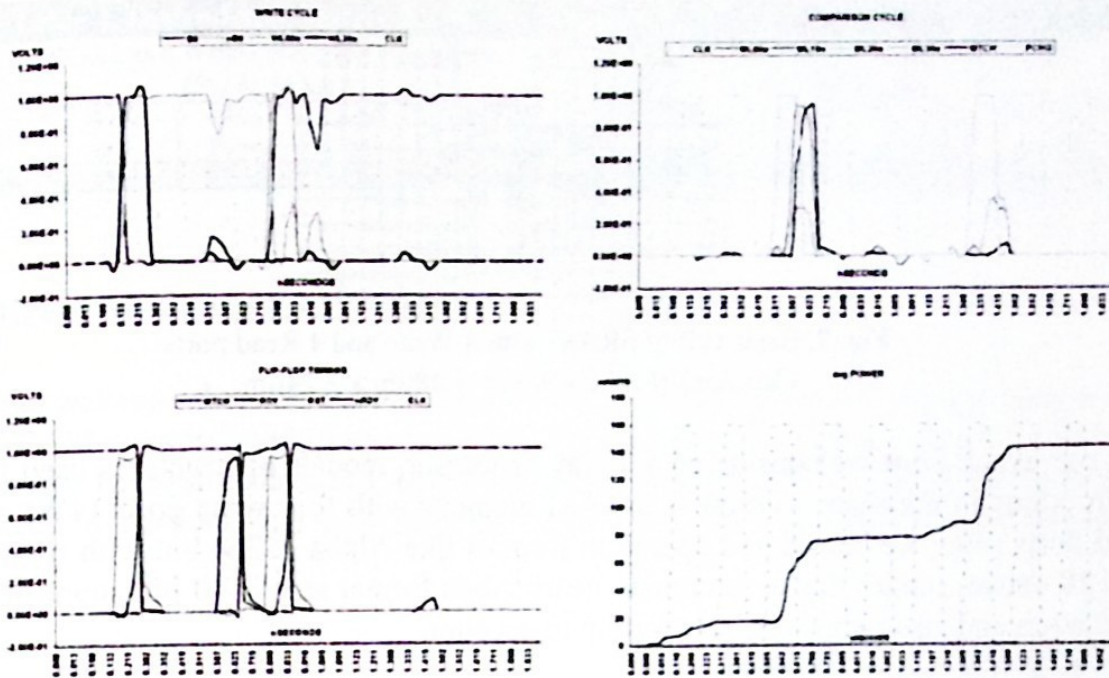


Fig. 6. Wave form of write cycle, comparison cycle, flip-flop timing and average power of CAM with 32 entries 4 Write ports and 6 Comparison ports

Table 2 shows the measure of timing and power from spice3, for timing measure the biggest delay between bit-line and word-line was taking (considering drivers delay). For write access, basic-cell stabilization delay was added and for comparison, MTCH signal generation delay was added. Power was measured on complete structures.

Table 2. CAM 70nm technology										
CAM entries	7 bits 4 Write Ports, 6 Comparison Ports ($l=31.36\mu\text{m}$)					7 bits 4 Write Ports, 8 Comparison Ports ($l=39.20\mu\text{m}$)				
	$h(\mu\text{m})$	Write (ps)	Avg Power	Comp (ps)	Avg Power	$h(\mu\text{m})$	Write (ps)	Avg Power	Comp (ps)	Avg Power
32	89.6	42.8	18.80mW	73.9	47.15mW	107.52	48.6	76.95mW	104.9	94.45mW
64	179.2	55.5	36.50mW	80.0	95.27mW	215.04	56.3	153.55mW	130.7	166.95mW

4.2 SRAM Issue Window

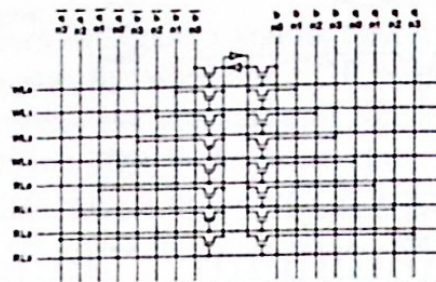


Fig. 7. Basic cell of SRAM with 4 Write and 4 Read ports
Cell Area ($l \times h$): 4W4C= $4.48\mu\text{m} \times 2.24\mu\text{m}$

For our evaluations we considered a 4-way processor, models of structures used for store instructions in the issue window is a RAM memory with four write ports (4W) and four read ports (4R), we considered operation formats like Alpha 21264 but with register files of 128 entries, under this assumptions instructions format size is 40 bits, control bits for allocation and issue logic are not taken in to account.

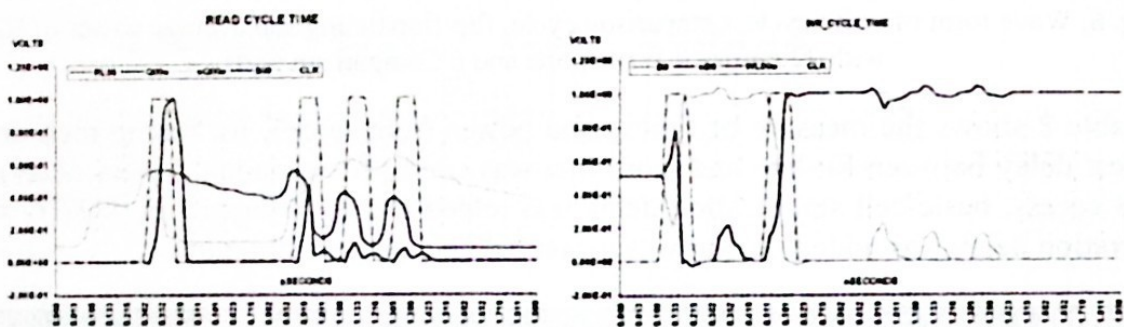


Fig. 8. Waveform of read cycle and write cycle time of RAM with 32 entries
4 Write ports and 4 Read ports

Table 3. Issue Queue RAM 70nm technology					
RAM entries	40 bits 4 Write Ports and 4 Read Ports ($l=156.8 \mu\text{m}$)				
	$h(\mu\text{m})$	Write (ps)	Avg Power	Read (ps)	Avg Power
32	71.68	56.5	6.14 mW	22.5	11.50 mW
64	143.36	71.5	9.85 mW	22.5	13.75 mW

4.3 RAM Dependency Matrix.

Dependency matrix structure is a RAM like shown in figure 7; this is a square matrix of $N \times N$ where N is queue size.

Table 4. Dependency Matrix RAM 70nm technology									
32X32 bits 4 Write Ports and 4 Read Ports ($l=143.36 \mu\text{m}$)					64X64 bits 4 Write Ports and 4 Read Ports ($l=286.72 \mu\text{m}$)				
$h(\mu\text{m})$	Write (ps)	Avg Power	Read (ps)	Avg Power	$h(\mu\text{m})$	Write (ps)	Avg Power	Read (ps)	Avg Power
71.63	48.6	6.94mW	22.5	8.90mW	146.36	65.5	19.00mW	22.5	22.00mW

5 Direct Wakeup Evaluation

Direct wakeup schema is show in figure 10. The goal of this mechanism is replace the use of CAM memory used in associative searches on wakeup stage by the use of pointer keep in two RAM structures, a Mapping table MT and a little Dependency Matrix MWT to wakeup instructions with multiple-dependents. Direct Wakeup Mechanism is proposed in [7].

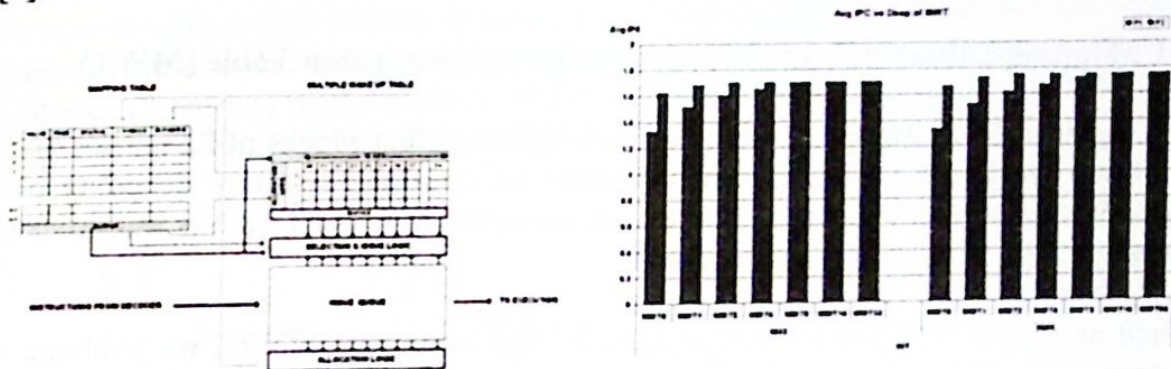


Fig. 9. Scheme of direct Wakeup Mechanism and Impact of MWT deep on performance

This MT allocates in each entry follows fields (example for one dependent-instruction-pointer): The free register bit, the value register (both two last field are part of register file), the status register, the consumer instruction pointer and a Pointer to Multiple-Wake-up-mechanism table. The status stage could be 00: No-dependent, 01: One dependent 10: Multiple-dependents, 11: Computed Value (Ready), for this case. The pointer for multiple wakes up mechanism is used depending of the third field value –If the register is not free and the status is 10: Multiple dependents. Mapping table is an extension of register file but with independent ports for reads, used on wakeup stage. Allocation of pointer is performance on decode stage, at checking time for ready operands. Wakeup is achieved reading the MT indexed by destination register file and reading MWT indexed by M-POINTER if instructions with multiple-dependents exist.

In order to evaluate the microarchitecture behavior of Direct-Wakeup scheme, we used SimpleScalar 3.0. the simulator has been modified to model separately integer, floating

point and load/store queues, and was build for Alpha configuration, as a workload. SPEC2000 programs was simulated (each of which ran for 200 million instruction). Figure 9 shows the impact of MWT deep on performance processor; right columns each group correspond to the maximum IPC possible for its configuration. MWT with entries using single pointer, experiment IPC loss of 4.8% for FP and 2% for INT. MWT with 4 entries, experiment IPC loss of 10% for FP and 4% for INT benchmark respectively.

For Spice3 simulation we use MWT with 16, 8, 4, 2 entries and single pointer. Considering 4-way Out Of Order processors, these structures require 8 write ports and read ports for MT and MWT. The status fields are independent state-machines with parallel inputs which can do change the state of machine (inputs Ored) on each access each physical register. The MWT, needed 8 1-bit write ports, and 4 full-bit (32/64) read ports.

5.1 Mapping Table (MT) and Multiple Wake-up Mechanism Table (MWT)

Organization of Mapping Table has been split into four blocks of 32 entries and these in 4 sub blocks of 8 entries, causing shorter bit lines and selective decoding block optimize power and cycle time. Organization require of one 4:1 18-bits multiplexer with delay of 22ps, for a single output data bus.

Such as MT of 128 Entries was divided in four symmetric blocks, we evaluate power consumption for writes (2.128mW/2.59mW) and reads (3.058mW/3.19mW) of a single entries per 10/11-bits block respectively, measures of only one cycle without activity report around of 0.3mW. In order to evaluate total power we have considered access power plus three times power on idle stage.

Table 5. MT RAM 70nm technology					
RAM 128 4 Banks of 32 Entries	10bits 8Write Ports and 6 Read Ports ($l=78.4 \mu m$)				
	$h(\mu m)$	Write(ps)	Avg Power (mW)	Read (ps)	Avg Power (mW)
	125.44	56.5	3.028 mW	45.0	3.958mW
	11bits 8Write Ports and 6 Read Ports ($l=86.24 \mu m$)				
	125.44	56.5	3.49mW	45.0	4.09mW

Multiple Wake up Table MWT is a RAM structure of 2, 4 or 8 entries, 8 write ports decoding the IQ pointers to write one bit in the corresponding entry which is reserved allocation logic, write access is validated by the status signals latched in the register read stage. 6 read ports are necessary in case of maximum possible number of instructions execution have it all multiple dependents.

Table 7. MWT RAM timing and power for 70nm technology

MWT entries	32bits, 8 Write and 6 Read Ports ($l=250.88 \mu\text{m}$)					64 bits, 8 Write and 6 Read Ports ($l=501.76 \mu\text{m}$)			
	$h(\mu\text{m})$	Write (ps)	Avg Power (mW)	Read (ps)	Avg Power (mW)	Write (ps)	Avg Power (mW)	Read (ps)	Avg Power (mW)
1	3.92	23.6	1.00 mW	22.0	4.70 mW	23.6	5.00 mW	22.0	9.51 mW
2	7.84	23.6	1.30 mW	22.0	4.73 mW	23.6	8.16 mW	22.0	9.51 mW
4	15.68	23.6	4.97 mW	22.0	7.10 mW	23.6	8.83 mW	22.0	13.42 mW
8	31.36	25.6	7.29 mW	22.0	8.93 mW	25.6	10.53 mW	22.0	15.05 mW
16	62.72	38.2	10.50 mW	25.0	11.50 mW	38.2	15.30 mW	25.0	19.43 mW

5.2 Decoder

Such as MT-memory is splitting into 4 blocks, decoder is designed using dynamic gates to optimize timing and power the address word is split to reduce the number of decode stages. Each basic decoder uses 8 row decoder like show in the figure 12c, decoding the three lowest address (A2-A0) bits to select one of eight entries of the memory sub block. Sub blocks are decoded using two next address bits (A4-A3) to selecting one of the four sub blocks (8 entries) of the block array (32 entries). Finally blocks are decoded with another similar 2 bits decode using (A6-A5).

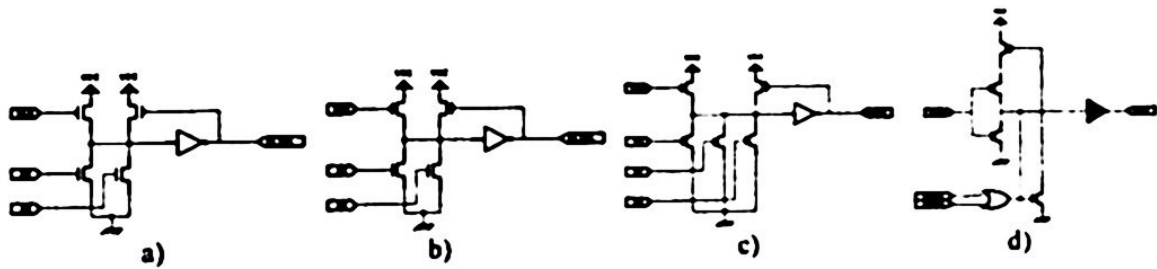


Fig. 10. a) Block decoder b) Sub block decoder c) ROW decoder and d) Enable WL circuit

The enable word line circuit shown in figure 12d is used to isolate the RAM array of the decoder. Selection block and Selection sub block signal enabling or disabling the WL. Table 6 show the delay and consumption of 7-bits address decoder.

Table 6. 7 Address bits DECODER 70nm technology

DECODE	7 Address bits Decoder		
	$h(\mu\text{m})$	Delay(ps)	Avg Power (mW)
128WL	125.44	32	8.5

6 Power Evaluation

Power dissipation is one of the most important factors on evaluation of VLSI designs [8], new approaches in low power require of accurate power consumption simulation new technology generations. Note that in general expression for power, the consumption depends strongly of the circuits capacitance $p = C_T V_{DD}^2 f$, however estimation of C_T requires not only identification of state-changing in logic gates, but also the effective capacitances in gate regions and drain regions a long with biasing effects, this is the analogical behavior of transistors. In order to achieve an accurate monitoring of power dissipation in VLSI circuits, we use Spice3 simulator tool. There is a sub-circuit to measure average power taking advantage of the voltage equation on one capacitor (see figure 11).

This meter is an independent subcircuit with a current-controlled current source and a parallel RC circuit. Power average in a dissipative element with fixed source voltage V_{DD} in a time interval $\Delta t = t_2 - t_1$ is given by:

(1) $P_{avg} = V_{DD} / t_2 - t_1 \int_1^2 i_{DD}(t) dt$ by using a current-controlled current source with current equal to $i_x = i_{DD}$, Voltage in a Capacitor C_y (from figure 11) in a time interval $\Delta t = t_2 - t_1$ is given by next equation

(2) $V_{C_y}(t) = \beta / C_y \int_1^2 i_{DD}(t) dt + v_o$ by running a transient analysis and reading V_{C_y} at time t_1 and t_2 the average power consumption of the circuit is monitored by the equation 2 if a value for β is chosen such that

(3) $\beta = C_y V_{DD} / t_2 - t_1$ or $\beta = C_y V_{DD} f$.

The time interval $t_2 - t_1$ should span a clock period or multiple integer of period of the circuit frequency operation. Measuring the power dissipation is equivalent to measuring the supply current flow during the transient analysis. The current pulses are very short, and the current waveform must be computed carefully to accurately compute the power. In order to produce better results we choice second order gear rather than trapezoidal integration method.

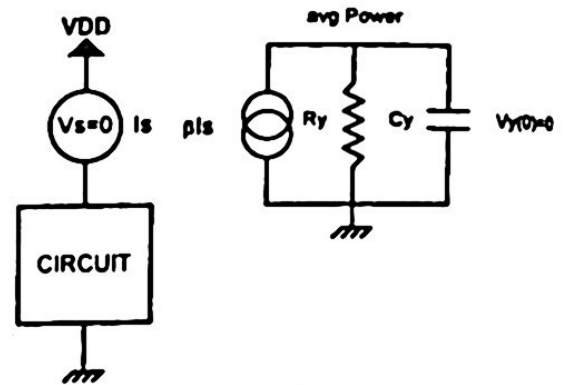


Fig. 11. Power meter

7 Summary and Conclusions

Power and timing evaluation is an important factor on evaluation in new proposals, in this work we evaluated traditional Issue Queue instruction wakeup mechanisms and compared its results with a direct wakeup scheme proposed in [7], Table 8 shows results of these evaluations. In order to evaluate the total impact of structures in timing [IQ32/IQ64] and power (IQ32/64IQ) - for instruction issue queue for 32- and 64-entries respectively - On overlapped access to different structures maximum time delay is considered, but power is supplementary.

