Table of Contents

Índice

Page/Pág.

Artificial Intelligence, Robotics, and Control

Computer Vision and Image Processing

Natural Language Processing and Music Processing

Software Engineering

Networked, Mobile, and Ubiquitous Computing

Educational Applications

Computer Architecture

Graphical Markov Model Learning with a Double-Stressed Genetic Algorithm (GML-2SGA)

Elva Díaz[1] , Eunice Ponce de León[1] , Felipe Padilla[1]

[1] Departamento de Sistemas Electrónicos
Centro de Ciencias Básicas
Universidad Autónoma de Aguascalientes
elva.diaz@itesm.mx
{eponce, fpadilla}@correo.uaa.mx

Abstract. The graphical model learning is considered as a multiobjective optimization problem and a genetic algorithm added with two stressors over the exploration and exploitation capabilities (GML-2SGA) is presented here and applied to obtain a Pareto front. To stress the exploitation of the GA a Metropolis step is added to the genetic operators, and to stress the exploitation, independent samples are taken from different combinations of the GA parameter values. An experimental design is executed using benchmarks of complexities sparse, medium and dense, generated with a Markov Model Random Sampler (MMRS). It is compared to GMS-MGA, and the performance is assessed using the mean number of cliques of the true model identified by the algorithms. As results, the algorithm presented here identifies more cliques of the true models than the one used to compare, in all complexity types, and the complexity of the models made a difference in the performance of the algorithms.

1 Introduction

A new paradigm known as Estimation of Distribution Algorithms (EDAs) in Evolutionary Computation [12] [17] makes use of graphical model estimation and selection (learning) to guide the search in the space of solutions in optimization problems. This has made model estimation and selection an important tool in Evolutionary Computation. However, graphical model selection is not an ease task. The space of models grows exponentially with the number of variables, and there is no polynomial time certificate for a model if one were available. So, one way to tackle the problem is to construct a heuristic.

The genetic algorithm has been used to learn models in Poli and Roverato [19] and Roverato and Paterlini [20]. In those papers the problem of learning a model was considered as an optimization one, using the Akaike index as a simple criterion to optimize. In Díaz and Ponce de Leon [6] the problem of learning models was considered a multiobjective optimization one. The two objectives are (1) the fitting of the model to the data measured by the Kullback-Leibler deviance, and (2) the simplicity of the model, measured by the number of edges of the graph. The two

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 29, 2007, pp. 3-14

Received 16/09/07
Accepted 19/10/07
Final version 24/10/07

objective functions are conflicting because the model that best fit the data is the one represented by a complete graph, whose fitting is zero. As the number of edges decrease the fitting grows. In Diaz and Ponce de Leon [6], a relative preference vector is used to convert the multiple objectives into a single one. In Ponce de Leon and Diaz [18] a Pareto front optimality criterion is used helped by a genetic algorithm (GMS-MGA).

Evolutionary Computation algorithms in search need to balance the extension of exploration of the space through recombination and mutation, with the extension of exploitation through the selection operator. If the solutions obtained are exploited too much, premature convergence is expected, but if too much stress is given on the search, the information obtained so far is not properly used, the execution time is enormous and the search exhibits a similar behavior to that of a random search. On the other hand, it is known that the convergence of a MOEA algorithm to the Pareto Front can not be assured by the performance of the genetic operators alone [5] [14]. This last author suggests introducing a Metropolis step to the genetic algorithm to assure the convergence. This step is really a local search stressor and the question is how the balance between exploration and exploitation is affected. To solve this problem, independent samples for a multistart method are used. Other related issue is confronted: the difficulty to design the genetic operators in such a way that the resulting offspring (a) is likely to inherit desirable properties of its parents, and (b) is capable of improving on its parents' solution quality. The GMS-MGA presented in [18], fulfill the (a) part. In the present paper the mutation and the recombination operators for discrete graphical models introduced in [6] and used in [18] are modified to fulfill both parts (a) and (b). The class of Evolutionary Algorithms obtained in this form could be put in the class of the some times called Memetic Algorithms and some other times Genetic Local Search Algorithms [14]. But the algorithm presented here is added with two stressors over the exploration and exploitation capability of the genetic algorithm, so, it will be named Graphical Markov Model Learning with a two Stressed Genetic Algorithm (GML-2SGA).

2 Content

The content of the paper is:
(1) to define the graphical Markov model representation, (Section 3)
(2) to define the multiobjective graphical Markov model learning problem, (Section 4)
(3) to define the stressing methods used: Multi-start method and Metropolis step with Boltzman distribution, (Section 5)
(4) to define the Graphical Markov Model Learning with a two Stressed Genetic Algorithm (GML-2SGA), (Section 6)
(5) to perform an experiment to compare the two algorithms, GMS-MGA and GML-2SGA (Section 7),
(6) to discuss the results and obtain conclusions (Sections 8, 9).

3 The Graphical Markov Model Class

In this section first, the graphical Markov model and its hypergraph representation are defined, and second, the hypergraph operators to handle the models are defined.

3.1 The Graphical Markov Model Class

A graphical Markov model consists of an undirected graph and a subclass of probabilistic distributions for random variables that can be represented by this undirected graph [13]. “Represented” means that the set of nodes of the graph are the random variables, and that an edge between two nodes is absent, when theses two random variables are conditionally independent given the rest of the variables in the graph. The graph that has this last property (represent the structure of interactions of the random variables) is known as Markov graph and the graph together with the class of distributions is known as a probabilistic graphical Markov model.

In this paper only discrete binary random variables are considered, but the probabilistic graphical model class used in this paper is unrestricted, meaning it that every simple graph can be used to represent the probabilistic conditional independences between the random variables of the probability model. In this type of models, it is necessary to use some iterative algorithm to perform the parameter estimation. In this paper a modification of the generalized iterative scaling [4] that will appear soon, is used. The concepts, definitions and properties of graphical models are taken from the book of Lauritzen [13].

3.2 Hypergraph Representation of a Discrete Graphical Markov model

The graphical models can be characterized by a set E= {E_1,...,E_k} of pair-wise incomparable (w.r.t. inclusion) subsets from V, named the generating class, that is to be interpreted as the maximal sets of permissible interactions [13].

To represent a graphical model in a convenient way a hypergraph structure is used. A hypergraph is a collection of subsets, $H = \{H_1, H_2, ..., H_k\}$ named hyper edges, of a finite set V of nodes of a graph. Each hyper edge corresponds to a clique of the graph. This collection of subsets satisfies the same property as the generating class, that is, they are pair-wise incomparable (w.r.t. inclusion) [2]. So, a one to one correspondence can be established between the generating class E and the hypergraph H. [13]

A discrete graphical Markov model can be represented by M = (H, P(X)) where H is a hypergraph and P(X) is a class of probability distributions [13].

The intersection between two hypergraphs H_1 and H_2, is the family of sets intersections taking one hyper edge from each of the two hypergraphs,

$$H_1 \wedge H_2 = \{h_i \wedge h_j, \forall h_i \in H_1 and \forall h_j \in H_2\} \qquad \textbf{(1)}$$

This operation between two hypergraphs will be used in the next section to handle with hypergraphs models.

3.3 Operators to Handle with Hypergraphs Models

To handle with hypergraph models, different types of operators are needed.

The Nabla menus ∇- operator is a hypergraph used to define a unary operator over a hypergraph. It is defined by two binary chains:

∇- = (b_i,b_j)', where b_i = (1 1 1 1 0 1 1 1 1), and where the number 0 is in the position i, and where b_j = (1 1 1 1 1 0 1 1), where the number 0 is in the position j.

Intersecting Nabla menus with an hypergraph, the edge (i, j) is taken out of the graph, that is,

∇- $\wedge$H is one edge less than H.