The average access power was computed taking into account all tree components of IQ operation: Allocation, issue, and Wakeup for all three mechanisms presented, values in brackets represent access time and values in parentheses represent access power to mentioned structures, for 32 and 64 entries respectively.

For CAM wakeup mechanism: on allocation stage writes of two source operands in CAM4W6C's [42.80ps/55.50ps] (18.80mW/25.50mW) and writes of the complete instruction in SRAM IQ are performed [56.50ps/71.50ps] (6.14mW/9.85mW). On Wakeup stage, comparison on two structures CAM are performed [73.9ps/80.0ps] (47.15mW/95.27mW) one for each operand in the IQ. On Issue stage instructions are read from SRAM IQ [22.5ps/22.5ps] (11.50mW/13.75mW).

For Dependency Matrix Wakeup mechanism: on allocation stage comparison of destination register are performed in one structure CAM4W8C [48.60ps/56.30ps] (76.95mW/153.55mW), writes of the complete instruction in SRAM IQ [56.50ps/71.50ps] (6.14mW/9.85 mW) and writes of dependency bits on matrix at time are performed [48.60ps/65.50ps] (6.94mW/19.00mW). On issue stage instructions are read from SRAM IQ [22.5ps/22.5ps] (11.50mW/13.75mW) and vector bit of matrix is read [22.5ps/22.5ps] (8.90mW/ 22.00mW) to wakeup instruction on next cycle.

For Direct Wakeup mechanism: we consider that in decode stage is known the address for allocation of instructions in the queue and processor check the register file for ready operands indexing it with the source operands of instructions. Then, on allocation stage writes of IQ-pointers in the MT [56.5ps/56.5ps] (3.028mW/3.49mW) after writes of M-pointers in the MWT 8-entries [25.6ps/25.6ps] (7.28mW/10.53mW) and writes of the complete instruction in SRAM IQ [56.50ps/71.50ps] (6.14mW/9.85mW) are performed, note that decode for write access is not necessary. On wakeup stage MT is indexed [32.00ps] (8.50mW) with the result-registers for read [45.0ps/45.0ps] (3.95/4.09mW) the IQ-pointer and MWT [22.00ps/22.00ps] (8.93mW/15.05mW) is indexed subsequent if instructions with multiple dependents are executed. On issue stage instructions are read from SRAM IQ [22.5ps/22.5ps] (11.50mW/13.75mW).

Table 8, shows the power and cycle time for all three models of instruction issue queue evaluated, cycle time and power is considered for each stage of issue logic, Direct Wakeup Scheme requires the longest cycle time for allocation and wakeup due for sequential access to the MWT, but this stages use half clock cycle. With Direct Wakeup Scheme save 67%/80% of total energy dissipated in a 32/64 CAM Wakeup Scheme.

Table 8. Issue Queue timing and power for 70nm technology							
IQ entries	CAM Wakeup						
	Allocation (ps)	Avg Power _A (mW)	Issue (ps)	Avg Power _I (mW)	Wakeup (ps)	Avg Power _W (mW)	Avg Power _T (mW)
32	56.5	43.74	22.5	11.50	73.90	94.30	149.54
64	71.5	120.65	22.5	13.75	80.0	190.54	325.14
IQ entries	Dependency Matrix Wakeup						
	Allocation (ps)	Avg Power _A (mW)	Issue (ps)	Avg Power _I (mW)	Wakeup (ps)	Avg Power _W (mW)	Avg Power _T (mW)
32	56.5	90.03	22.5	11.50	22.5	8.90	110.70
64	71.5	182.40	22.5	13.75	22.5	22.00	218.15
IQ entries	Direct Wakeup						
	Allocation (ps)	Avg Power _A (mW)	Issue (ps)	Avg Power _I (mW)	Wakeup (ps)	Avg Power _W (mW)	Avg Power _T (mW)
32	82.1	16.44	22.5	11.50	99.0*	21.35*	49.32
64	97.1	23.67	22.5	13.75	99.0*	27.64*	65.26

*In this table we are considering the worst case, this is like all instructions having multiple dependents

References

- [1] Ron Ho, Kenneth W. Mai, Mark A. Horowitz "The Future of Wires" proceeding of IEEE Vol.89, No.4, April 2001.
- [2] Vikas Agarwal, Stephen W. Keckler and Doug Burger "The effect of technology Scaling on Micro-architectural Structures" Tech Report TR2000-02
- [3] S. R. Kunkel and J. E. Smith "optimal pipelining in supercomputer" Proceeding of 13th International Symposium on Computer Architecture, 1986
- [4] Masahiro Goshima Kengo Nishino, Yasuhiko Nakashima, Shin-ichiro Mori, Tashiaki Kitamura and Shinji Tomita "A High Speed Dynamic Instruction Scheduling Scheme for Superscalar Processors" proceeding of 34th ACM/IEEE International Symposium on Microarchitecture, December 1-5, 2001, pag. 225-236
- [5] Daniel Folegnani and Antonio Gonzalez, "Energy Effective Issue Logic", Proceedings of 28th Annual of International Symposium on Computer Architecture, 2001. Page(s): 230-239, Göteborg Sweden.
- [6] Marco A. Ramírez, Adrian Cristal, Alexander V. Veidenbaum, Luis Villa and Mateo Valero "A Simple Low-Energy Instruction Wakeup Mechanism" Proceeding of 5th International Symposium on High Performance Computing, Tokio-Odaiba, Japan. October 2003.
- [7] Marco A. Ramírez, Adrian Cristal, Alexander V. Veidenbaum, Luis Villa and Mateo Valero "Direct Instruction Wakeup for OOO processors" Proceeding of International Workshop on Innovative Architecture for Future Generation High Performance Processor and Systems IWIA-2004.

- [8] Sun Mo Kang, "*Accurate Simulation of Power Disipation in VLSI Circuits*", IEEE Journal of Solid-State Circuits, vol. SC-21, No. 5, October 1996.
- [9] Gregory J. Fisher, "*An Enhanced Power Meter for Spice2 Circuit Simulation*", IEEE Transaction on Computer-Aided Design. Vol.7 No. 5 May 1988.

Programming Languages

Prepare Yourself... The Tiger Will Be Unleashed Soon

María Lucía Barrón Estrada¹, Ryan Stansifer², and Ramón Zatarain Cabada¹

¹Instituto Tecnológico de Culiacán, Av. Juan de Dios Batiz s/n Col. Guadalupe, Culiacán, Sin.
80220 México Tel. 667-713 3804

mbarron@itc.cu, rzc777@hotmail.com

²Florida Institute of Technology, 150 W. University Blvd. Melbourne, FL, 32901 USA
ryan@cs.fit.edu

Abstract. The Java programming language has been, for the last several years, in the process to adding many features. The newest version of Java is called Tiger and will include new features to simplify software development. The watchwords of the Java 2 Standard Edition (J2SE) version 5 are "*easy of development*" and "*easy to use*". The most significant addition to the language is the ability to create and manipulate generic types. Despite major changes the language syntax had to be changed only slightly and the Java virtual machine remains the same, thus insuring compatibility with existing code. Others features in Tiger allow developers to annotate programs and reduce the amount of code they need to write. The new features of Java will have a major impact in the use of Java and in the curricula of Computer Sciences and Informatics. In particular, Java's generics will put parametric polymorphism back into the toolkit of students, teachers, and practitioners. In this paper, we describe several of the changes in Java and provide examples illustrating their use.

Keywords: Java, generic types, typesafe enums, foreach, programming, data structures.

1 Introduction

Java [AG 98] is one of the most popular programming languages. The balance of simplicity and power that Java offers allows developers to simplify the construction of robust and maintainable programs. However there is room for improvement. Java has been constantly evolving since it catapulted in the programming arena in 1995. After its first release, nested classes were added, event handling in GUIs was revised, numerous speciality APIs were added, and the collection class hierarchy was designed. Many changes to Java are requested by users through the Java Community Process. One of the most requested features is some sort of parametric polymorphism (generics). A formal document [JSR014] outlines the proposal. With the next release of Java, Tiger, the proposal becomes reality.

Adding parametric polymorphism to Java has been the focus of many researchers as well [BCK+ 03, BOSW 98a, BOSW 98b, CS 98, MBL 97, and OW 97]. After many proposals

were analyzed, a new version of the language including this and other features is now under review process. Java 1.5 is named Tiger, several beta versions have been released since last year; the latest version is beta 3, which is available for developers since July 19, 2004. Sun plans to release the final version later this year.

Many universities all over the world have adopted Java as the programming language to be taught in several courses of different curricula. The new features included in Java 1.5 will impact this process in several ways. Some of these additions will release the programmer of writing repetitive code that could be automatically generated. Other features will support the "*easy of development*" theme providing enumerations, autoboxing, enhanced *for* loops, and static import. All these changes are described in [JSR 201].

In this paper we present a list of features that are part of the new version of Java. We show some examples of code in the current version y contrast them with code in Tiger. The paper is organized as follows: section 2 explains the benefits of static import, section 3 describes the automatic conversion of primitive types to they corresponding reference types (wrapper classes), this process is known as autoboxing. Section 4 describes the new form of *for* loop designed to iterate over collections. Section 5 explains the enumeration facility provided in the language and its differences with respect to other languages. The most important feature added in this version is presented in section 6, it contains three subsections describing how to use generics, how to implement generics and the problems associated to them. Metadata, a declarative way of programming using annotations is shown in section 7. Section 8 shows an example of a procedure with a variable number of arguments. API's to get formatted input/output are presented in section 9. Finally conclusions are offered in section 10.

2 Static Import

In the current version of Java, class members defined as static can be imported by other classes that want to use them but it is necessary to refer to them using fully qualified names (*ClassName.member* or *ClassName.member()*). In other languages like Pascal or C, imported members can be directly referred to without using qualifications.

Java 1.5 includes variants of the import statements to allow importation of static members (fields and methods) in the same way classes and interfaces can be imported.

Example of the two new import static declarations

```
import static TypeName.Identifier;
import static TypeName.*;
```

In the example shown above the first declaration imports into the current unit the specific *Identifier*, which must be a static member of the class or interface named *TypeName*. The second declaration should import into the current unit all the static members (fields and methods) of the class or interface named *TypeName*.

Next we show an example of the use of static import declarations in Java 1.5 and contrast it with the current version of Java.

Current version

```
import java.lang.Math;
y = Math.sqrt(r) * Math.PI;
```

Tiger Java 1.5

```
import static java.lang.Math.*;
y = sqrt(r) * PI;
```

3 Automatic Boxing and Unboxing

Frequently programmers need to wrap primitive types into objects. Java provides wrapper classes that can be used by the programmer to convert primitive types into reference types. However this is a process that could be automatically done by the compiler and it is commonly known as *boxing*. Java 1.5 offers this feature facilitating the integration of generic types into the language.

The rules for boxing are straightforward. A value of a primitive type is converted into its corresponding reference type. Corresponding types are shown in table 1. An opposite process, known as unboxing, converts a reference type into a value of its corresponding primitive type.

Table 1. Correspondence between reference types and primitive

Primitive type	boolean	byte	double	short	int	long	float
Reference type	Boolean	Byte	Double	Short	Integer	Long	Float

An example of the use of automatic boxing and unboxing is presented in the code fragment below. A code fragment with the same functionality using the actual version of Java is presented before.

Current version. Wrapping primitive type into reference types

```
public static void main (String [] args) {
    Stack x = new Stack();
    x.push(new Integer(17));.           // wrap 17
    Integer y = (Integer) x.pop();
    int num = y.value(); // get value
}
```


Tiger Java 1.5. Using automatic boxing/unboxing with parameterized classes

```
public static void main(String args[ ] ) {
    Stack<Integer> x = new Stack<Integer>();
    x.push(17);                // boxing
    Integer y = x.pop();
    int num = y;                // unboxing
}
```

4 Foreach

An iterator provides a mechanism to navigate sequentially through a collection of elements. Using the current version of Java, boilerplate code is needed to iterate through a collection defining explicitly an iterator. This is shown in the first code fragment below. Tiger provides a new form of *for* loop commonly known as *foreach* loop. The *foreach* loop reduces the need of boilerplate iteration code, and simplifies the code reducing the chances of errors. The code presented in the right part of figure 4 is using a generic type to define a type argument of a Collection received as argument. This code is based on a similar example presented in [BG 03]. As we can appreciate iterating over the elements of the collection is simpler shorter and safer in the code using Tiger than the code using the current version. Both code fragments are shown below. The enhanced for loop can be used to iterate through array elements also.

Current version. An example of iterating over a collection.

```
void cancelAllElements(Collection c) {
    for( Iterator Ii = c.iterator(); i.hasNext() ; ) {
        TimerTask timer = (TimerTask) i.next();
        timer.cancel();
    }
}
```

Tiger Java 1.5. An example of foreach using generics and collections

```
void cancelAllElements(Collection<TimerTask> c) {
    for( TimerTask timer : c)
        timer.cancel();
}
```

5 Typesafe Enumerated Types

Enumerations were widely known in Pascal and C/C++. This construct allows the definition of names to help document and clarify code. Java didn't have an enumeration

construct and programmers were forced to workaroud to create this pattern. This is not an easy task if type safety is required and many programmers fail to implement the correct pattern. In his book [B 01] Bloch discuss the importance of implement a correct enumeration but the amount of code needed increases the chances of errors. Tiger offers a typesafe enum facility that combines power, performance and it is easy to use. Enums have all the advantages of the typesafe enum patter described in [B 01]. Enums are a special kind of classes. They introduce a new keyword into the language. Its simplest definition looks like a C/C++ enum declaration, but it doesn't have its disadvantages. Figure 5 shows two examples of enum declarations.

Example of the typesafe enumeration pattern.

```
public class City {
    private final String name;
    private Suit(String name) { this.name = name; }
    public String toString() { return name; }
    public static final City MIAMI = new City("Miami");
    public static final City ORLANDO = new City("Orlando");
    public static final City MELBOURNE = new
City("Melbourne");
}
```

Two examples of the type safe enumeration facility provided in Tiger.

```
public enum City { Miami, Orlando, Melbourne; }
public enum Ticket {
    Plateau(1), General(5), Number(10), VIP(100);
    Ticket(int value) { this.value = value; }
    private final int value;
    public int value() { return value; }
}
```

Class modifiers in enum declarations contain restrictions. Some of them are: all enum declarations are implicitly final and can not be abstract unless they contain constant-specific class bodies for every constant, and members of enum classes are implicitly static.

6 Generic Types

The absence of generics in Java forces the programmer to use subtype polymorphism to workaroud writing code with cast operations that could fail at runtime. Tiger provides generics enhancing the expressiveness of the language and improving safety because more errors can be detected at compile time. A detailed description of generics can be found in

[B 04]. Generic types in Java are not like templates in C++. They are compiled once and for all, they are type checked at compile time and they are translated into a homogenous piece of code.

6.1 Generic Libraries

Generic types are widely used in the Collections API. In Tiger, collection types like `List` or `ArrayList`, are part of the Collection API and can be parameterized by a type to specify the type of elements the collection will contain. An example of the `List` class is presented next in both version.

Current version

```
List xs = new LinkedList();
xs.add(new Integer(0));           // wrapping needed
Integer x = (Integer) xs.iterator().next(); // cast needed
```

Example of an instantiation of a generic class in Tiger Java 1.5

```
List<Integer> xs = new LinkedList<Integer>();
xs.add(0);           // automatic boxing
Integer x = xs.iterator().next(); // no cast
```

6.2 Implementing Generic Types

It is easy to define generic classes and interfaces. A generic class definition contains a list of type parameters after the class identifier. The code fragment below shows an example of a generic class with two type parameters. The main function of this code fragment shows an example of how instantiate and use generic classes.

A generic class declaration and instantiation.

```
public class Pair<T,U> {
    private T first;
    private U second;

    public Pair(T f, U s) {
        first = f;
        second = s;
    }

    T getfirst () { return first; }
    U getsecond() { return second; }
    String toString() {...}
    ...
}
```

```

public static void main(String args[ ] ) {
    Pair<Integer, String> x = new Pair<Integer, String>
(1, "Aaron");

    List<String> l = new List<String>();

    int tot = x.getFirst() + 1;
    l.add(x.getsecond());
}
}

```

6.3 Problems with generics

The translation approach used for generic classes requires that the type parameters must be reference types. Primitive types cannot be used to instantiate generic classes or interfaces. Tiger provides a feature to ameliorate this problem, automatic boxing and unboxing of primitive types. F-bounded polymorphism is used to define constrained type parameters but it cannot be combined smoothly with inheritance in the presence of binary methods as noted in [BS 03].

7 Metadata

This feature allows annotating classes, interfaces, fields, and methods as having particular attributes. The current version of Java has a limited implementation of metadata using tags. The new version of Java contains six built-in annotations and users can define custom made annotations to decorate their types controlling their availability. With this new feature developers are going to avoid writing boilerplate code that may be automatically generated and updated by tools maintaining all the information is the source file.

7.1 Built-in annotations

The built-in annotations available in the new version of Java are:

- a) `java.lang.Overrides` - indicates that a method declaration in a class intent to override a method from its superclas.
- b) `java.lang.annotation.Documented` - indicates that javadoc documents the annotated element. It can be ignored by the tool.
- c) `java.lang.annotation.Deprecated` - The java compiler can warns the user if he/she uses the annotated element.
- d) `java.lang.annotation.Inherited` - A class decorated with an annotation that contains the annotation `Inherit`, will inherit the annotation to all derived classes.

- e) `java.lang.annotation.Retention` - used to determine the annotation availability.
- f) `java.lang.annotation.Target` - indicates to which kind of element (class, method, or field) the annotation is applicable.

An example of an annotation is shown in the code fragment below.

Current version

```
public interface PingIF extends Remote {
    public void ping() throws RemoteException;
}

public class Ping implements PingIF{
    public void ping() { ... }
}
```

An example of metadata annotations using Tiger Java 1.5

```
public class Ping {
    public @Remote void ping() { ... }
}
```

8 Variable Arguments

In the actual version of Java, the number of parameters in a method is fixed. Generally when a method needs an arbitrary number of parameters, it uses an array to store all the arguments. The new version of Java allows defining methods with a variable number of arguments without using arrays to store them. The code fragments below show an example of this in both the current version and Tiger.

Current version

```
Object [ ] arguments = {
    new Integer(7),
    new Date(),
    " a disturbance in the force"
}

String result = MessageFormat.format (
    "At {1,time} on {1,date}, there was {2} on planet" +
    "{0,number,integer}.",    arguments);
```

reused without recompilation. In this regard the big problem was with the collection classes. The new generic collection classes are translated in a way that produces the same Java bytecode as the old (non-generic) collection classes. This is not without its own drawbacks, however.

The collections classes: lists, sets, dictionaries, and so on, are by their nature polymorphic. With generics in Java it will be possible to program these data structures naturally without attention to the type of the elements. This should be a great help in the instruction of data structures using the Java programming language.

Automatic boxing and unboxing also contributes to a more natural view of data structures by erasing the difference between, say, stacks of integers and stacks of strings.

Input and output has been greatly improved. Although quite logically designed, I/O has been cumbersome in Java. It has been the cause of much frustration to developers, students, and instructors alike. The new I/O API with its C style formatted output will make common tasks like printing numbers in columns simple again.

References

1. [AG 98] Ken Arnold and James Gosling. *The Java™ Programming Language*. Addison Wesley. 1998
2. [B 01] Joshua Bloch. *Effective Java™* Addison Wesley. 2001.
3. [B 04] Gilad Bracha. Generics in the Java Programming Language. March 9, 2004. Available online at <http://java.sun.com/j2se/1.5.0/lang.html>
4. [BCK+ 03] Gilad Bracha, Norman Cohen, Christian Kemper, Martin Odersky, David Stoutamire, Kresten Thorup, and Philip Wadler. Adding generics to the Java™ Programming Language: Public Draft Specification, Version 2.0. June 23, 2003. Available online at <http://java.sun.com>
5. [BG 03] Joshua Bloch and Neal Gafter. Forthcoming Java™ Programming Language Features. Presentation in JavaOne Conference. June 2003.
6. [BOSW 98a] Gilad Bracha, Martin Odersky, David Stoutamire, and Philip Wadler. GJ: Extending the Java Programming Language with type parameters. Manuscript, March 1998, revised August 1998.
7. [BOSW 98b] Gilad Bracha, Martin Odersky, David Stoutamire and Philip Wadler. Making the future safe for the past: Adding Genericity to the Java Programming Language (GJ) In *Proceedings of OOPSLA '98, Conference on Object-Oriented Programming, Systems, Languages and Applications*, Vancouver, British Columbia, Canada. October 1998.
8. [BS 03] Maria Lucia Barron Estrada and Ryan Stansifer. Inheritance, Genericity and Binary Methods in Java. In *Computacion y Sistemas*, Volumen VII, numero 2. Mexico, DF. December 2003.
9. [CS 98] Robert Cartwright and Guy L. Steele. Compatible Genericity with Run-Time Types for the Java™ Programming Language. (NextGen) In *Proceedings of OOPSLA '98, Conference on Object-Oriented Programming, Systems, Languages and Applications*, Vancouver, British Columbia, Canada. October 1998.

reused without recompilation. In this regard the big problem was with the collection classes. The new generic collection classes are translated in a way that produces the same Java bytecode as the old (non-generic) collection classes. This is not without its own drawbacks, however.

The collections classes: lists, sets, dictionaries, and so on, are by their nature polymorphic. With generics in Java it will be possible to program these data structures naturally without attention to the type of the elements. This should be a great help in the instruction of data structures using the Java programming language.

Automatic boxing and unboxing also contributes to a more natural view of data structures by erasing the difference between, say, stacks of integers and stacks of strings.

Input and output has been greatly improved. Although quite logically designed, I/O has been cumbersome in Java. It has been the cause of much frustration to developers, students, and instructors alike. The new I/O API with its C style formatted output will make common tasks like printing numbers in columns simple again.

References

1. [AG 98] Ken Arnold and James Gosling. *The Java™ Programming Language*. Addison Wesley. 1998
2. [B 01] Joshua Bloch. *Effective Java™* Addison Wesley. 2001.
3. [B 04] Gilad Bracha. Generics in the Java Programming Language. March 9, 2004. Available online at <http://java.sun.com/j2se/1.5.0/lang.html>
4. [BCK+ 03] Gilad Bracha, Norman Cohen, Christian Kemper, Martin Odersky, David Stoutamire, Kresten Thorup, and Philip Wadler. Adding generics to the Java™ Programming Language: Public Draft Specification, Version 2.0. June 23, 2003. Available online at <http://java.sun.com>
5. [BG 03] Joshua Bloch and Neal Gafter. Forthcoming Java™ Programming Language Features. Presentation in JavaOne Conference. June 2003.
6. [BOSW 98a] Gilad Bracha, Martin Odersky, David Stoutamire, and Philip Wadler. GJ: Extending the Java Programming Language with type parameters. Manuscript, March 1998, revised August 1998.
7. [BOSW 98b] Gilad Bracha, Martin Odersky, David Stoutamire and Philip Wadler. Making the future safe for the past: Adding Genericity to the Java Programming Language (GJ) In *Proceedings of OOPSLA '98, Conference on Object-Oriented Programming, Systems, Languages and Applications*, Vancouver, British Columbia, Canada. October 1998.
8. [BS 03] Maria Lucia Barron Estrada and Ryan Stansifer. Inheritance, Genericity and Binary Methods in Java. In *Computacion y Sistemas*, Volumen VII, numero 2. Mexico, DF. December 2003.
9. [CS 98] Robert Cartwright and Guy L. Steele. Compatible Genericity with Run-Time Types for the Java™ Programming Language. (NextGen) In *Proceedings of OOPSLA '98, Conference on Object-Oriented Programming, Systems, Languages and Applications*, Vancouver, British Columbia, Canada. October 1998.

10. [JSR 014] Sun Microsystems. Adding Generic Types to the Java™ Programming Language. Java Specification Request JSR-000014, 1998. [Online] URL <http://www.jcp.org/en/jsr/detail?id=14> Approved in 1999.
11. [JSR 176] J2SE™ 5.0 (Tiger) Release Contents <http://www.jcp.org/en/jsr/detail?id=176>
12. [JSR 201] Sun Microsystems. Extending the Java Programming Language with Enumerations, Autoboxing, Enhanced for loops and Static Import. Java Specification Request JSR 201. Available online at <http://jcp.org/en/jsr/detail?id=201>
13. [MBL 97] Andrew C. Myers, Joseph A. Bank, and Barbara Liskov Parameterized Types for Java (PolyJ). In *Proceedings 24th ACM Symposium on Principles of Programming Languages*, pages 132-145, Paris, France, January 1997.
14. [OW 97] Martin Odersky, Philip Wadler. Pizza into Java: Translating theory into practice. In *Proceedings 24th ACM Symposium on Principles of Programming Languages*, pages 146–159, Paris, France, January 1997.

Implementing a CPS Compiler for Functional Languages with Zero Overhead Exception Handling

Ramón Zatarain Cabada,¹ Ryan Stansifer,² and
María Lucía Barrón Estrada¹

¹Instituto tecnológico de Culiacán, Av. Juan de Dios Batiz s/n, Col. Guadalupe, Culiacán,
Sin. CP 80220 Mexico

rzatarain@itculiacan.edu.mx mharron@fit.edu

²Florida Institute of Technology, 150 W. University Blvd. Melbourne, FL. 30901 USA
ryan@cs.fit.edu

Abstract. We have implemented a basic CPS compiler for functional languages with exception handling. With it we were able to implement a new approach to exception handling and compare it side-by-side with the approach taken by the SML of New Jersey compiler. The new approach uses two continuations instead of the one continuation. One continuation encapsulates the rest of the normal computation as usual. A second continuation is used for passing the abnormal computation. The new approach and an experiment were shown in [9,10] where we concluded that programs with exception handling using our new approach do not produce overhead. In this paper we show more details about the compiler. This includes lambda, CPS and abstract machine code generation. At the end, we show some results of the experiments.

1 Introduction

Functional languages focus on data values described by expressions (built from function applications and definitions of functions) with automatic evaluation of expressions. Programs can be viewed as descriptions declaring information about values rather than instructions for the computation of values or of effects [11]. ML [8] is a general-purpose functional programming language designed for large projects. Every expression has a statically determined type and will only evaluate to values of that type. Standard ML of New Jersey (abbreviated SML/NJ) is a compiler and programming environment for ML written in ML with associated libraries, tools, and documentation [3]. The compiler translates a source program into a target machine language program in several phases. Compilers of imperative languages like Java, Ada, and C++ implement exception handling without imposing overhead on normal execution [4, 5, 6]. When a program defines an exception handler, the runtime performance of that program would be the same without exception handler definition. We can say that there is no runtime penalty for defining an exception handler, which is never used. In other words, no runtime overhead occurs in the case in which no exceptions are raised. However, compilers of

functional languages like SML/NJ or CAML [7] produce code that has exception-handling overhead [10].

2 A Model of CPS Translation

The middle part and key transformation in some functional language compilers is the conversion to CPS (continuation-passing style) language. We use CPS as our intermediate representation in our functional language compiler (figure 1). The compiler first translates a source program written in lambda code into CPS code. Then, it translates the CPS code into a one-function CPS program named flat CPS. Finally, flat CPS code is translated into abstract machine code (AMC).

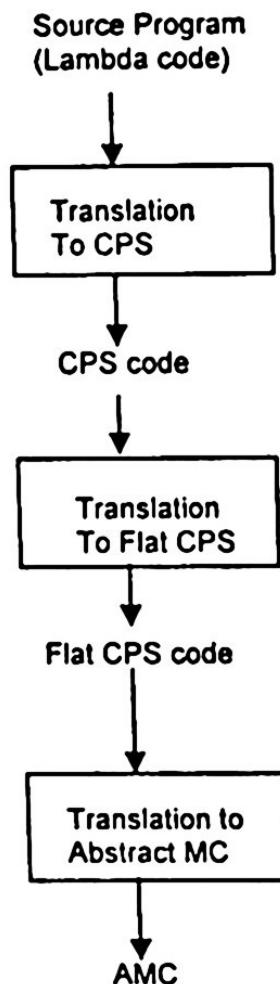


Fig. 1. Overview of the compiler

2.1 The Lambda Language

Our source language is written using lambda code. The next SML code shows the definition of a lambda expression. We start here to avoid issues of parsing a concrete source language.

```

datatype lexp=
  VAR of var | INT of int | STRING of string
  | FN of var * lexp
  | FIX of var list * lexp list * lexp
  | APP of lexp * lexp
  | PLUS | SUB | MULT | LESS | EQ
  | MAKEREF | RAISE of lexp
  | HANDLE of lexp * lexp
  | COND of lexp * lexp * lexp

```

In this case, each value (a constructor) of type `lexp` can represent:

- A variable (VAR), an integer (INT), or a string (STRING).
- An anonymous (lambda) function (FN).
- A function declaration (FIX) where function names (var list) are bound to anonymous functions (lexp list) under the scope of a lambda expression.
- A function-calling construct (APP).
- A set of primitive operations for making arithmetic (PLUS, SUB, and MULT); comparisons (LESS, and EQ); and creation of references to memory (we use them when exceptions are declared).
- A primitive operation to evaluate an expression of type **exception** and to throw a user-defined or system exception (RAISE).
- A primitive operation **HANDLE** that evaluates the first argument, and if an exception is raised, then applies the second argument (handler) to the exception.
- A primitive operator **COND** used to test conditions **EQ** and **LESS**. Besides normal testing, this primitive is very important when a **HANDLE** tests for a given exception.

2.2 The CPS Language

The CPS language used in our translator has three big differences with respect to those traditional compilers, which use also CPS as an intermediate representation [2]:

- Every function has a name.
- There is an operator for defining mutually recursive functions (instead of fixed point function).
- There are n -tuple primitive operators.

Besides that, we use the ML datatype declaration in order to prohibit ill-formed expressions. One important property of CPS is that every intermediate value of a computation is given a name. This makes easier the translation later, to any kind of machine code. For example the SML expression $289 - (17 * 17)$ is translated to

```

PRIMOP(*,[INT 17,INT 17],[w2],
  [PRIMOP(-,[INT 289,VAR w2],
    [w1],[APP (VAR k,[VAR w1])])])

```

in CPS notation, where `w1` and `w2` are intermediate names produced by the translator.

The next SML code shows the definition of a CPS expression.

```
datatype primop=
  gethdlr | sethdlr | + | - | * | < | equal | makeref
type var=string
datatype value =
  VAR of var | INT of int | STRING of string
datatype cexp=
  APP of value * value list
  | FIX of (var * var list * cexp) list * cexp
  | PRIMOP of primop * value list * var list * cexp list
```

A primitive operator can be:

- **gethdlr** and **sethdlr**. Both are used for handling exceptions. The operator **gethdlr** obtains the current exception handler (or saving the old handler), and **sethdlr** updates the store with a current handler (re-install a new handler).
- **+**, **-**, *****. Arithmetic operators for adding, subtracting, and multiplying two arguments.
- **<**, **equal**. Testing (comparison) operators for less than and equal to.
- **makeref**. This operator is used to create a reference (a pointer) to memory (specially for declaring an exception).

Datatype **value** is defined as all the different kind of atomic arguments that can be used in a CPS operator. A value or argument can be a variable (**VAR**), an integer (**INT**), or a string constant (**STRING**).

Our CPS language has just three different kinds of expressions. They are:

- **APP**. It is used for calling a function (whose name is of type **value**), passing one or more arguments (using a list of values).
- **FIX**. Like we established before, in CPS all functions have a name. There are no anonymous functions. **FIX** is used to define a general-purpose mutually recursive function definition. The syntax of **FIX** defines a list of zero or more functions, with a name (type **var**), arguments (type **var list**), and bodies (type **cexp**). All of these functions can be called (using the **APP** operator), from each body of the function or from the main body of the **FIX** expression (type **cexp**).
- **PRIMOP**. This stands for **primitive operator**. All primitives like handling exception, arithmetic, testing, and references, are built by using this constructor. The first field is the primitive name (**primop** type), the second and third fields are used for arguments and/or result names, and the fourth field is the continuation expression of the primitive operator.

2.3 The Abstract Machine Code (AMC)

Continuation-passing style is used because it is closely related to Church's lambda calculus and to the model of von Neumann, here represented by our target abstract machine language (see figure 2). Each operator of the CPS corresponds to one opera-

tor in our target abstract machine code. In order to test the performance of the CPS code we implemented an abstract machine.

The machine has an instruction set, a register set and a model of memory, and executes programs written in abstract machine code. Figure 2 illustrates the components of the abstract machine.

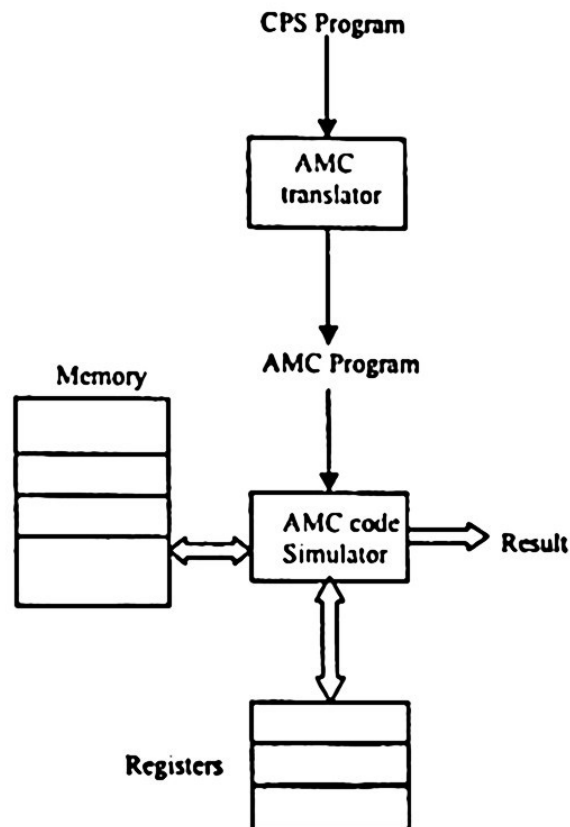


Fig. 2. Components of the abstract machine

During the compilation process, lambda expressions are translated into corresponding CPS expressions. Then, CPS expressions are translated into abstract machine code.

The AMC is essentially an assembly-language code, and like any abstract machine it has some advantages with respect to a real machine: first, creating a simulator for the abstract machine is no big deal; and second making performance analysis is very convenient (we can include code for performance analysis inside the machine). The next SML code shows the definition of the format for AMC instructions and expressions.

```

datatype instruction =
  LABEL of string
  | JUMP of string
  | CJUMP of relop * exp * exp * string * string
  | LOAD of exp * exp
  | STORE of exp * exp
  
```

```

|ADD of exp * exp * exp
|SUB of exp * exp * exp
|MUL of exp * exp * exp

```

```

and      exp=
          MEM of string
          |NAME of string
          |CONST of int
          |STRING of string
          |REG of int

```

```

and      relop=  EQ | LT

```

An abstract machine instruction can be:

- A label that represents an address. Whenever the simulator finds a label it just increases the program pointer, in order to read the next instruction.
- A jump instruction is an unconditional branch to any label.
- A CJUMP is a conditional jump to one of two labels, depending of the result of the test.
- A load from memory into a register.
- A store from a register or a string into a memory address.
- Arithmetic operations to add, subtract, or multiply two values, producing a result, which is stored into memory.

and a expression can be:

- An address of memory represented by a name (string) of a register, variable, etc.
- The name of a label, which represents an address.
- A constant for an integer data.
- A string data.
- A register (registers have a unique number).