The Nabla plus is defined in a similar but more complicated way (See [6]). The intersection of two hyper graphs is used as the crossover operator, and the Nabla menus and Nabla plus operators are used as mutation operators to take an edge out and to put an edge in, in a graph [6].

4 Multiobjective Graphical Markov Model Selection

4.1 Model Learning from Data: A Multiobjective Optimization Problem

Model learning from data problem is an optimization problem with two objective functions. First, the learned model must be as near to the data sample as possible (best fitting), and second, the model must be, as simple as possible to evade the over fitting. Best fitting to the data means, that the Kullback-Leibler deviance [1], [11] from the model to the sample is minimum. The Kullback-Leibler deviance from the complete graph to the sample is null, but, at the same time this model contains all the data noise. Objective functions for each of the tasks are needed, that is, an objective vector function with two components must be considered. Let denote it as O= (O_1, O_2). For the first task, the Kullback-Leibler divergence, or relative entropy is used, and for the second task a measure of complexity, the number of edges, is used. The problem is, to minimize the two elements of O. The two objective functions are conflicting because the model that best fit the data is the complete graph, and the Kullback-Leibler divergence grows when the edges number decreases.

Classical multiobjective optimization methods convert multiple objectives into a single one by using a relative preference vector of preferences [6]. Unless a reliable and accurate preference vector is available, the optimal solution obtained by such methods is highly subjective. One way out of this problem is to use the Pareto optimality. The Pareto optimality is based on the concept of dominance [5]. A model is not dominated when there is no other model in a class that is better in all the objectives. The subclass of non dominated models in a class is the Pareto front. The

Pareto front is a solution set and not a single solution. So, a multiobjective optimization problem requires multiple trade-off solutions to be found. The task is then, to find as many different trade-off solutions as possible. A way to find many different solutions is to use an Evolutionary Algorithm (EA) that works with a population of solutions. This ability of an EA makes it unique in solving multiobjective optimization problems. The ability of a Genetic Algorithm (GA) to obtain multiple optimal solutions in one single simulation run makes it especially well prepared to play an important role in finding the Pareto front.

In this paper a multi-start GA is used to generate as many different trade-off solutions as possible, and then a multiobjective selection algorithm is used to obtain a Pareto front. In order to use a GA, the hypergraph model representation, and the operators to manipulate models defined in section 3 are used.

4.2. Objective Function Definition

Let define the components of the objective function O= (O_1, O_2).

The fitting function O_1, is defined based on Information Theory criteria [11], [14].

Let $L(\hat{m}_n^M|x)$ be the likelihood ratio statistic for a sample of size n from a multinomial distribution, where $\hat{m}_n^M|x$ is the maximum likelihood estimate [3], [4], [9] assuming that the true model is M . The log likelihood ratio in the multinomial case is

$$G^2(M) = \log L(\hat{m}_n^M|x) = -2\sum x_i \log \frac{\hat{m}_i^M}{x_i} \qquad (2)$$

Then the objective function to minimize is:

$$O = (O_1, O_2) = (G^2(M), l(M)) \qquad (3)$$

where l(M) is the number of edges of the model M.

Let

$$G(M, M_0) = \frac{G^2(M_0) - G^2(M)}{G^2(M_0)} \qquad (4)$$

where M_0 is the equiprobability model.

Let

$$l(v, M) = \frac{\max l_v - l(M)}{\max l_v} \qquad (5)$$

where $\max l_v$ is the maximal number of edges formed with v vertices.

The aggregated convex fitting index for the model M is defined as in [6] by

$$CFI(M) = p(G(M,M_0)) + (1-p)(l(v,M)) \tag{6}$$

This index is used as a tool in the genetic algorithm.

5 Stressing the GA

5.1 Multi-start Method

The multi-start method consists of a strategy for searching parts of the space, starting from different initial solutions [10]. The objective of these methods is to avoid falling in a local optimum. These methods provide an appropriate tool to introduce diversity to the population of the evolutionary algorithm. In this paper the objective of a multi-start method is to visit different parts of the search space, widening the exploration. The parameter values of the genetic algorithm are used to define the multi-start method. Fixing the per cent selected to reproduce, the probability to mutate, and the weight assigned to fitting against the simplicity of the models, the genetic algorithm is repeated for each combination of values. (See Table No. 1). The exploitation is widened when more values of the parameters are tested. The multi-start method was tested with three values for each parameter (27 start points) and with two values for each parameter (8 start points) and there was no difference in the results. So, the method with two values for each parameter, were used to perform the full experiment. For each combination of the parameters values, 5 random samples are generated. A total of 40 independent samples are taken.

5.2 Metropolis Step and the Boltzman Distribution

The well-known Metropolis step is a fundamental part of the Simulated Annealing (SA) algorithm. It was first introduced by Metropolis [16] to simulate the physical annealing process of solids. With the same words of Metropolis, let E(X) be the energy of a solid X, then the solid is perturbed, let E(X') be the energy of the perturbed solid and the objective of the algorithm is to accept a new state X' if its energy E(X') is lower than the energy E(X). The decision of accepting a new state is made by the α criterion defined as:

$$\alpha = \exp\left(-\frac{\delta E(X)}{kT}\right) \tag{7}$$

where

$$\delta E(X) = E(X') - E(X) \tag{8}$$

T denotes the temperature and k is the Boltzman constant. For k=1, at each T the SA algorithm aims to draw samples from the Boltzman equilibrium distribution:

$$\pi_T(x) \propto \exp(-\delta(E(X))/kT) \tag{9}$$

For the experiments performed in this research k=1 and T=1 are used.

6 Two Stressed Genetic Algorithm (GML-2SGA)

In this section a description and a pseudocode of the GML-2SGA algorithm and each of its parts are given.

6.1 The Main Stressed Algorithm

The main stressed genetic algorithm is sketched in Algorithm 1.

To learn a graphic Markov model an approximate Pareto front is initialized and updated at each step of the stressed genetic algorithm, to obtain after k generations a list of models (the approximate Pareto front) that contains the approximate best model for each number of edges. To diversify the searching part of the algorithm (stress exploring), the genetic algorithm is added with a multi-start method that obtains a sample with one of a list of genetic parameters combinations. The genetic populations obtained are used to update the approximated Pareto front [5]. To stress the exploitation part of the algorithm an M-step algorithm is added to the genetic operators.

Algorithm 1. Main stressed GA (GML-2SGA)

```
begin
   Initialize the Pareto Front
   repeat
      Initialize the diversity parameters
      Choose an initial population
      Calculate the number of edges and the fitness of each model
      repeat
         Perform truncation selection for the population i
         Perform M-step crossover or mutation
         Calculate the number of edges and the fitness of each model
         Actualize the Pareto front
      until (Some stopping criterion applies (k populations))
   until (Multi-start parameters combination are over)
end
```

6.2 The M step Crossover and Mutation Operators

The hybridized crossover and mutation operators include an M-Step. They leave the class of graphical Markov models closed, and conserve (specially the crossover operator) the heritability of the common interaction structures of the parents.

Algorithm 2. M-Step crossover algorithm

```
begin
   Select  X, Y from Pop
   repeat
       $Z \leftarrow$ crossover(X, Y)
       $\delta E(Z|X,Y) \leftarrow E(Z)-E(X,Y)$
       $U \leftarrow$ rand(0,1)
       if $(U < \min\{1,\exp(-\delta E(Z|X,Y))\})$ then $Pop \leftarrow Z$
   until (Some stopping criteria applies)
end
```

Algorithm 3. M-Step mutation algorithm

```
begin
   Select  X from Pop
   repeat
      $Z \leftarrow$ mutation(X)
      $\delta E(Z|X) \leftarrow E(Z)-E(X)$
      $U \leftarrow$ rand(0,1)
      if $(U < \min\{1,\exp(-\delta E(Z|X))\})$ then $Pop \leftarrow Z$
   until (some stopping criteria applies)
end
```

Algorithm 4. M-step crossover or mutation

```
begin
    With probability p select two parents: obtain an
    offspring by crossover
    Assess the offspring fitness
    if (The offspring is not dominated by one of the
        fathers)
        then Add it to the new population
        else Add it to the new population with M step
             criterion for crossover
    With probability 1-p select one parent: with
    probability q take one edge out
    Assess the offspring fitness
    if (The child is not dominated by the father)
        then Add it to the new population
        else Add it to the new population with M step
             criterion for mutation
    With probability 1- q: put one edge in
    Assess the offspring fitness
    if (The child is not dominated by the father)
        then Add it to the new population
```

```
       else Add it to the new population with M step
            criterion for mutation
end
```

6.3 Algorithm Components Description

Multi-start estates: The parameter combinations of the genetic algorithm are used as diversity factors. They are: τ % of individuals selected to reproduce (25, 50); p % of individuals selected to mutate (25, 45). Parameter of the convex index (0.60, 0.45) Initial population: We form the initial population taking one edge out from the saturated model. To attain this objective the Nabla menus operator is used. Then, ξ % of the best evaluated models is taken to begin the genetic algorithm.

Fitting: The convex fitting criterion acts as the genetic algorithm fitness.

The Pareto front: for each number of edges, the best adjusted model is saved at each population of the genetic algorithm.

In a random fashion the algorithm decides to do crossover with probability p or mutation with probability 1- p. The crossover operation of two models is defined by the binary intersection operator.

The mutation operator could take one edge out at random with probability 0.7 (using the operator Nabla menus) or put an edge in (using the operator Nabla plus) at random with probability 0.3.

7 Experimental Design

To compare the two algorithms with models of different complexities [8], an experiment was designed and run. A conditional independence restrictions structure was selected at random of the type dense, mean and sparse. The structures are defined by its cliques (See table 2). To asses the performance of the algorithms, simulated samples from known models of 10 and 12 variables, sparse, medium and dense (see Table 2), are generated with a Markov Structure Random Sampler (MSRS) presented in [7].

Each algorithm is run 6 times with each type of model. The experiment and the algorithm is programmed in C++ and executed in a Pentium IV PC at 1.6 MHz.

Table 1. Starting values

Per cent selected to reproduce	Per cent to mutate	Convex fitting index parameter
25	25	0.60
50	45	0.45

8 Results and Discussion

The mean execution time of the algorithm proposed in this paper is ≤ 5.12 minutes for 10 variables, and ≤ 14.28 minutes for 12 variables in C++, over a PC at 1.6 GHz. The sparse models are more difficult to identify because they have few edges, and are immersed in a huge space like a "Needle in a Haystack". The mean execution time for dense models is 21 seconds more than for medium models in the case of 10 variables, and 3.87 minutes more in the case 12 variables, indicating that the complexity of the models make the computations time longer. Some explanation for that could be that the independent sample sizes are not enough to produce convergence in more complex models.

The performance of the GML-2SGA algorithm is better than the performance of the GMS-MGA, in the case of the sparse and the medium models, in the case of the dense model the performance are essentially the same (See Table No. 2).

Table 2. GMS-MGA vs. GML-2SGA

MODELS	MODEL'S MAXIMAL CLIQUES	GMS-MGA Correct Cliques Mean	GML-2SGA Correct Cliques Mean	Mean Run time
10 VAR. SPARSE	AB BC CD DE EF FG GH HI IJ (9 cliques)	6.65	7.35	5 min.
10 VAR. MEDIUM	ABC CDE EFG GHI JA (5 cliques)	3.95	4.92	4.91 min.
10 VAR. DENSE	ABCD DEFG FGHI HIJA (4 cliques)	2.75	2.75	5.12 min.
12 VAR. SPARSE	AB BC CD DE EF FG GH HI IJ JK KL (11 cliques)	9.43	10.12	11.71 min.
12 VAR. MEDIUM	ABC CDE EFG GHI IJK JKL (6 cliques)	4.11	5.83	10.41 min.
12 VAR. DENSE	ABCD DEFG FGHI HIJK IJKL (5 cliques)	3.98	4.27	14.28 min.

9 Conclusions and Recommendations

The sparse models are more difficult to identify because they have few edges, and are immersed in a huge space like a “Needle in a Haystack”. The GML-2SGA for binary variables has a better performance than the GMS-MGA because the sparse models are not more difficult to determine than the medium and dense ones as was the case with the GMS-MGA. The medium complexity models are almost always completely determined. The execution mean run time grows with the number of nodes (variables) in the model and with the complexity (sparse, medium, and dense) of the graphical model.

References

1. Akaike, H: A new look at the statistical model identification, IEEE Transactions on Automatic Control 19 (1974) 716-23.
2. Berge, C.: Graphs and Hypergraphs, North- Holland (1973).
3. Birch, M. W.: Maximum-likelihood in three way contingency tables, Journal of the Royal Statistical Society, Series B 25 (1963) 220-33.
4. Darroch, J. N., Ratcliff, D.: Generalized iterative scaling for log-linear models. Annals of Mathematical Statistics, 43 (1972) 1470-80.
5. Deb, K.: Multi-objective optimization using evolutionary algorithms, Wiley, Chichester, UK, (2001).
6. Díaz, E., Ponce de León, E.: Discrete Markov Model Selection by a Genetic Algorithm, Avances en Ciencias de la Computación e Ingeniería de Cómputo Vol II, Ed. Sossa Azuela, Aguilar Ibáñez, Alvarado Mentado, Gelbukh Khan, IPN, (2002) 315-324.
7. Díaz, E., Ponce de León, E.: Markov Structure Random Sampler Algorithm (MSRS) from unrestricted Discrete Models, Proceedings Fifth Mexican International Conference on Artificial Intelligence MICAI 2006, Special Session IEEE Computer Society, Eds. Alexander Gelbukh, Carlos Alberto Reyes García (2006) 199-206.
8. Díaz, E., Ponce de León, E.: Modelling and Simulation Complexity for Discrete Graphic Markov Models: an Experimental Study, WSEAS Transactions on Mathematics, Issue 1 Volume 3 January (2004) 288-291.
9. Díaz, E., Ponce de León, E.: Discrete Graphic Markov Model Selection by a Genetic algorithm based on Different Estimation of Distribution Algorithms, WSEAS Transactions on Mathematics, Issue 2 Volume 4 April (2005) 110-116.
10. Hickernell, F.J.,Yuan, Y.: A Simple Multistart Algorithm for Global Optimization, OR Transactions, Vol 1 (1997) 1-11.
11. Kullback, S., Leibler, R. A.: On information and sufficiency, Ann. of Math. Statistics, 22, (1951) 79-86.
12. Larrañaga, P., Lozano, J.A.: Estimation of Distribution Algorithm: A new tool for Evolutionary Computation, Kluwer Academic Publisher (2001).
13. Lauritzen, S.L.: Graphical Models, Oxford Science Publications, (1996).
14. Mc Kay, David J. C.: Information Theory, Inference and Learning Algorithms, Cambridge University Press (2003).
15. Merz, P., Freisleben, B.: Fitness Landscape and Memetic algorithm Design, In New Ideas in Optimization. Ed. By D. Corne, M. Dorigo, and F. Glover, Mc Graw Hill (1999) 245-260.