Example:

We illustrate all the different code representations with a complete program in SML, Lambda, CPS, and abstract machine code.

SML

```

let
  fun f(x) = x*5
in
  f(4)
end

```

LAMBDA

```

FIX(["f"], [FN ("x", APP(b.MULT, RECORD [VAR "x", INT 5]))],
      APP(VAR "f", INT 4))

```

CPS**FIX**

```

(((("f",["x","w1"],
  PRIMOP (*,[VAR "x",INT 5],["w2"],
    [APP (VAR "w1",[VAR "w2"])]))),
  FIX
    (((("r3",["x4"],APP (VAR "initialNormalCont",[VAR "x4"]))),
      APP (VAR "f",[INT 4,VAR "r3"])))

```

AMC

```

0          LOAD      Const 4,Reg 1
1          LOAD      Mem r3,Reg 2
2          JUMP      Name f
3      LAB f:
4          STORE     Reg 1,Mem x
5          STORE     Reg 2,Mem w1
6          MUL       Mem x,Const 5,Mem w2
7          LOAD      Mem w2,Reg 1
8          JUMP      Mem w1
9      LAB r3:
10         STORE     Reg 1,Mem x4
11         LOAD      Mem x4,Reg 1
12         JUMP      Mem initialNormalCont
13      LAB end:

```

The code in the AMC performs the following operations:

- Instructions 0 and 1 pass the parameters in registers 1 and 2.
- Instruction 2 is a jump to label f.
- Instructions 4 and 5 store both parameters (constant 4 and register 3) in memory.
- Instruction 6 multiplies first parameter (constant 4) by constant 5.
- Instruction 7 passes as a parameter the result of the multiplication in register 1.
- Instruction 8 is a jump to address r3 (the value of variable w1).
- Instruction 10 stores the parameter into memory address x4.
- Instruction 11 passes the value of x4 into register 1. This register always keeps the final result.
- Instruction 12 jumps to the initial continuation `initialNormalCont`, which is a fixed address or constant in memory that represents the end of any program executed in the abstract machine.

2.4 A Simulator of the AMC

The simulator is a program, which simulates a real computer. It is a piece of software that runs an AMC program. In order to simulate a real computer it uses three data structures, which mimic a memory for data values, a memory for code, and a set of registers (see figure 2). It also uses two variables that keep the current program pointer (PC) for the code, and the current stack pointer (SP) for the data. The main routine of the simulator is a recursive function that keeps reading instructions from the AMC

program. Next, we describe the algorithm used for the simulation of an AMC program.

Input: An AMC program, which is kept as a list of instructions.

Output: A value or result after executing the AMC program.

Method:

- Convert the instructions kept as a list into a different data structure: an array. The array is more convenient for the simulation.
- Initialize PC and memory pointers with initial address. PC points to first AMC instruction and memory pointer to address zero in memory.
- Start main function, which keeps reading instructions pointed by PC, executing the operations (storing, loading, jumping, adding, etc.), and updating the value of PC and memory.

3 A New Approach for Implementing Exception Handling

Implementing exception handling in SML/NJ and OCAML produce runtime overhead [10]. The implementation of our compiler generates also exception-handling overhead [9]. We have implemented a new technique for zero overhead exception handling. A more detailed explanation of the source of that overhead can be found in [9,10].

3.1 Experiments

This section presents one example of the experiments we made in order to test the performance of the new technique.

In section 2.4 we gave an explanation of the implementation of a simulator for a real machine. The simulator “executes” a program in abstract machine code, getting then, information like the number of instructions performed by that program. The abstract machine code can be produced by three different compilers. One using the original approach (the method used in the SML/NJ compiler), a second that uses a two-continuations approach, and a third that uses a one-continuation-displacement approach [9]. Some comparisons in the experiments are between programs that declare and use exception handlers; and some are between programs that declare and never use exception handlers which is, of course, the most important situation for our research. All measurement in the experiments are made by counting the number of instructions executed by the simulated programs.

The program for the experiment contains two functions: *f* and *run*. At the beginning function *run* is called passing a value of zero as a parameter. Function *run* loops 10 times (we later increase this value), calling function *f* and making computations. The result of the computation in function *f* never raises an exception so the handler *ovfl* is never evaluated. The program was translated by three different compilers: the SML/NJ compiler (old), our compiler with the one-continuation approach, and our compiler with the two-continuation approach

SML code

```

let
  fun f(n)=n*n handle ovfl=>n
  fun run(x)=if x>10 then f(17) else (run(x+f(17)-288))
in
  run(0)
end

```

We showed in [9] that this program produces exception handling overhead. Figures 3, 4, and 5 illustrate the performance of the program on its four different versions: with an exception handler in function *f*, with no exception handler (for the old and new approach), and with an exception handler using the two continuation approach [9,10]. The graphs show that the curves of version 1 and version 4 are "tied" or follow exactly the same direction. That means, that there is no exception handling overhead in the program that uses the one-continuation-displacement technique.

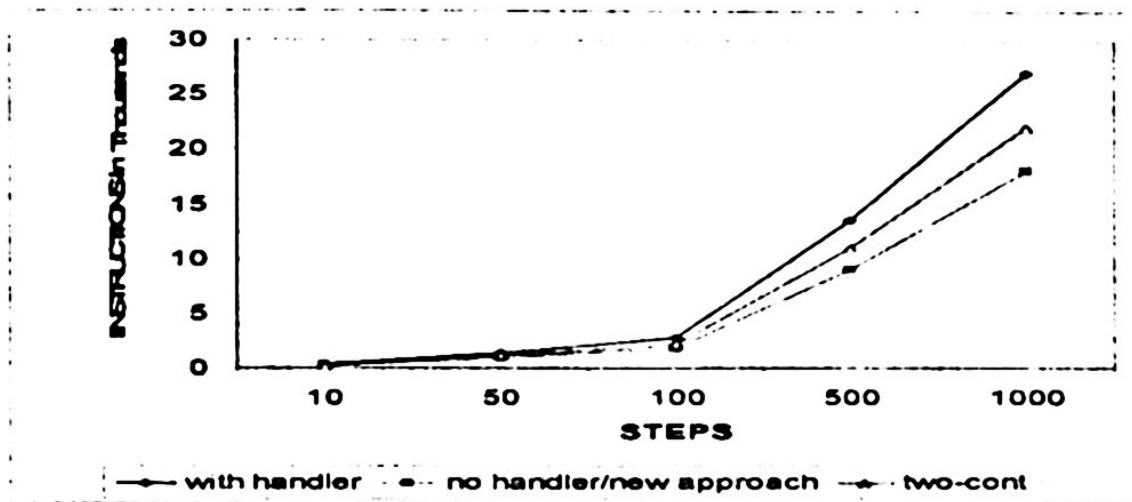


Fig. 3. Performance of program from 10 to 1000 steps

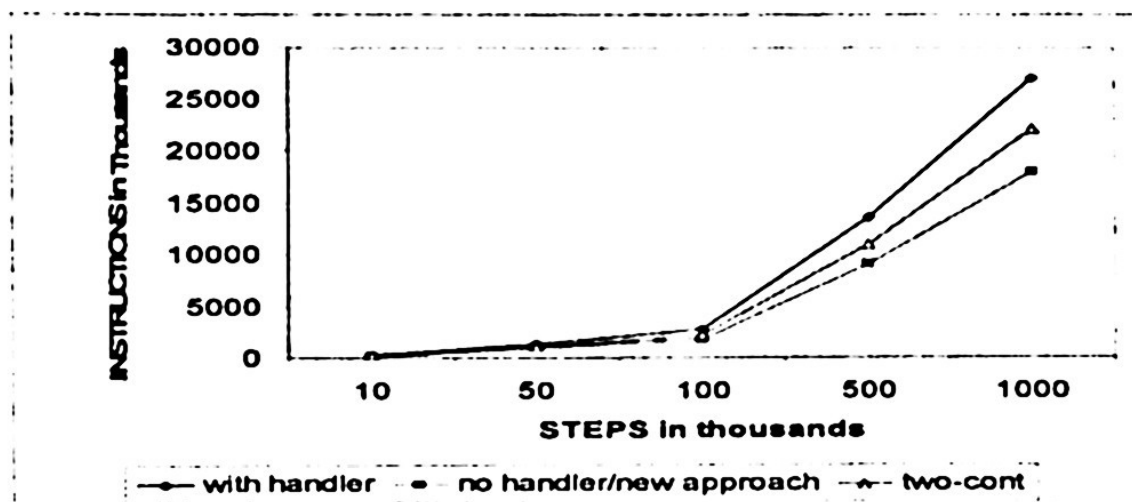


Fig. 4. Performance of program from 10000 to 1000000 steps

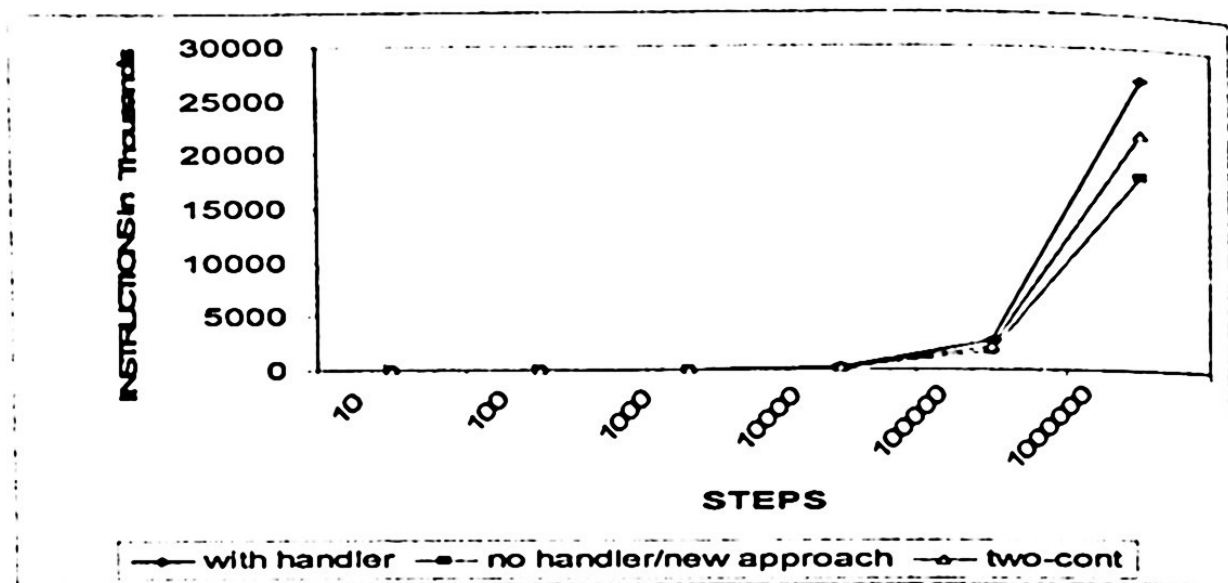


Fig. 5. Performance of program from 10 to 1000000 steps

4 Conclusions

We have implemented a basic CPS compiler for functional languages with exception handling. With it we were able to implement the new approach to exception handling and compare it with the approach taken by the SML of New Jersey compiler. Our model of translation and execution allows a programmer (or student/teacher) to write, translates, and executes programs in a source functional language (an extended lambda language) and a target abstract machine language. The model can be seen as a framework that can be used to execute programs, allowing studying a wide range of performance assessments. Last, we designed and implemented a new technique where the entire overhead is moved from the normal flow of control to the code executed when an exception is raised.

References

1. *American National Standard Programming Language PL/I*. ANSI X3.53-1976. American National Standards Institute, New York.
2. Andrew W. Appel. *Compiling with Continuations*. Cambridge University Press, Cambridge, England, 1992.
3. Andrew Appel and David B. MacQueen. A standard ML compiler. In Gilles Kahn, editor, *Functional Programming Languages and Computer Architecture*, pages 301-324. Springer-Verlag, Berlin, 1987. Lecture Notes in Computer Science 274; Proceedings of Conference held at Portland, Oregon.
4. Baker, T.P. and Riccardi, G. A. Implementing Ada Exceptions. *IEEE software*, September 1986, 42-5.

5. Dinechin, C. de. C++ Exception Handling. *IEEE Concurrency*, Vol. 8, No.4, October-December 2000, Pages 72-79.
6. Venners, B. Inside the Java 2 Virtual Machine. McGraw-Hill, second edition, 1999.
7. Leroy (Xavier), Rémy (Didier), Vouillon (Jérôme) et Doligez (Damien). — *The Objective Caml system release 2.04*. — Rapport technique, INRIA, décembre 1999.
8. Milner, R., Tofte, M., and Harper Jr., R.W. *The Definition of Standard ML*. MIT Press, Cambridge, Massachussetts, 1990.
9. Ramon Zatarain. Design and Implementation of Exception Handling with Zero Overhead in Functional Languages. *PhD Dissertation*, Florida Institute of Technology, 2003.
10. Ramon Zatarain and Ryan Stansifer. Exception Handling for CPS Compilers. *Proceedings of the 41st ACM Southeast Regional Conference (ACMSE'03)*, Savannah, Georgia.
11. Reade, Ch. *Elements of Functional Languages*. Addison-Wesley, 1989.

MOOL: an Object-Oriented Language with Generics and Modules

Maria Lucia Barron Estrada¹, Ramón Zatarain Cabada¹, and Ryan Stansifer²

¹Instituto Tecnológico de Culiacán, Av. Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán, Sin. CP 80220 México

mbarron@itc.edu

²Florida Institute of Technology, 150 W. University Blvd. Melbourne, FL. 30901 USA
ryan@cs.fit.edu

Abstract. Though most developers are concerned with only a few languages, programming languages continue to evolve. Researchers continue to develop new features and experiment with new combinations of features in order to design languages that are easier to program in. In this paper we describe the programming language called MOOL (Modular Object-Oriented language). MOOL is a simple class-based object-oriented language that supports generic programming by parameterized classes and interfaces [4]. It contains an independent module mechanism to allow the development of large programs in a safe manner. It is this combination of features that differentiates MOOL from C++, C#, and Java. A prototype compiler is under development that will translate MOOL to both the MSIL for .NET platform and bytecode for Java.

1 Introduction

There are many language features that programmers expect to find in modern programming languages. Every programming language provides some of them, usually in different combinations and variations. It is important to begin with a rough list of these features.

- *Control constructs
- *Concurrency
- *Input/output
- *Aggregation for structuring large programs
- *Generics/templates for convenient code-reuse
- *Inheritance for convenient code-reuse

We are not concerned with the first three in this paper. Rather we focus on the last three. Our Goal is to design an imperative, type-safe, class-based language with aggregation, generics, and inheritance. We want a language that is as simple and natural as possible.

We look at several of the major languages and how they approach aggregation, generics, and inheritance.

Ada [1] and SML [8] both have a well-defined module mechanism. Both support generic programming. Support for inheritance was added later to Ada.

Java [3] supports inheritance and classes and it is planned to add support for generics programming. The situation for C# [2] is similar.

Both Java and C# relegate aggregation to a minor role.

Only Modula-3 [9] was designed with all three (aggregation, generics, and inheritance) in mind from the beginning.

With respect to our goal all these language fall short.

Ada and Modula-3 are deficient (from our point of view) because objects/classes are subordinate.

SML is inspirational in many ways, in particular with universal polymorphism type reconstruction, and modules, but it is not a traditional, imperative language. And it does not have classes, though its cousin OCAML [10] does.

In Java, C#, C++ [11] modules play a reduced role. In these languages the class construct is overburdened.

Classes play many roles in languages that do not support another mechanism to structure programs [12]. The separation of classes and modules in two distinct elements allows using them independently; it also permits to use them to generate applications without forcing the developer to use a specific paradigm. However, separate classes and modules in two distinct constructs is not an easy task. Some language designers have accomplished this task but not without sacrificing other elements, i.e. MOBY [7] and OCAML [10]. These two languages are descendants of SML and they incorporate classes into the language carrying on all the elements previously defined in it.

The absence of parametric polymorphism in object-oriented languages like Java and C# has annoyed developers to some extent that both languages are in the process to get a mechanism to support parametric polymorphism. MOOL allows the definition of parameterized classes, class interfaces, methods and functions. Parameterized types and functions allow developers to abstract over types. This will enhance code reuse in a safe manner.

MOOL provides different constructs for classes and modules. Modules are containers, which are used to generate programs or code fragments. Classes on the other hand, are useful to create programs using the object-oriented style. Thus both imperative and object-oriented programs can be designed without forcing the developer to use any style. A complete definition of the language can be found at [4]. The grammar of MOOL is presented in BNF notation as an appendix in [4].

2 MOOL

MOOL - Modular Object-Oriented Language - is a simple, general-purpose, statically typed, class-based, object-oriented programming language. It provides a module construct with interface and implementation separated to create large programs, class construct to define and generate objects, and supports the definition of generic code using parameterized types.

MOOL provides two kinds of universal polymorphism. Parametric polymorphism is supported with parameterized types and type variables. Subtype polymorphism is

provided to be able to use an object of a subtype where an object of its supertype is expected.

There is only one kind of type in MOOL: reference types. There is a hierarchy of types defined with object at the top. Everything is a reference to an object of certain type. Automatic boxing and unboxing is provided to use the values contained in certain objects.

MOOL includes classes, inheritance, polymorphism, dynamic dispatch, and late binding to support the object-oriented programming style. It also includes functions to create procedural programs that do not require the use of objects.

2.1 Definitions

Program. A MOOL program is a set of compilation units. A compilation unit is either a *module interface* or a *module implementation*. A program specifies a sequence of statements to be executed in some order.

Identifiers. Identifiers are names used to define and refer to some elements in a program such as variables, functions, types, etc.

Expression. An expression is a construct in the language in which a combination of operators and operands specify a computation that produces a value.

Predefined types hold numeric or boolean values. They are integer, float, and boolean. There is also a special reference value called null. The user can create other reference types using classes, class interfaces, functions and arrays.

2.2 Module Interface and Implementation

A module is the basic unit to create a simple program or to create a code fragment that can be combined with other modules to create larger programs. Functions, classes and class interfaces are part of modules and cannot be defined independently.