16. Metropolis N., Rosenbluth A. E., Rosenbluth M. N., Teller A. H., Teller, E.: Equation of state calculations by fast computing machines, J. Chem Phys 21 (1953) 1087-1092.
17. Ponce de León, E., Díaz, E., Padilla, F.: Evolutionary Algorithm based on Markov graphical models of promising solutions, Research on Computing Science. Advances in Artificial Intelligence, Computing Science and Computer Engineering, Vol. 10: Eds. Jesús Figueroa Nazuno, Alexander Gelbukh Khan, Cornelio Yánez Márquez, Oscar Camacho Nieto, IPN (2004) 341-347.
18. Ponce de León, E., Díaz, E.: Discrete Graphic Markov Model Selection by Multiobjective Genetic Algorithm (GMS-MGA). Proceedings 15th Internacional Conference on Computing CIC 2006, IEEE Computer Society, Eds. Alexander Gelbukh, Sergio Suarez Guerra (2006) 18-23.
19. Poli, I., Roverato, A.: A genetic algorithm for graphical model selection, J. Italian Statist. Soc. (2) (1998) 197-208.
20. Roverato, A., Paterlini, S.: Technological modeling for graphical models: an approach based on genetic algorithms, Comp.Statistic & Data Analysis 47, (2004) 323-337.

Modeling and Simulation of a 5 DOF Robot Manipulator for Assembly Operations Using Artificial Vision

Luis Felipe Mejía Rodríguez[1], Jesús Carlos Pedraza Ortega[2],
Joaquín Santoyo Rodríguez[1]

[1]Universidad Tecnológica de Querétaro, Av. Pie de la Cuesta S/N,
Col. Lomas de San Pedrito Peñuelas, Querétaro, Qro., C.P. 76148, México
lmejia@uteq.edu,mx, jsantoyo@uteq.edu.mx
[2]Centro de Ingeniería y Desarrollo Industrial, Av. Pie de la Cuesta # 702,
Col. Desarrollo San Pablo, Querétaro, Qro., C.P. 76130, México
jpedraza@cidesi.mx

Abstract. The purpose of this paper is to show the applied methodology for modeling, simulation and automatic program generation for basic assembly operations using a 5 degree of freedom robot manipulator and a vision system. The robot used was a Pegasus II from Amatrol, and the vision system consisted of a web cam and a personal computer. The forward and inverse kinematics of the manipulator is developed, and the results are used for simulation purposes. The main goal is to manipulate small rods and put them into holes drilled in a mounting board, with a program for the robot generated using visual information. The software tools used are Visual C++ 6.0, the OpenGL graphics library and Matlab 7.

1 Introduction

Flexible manufacturing (FM) must have a fast response to changes in production requirements. In order to implement a FM cell, it is common to use robots, automatic conveyors and storage, and CNC machinery. Despite this, it is still a common practice to program the computers (including PLCs and robot controllers) “by hand”, delaying the adjustment of the cell. One way for the solution of this problem is the automatic generation of programs for the computers, considering the physical characteristics of the operations required, like size, form, position, etc.

Computing power and costs enables us to process vision data for different tasks of increasing complexity. One of such tasks is the automatic generation of programs for robot controllers, using vision information to obtain the points that must be followed by a robot, and thus allowing us to do the trajectory planning.

Robot modeling requires a deep understanding of the structure and dimensions of the robot. Knowing the dimensions and movement restrictions of the robot, it is possible to apply the Denavit-Hartenberg method to do the direct kinematics, and using some positions constraints, based on the task required for the robot, the inverse kinematics can be calculated.

Received 10/09/07
Accepted 19/10/07
Final version 24/10/07

This work shows the development of the direct and inverse kinematics for an educational robot of industrial quality, the Pegasus II from Amatrol, and its application for simulation purposes. This work also shows an alternative for the application of artificial vision and image processing in the extraction of positions where assembly operations are required, and the generation of the associated program.

The software tools picked were Visual C++, OpenGL libraries and Matlab. C++ is a general-purpose object-oriented language, very powerful and common in software development. Matlab is a powerful mathematical environment, with toolboxes specific for some kinds of applications, such as image acquisition and processing, robotics, fuzzy control and neural networks. With the right mix of C++ and Matlab, the developer can optimize the life cycle of software development, and with the use of OpenGL a virtual representation can be obtained.

Some related works are [1], [2] and [5].

2 Development

In order to reach the goals of this work, several activities are necessary. This activities are detailed in the next sections

2.1 System Design and Operational Environment

The general structure of the system is shown in the following figure.

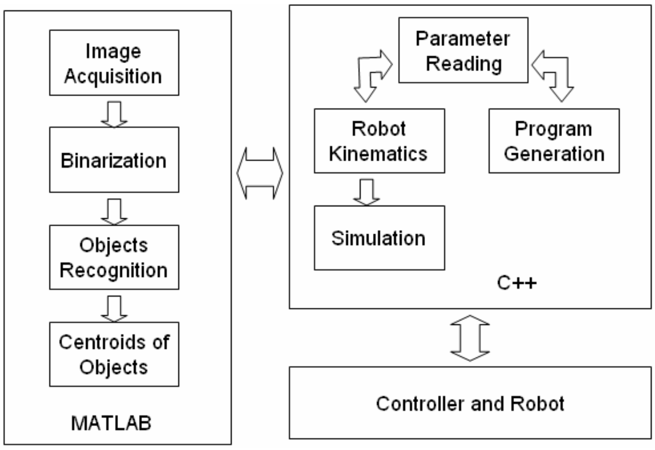


Fig. 1. System design and operational environment.

The tasks that involve images are carried out in Matlab. The outcome of this stage is an array of coordinates of the centers of the holes in a mounting board, stored in a text file. The tasks of simulation and robot program generation are carried out in C++. The outcome of this stage is a virtual robot doing the assembly and a program for the robot controller that can do the actual assembly.

2.2 Specifications and Characteristics of the Pegasus II from Amatrol

The robot manipulator used was a Pegasus II from Amatrol. This is an educational robot, but it has many common characteristics with an industrial one.

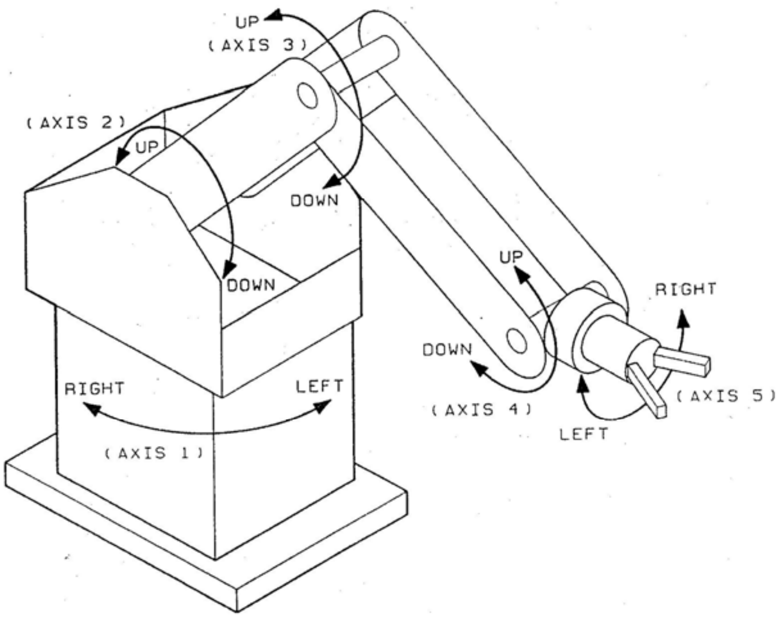


Fig. 2. General structure of the Pegasus II

The robot has 5 degree of freedom, with the following rotation restrictions.

Table 1. Manipulator restrictions.

Articulation	Angle of rotation
1 (waist)	345°
2 (shoulder)	220°
3 (elbow)	270°
4 (pitch)	270°
5 (roll)	Unlimited

2.3 Modeling

The Denavit-Hartenberg parameters of the robot are as follows.