Modules are static units to encapsulate elements, hide information and separate compilation. Modules contain two parts: a **module interface** that describes the signature of the module and the **module implementation** that contains the implementation of the signature. Modules define the namespace structure to refer to qualified names. They define two scopes: internal and external.

In this section we present the two parts of the module construct.

Module Interface

A module interface is a specification of the services a given module provides to others. A module interface reveals the public parts of a module. Information hiding can be achieved by restricting the interface to contain only a subset of the elements defined in the module implementation. By default all members of a module interface are public.

The language contains a module interface called *IMain*, which contains the *main* function. The *main* function receives an array of string elements and its result is void.

Any module may implement *IMain* providing code for the *main* function. The *main* function is the point where the program starts execution.

Example of the module interface *IMain* containing the definition of the *main* function.

```
module interface IMain {
    void main (String [] args);
}
```

Module Implementation

A module implementation contains the definition of all the elements shown in the module interface as well as some other elements that are only visible inside the module. A module implementation can contain constants, variables, types (functions, classes, and interfaces) and an initialization part (init Block), which is used to initialize the elements of the module before they are loaded to execution.

The elements declared inside a module are valid in the scope they are declared. All elements listed in the module interface are public elements unless they are annotated as protected. The module exports the interfaces listed in the *ModuleInterfaces* part. A module implementation has the form:

```
module Identifier ModuleInterfaces ModuleBlock
```

Example of of a module implementing the *IMain* module interface.

```
module Hello implements IMain {
    import System;
    void main (String [] args){
        IO.println('Hello world!');
    }
}
```

Scope

Modules define two scopes; internal and external. The combination of modules and classes provides control over class members' visibility. Listing a class interface in a module interface allows hiding some members of the class in the module implementation. A class declared in the module interface can annotate its members as protected. By default, all members are public. The combination of members that are listed or not listed in the module interface gave us several views of them in different scopes.

Example of a module interface and its implementation defining different views of elements.

```
module interface IM1 {
    class interface IC1 {
        void m1( );
        protected void m2( );
    }
    class C1 extends object implements IC1{
```

```

        constructors
            C1 ( );
    }
}
module M1 implements IM1 {
    class C1 {
        fields
            --
        constructors C
            C1 ( ) { ...}
        methods
            void m1 {...}
            void m2 {...}
            void m3 {...}
    }
    // other elements of module M1
}

```

The module interface *IM1* contains two declarations, a class interface and a class. In the class interface *IC1* two methods with different access (public and a protected) are declared. The module implementation *M1* contains the complete definition of class *C1*. Class *C1* contains three methods, but only two of them were listed in the class interface *IC1* in module interface *IM1*. All members of a class are visible inside the module and they are available for objects and derived classes. A protected member of the class listed in the class interface is visible outside the module only for derived classes.

The example above contains a definition of class *C1* with three methods: *m1*, *m2*, and *m3*. Tables 1 and 2 show how these members are available for users and derived classes inside and outside the module implementation.

Table 1. Visibility of methods of class *C1* inside the module implementation

	Derived classes	Users
Member <i>m1</i>	visible	visible
Member <i>m2</i>	visible	visible
Member <i>m3</i>	visible	visible

Table 2. Visibility of methods of class *C1* outside the module implementation

	Derived classes	Users
Member <i>m1</i>	visible	visible
Member <i>m2</i>	visible	non-visible
Member <i>m3</i>	non-visible	non-visible

2.3 Class Interface and Class Definition

MOOL provides single implementation inheritance and multiple interface inheritance. Classes are organized in a hierarchy with class object at the top. The class hierarchy is build with the definition of new classes and the specialization of existing ones. A class inherits from another class and implements one or more class interfaces. Classes that do not explicitly extend another class, implicitly inherit from object.

The class hierarchy does not organize the structure of a program; it is defined to allow code reuse and incremental definition of classes. The class mechanism is not used to support namespace management nor visibility control.

MOOL uses nominal subtyping, which means that classes define types and subclasses define subtypes.

Class Interface

A *class interface* is a type declaration that provides a specification rather than an implementation for its members. *Class interface* types are used to provide multiple inheritance in MOOL. Any class interface implemented by a class is a supertype of that class. A class interface declaration has the form:

```
class interface Identifier [TypeParameters][ExtendsInterfaces] InterfaceBodyDec
```

The class interface identifier must be unique in the module where it is defined. The identifier may be followed by an optional list of type parameters to declare a parameterized interface type which are presented in section 2.4.

Example of of a class interface definition.

```
class interface IFigure {
    void move (integer dx,dy);
    void draw();
}
```

Class Definition

MOOL contains a construct to define classes as extensible templates that encapsulate state and behavior. Classes in MOOL have three distinct roles: class definition, class specialization, and object creation. A class may inherit from another class and it may implement one or more class interfaces. A derived class can override an inherited method but it must be explicitly declared. It can also shadow some members but it must be explicitly declared to avoid unintentional shadowing of members.

Example of two class definitions using inheritance.

```
class Figure implements IFigure {
    fields
        Point center;
    constructors
        Figure () {center.x = 0; center.y =0;}
```

```

    Figure (integer x, integer y) {
        center.x = x; center.y =y;
    }
    methods
        void move (integer dx, integer dy) {...}
        void draw() {... }
}
class Circle extends Figure implements ICircle {
    fields
        integer ratio;
    constructors
        Circle () { this(0,0,0) }
        Circle (integer r) { this(0,0,r); }
        Circle (integer x, y, r) { super(x,y);
                                   this.ratio=r;}
    methods
        float area () { // implementation of area }
        override void draw() { //new impl. of draw }
}

```

A class declaration provides a class type that can be used to declare object variables of that type. Classes are used to generate objects dynamically. All objects created with a specific class have the same behavior at runtime and it cannot be modified. Objects are created applying the new operator to a class constructor. A class declaration has the form:

```
class Identifier [TypeParameters] [SuperClass] Interfaces ClassBodyDec
```

TypeParameters is an optional part that specifies that the class is generic. Generic classes are explained in detail in section 2.4. *SuperClass* is an optional part that specifies the direct superclass of the class. *Interfaces* specifies the list of interfaces that are implemented by the class. *ClassBodyDec* contains the declarations of the members of the class and the implementation of its constructors and methods. Classes have four kinds of members: class variables, fields, constructors, and methods.

Class variables. Class variables are special members that are shared by all instances of the class. They are allocated once for the lifetime of the program.

Fields. Fields are also called instance variables. Each object has a copy of the fields declared in the class. A field declaration can hide an inherited field if it has the same name and type but the declaration has to be preceded by the shadow access modifier.

Constructors. A constructor is a special function that has the same name as the class and does not specify a return type. A constructor is used in the creation of instances of the class. A class can contain many constructors with different signatures. Constructors must be part of a class declaration in a module interface if they are meant to be available for users or specializers.

Methods. Methods are functions defined inside a class that are always dynamically dispatched. They implement the behavior of objects. All methods of a class are available inside the module that contains the class definition. A class can contain two or more methods with the same name if their signatures are different. A method with the same name and signature than one inherited may override it, if it is annotated as *override*. A method can hide an inherited method with the same name and signature if it is annotated as *shadow* and not *override*. Both methods will be available using a complete qualified name. By default all methods can be overridden in subclasses.

Modifiers

Access modifier. There is one access modifier called *protected*. Any element of a class interface annotated as *protected* can be used in derived classes. Protected members are not available for clients.

Member modifier. There is one member modifier called *shadow*. A field or method of a class can be annotated as *shadow* if it has the same name as one inherited. It is used to hide the inherited member. Both members are available for access. The member defined in the parent class can be accessed using a fully qualified name, casting the object to its parent class, or using *super*. The new member can be accessed with the dot notation.

Method modifier. There is one method modifier called *override*. A method annotated as *override*, overrides an inherited method. The signature of the method must follow the subtyping rules defined in section 6.2.5.

Subtyping for classes and class interfaces. The subtyping relationship between classes is defined explicitly when a class is declared. A class that extends another class is a subtype of the extended class. If a class doesn't extend another class, it implicitly extends *object*. A class is also a subtype of any class interface it implements.

Derived classes must follow the subtype rule for functions when a method is overridden to ensure type safety.

MOOL allows changes of types in subclasses as follows: *invariant* – no type changes are allowed for fields, *covariant* changes are allowed for result types of functions, and *contravariant* changes for function arguments.

2.4 Generic Classes and Class Interface

Generics are abstractions over types. MOOL provides support for the definition of generic types, and type variables.

In this section we present the definition of type variables, and generic types with different kinds of constraints, and how they can be used to create instances of them.

Type variables

A type variable is an identifier with the same features as other identifiers but it stands for a type. Type variables are introduced in parameterized types to represent a

type parameter. They are defined after the identifier of the type declaration and they can be bounded to other types to constraint the type that can be used in instantiations.

Type constraints

Generic code can be defined for all the types available in the system is called *unconstrained genericity* or for some types that hold some properties which is called *constrained genericity*. A bound is declared using the `implements` keyword. Type parameters may contain recursive bounds as in the example shown next.

Example of a parameterized class with a recursively bound type parameter.

```
class interface IOrderable <T> {
    integer compareTo (T elem);
}
class interface IOrderedList < T > {
    T remove();
    void insert (T elem);
}
class OrderList <T implements IOrderable <T>>
    implements IordeList<T> {
    // class implementation...
}
```

Generic types

In MOOL classes, class interfaces, and functions can be defined to be generic. A generic type contains a list of type parameters with specific bounds. The bounds of the type parameters restrict the types of the actual parameters when an instance of the generic type wants to be created.

Generic functions. A generic function is a function that has a list of type parameters. It is called in a similar way to that of a non-generic function, except for the type parameters. A generic function named *swap* that receives a type parameter called *T* and three formal parameters is shown next.

Example of of a generic function called *swap*.

```
void swap <T> ( T [] a, integer i, integer j) {
    T temp = a[i];
    a[i] = a[j];
    a[j] = temp
}
```

Generic classes. A generic class contains a list of type parameters enclosed in `< >`. The type parameters can be bounded to other types.

Example of of a generic class interface and its generic class.

```

class interface IList <T> {
    T head ( );
    void cons(T elem);
}
class List < T > implements IList < T > {
    fields
        ...
    constructors
        List<T> () {...}
    methods
        T head ( ) { ... }
        void cons (T elem) {... }
        ...
}

```

Generic class interfaces. A generic class interface contains a list of type parameters enclosed in <>. The type parameters can be bounded to other types.

2.5 Declarations

A declaration introduces a name for a variable, a constant, a function, or a type that is valid in a scope delimited by the block that contains it. Repeated names for variables are not allowed in the same scope. There are four kinds of declarations: constants, variables, functions, and types.

2.6 Statements

Statements execute actions. They are used to control the flow of execution of a program. Some statements are simple and some others contain other statements as part of their structure. Statements in MOOL are very similar to those in Java, C# and other languages. In this section we present some of the statements supported in MOOL without describing them exhaustively due to their similarity with other languages..

Assignment

An assignment statement has the form LHS = RHS. It requires checking type compatibility between the expression at the LHS and the value generated by the expression at the RHS.

Function Call

A function call could be an expression or part of it if it returns a value.

Continue, return, and break

These statements are used to break the execution flow and continue the execution with the next statement.

Block

A block statement is delimited by curly brackets and may contain local variable declarations and a sequence of statements. It defines a scope where local variables declared inside are valid.

For

The for statement contains a controlling-loop part and a block.

While

The while statement is a conditional loop that executes the block while the value of the expression is *true*.

If

The if statement contains an expression and a body, delimited by curly brackets. The body contains a statement and an optional else part.

Switch

A switch statement contains an expression and a body. The body defines a set of cases for which specific actions are defined and a default clause.

3 Translation

We are implementing a compiler for the MOOL language. The compiler produces target code for the .NET and JVM platforms.

The compiler reads the input file (MOOL source code) and then, in one step, executes lexical, syntactic, and semantic analysis, and intermediate code generation (abstract syntax tree). Another step traverses the abstract tree, producing the target code (.net or bytecode) as the output. We also are implementing a preprocessor that takes a MOOL program as an input and produces C# or Java code as an output (target code). Both schemes allow us to test different types of programs using all the features of the source and target languages.

4 Conclusions

We have presented MOOL, which is a new general-purpose programming language where the roles of classes and modules are separated and generic programming is supported.

MOOL enables object-oriented programming defining hierarchies of classes with single implementation inheritance and multiple interface inheritance. MOOL enables

also the implementation of large programs providing modules - static units of encapsulation, information hiding, and reuse - and module interfaces to describe their interconnection. Generic programming is sustained by parameterized types.

Our language is similar to other programming languages in many ways. We adopted a related Java and C# syntax which both descend from C. We can say that MOOL's module system is based on the module system of Modula-3 and the class mechanism is a simpler version of Java and C# classes.

MOOL is not an extension of any other language despite of the similarities whit other languages.

References

1. United States Department of Defense. *Reference manual for the Ada programming Language*. GPO 008-000-00354-8, 1980.
2. Standard ECMA-334. *C# Language Specification* [Online] <http://www.ecma.ch> December 2001.
3. Ken Arnold and James Gosling. *The Java™ Programming Language*. Addison Wesley, 1998.
4. Barron-Estrada, M. L., *MOOL: an Object-Oriented Language with Generics and Modules*. Ph.D. Dissertation. Florida Institute of Technology, Melbourne, Florida, USA, May 2004.
5. Kim Bruce. *Foundations of Object-Oriented Languages Types and Semantics*. MIT-Press 2002.
6. Kim B. Bruce, Luca Cardelli, Giuseppe Castagna, the Hopkins Objects Group, Gary T. Leavens, and Benjamin Pierce. "On binary methods." In *Theory and Practice of Object Systems*, 1(3): 221-242, 1996.
7. Kathleen Fisher and John Reppy. *Foundations for MOBY classes*. Technical Memorandum, Bell Labs, Lucent Technologies, Murray Hill, NJ, February 1999.
8. Robin Milner, Mads Tofte, and Robert Harper. *The Definition of Standard ML*. The MIT Press, Cambridge, Massachusetts. 1990.
9. Greg Nelson, editor. *Systems Programming with Modula-3*. Prentice-Hall, Englewood Cliffs, NJ, 1991.
10. Didier Rémy. *Using, Understanding, and Unraveling the OCaml Language*. In Gilles Barthe, editor, *Applied Semantics. Advanced Lectures. Volume 2395 of Lecture Notes in Computer Science*, pages 413–537. Springer Verlag, 2002.
11. Bjarne Stroustrup. *The C++ Programming Language*. Addison-Wesley, 1991
12. Clements Szypersky. Import is not inheritance; why we need both: modules and classes. In *Proceedings of ECOOP '92, European Conference on Object-Oriented Programming*, Utrecht, The Netherlands, June/July 1992. Volume 615 of *Lecture Notes in Computer Science*, pages 19-32, Springer-Verlag Berlin Heidelberg 1992.

AID: an Object-Relational Schema Design-Tool

Hassan Badir¹, Etienne Pichat, He Xiaojun

LIRIS Laboratory, UFR/Info, UCBL, Lyon
8 Niel Bohr 69622, Villeurbanne, France
{hbadir}@liris.cnrs.fr

Abstract. Schema design is a complex activity that makes use of a variety of knowledge. It should take into account foreseen processing. Nowadays, interaction of processing with data (static) is not taken into account by our AID tool that is limited to offer a graphical interface to conceive in various models (UML Class diagrams, Universal relationship and object attributes forest), and transform the input diagram into the two other models. This tool highlights the complementarities of these different models, especially object attributes forest able to take into account designer knowledge of the processing.

Keywords. Object attributes Forest, Universal relationship with inclusions, Class diagrams, stereotype, SQL3, XML.

1 Introduction

Database design based-tools aim essentially modeling, schema conceptual design, transforming these schemas into implementation structures, with an optimization and querying aim. Nowadays, (Entity-relationship [6], UML diagrams [16], relational data model, Universal relationship with inclusions [18]...), database design methods (Object-Merise [19], UML, ...) and platforms (Super [10], Tramis [12], Power Designer [7], ConceptBase [13], Rational Rose [17], etc.) are widely available in the computer science community, by the way of published papers, and books or softwares. However, despite all these various tools; actually database applications are not designed in an automatic manner, not well documented, not well structured, and incoherent and not easy to maintain. Numbers of models and methods are proposed to the public, but insufficient to answer all their needs. Commercial Design tools are often suitable for a specific field and inadequate for the others or are not well-used.

In the framework of object information systems design, the aim of this paper is to present an advanced visual graphical environment and sufficiently rich semantically, that allows express non-expert user needs. Conceptually, it relies on object attributes forest constructed with an interactive help. This expressive representation is a set of attributes trees where leaves refer to a tree. We then transform object attributes forest into a

database stereotyped UML class diagram which uses UML-DB profile [3] [15] completed in [4].

This paper is structured as follows: section 2 discuss our work with respect to our global project, section 3 treat general principals of the used model Object Attributes Forest, section 4 discuss the derivation of class diagrams from object attributes forest. We conclude with an overview of the main contributions of this paper, followed by the future perspectives.