Table 2. Denavit-Hartenberg parameters of the robot.

Join	θ	D	a	α
1	q_1	L_1	0	π/2
2	q_2	0	L_2	0
3	q_3	0	L_3	0
4	q_4	0	0	π/2
5	q_5	L_4	0	0

Where L_1 = 324 mm, L_2 = 230 mm, L_3 = 230 mm and L_4 = 42 mm.

For the inverse kinematics there exists some constraints, derived from the tasks characteristics:

- The gripper will always be oriented downwards (normal to the mounting surface of the robot), so the roll will be 0° ($q_5 = 0$).
- The pitch will be a function of q_2 and q_3: $q_4 = q_3 - q_2$.
- Angle q_1 is free.
- Angles q_2 and q_3 are determined in terms of x, y and z, and in general there will be 2 possible solutions.
- The z value for the pick and place coordinates is fixed, but the actual z value will change while the robot is moving between the storage and a mounting hole.

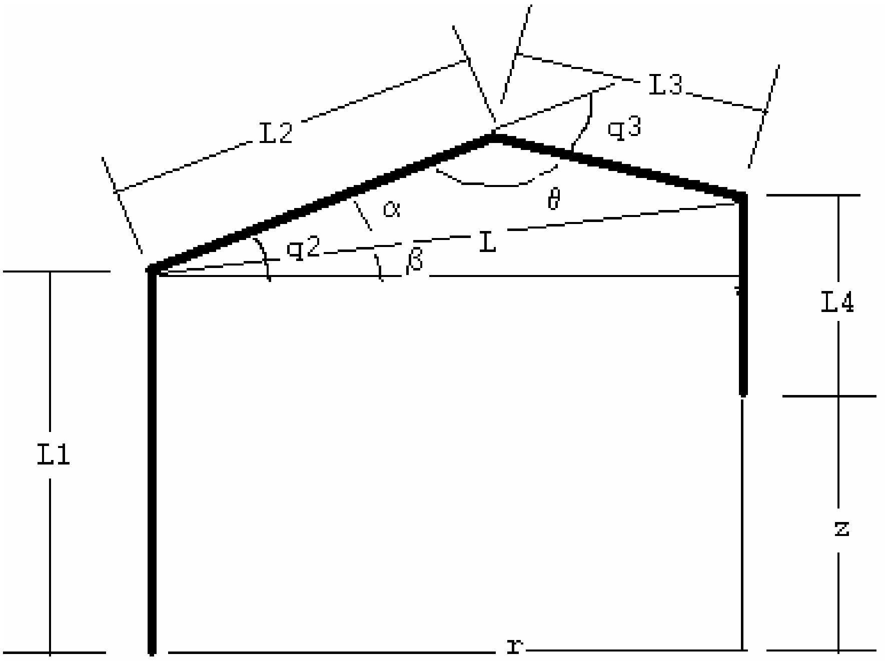


Fig. 3. Free body diagram for the inverse kinematics.

With these diagram, we can obtain the next equations, solving the inverse kinematics:

$$q_1 = \arctan\left(\frac{y}{x}\right) \tag{1}$$

$$r = \sqrt{x^2 + y^2} \tag{2}$$

$$\theta = \pi + q_3, \quad \cos\theta = -\cos q_3, \quad \sin\theta = -\sin q_3 \tag{3}$$

$$L^2 = x^2 + y^2 + \left(L_4 + z - L_1\right)^2 \tag{4}$$

$$L^2 = {L_2}^2 + {L_3}^2 - 2L_2L_3\cos\theta = {L_2}^2 + {L_3}^2 + 2L_2L_3\cos q_3 \tag{5}$$

$$q_3 = \arccos\left(\frac{L^2 - {L_2}^2 - {L_3}^2}{2L_2L_3}\right) \tag{6}$$

$$\beta = \arctan\left(\frac{L_4 + z - L_1}{r}\right) \tag{7}$$

$$\frac{\sin\theta}{L} = \frac{\sin\alpha}{L_3} = \frac{-\sin q_3}{L} \tag{8}$$

$$\alpha = \arcsin\left(\frac{-L_3\sin q_3}{L}\right) \tag{9}$$

$$q_2 = \alpha + \beta \tag{10}$$

$$q_4 = q_2 - q_3 - \pi/2 \tag{11}$$

$$q_5 = 0 \tag{12}$$

2.4 Simulation

A basic solid model corresponding to the links of the manipulator is used at this time. Given a sequence of desired mounting locations, the robot can follow the trajectories between a fixed position (the storage) and a mounting hole.

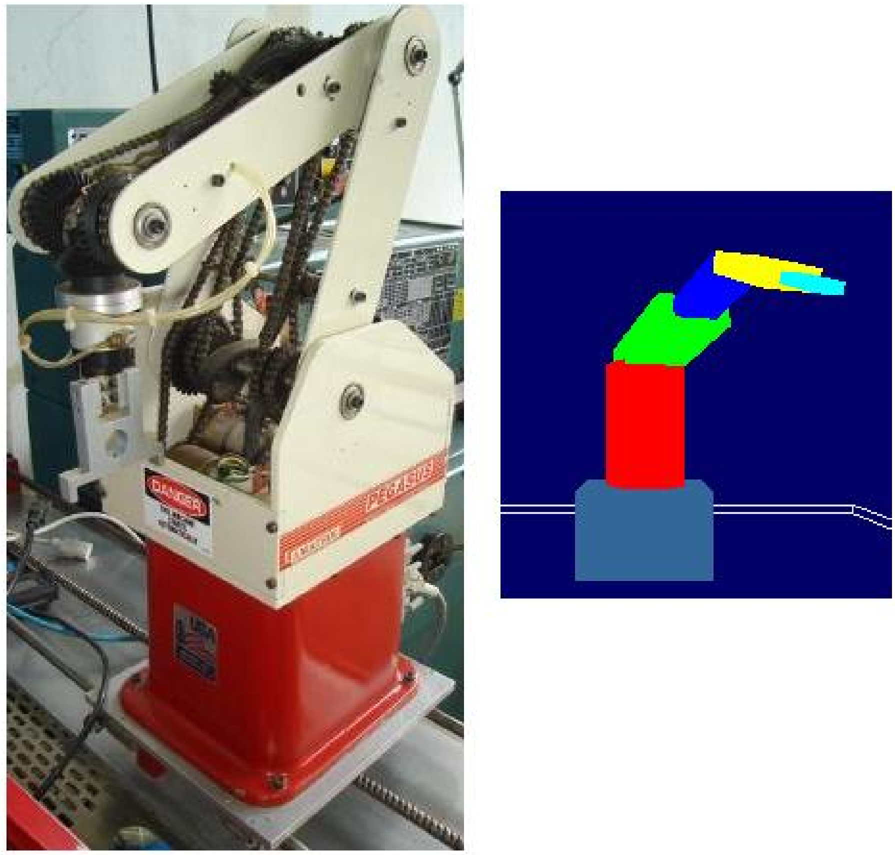


Fig. 4. Robot and virtual model.

2.5 Vision System

The vision system in use is a basic one, but the results obtained with it are good enough, and it wasn’t necessary to spend money buying specialized hardware. A generic web cam and a PC are used. It is desired to have a good standard illumination, but small variations can be compensated by the processing algorithm.

2.6 Image Processing

With the image of the assembly that is required already in a file, the next step is the binarization, based on the optimum threshold.

With the binary image it is possible to recognize the objects in the image applying a cluster analysis. In this case the objects of interest are the holes for the mounting operation. This can be carried out using 4-connected or 8-connected components.

Now each component (object) is processed to obtain the coordinates of its centroid, using the formulas:

$$x = \frac{\sum_{i=1}^{n} x_i}{n} \tag{13}$$

$$y = \frac{\sum_{i=1}^{n} y_i}{n} \tag{14}$$

Where (x_i, y_i) are the coordinates of each pixel that belongs to a component and n is the total number of pixels. Next, a factor is applied to map the pixel coordinates with absolute coordinates in mm. This factor is determined by the physical size of the board and the position where it will be put in front of the robot for the assembly.

2.7 Robot Program Generation

The Pegasus robot controller allows the use of Cartesian coordinates in the application programs. In a structured world, it is possible to determine the position of the centroid of the mounting holes with the required precision, and to use these points to generate the trajectory of the manipulator.

A Cartesian coordinate for the Pegasus robot has 5 components: X, Y, Z, P (pitch) and R (roll). X, Y and Z can be expressed as inches or millimeters, and the angles normally are expressed in degrees. The Cartesian coordinates for the home position are X=0”, Y=16.0”, Z=21.5”, P=–90°, R=0°.

The commands that can be used to move the robot and their syntax are:

- PMOVE(X coordinate, Y coordinate, Z coordinate, Pitch, Roll)
- TMOVE(X coordinate, Y coordinate, Z coordinate, Pitch, Roll)

The units are expressed as follows:

- Thousandths of an inch (X=7000 means X=7”).
- Tenths of a millimeter (X=100 means X=10 mm).
- Hundreths of a degree (P=900 means P=9 degrees).

A sample program in MCL (Manufacturing Control Language) generated by the system is (4 mounting holes):

```
PMOVE <-3000,100,2000,-1221,0>
PMOVE <-3000,100,800,-942,0>
GRASP
PMOVE <-3000,100,2000,-1221,0>
PMOVE <-689,3530,2000,-1138,0>
PMOVE <-689,3530,800,-869,0>
RELEASE
PMOVE <-689,3530,2000,-1138,0>

PMOVE <-3000,100,2000,-1221,0>
PMOVE <-3000,100,800,-942,0>
GRASP
PMOVE <-3000,100,2000,-1221,0>
PMOVE <-290,2851,2000,-1236,0>
PMOVE <-290,2851,800,-951,0>
RELEASE
PMOVE <-290,2851,2000,-1236,0>

PMOVE <-3000,100,2000,-1221,0>
PMOVE <-3000,100,800,-942,0>
GRASP
PMOVE <-3000,100,2000,-1221,0>
PMOVE <96,3542,2000,-1147,0>
PMOVE <96,3542,800,-878,0>
RELEASE
PMOVE <96,3542,2000,-1147,0>

PMOVE <-3000,100,2000,-1221,0>
PMOVE <-3000,100,800,-942,0>
GRASP
PMOVE <-3000,100,2000,-1221,0>
PMOVE <495,2862,2000,-1232,0>
PMOVE <495,2862,800,-949,0>
RELEASE
PMOVE <495,2862,2000,-1232,0>

STOP
```

3 Experimental Work

So far the model for the robot is complete and the programs for simulation, image acquisition and processing, and robot program generation are also complete.

An example mounting board is shown in the next figure.

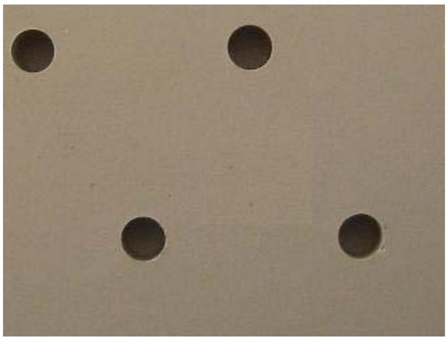


Fig. 5. Acquired image (the actual color of the board is gray).

The binary image after processing is:

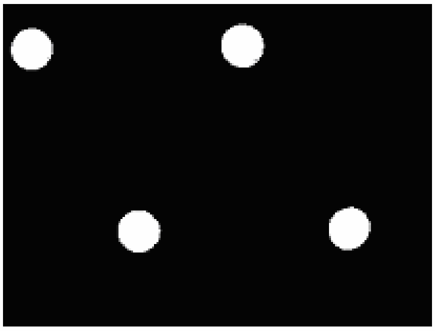


Fig. 6. Binarized image.

The parameters obtained using the algorithm, expressed in pixels, are:

Centroid 1 :	22.1160	34.0973
Centroid 2:	102.0321	169.8586
Centroid 3:	179.2522	31.5304
Centroid 4:	258.9001	167.6744

4 Conclusions and Future Work

At this moment, only basic assembly operations have been done, but in the future it will be possible to add another camera to obtain new data from the assembly operation required, allowing the execution of more advanced assembly between parts.

The robot used, despite the industrial quality of its components, is intended for educational purposes. It will be possible to replace the robot with an industrial one, and with the appropriate changes in the parameters of the robot and the syntax of the instructions, the simulation and program generation will be possible.

The web cam can also be replaced with one designed for industrial use, using a good illumination scheme, which can lead to better parameter determination for the assembly.

For simulation purposes it is possible to do a CAD definition of the robot, export it to a VRML or OBJ format, and use the file to obtain a better approximation to the real world.

References

1. Soto Cajiga J. A., Vargas Soto J. E., Pedraza Ortega J. C.: Trajectories generation for a robot manipulator using image processing and splines. Segundo Congreso Internacional de Ingeniería, Querétaro, Qro., México (2006).
2. Gamiño M., Pedraza J. C., Ramos J. M., Gorrostieta E.: Matlab–C++ interface for a flexible arm manipulator simulation using multi-language techniques. Proceedings of the fifth Mexican international conference on artificial intelligence (MICAI'06). México (2006).
3. Myler H., Weeks A.: Computer imaging recipes in C. Prentice-Hall, USA (1993).
4. Kurfess, Thomas R. (Editor): Robotics and automation handbook. 1st edition. CRC Press, USA (2005).
5. Viramontes Reyna, J. L., Pedraza Ortega, J. Carlos, González Galván, Emilio J.: Vision-based manipulator control using a monocular vision system. 5th Internatiomal Simposium on Robotics and Automation, San Miguel Regla, Hidalgo, México (2006).
6. Gonzalez, Rafael C., Woods, Richard E.: Digital image processing, 2nd edition. Prentice-Hall, USA (2002).

Stabilization of the Inverted Pendulum by means of a nested saturation function

Carlos Aguilar Ibañez.[1] and Oscar O. Gutiérrez F.[1]

Instituto Politécnico Nacional,
Centro de Investigación en Computación
México, D.F., México

Abstract. In this letter we present a technique to stabilize the inverted pendulum mounted on a cart. The pendulum is brought to its top position with zero displacement of the cart by using a nested saturation function. This can be done because the original system can be expressed as a chain of integrator with an additional nonlinear perturbation. Under the assumption that the pendulum angle is initialized above the upper-half plane, the obtained closed-loop system is semi-global asymptotically and locally exponentially stable.

1 Introduction

The stabilization of the inverted pendulum on a cart (**IPC**) is a very interesting problems. This device consists of a free vertical rotating pendulum with a pivot point mounted on a cart. The cart can be moved horizontally by means of an horizontal force, which is the control of the system. Because the pendulum angular acceleration can not be directly controlled, this system is a classic example of an under-actuated mechanical system, that is, it has fewer actuators than degrees-of-freedom. For this reason the majority of fully-actuated systems control techniques cannot be directly applied to stabilize this kind of systems. Maneuvers such as the stabilization around the unstable vertical position and others related to the stabilization around its homoclinic orbits are almost impossible to achieve (see[14],[9],[16]) because the **IPC** is not input-output linearizable using a static feedback [7]. Also, when the pendulum moves through the horizontal plane it looses controllability and other geometric properties [9]. On the other hand, a linearized model of the **IPC** is is locally controllable around the unstable equilibrium point and can be locally stabilized by a direct pole placement procedure [15].