2 AID Architecture

In order to achieve the expected objectives of our work, we integrate three formalisms representing three different models of our tool called AID. AID is an edition and object, relational, and object-relational conception-assisted platform. Its roles can be enumerated as follows:

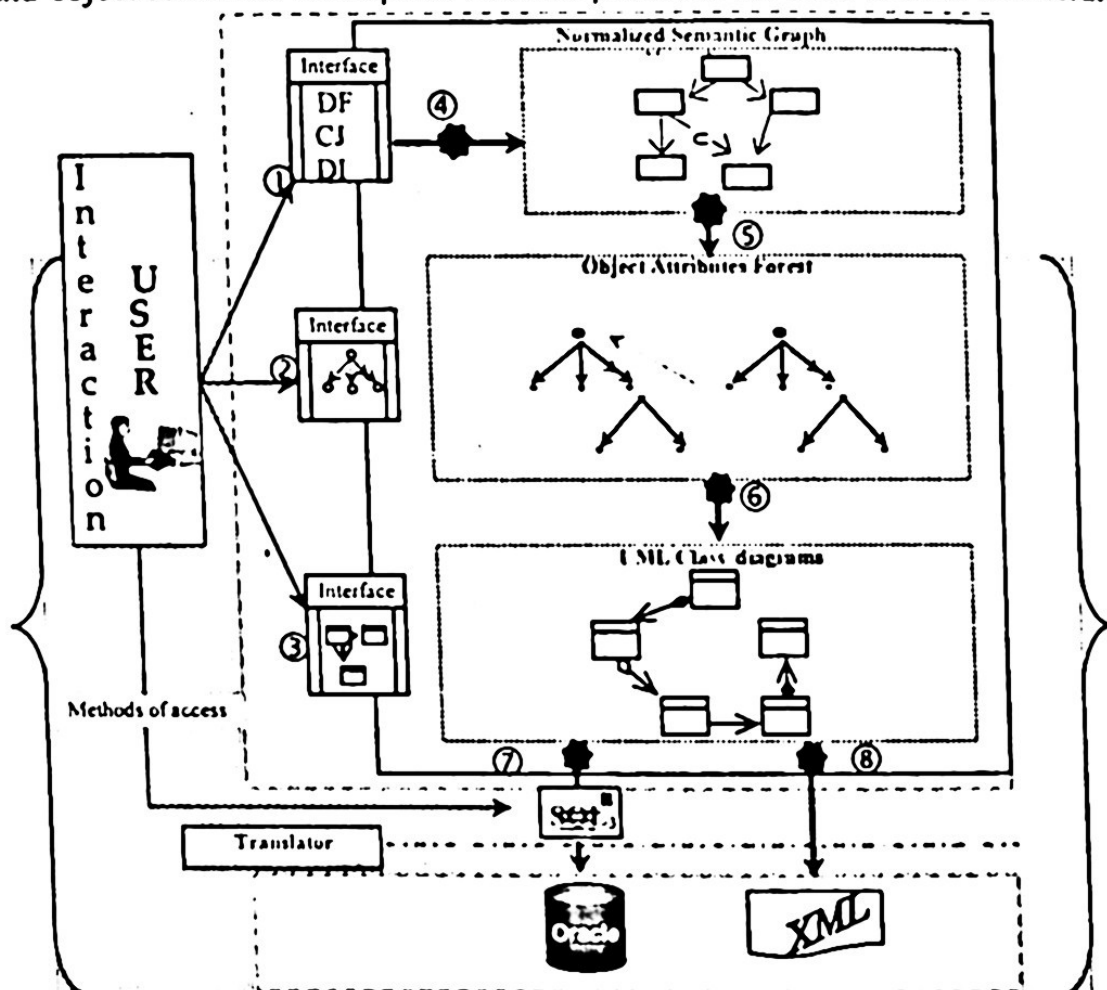


Fig. 1. AID architecture

- User can describe schema of his choice through on the following models (Fig. 1): Universal relationship with inclusions (URI) [18], object attribute forest and DB stereotyped UML classes.
- to go from a model to another by using transformation algorithms, it allow as well SQL3 description generation or XML schema,
- User can normalize; transform a representation into relationships related by inclusion dependencies in a Normalized Semantic Graph (NSG). This normalization could be partial, and keep unchanged some complex object structures fixed by user without any further normalization.
- To optimize the obtained conceptual schema with respect to the foreseen processing, introducing hence access methods, even to denormalize it.

This tool is composed of three windows or principales interfaces. Each window is reserved for representing different formalism. It is constituted of five modules of transformation:

- Interactive interface for specifying normalized semantic graph by applying a normalization algorithm on a set of functional dependences, jointure composantes, and inclusions dependences (Fig. 1)(1).
- Interactive interface to specify object attributes forest allowing describe complex data (Fig. 1)(2).
- Interactive interface for specifying database stereotyped UML class diagrams (3).
- Transformation module of an ordered normalized semantic graph (ONSG)[14] into an object attribute forest. The corresponding algorithm is given by [14] (Fig. 1)(5).
- Transformation module (Fig. 1)(6) of object attribute forest into database stereotyped UML class diagrams. The corresponding algorithm is described in section 4.
- Module for generating SQL3 description from a UML classes diagram (Fig. 1)(7).
- Module for generating XML schema from a UML classes diagrams (Fig. 1)(8).

In this paper, we will restrict our study at the shaded part which is composed of two interfaces ((Fig. 1)(2 and 3) and a transformation module ((Fig. 1)(6 and 7).

3 Object Attributes Forest (OAF)







The main advantage of the relational model is its great simplicity in representing data in the form of relation (table). In this model, the schema of a relation is composed of a set of attributes taking values from atomic domains: it is the first normal form (1NF). However, such an inherent constraint requires serious limitations in terms of modeling. In order to overpass such a limit, a model with complex values (structured values, or complex objects), very simple, has been introduced to propose to user a maximum of possibilities in designing his reality. This model uses the least set of modeling operators. It's in this objective that has been used OAF, conceptualization of complex object [7][2].

3.1 Elements composition Object Attributes Forest

Elements composing Object Attributes Forest are attributes tree and reference edge. Recall that a tree is a non-cycled oriented graph having one and only one root (or node without predecessor). Its other nodes called intermediates if they have at least one successor and leaf otherwise.

Attributes tree is a tree where each node is has a named attribute. The root name is also the tree name and will be the name of the associated class. Each intermediate node or root is unshared and its lifetime is of the composed class. Each arc of attributes tree is:

- Univoque (total or partial): to each value of its original attribute (node) is associated at most a value of its extremity attribute. It's represented by a simple arrow, or.
- Multivoque (total or partial): to each value of its original attribute could be associated a set of values to its extremity attribute. It's represented by an arc with a double arrow in the same direction (table. 1)

Table 1. Representation of the bows of a object of attributes forest					
					
Univoque (1,1) - (1,n)	set (0,1) - (1,n)	Multivoque (1,n) - (1,n)	set (0,n) - (1,n)	Reference (heritage)	Inclusions dependences

A OAF is a set of attributes trees and reference arcs. A reference arc relates a leaf node to an attributes tree. It indicate that each value of leaf attribute is a reference (a pointer) to at most one instance or value of the root of the referenced attributes tree. It is represented by discontinued fat arrow.

The naming of the attributes forest nodes should verify the following constraints:

- The name of the attributes tree should be unique among the attributes trees names.
- The node name should be unique among the nodes names of the same immediate predecessor.

(Fig.2) represent an object attribute forest of the Olympic-Games modeling athletes, coaches, and sportive events.

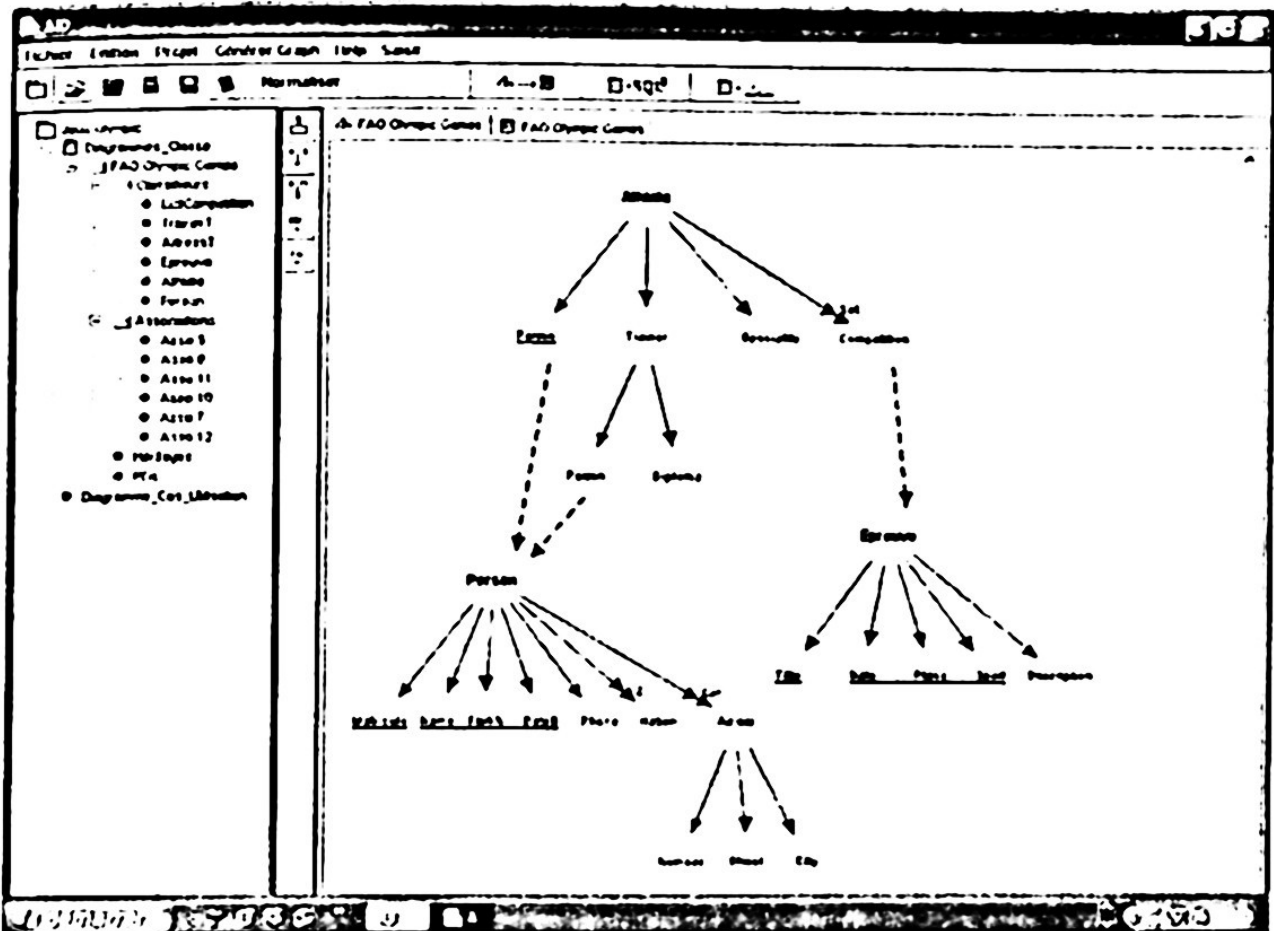


Fig. 2. OAF of Olympic-games where for Person the second key Name FirstN Tel not displayed

In an object attribute forest, each attributes tree has a color chosen hazardly by AID or clarified by user. This color will be inherited by UML class diagram in the transformation phase.

3.2 Keys in a OAF

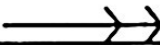
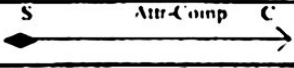
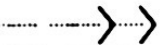
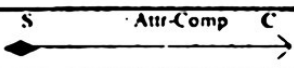
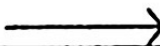
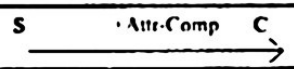

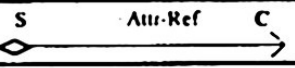
Object identifier in object data modeling is provided at the same time of the object creation. It is not often visible to users (or applications). Object identity is insufficient. It's important to give users the possibility to access objects using their contents and to avoid double insertion (equal objects or having the same key value). The key concept is an essential semantic concept in data modeling, introduced later by object DBMS. [5] A complex class or attributes tree could have one or several keys. For example (Fig. 2), Person has as keys *Matricule*, *First-Name Second-Name BirthDate* and *First-Name Second-Name Phone-Number*.

4 Database Stereotyped UML Schema Derivation from a Complex Tree

This section relates complex data representation and UML class diagram, which represent the cornerstone of UML method. Recall that a class diagram is composed of classes and relations between these classes, with their attributes and methods. We give at first a algorithm for transforming object attributes forest into classes diagram, then after some explanations about the used UML-DB profile, we explain how stereotypes are generated from attributes tree keys.

4.1 Algorithm for transforming OAF into classes diagram

This algorithm (Algo.1) constructs persistent classes from roots and intermediate nodes of OAF, adds to them attributes from leaves, and defines compositions from arcs and associations from references. It highlights the recapitulated correspondences (Table 2):

Table 2 Correspondences between arcs of OAF and associations of UML classes diagram		
Types	Arcs	Relations
Illimited multivoque composed		
Borned multivoque composed		
Univoque composed		
Reference [univoque]		

Algorithm 1: Algorithm of transformation of a OAF in a Classes Diagram

Input : Object Attributes Forest.

Output : DB Stereotyped UML classes Diagram.

For All each tree (of root R) **do**

If its class (or those of it root) R doesn't exist **Then**

 CreateClass(R)

End If

For All Arc(S_o, S_e) in an order that predecessor has been examined **do**

If (S_o, S_e) is univoque **Then**

If S_e is intermediate **Then**

 CreateClass(S_e)

 Compose univoquely this class to class S_o .

End If

```

If  $S_e$  is a leaf and reference the tree  $S$  Then
If referenced class doesn't exist Then
    CreateClass( $S$ )
End If
    Associate referenced class  $S$  to class  $S_o$  with role  $S_e$ 
End If
If  $S_e$  is a leaf not referenced Then
     $\text{Attr}(S_o) = \text{Attr}(S_o) \cup \{S_e\}$  {Associate attribute  $S_e$  to
    classe  $S_o$ }
End If
Else
If  $(S_o, S_e)$  is multivoque Then
    CreateClass( $TasS_e$ )
    Compose multivoquely this class to classe  $S_o$ 
If  $S_e$  is a leaf and referenced  $S$  Then
If referenced class doesn't exist Then
    CreateClass( $S$ )
End If
    Associate referenced class  $S$  to classe  $TasS_e$  with
    role  $S_e$ 
End If
If  $S_e$  is a leaf not referenced Then
    {Associate attribute  $S_e$ } to class  $TasS_e$ 
End If
If  $S_e$  is intermediate Then
    CreateClass( $TasS_e$ )
    Compose multivoquely this class to classe  $S_o$ 
End If
End If
End If
End If
End If

```

Algorithm execution applied to OAF of (table 2) gives the UML class of (Fig.3). Therefore, the node *Address*, which has an entering arc (*Person*, *Address*) of labeled multivalued NT type, generate between the two classes *Person* and *Address* the composition arrow into *TasAddress* taking the +*Address* role *TasAddress* side, the competition of the going node of reference type, generate the composition arrow of Competition into *TasCompetition* and aggregation arrow of *TasCompetition* into *Epreuve* having the +*Competition* role side *Epreuve*.

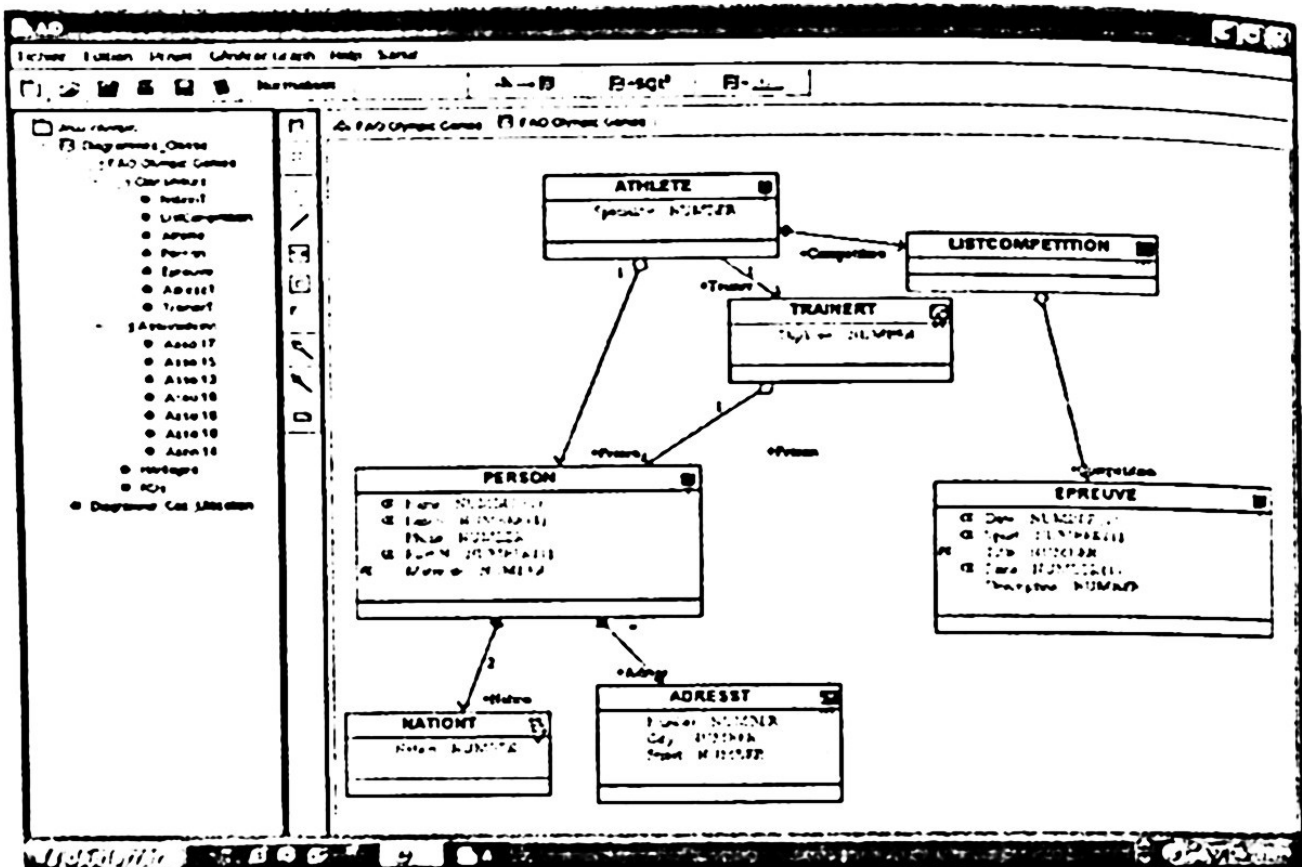


Fig. 3. DB stereotyped classes diagram obtained from The FAO of the (Fig 2)

4.2 UML-DB profile

The profile concept is essential in model engineering. This concept is not neutral; it descends principally from *hypergenericity* [9] concept. The profile concept has been standardized in UML 1.4 in 2000 and reinforced in UML 2.0. A profile is a UML extension that keeps intact the UML metamodel [17]. It's composed of a set of stereotypes, of labeled values attached to these stereotypes and constraints (table 3). In the field of databases, we have extended the UML-DB [4] profile to composed or simple candidate keys. This profile has also icons stereotypes representing nested table, array, type and table.

4.3 Keys processing

To transform keys of objects attributes forest into keys in classes diagram, we have used UML-DB profile for primary, secondary, simple, and composed keys specifications. We have used stereotypes to express keys in classes diagram. In the design phase, we can associate several keys to one class, and each key can contain several attributes. One of these keys could be specified as primary: we stereotype it as PK as we can see it in (fig 4).

Stereotyping the others keys often called candidates, is more difficult. Each key is identified by an index that will be associated with its attributes. Thus, each candidate key is stereotyped CK, and each one of its attributes carries on its index as showed in (fig 4). This stereotyping generalizes that adopted by Rose [17].

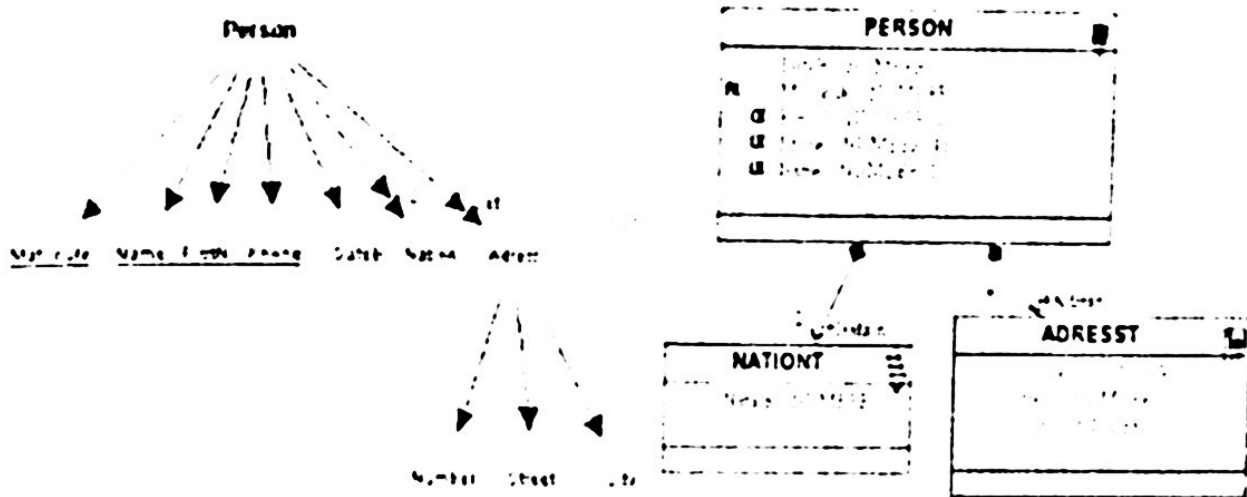


Fig. 4. Computing and transforming complex keys 3

The following table resumes a part of the UML-DB profile with stereotypes for Oracle9i:

Stereotype	Icon	Description	Constraint
<<Object-type>>		Allows the representation of new user defined data types. It corresponds to the structured type in SQL:1999	It can only be used to define value types
<<Object-table>>		It represents a class of the database schema that should be defined as a table of an object type. It corresponds to the typed table in SQL:1999	A typed table implies the definition of a structured type, which is the type of the table
<<Nested-Table>>		Represents an non-indexed and unbounded collection type	The elements of a NT can be of any data type except another collection type
<<Varray>>		Represents an indexed and bounded collection type. Its corresponds to the array type in SQL:1999.	The elements of a VARRAY can be of any data type except another collection type

Table 3. UML-DB Profile for Oracle9i

Algorithm 2 Algorithm of transformation of keys


```

For All each root  $S_0$  do
  For All each arc  $(S_0; S_e)$  do
    If  $(S_0; S_e)$  is univoque then
      If  $S_e$  is leaf not refernced then
        If  $S_e$  2 an Primary Key PK then
          Stéréotype  $S_e$  = «PK»
        End If
      For  $j = 1$  to  $\text{card}(\text{setofCandidatesKey})$  do
        If  $S_e$  2 an Primary Key PK then
          Stéréotype  $S_e$  = Stéréotype  $S_e$  + «PK»
        Else
          Stéréotype  $S_e$  = «CK»
          Contrainte  $S_e$  = Contrainte  $S_e \cup \{j\}$ 
        End If
      End For
    End If
  End For
End For

```

In an objects attributes forest, keys could be represented in the same time with PK, CK, and index symbols. Such a representation seems to us less visual. Thus, we have adopted the underlining like entity-association diagrams, that supposes doubling attributes belonging to several keys.

5 Conclusions

In our paper, we have addressed quality improvement in designing complexes relational-object database schemas and the construction of UML classes diagram from an objects attributes forest. We have presented an algorithm for transforming a complex tree into a UML DB stereotyped classes diagram for conceiving object-relational schema. Our choice for using objects attributes forest as conceptual data model is justified by its simplicity in representing complex data, and data modeling. The enrichment, by keys and references that we have added, leads to a better complex data design. In other hand, we have chosen as a final conceptual model, UML class diagram thanks to its expressive semantic expressiveness, and its standardization, after extending it in order to support object-relational and relational database design. We have developed an intuitive interface for assisting design. The goal is to make the conception process very simple and very easy for non-specialists users. Thanks to this interface, user conceive his OAF progressively. While he refines his schema, user can eventually come back, and go in other direction.

Interface offers flexibility and conviviality to express graphically trees, and inheritance management by its references. It will be very promising to derive object-relational schemas from relational schemas. Other possible objective is to extend our tool for automatic transformation of relational model (Normalized Semantic Graph) into an object-relational model (UML class diagram). Thus to improve our tool (processing, offering a graphical query interface for precise specifications).

References

1. Abitboul S., Hull R., Vianu V., Foundations of Databases, Addison-Wesley, 1995.
2. Abitboul S., Hull R., Buneman P., Suciu D., Data on the Web, Addison-Wesley, 1999.
3. Ambler S., Persistence a UML Profile for Data Modeling, www.agiledata.org, 2002.
4. Badir H., Pichat E., Beqqali O., Utilisation d'UML pour concevoir une base de données objet ou objet-relationnelle IEEE SETIT, 2003.
5. Cattell R.G.G., Skeen J., The Object Database Standard, Morgan Kaufmann Publishers, 1994, ACM Transaction.
6. Chen P. P.: The Entity-Relationship-Model-Toward a unified view of data, ACM Transaction on Database System 1 (1), 1976.
7. DATE C.J., An Introduction to Database Systems, Editions Addison Wesley, 2000.
8. Dellobel C. Bancilhon R., Building an Object-Oriented Database System- The story of O2, Morgan Kaufmann Publishers, 1992.
9. Desfray P., Object Engineering, The fourth dimension, Addison Wesley 1994.
10. Dennebouy Y., Andersson M., Auddino A., Dupont Y., Fontana E., Gentile M., and Spaccapietra S, SUPER: visual interfaces for object + relationship data models, Journal of visual languages and computing, 6(1):27 -52, 1995.
11. Fallouh F., Données complexes et relation Universelle avec inclusions. Une aide à la conception à l'interrogation des bases de données, thèse de doctorat de l'université UCB Lyon1, 1994.
12. Ilick J-M, Englebert V., Henrard J., Roland D., Hainaut L.-L., The DB-Main Data-Base Engineering CASE Tool (Version 6) - Functions Overview, Technical manual, Institut d'Informatique, FUNDP, November 2000.
13. Jarke M., Gallersdörfer R., Jeusfeld M.A., Staudt M., Eherer S., ConceptBase – a deductive object base for meta data management, In Journal of IIS, Vol. 4, No. 2, 1998
14. Manea A., Towards an object Oriented Design Method. Application to Hydrological Database, DEXA 1995: London, United Kingdom.
15. Marcos E., Vela B. and Cavero J. M., Extending UML for Database Design, Fourth International Conference on the Unified Modeling Language: UML 2001, Toronto, Canada.
16. Muller P.A., Gaertner N., Modélisation objet avec UML, Eyrolles, 2e édition, 2000.
17. Naiburg E. J., Maksimchuk R. A., UML for Database Design, Addison-Wesley, 2001.
18. Pichat E., Bodin R., Ingénierie des données, Masson, 1990.
19. Rochfeld A., Rigaux P., Traité de modélisation objet, Editions Eyrolles, 2002.

A Preprocessor Based Parsing System

Pradip Peter Dey, Mohammad N. Amin, and Thomas M. Gatton

**School of Engineering and Technology
National University**

**11255 North Torrey Pines Road, La Jolla, CA 92037, U.S.A.
{pdey, tgatton, mamin}@nu.edu**

Abstract. A preprocessor based parsing system for Tree Adjoining Grammars is presented. The preprocessor is used primarily for organizing the data structures required for parsing the grammar efficiently. The preprocessor works hard in order to reduce the runtime processing load so that the parser executes fast. At least one model of Tree-Adjoining Grammars allows transfer of significant processing load from the parser to the preprocessor since the adjoining process can be optionally applied before runtime on tree structures. A parallel parsing algorithm is presented that takes advantage of the preprocessor.

1 Introduction

A preprocessor based parsing system is described in this paper. A parsing system basically performs syntactic analyses in natural language processing [1, 2]. The parsing system described here is based on a grammatical formalism called Tree Adjoining Grammar (TAG) developed by Joshi [3, 4, 5]. TAGs seem to be appropriate for our purposes because: (a) they can be easily decomposed into independent modules which can be processed concurrently, (b) they can be developed incrementally since they represent information about the language in a highly modular fashion, (c) they seem to have the formal properties required for processing natural languages [5, 6] and most importantly, (d) the significant processing load of the system can be transferred from the runtime module to the preprocessor. This paper describes the parsing system with a model of TAGs that is appropriate for preprocessor based parsing strategies.

Two finite sets of trees are defined by a TAG along with a composition operation called an adjoin that recursively generates new trees by combining existing trees [3, 4]. The trees serve as structural descriptions of sentences or sentence fragments. The central module of the parser is a pattern matcher that finds every tree that matches the input string. If the input matches two or more trees, then it is structurally ambiguous. If there is no perfect match then the best available approximate match can be determined according to some heuristics. This paper is not concerned with this latter inexact match by which semi-grammatical sentences are parsed. It considers only grammatical sentences. The lexical categories of the input string are determined by a lexical search. These categories are matched with the leaves of the trees stored in TREE BANK or produced dynamically by the adjoin process as shown in Figure 1.

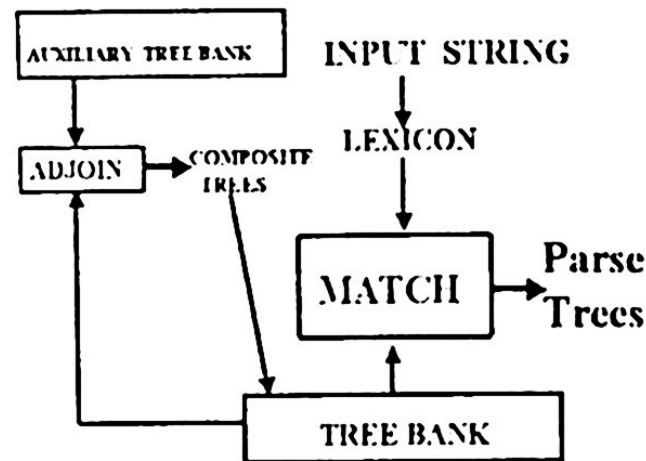


Figure 1: Major Components of a TAG Parser

The tree-bank has a set of trees that correspond to the structural descriptions of sentences of the language. Initially, it has a well-defined set of minimal trees called initial trees that correspond to "simple" sentences of the language. The auxiliary tree-bank has a set of trees called auxiliary trees. Auxiliary trees are used in the adjoining operation to account for recursion. They do not occur independently in the language. In order to improve the performance of the parser, three basic strategies are adopted: (a) Adjoin is applied before runtime through a preprocessor in order to obtain an inflated tree-bank which is sorted according to the length of the trees; (b) the lexicon is ordered so that a parallel binary search can be efficiently applied; (c) the match operation is parallelized so that millions of trees can be searched efficiently. Suppose the input is a string such as "The girl danced". The parser first searches the lexicon and finds that "The" is a determiner (D), "girl" is a noun (N), and "danced" is a verb (V). The category sequence, DNV, for this sentence is then matched with the leaves of the trees from the tree-bank. The tree given in Figure 2. matches the category sequence because it has exactly the same leaves. The result of the match is the top tree in Figure 3. For this tree and for future references, assume N = Noun, V = Verb, A = Adjective, NP = Noun Phrase, VP = Verb Phrase, R = Relative-Pronoun, and S = Sentence.

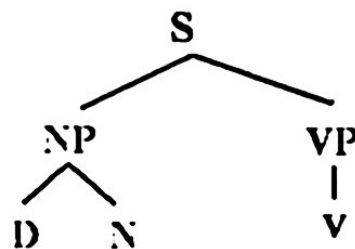


Figure 2. A tree that matches the string "The girl danced"

MATCH simply asks if the categories of the input string are equal to the leaf-nodes of a tree. In the proposed parser, a large number of trees are pre-generated by a preprocessor before runtime in order to reduce runtime processing loads. The TAG parser has at least two advantages over classical parsers: (i) the trees are not built at run time, so it is fast, (ii) the parser can be developed incrementally by adding new trees to the tree-bank. Joshi and Vijay-Shankar described a sequential algorithm that takes $O(n^6)$ time, where n is the length of the input string [6]. We describe an algorithm that takes $O(n)$ time. We present a parallel version of the algorithm that achieves almost linear speed-up. TAGs can be processed very fast with appropriate combination of parallel processing, preprocessing of certain information and heuristic search. It is suggested that natural language systems should be designed to process short sentences very fast, because, in ordinary interactions, a sentence will be approximately a dozen words long. The proposed parser does not fail to process long strings, but their processing is slower because additional runtime processing is required for them where aids from the preprocessor are minimal. Specifically, it has to apply adjoin in runtime to generate large trees for long input.

2 TAGs For Natural Language Parsing

Natural language parsing systems are usually difficult to build because they are expected to achieve computational efficiency and linguistic adequacy. TAGs seem to be appropriate for natural language parsing because they help in achieving these two goals. TAGs have two types of elementary trees: initial trees (IT) and auxiliary trees (AT). All elementary trees are minimal in the sense that they do not exhibit recursion. Recursion is factored by the composition operation adjoin that produces composite trees by combining auxiliary trees with initial or composite trees. The trees are stored in the parser without words or lexical materials. If an input string matches a tree then that tree is returned as an output of the parser after inserting the words of the input string into the tree. A tree, IT_1 , is an initial tree if its root is labeled S and no other node is labeled S . A tree, AT_1 , is an auxiliary tree if its root and one of the leaf-nodes (frontier nodes) are labeled by the same non-terminal symbol, not necessarily S . The leaf-node that has the same label as the root is called the hook node of AT_1 . The adjoining operation can only adjoin an auxiliary tree to a non-auxiliary tree (initial or composite) that has at least one node whose label is the same as that of the root of the auxiliary. This matching node in the initial or composite tree is called the target node.

An example of adjoining from English is given in Figures 3-5 for revealing its application. An initial tree that matches English sentences like "The girl danced" is given at the top of Figure 3. An auxiliary tree that corresponds to an embedded clause like "who ate fish" is given at the bottom of Figure 3. The auxiliary tree of Figure 3 can be adjoined at the NP node of the initial tree of Figure 3, because it matches with the root of the auxiliary tree.

The ultimate result of this adjoining is the composite tree given in Figure 5 which corresponds to the English sentence "The girl who ate fish danced". However, adjoining is a two step process in which the first step is shown in Figure 4. In the first step of adjoining, the sub-trees below the target NP node of the initial tree are

moved to the hook-node NP of the auxiliary tree resulting in the trees of Figure 4. In the next step of adjoining, the root of the auxiliary tree is pasted on the target node of the initial tree. The resulting composite tree is given in Figure 5. The lexical items are inserted into each of the trees to illustrate the correspondence between trees and strings. The composition operation is actually applied on skeletal trees without lexical items or words.