There are two important problems related to the stabilization of this device. The first is swinging the pendulum up from the hanging position to the upright vertical position. An energy control strategy is usually applied for this purpose. Once the system is close to the desired top position with low enough speed (the inverted pendulum remains swinging while getting closer and closer to the origin), and suddenly, by means of a simple change in the controller, from the non-linear to the linear controller, it is possible to keep the pendulum in the

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 29, 2007, pp. 25-36

Received 05/07/07
Accepted 19/10/07
Final versioinn 24/10/07

desired equilibrium [3],[6],[9],[16],[14]. The second problem in importance consists in stabilization of the **IPC** around its unstable equilibrium point (which is defined when the angle position and the displacement of the cart are zero), assuming that the pendulum is initially above the horizontal plane, or lies inside an open vicinity of zero. In general, this vicinity defines a stability domain for the closed-loop system. Exist many works related with the second problem but a detailed review of the state of the art of the problem here treated is beyond the scope of this work. However, we refer the interested reader to the following references: [11],[19],[1],[2],[13],[12], [5].

In this paper we develop a simple strategy for the stabilization to the **IPC**. We transform the original system into a four-order integrator chain plus an additional nonlinear perturbation based in the procedure presented in [13]. Then, by applying the simple Lyapunov method, a controller based on nested saturated functions is introduced. Next, we show that the closed-loop solution is bounded, which allows to prove that the system is locally exponentially stable. The stability analysis of the whole four order system is fairly simple, as opposite to [13],[10], because, we neither use a fixed-point equation, nor a highly complex domain of attraction estimation. Also, we do not use the contraction mapping theorem to verify the convergence of all states to zero.

The remaining of this paper is organized as follows. Section 2 presents the dynamical model of the **IPC** and how this system is converted into an integrators chain, by means of some suitable transformations. In section 3 we present a stabilizing nonlinear controller for the **IPC**. The corresponding stability and convergence analysis is carried out in the same section. Section 4 presents some computer simulations and the conclusions are given in section 5.

2 Nonlinear Model

Consider the traditional **IPC**, as shown in Figure 1. This system is described by the set of normalized differential equations [12]:

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

where x is the normalized displacement of the cart, θ is the angle that the pendulum forms with the vertical, u is the horizontal normalized force applied to the cart, and $\delta > 0$ is a constant depending directly on the cart mass and the pendulum mass, respectively. Defining $v = \ddot{\theta}$ and canceling $\ddot{x}$ from the last two differential equations, we have after substituting:

$$u = (1+\delta)\tan\theta_1 - \theta_2^2\sin\theta_1 + v(\cos\theta_1 - \frac{1+\delta}{\cos\theta_1}),$$

into system (1) the following

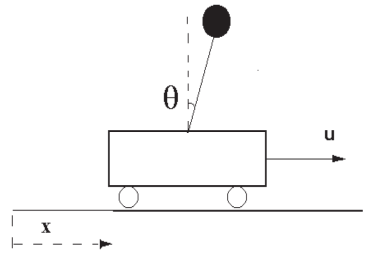


Fig. 1. The inverted pendulum cart system

$$\begin{aligned} \dot{x}_1 &= x_2, \\ \dot{x}_2 &= \tan\theta_1 - \tfrac{v}{\cos\theta_1}, \\ \dot{\theta}_1 &= \theta_2, \\ \dot{\theta}_2 &= v. \end{aligned} \tag{2}$$

Where v is a fictitious controller acting in the coordinate θ_1. Of course, the above representation is validated for all $\theta_1 \in (-\pi/2, \pi/2)$. *Hereafter, we refer to this restriction as assumption **A1** and system (2) as a partial linearized model of the **IPC**.*

The main objective is to control the partial linearized system (2) under assumption **A1**. In other words we want to bring, both, the pendulum angle position and the cart displacement to zero.

2.1 Transforming the partial linear model into a chain of integrators:

From [13] we introduce

$$z_1 = g(\theta_1) + x_1 \; z_2 = g'(\theta_1)\theta_2 + x_2 \tag{3}$$

where the function g is selected such that the derivative of the variable z_2 does not depend directly on the control v, that is,

$$\dot{z}_2 = \tan\theta_1 + v(g'(\theta_1) - \frac{1}{\cos\theta_1}) + \theta_2^2 g''(\theta_1). \tag{4}$$

Consequently,

$$g'(\theta_1) = \frac{1}{\cos\theta_1}; \quad g(\theta_1) = \log\left(\frac{1+\tan(\theta_1/2)}{1-\tan(\theta_1/2)}\right), \tag{5}$$

these relations are well defined for $|\theta_1| < \pi/2$.

Now, from (3) and (5), we can write system (2), as follows,

$$\begin{aligned}\dot{z}_1 &= z_2,\\ \dot{z}_2 &= \tan(\theta_1)(1 + \tfrac{\theta_2^2}{\cos\theta_1}),\\ \dot{\theta}_1 &= \theta_2,\\ \dot{\theta}_2 &= v.\end{aligned} \tag{6}$$

In order to express the above system as an integrators chain plus a nonlinear perturbation, the following global nonlinear transformations is introduced

$$\begin{aligned} w_1 &= \tan\theta_1, \quad w_2 = \sec^2\theta_1\theta_2,\\ v_f &= \sec^2\theta_1 v + 2\theta_2^2\tan\theta_1\sec^2\theta_1\end{aligned} \tag{7}$$

which leads to

$$\begin{aligned}\dot{z}_1 &= z_2,\\ \dot{z}_2 &= w_1 + \tfrac{w_1 w_2^2}{(1+w_1^2)^{3/2}},\\ \dot{w}_1 &= w_2,\\ \dot{w}_2 &= v_f.\end{aligned} \tag{8}$$

2.2 Comment

A similar representation of model (8) was proposed in [10]. There, the control and the non-actuated coordinate are not completely uncoupled.That is the control acts directly on the additional nonlinear perturbation, and as consequence, the resulting closed-loop system has a more restrictive domain of attraction. In our case, the control action is completely uncoupled, so that, it is possible to increase the stability domain for all the initial conditions that belongs in the upper half plane.

3 Control strategy

A nested saturation function is suggested to use to control a nonlinear system that can be expressed, approximately, as a chain of integrators with a nonlinear perturbation. This technique, introduced in [18], has been used for the stabilization of a linear integrators chain and controlling mini-flying machines [20]. Thus, our stability problem will be solved as follows. First, a linear transformation is used to directly propose a stabilizing controller; then, it is shown that the proposed controller guarantees the boundedness of all states and, after a finite time, it is possible to ensure that all states converge to zero.
Definition: $\sigma_m(s) : R \to R$ is a linear saturation function, if it satisfies

$$\sigma_m(s) = \begin{cases} s \quad if \ |s| \leq m \\ m \ sign(s) \quad if \ |s| > m \end{cases}. \tag{9}$$

3.1 A feedback controller

Let us introduce the following linear transformations:

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} = \begin{bmatrix} 1\,3\,3\,1 \\ 0\,1\,2\,1 \\ 0\,0\,1\,1 \\ 0\,0\,0\,1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ w_1 \\ w_2 \end{bmatrix}, \tag{10}$$

then system (8) is transformed as,

$$\begin{array}{l} \dot{q}_1 = v_f + q_2 + q_3 + q_4 + 3\delta_a(q) \\ \dot{q}_2 = v_f + q_3 + q_4 + \delta_a(q) \\ \dot{q}_3 = v_f + q_4 \\ \dot{q}_4 = v_f \end{array} \tag{11}$$

where the perturbation δ_a is given by

$$\delta_a(q) = q_4^2 G(q_3 - q_4), \tag{12}$$

and

$$G(w) = \frac{w}{(1+w^2)^{3/2}}, \tag{13}$$

for simplicity, we stand for $q = [q_1, q_2, q_3, q_4]$.
Remark 1: *Note that* $\max |G(w)| \leq k_0 = \frac{2}{3^{3/2}}$, *and it is achieved when* $w = 1/\sqrt{2}$.