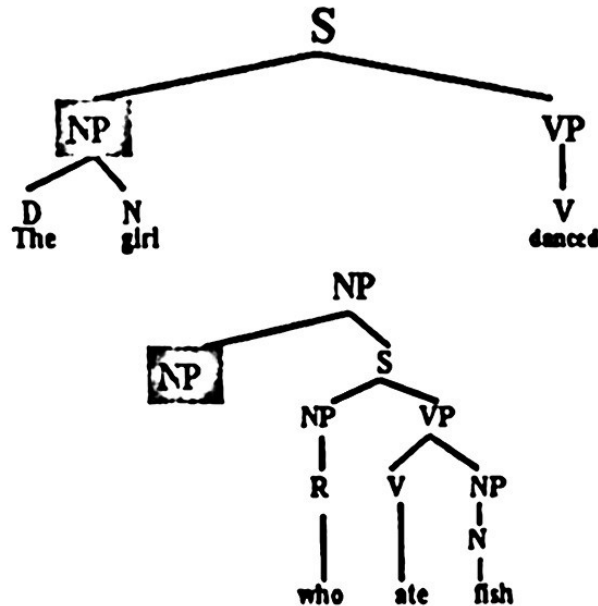


Figure 3. An initial tree at the top and an auxiliary tree at the bottom

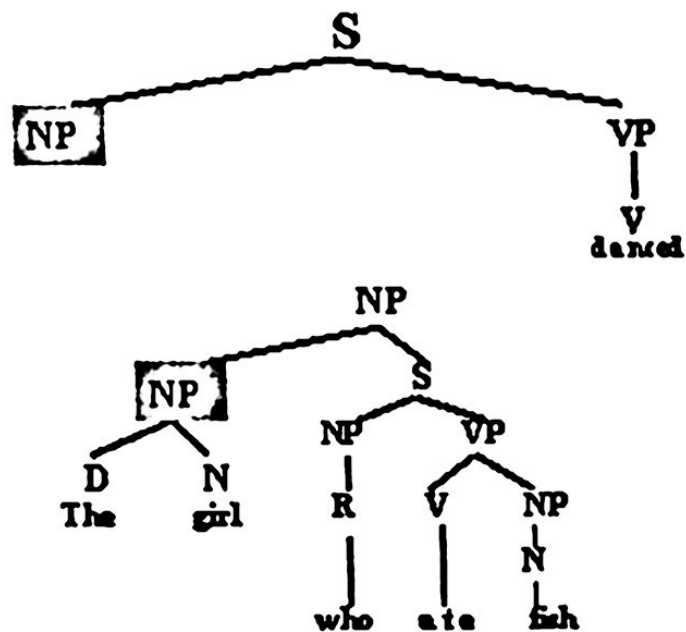


Figure 4. Sub-trees below the target node of the non-auxiliary tree are moved to the hook node of the auxiliary tree

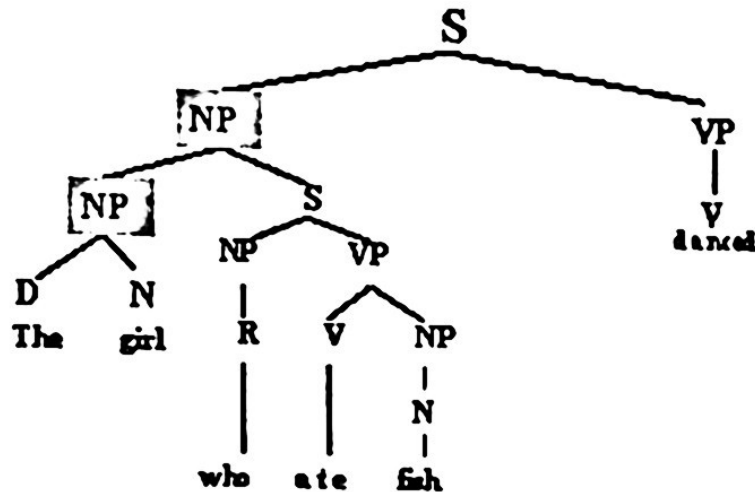


Figure 5: The composite tree produced from the adjoining process

The lexicon is a database in which each word is a key and the lexical category of the word is its main entry. Some words belong to more than one lexical categories resulting lexical ambiguity [7, 8, 9].

The lexical items are inserted into each of the trees to illustrate the correspondence between trees and strings. The composition operation is actually applied on skeletal trees without lexical items or words. The lexicon is a database in which each word is a key and the lexical category of the word is its main entry. Some words belong to more than one lexical categories resulting lexical ambiguity [7, 8, 9].

3 Parsing Algorithms

In order to increase the speed of processing sentences of reasonable length, we maintain a sorted tree-bank with initial and pre-generated composite trees. That is, all trees a maximum of 7 adjoining are pre-generated before runtime and stored in the tree-bank. The trees are ordered in the tree-bank according to their length. All trees of the same length are stored in the same sub-tree-bank marked by the length of the trees. The length of a tree is the number of leaves it has. Depending on the length of the input string, only one of the sub-tree-banks is selected for an exhaustive search. The length of each tree in that sub-tree-bank must be equal to the input length. This strategy along with some heuristics allows efficient searching of the tree-bank. The tree bank (TB) is managed by the procedure PARALLEL-PREGENERATE. It pre-generates composite trees with a maximum of M adjoining and inserts them in the sorted tree bank, where M is an integer. We initially set M to 7. The ADJOIN procedure follows the steps outlined informally in section 2. It takes an auxiliary tree and a non-auxiliary (initial or composite) tree as arguments and returns z composite trees by adjoining the auxiliary tree at z occurrences of the root of the auxiliary tree in the non-auxiliary tree. In order to pre-generate a large number of composite trees and maintain them in a sorted tree-bank, can also be used during runtime. It is an asynchronous algorithm.

Procedure PARALLEL-PREGENERATE(FRONT, AT, TB, M)

FRONT = $\{I_1, I_2, \dots, I_c\}$, a finite set of non-auxiliary trees. Initially FRONT has only initial trees. AT = $\{AT_1, AT_2, \dots, AT_b\}$, a finite set of auxiliary trees. TB = (T_1, T_2, \dots, T_x) , an ordered set of non-auxiliary trees. Initially we present the PARALLEL-PREGENERATE procedure, which is usually applied before runtime although it TB has only initial trees. M = The maximal number of adjoining desired in a tree.

begin

1 OPEN \leftarrow FRONT // FRONT is copied to OPEN

2 For d = 1 to M do

begin

3 FRONT \leftarrow ()

4 For j = 1 to |OPEN| create process j

 // create a process for each tree in OPEN

5 In each process j :

5.1 r \leftarrow OPEN_j

5.2 For h = 1 to |AT| Create process h

5.3 In each process h:

5.3.1 ct \leftarrow ADJOIN(AT_h, r)

5.3.2 For i = 1 to |ct| Create process i

5.3.3 In each process i :

5.3.3.1 TB \leftarrow UPDATE(ct_i, TB, FRONT)

 if ct_i is not in TB then TB \leftarrow INSERT(ct_i,
TB) & FRONT \leftarrow FRONT \cup ct_i

6 Synchronize

7 OPEN \leftarrow FRONT

end

8 Return TB

end

Parallelism in the PARALLEL-PREGENERATE procedure is achieved mainly by data partitioning. It will generate parse trees with desired number of adjoining before runtime. For efficient runtime execution of the parser the PARALLELPARSE procedure is given below.

Procedure PARALLELPARSE(FRONT, AT, TB, W, L, M)

FRONT = $\{I_1, I_2, \dots, I_c\}$, a finite set of non-auxiliary trees. AT = $\{AT_1, AT_2, \dots, AT_b\}$, a finite set of auxiliary trees. TB = (T_1, T_2, \dots, T_x) , an ordered set of non-auxiliary trees. FRONT is a subset of TB. W = $\{W_1, W_2, \dots, W_n\}$, the input string. L = $\{L_1, L_2, \dots, L_y\}$ an ordered set of words called lexicon, M = The maximal number of adjoining desired in a tree.

Output: AS = Ambiguity set, i.e., a set of trees that match W

```

begin
1  CAT  $\leftarrow$  CATEGORIES(W, L)
    // CATEGORIES returns a sequence of lexical categories
2  n  $\leftarrow$  LENGTH(W) // The input length is assigned to n
3  SUBTB  $\leftarrow$  BINARY(TB, n)
    // BINARY returns a subset of TB containing trees of length n
4  AS  $\leftarrow$  MATCH(SUBTB, CAT)
    // MATCH returns a subset of SUBTB that matches CAT
5  DYN  $\leftarrow$  PRUNE(FRONT, CAT)
    // PRUNE returns a subset of FRONT appropriate for CAT
6  ATB  $\leftarrow$  PRUNE(AT, CAT)
    // PRUNE returns a subset of AT
7  If DYN and ATB are both non-empty then do:
    begin
8      LOOP:
9          TEMP  $\leftarrow$  ( ) // TEMP is initially empty
10         Create a process j for each tree in DYN
11         In each process j :
11.1             Create a processes h for each tree in ATB
11.2             In each process h:
11.2.1                 CT  $\leftarrow$  ADJOIN(ATh, DYNj)
11.2.2                 CTS  $\leftarrow$  MATCH(CT, CAT)
11.2.3                 AS  $\leftarrow$  AS  $\cup$  CTS
11.2.4                 CT  $\leftarrow$  CT - CTS
11.2.5                 TEMP  $\leftarrow$  TEMP  $\cup$  REVISE(CT, CAT)
                        // REVISE returns a subset of CT
12  Synchronize
13  DYN  $\leftarrow$  TEMP
14  if (DYN  $\neq$  nil) then Go to LOOP
    // If DYN is non-empty, then go to LOOP
    end
15  return AS
end

```

For input strings of reasonable length, steps 8 through 14 are not executed. For these strings, step 4 determines the trees that match them and these trees are returned in step 15. Steps 8 through 14 are used for dynamic adjoin which is avoided as much as possible, because dynamic adjoin is not efficient. However, if dynamic adjoin is used, then it stops at step 14 when DYN becomes empty. If the dynamic adjoin is successfully avoided then the worst case computational time is proportional to the number of trees in the tree-bank. It is expected that an implementation of PARALLELPARSE will achieve almost linear speed up. This expectation is based on the assumption that the synchronization and inter-process communication overheads can be kept at a minimal level, since each process can perform its work independently.

4 Concluding Remarks

Performance of natural language parsing systems can be improved if the input string length is restricted to a reasonable number. If the input string length is not restricted, then the proposed parser will take additional time for processing. For any reasonable application, parsing natural language demands both computational efficiency and linguistic adequacy. Computational efficiency can be achieved by parallel processing of TAGs with pre-runtime processing of certain information, and heuristic strategies. Linguistic adequacy can be achieved by a TAG with its formalized structure generating mechanism and incremental development of language based structures. Syntactic structures or trees can be added incrementally to the tree-bank of the TAG parser anytime. Recent advances in semantic processing of TAGs suggest that TAGs present a viable alternative for application development using natural languages such as English [3].

Acknowledgement:

We are grateful to Juan Espana, Jose Contreras and many others for their help, comments and encouragements.

References

- [1] Jurafsky, D. et al, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Prentice Hall, 2000.
- [2] Allen, J. *Natural Language Understanding* (2nd Ed.), Addison-Wesley, 1995.
- [3] Abeillé, A., Rambow, O., *Tree Adjoining Grammars*, University of Chicago Press, 2001.
- [4] Joshi, A. K., Levy, L. S. "Tree Adjoining Grammars", *Journal of the Computer and System Sciences*, 1975, 10: 136-163.
- [5] Joshi, A. K. "Tree Adjoining Grammars: How Much Context Sensitivity is Required to Provide Reasonable Structural Descriptions?", 1985, in Dowty et al (eds), *Natural Language Parsing*.
- [6] Vijay-Shankar, K., Joshi, A. K. "Some Computational Properties of Tree Adjoining Grammars", *Proceedings of the 23rd Annual Meeting of the ACL*, 1985, 82-93.
- [7] Dey, P., Bryant, B., Takaoka, T. "Lexical Ambiguity in Tree Adjoining Grammars", *Information Processing Letters*, 34, 1990, 65-69
- [8] Schabes, Yves, Waters, Richard C. : Tree Insertion Grammar: Cubic-Time, Parsable Formalism that Lexicalizes Context-Free Grammar without Changing the Trees Produced. *Computational Linguistics* 21(4), 1995, : 479-513
- [9] Joshi, Aravind K., Schabes, Yves: Tree-adjoining grammars and lexicalized grammars.

Índice de Autor /Autor Index

	Pág./Page		Pág./Page
A			
Aguilar Ibáñez	211, 221, 231	Filatov Denis	349
Aguilar José Luis	305	Flores Fortunato	231
Algorri María Elena	17	Fu Lihua	3
Alor Hernández Giner	255	G	
Altamirano Leopoldo	79	Gallegos Funes Francisco	359
Amin Mohammad N.	495	Garrido M. Rubén	211, 231
B		Gatton Thomas M.	495
Badir Hassan	483	Gómez Juan M.	315
Balderas C. Tomás	403	Gutiérrez F. Octavio	221
Barrón Estrada M. Lucía	447, 459, 471	Guzmán G.	167, 201, 265
Barrón Héctor	79	Guzmán Ramírez Enrique	373
Bautista Thompson E.	43, 331	H	
Bolshakov Igor A.	147, 157	Herrera Oscar	243
Bussler Christoph	315	J	
C		Junwu Zhu	127
Calvo Hiram	147	Jiménez Hernández H.	29, 103
Camacho Nieto Oscar	415	Jiménez Salazar Héctor	139
Carreto, Chadwick	265	K	
Castro Sánchez Noé	157	Kinsella Stephen	315
Alejandro	17	Kuri, Ángel F.	243
Cornejo Alejandro	17	L	
Cristal Adrián	429	Landassuri Moreno V.	189
Cruz Janeth	79	Levachkine Serguei	177
Cuayáhuatl Heriberto	291	M	
Cumplido Parra René A.	403	Manrique Ramírez Pablo	373
D		Martínez G. A. Rafael	211, 231
Dey Pradip Peter	495	Menchaca R.	201
Díaz de León S. Juan L.	91	Mendoza C. Juan	211
Díaz Elva	341	Mendoza Mendiola J.	67
F		Silvano	
Felipe Riverón Edgardo	67	Mitre José	277
Fensel Dieter	315	Morales Silva Erick	67
Figuroa Nazuno Jesús	29, 43, 103, 189, 331	Moreno M.	Pág./Page 167, 177,

Moyotl Hernández Edgar 201, 265
139

N

Navarro Moldes Leandro 277

O

Olmedo Aguirre J. Oscar 255
Oropeza Rodríguez José 115
Luis
Orozco del Castillo M. 55
Ortiz Cornejo Ángel 291

P

Padilla Felipe 341
Pérez Corona Carlos 291
Pichat Etienne 483
Pogrebnyak Oleksiy 373
Ponce de León Eunice 341
Ponomaryov Volodymyr 359

Q

Quintero Rolando 167, 177,
201, 265

R

Ramírez Marco A. 429
Ramos Fonseca A. 103
Rosales Silva Alberto 359
Ruíz Ramírez Jorge I. 67

S

Sánchez Fernández Luis 387
Pastor
Sánchez Garfias Flavio A. 67, 91
Sánchez H. Jorge 231
Sidorov Grigori 157
Silva Leyva David 67
Solís Estrella H. 331
Stansifer Ryan 447, 459,
471
Suárez C. Miguel S. 231
Suárez Guerra Sergio 115

T

Torres M. 167, 177,
201, 265,

V

Valero Mateo 429
Veidenbaum Alexander 429
V.
Vera Félix José Ángel 157
Villa Vargas Luis 415, 429
Villar Briones Alejandro 415

W

Wang Gaofeng 3

X

Xiaojun He 483

Y

Yáñez Márquez Cornelio 91
Yi Jiang 127

Z

Zatarain Cabada Ramón 447, 459,
471
Zhang Meng 3

Organización / Organization

Programm Committee Co-Chairs

Dr. Jesús Figueroa
Dr. Alexander Gelbukh
Dr. Cornelio Yáñez
Dr. Oscar Camacho

Comité / Programm Committee

Dr. Francisco Jesús Sánchez, (Intel Labs), España.
Dr. Oscar Iván Lepe A., (CICESE), México.
Dr. Alex Ramírez, (UPC), España.
Dr. Juan Luis Díaz de León S., (CIC-IPN), México.
Dr. Sergio Suárez G., (CIC-IPN), México e (ICIMAF), Cuba.
Dr. Klaus M. Linding Boss, CIDETEC-IPN), México.
Dr. Mateo Valero, (UPC), España.
Dr. Ramón Beivide, (U. Cantabria), España.
Dr. Oscar Camacho N., (CIC-IPN), México.
Dr. Luis Villa, (IMP), México.
Dr. Alexander V. Veindenbaum, (U. California), Estados Unidos.
Dr. Luis P. Sánchez, (CIC-IPN), México.
Dr. Cornelio Yáñez M., (CIC-IPN), México.
Dr. Manuel Guzmán R., (CINVESTAV), México.
Dr. Oleksy Progrenbyak, (CIC-IPN), México.
Dr. Carlos Aguilar I., (CIC-IPN), México.
Dr. Pedro Marcuello P., (Intel Labs), España.
Dr. Marcos R. De Alba, (ITESM-CEM), México.
Dr. Alexander Gelbukh, (CIC-IPN), México.
Dr. Jesús Figueroa N., (CIC-IPN), México.
Dr. Igor Bolshakov, (CIC-IPN), México.
Dr. Grigori Sidorov, (CIC-IPN), México.
Dr. Dennis Filatov, (CIC-IPN), México.
Dra. Sofía Galicia Haro, México.

Organización / Organization

Árbitros Adicionales / Additional Reviewers

iram Calvo Castro.
icente Cubells Nonel.

Comité de Organización / Organizing Committee

✓ic. Luis Hernández Lara.
✓.P. Raquel Torres Frausto.
✓ic. Miguel Ranferi Silva.
ng. Arturo Arreola Ceja
 . Martín Moreno Ramírez
✓ic. Carmen Rodríguez Aparicio
✓ic. Elvia Cruz Morales
 . Ignacio García Araoz
✓ic. Oralia del Carmen Pérez
 . Juan Antonio Ayala
Gra. Juana Galicia
Gra. Leticia Andrade

**Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Octubre de 2004.
Edición 500 ejemplares.**

Information processing is the base of modern society, industry, science, and education. It is perhaps the most rapidly growing, most dynamic and influencing area of research and development, actively penetrating in all other research fields and in virtually all areas of human activity.

This volume contains 43 carefully selected papers by 96 authors from China, France, Ireland, Mexico, Spain, USA, and Venezuela, presenting the most recent developments in a wide variety of areas related to information processing and computing. The papers are arranged into three large thematic groups:

Artificial Intelligence, including such topics as pattern recognition and image processing, natural language processing, and geoprocessing;

Computing Science, including such topics as programming languages, scientific computation, networking and distributed systems, and modeling and simulation;

Computer Engineering, including such topics as digital signal processing and digital system design.

The editorial effort resulting in this volume was supported by the ACM (Association for Computing Machinery) and CyT Americas (Science and Technology of the Americas Foundation).

ISBN:970-36-0194-4

ISSN: 1665-9899