Finally, a stabilizing controller may be readily proposed as:

$$v_f = -q_4 - k\sigma_\alpha(\frac{q_3 + \sigma_\beta(q_2 + \sigma_\gamma(q_1))}{k}). \tag{14}$$

where k is positive constant.

3.2 Boundedness of all states

We show in four steps that the proposed controller (14) ensures that all the states are bounded; moreover, the bound of each state depends directly on the controller parameters.[1]

First step. Define the positive definite function $V_4 = q_4^2/2$. Then the time derivative of V_4 is given by,

$$\dot{V}_4 = -q_4^2 - kq_4\sigma_\alpha(q_3/k + \sigma_\beta(q_2 + \sigma_\gamma(q_1))/k). \tag{15}$$

[1] Note that $|q_4(t)| \leq q_4(0)e^{-t} + \alpha$ and $|G(q_3 - q_4)| \leq k_0$. Therefore, the right hand of the closed loop system (14) and (11) is locally Lipschitz. Consequently, the states $\{q_1, q_2, q_3\}$ cannot have a finite time scape [8].

It is clear that $\dot{V}_4 < 0$, when $|q_4| \geq \alpha k$. Consequently, there is a finite time $T_1 > 0$ such that

$$|q_4(t)| < \alpha k \quad \forall t > T_1. \tag{16}$$

Second step. Let us analyze the behavior of the state q_3. Consider the definite positive function $V_3 = q_3^2/2$. Differentiating V_3, we obtain after substituting (14) into the third differential equation of (11):

$$\dot{V}_3 = -q_3 k \sigma_\alpha (q_3/k + \sigma_\beta(q_2 + \sigma_\gamma(q_1))/k), \tag{17}$$

where α and β are selected such that $\alpha > 2\beta$. Clearly, if $q_3 > \beta$, then $\dot{V}_3 < 0$ and there is a finite time $T_2 > T_1$ after which

$$|q_3(t)| < \beta \;\; forallt > T_2. \tag{18}$$

When the above condition is satisfied, the control v_f turns out to be

$$v_f = -q_4 - q_3 - \sigma_\beta(q_2 + \sigma_\gamma(q_1)) \; \forall t > T_2. \tag{19}$$

Third step. Substituting (19) into the second equation of (11), we obtain

$$\dot{q}_2 = -\sigma_\beta(q_2 + \sigma_\gamma(q_1)) + \delta_a(q). \tag{20}$$

Define a positive definite function $V_2 = q_2^2/2$. Differentiating V_2 along of the trajectories of (20) yields[2]

$$\dot{V}_2 = -q_2 \left(\sigma_\beta(q_2 + \sigma_\gamma(q_1)) + \delta_a(q)\right) \tag{21}$$

where β and γ must satisfy $\beta > 2\gamma + k_0\alpha^2 k^2$. Obviously, if $|q_2| > \gamma + k_0\alpha^2 k^2$ then $\dot{V}_2 < 0$. Hence, there exist a finite time $T_3 > T_2$ after which

$$|q_2| < \gamma + k_0 k^2 \alpha^2, \;\; \forall t > T_3. \tag{22}$$

Consequently, q_2 is bounded and the control v_f becomes

$$v_f = -q_4 - q_3 - q_2 - \sigma_\gamma(q_1), \;\; \forall t > T_3. \tag{23}$$

*Fourth step***.** Substituting (23) into the first equation of (11), we obtain

$$\dot{q}_1 = -\sigma_\gamma(q_1) - 3\delta_a(q). \tag{24}$$

Now, define a positive definite function $V_1 = q_1^2/2$. By differentiating V_1 along of the trajectories of (24), we obtain

$$\dot{V}_1 = -q_1(\sigma_\gamma(q_1) + 3\delta_a(q)), \tag{25}$$

where parameter γ must be chosen such that $\gamma > 3k_0\alpha^2 k^2$. If $q_1 > 3k_0\alpha^2 k^2$, then $\dot{V}_1 < 0$, thus there exits a finite time $T_4 > T_3$ afterwards

$$|q_1| < 3k_0\alpha^2 k^2, \;\; \forall t > T_4. \tag{26}$$

[2] Recalling that after $t > T_3$, it has $|\delta_a(q)| \leq k_0\alpha^2 k^2$.

Consequently q_1 is also bounded. So, all the previous constraints on parameters α, β and γ can be summarized as

$$\alpha > 2\beta,\ \beta > 2\gamma + k_0 k^2 \alpha^2,\ \gamma > 3k_0 k^2 \alpha^2. \tag{27}$$

Manipulating the last inequalities, we have that

$$\alpha < 1/(14k_0 k^2). \tag{28}$$

Thus, parameter k may be taken as $14k_0k^2 = 1$ and the set of control parameters may be selected as

$$\alpha = r,\ \beta = r/2,\ \gamma = 3r/14, \tag{29}$$

for all $0 < r \leq 1$.

3.3 Convergence of all states to zero

We will prove that the closed-loop system given by (11) and (14) is asymptotically stable and locally exponentially stable, provided that the controller parameter satisfies (27).

Note that after $t > T_4$, the control law is no longer saturated, that is,

$$v_f = -q_1 - q_2 - q_3 - q_4,$$

and the closed-loop system turns out to be, as

$$\begin{aligned} \dot{q}_1 &= -q_1 + 3\delta_a(q), \\ \dot{q}_2 &= -q_1 - q_2 + \delta_a(q), \\ \dot{q}_3 &= -q_1 - q_2 - q_3, \\ \dot{q}_4 &= -q_1 - q_2 - q_3 - q_4, \end{aligned} \tag{30}$$

with δ defined in (12) . Let us define the following Lyapunov function

$$V = \frac{1}{2} q^T q, \tag{31}$$

Now, differentiating V along the trajectories of (30), we obtain

$$\dot{V} = -q^T M q + (3q_1 + q_2)\delta_a(q) \tag{32}$$

where

$$M = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}. \tag{33}$$

being M positive definite with $\lambda_{\min}\{M\} = 1/2$.

From **Remark 1** and ((12), we easily have that the second term of the right hand of(32) satisfies

$$\begin{aligned} |(3q_1+q_2)\delta(q)| &< \tfrac{k_0}{2}\left|(3q_1+q_2)q_4^2\right|; \\ &< \tfrac{k_0}{2}(q_4^4+(3q_1+q_2)^2). \end{aligned} \tag{34}$$

So, $\dot{V}$ fulfills

$$\dot{V} < -\frac{1}{2}\left[q_1^2+q_2^2-k_0(3q_1+q_2)^2\right] - \frac{q_4^2}{2}(1-k_0q_4^2) - \frac{1}{2}q_3^2. \tag{35}$$

From definition of k_0 and recalling that $14k_0k^2 = 1$, we obtain that the previous inequality is strictly negative definite, since

$$q_1^2+q_2^2-k_0(3q_1+q_2)^2 > 0, \tag{36}$$

and

$$-1+k_0q_4^2 \leq -1+4k_0k^2 < 0. \tag{37}$$

Therefore, $\dot{V}$ is strictly negative definite, and the vector state q locally exponentially converges to zero, after $t > T_4$.

It should be noticed that, proceeding as described, we obtain that the system (11) in closed-loop with the controller (14) is globally asymptotically stable and locally exponentially stable, when the parameters satisfy the restriction (27). However, we can only assure converge to zero of the original states $(x, \theta, \dot{x}, \dot{\theta})$, assuming that the initial angle of the pendulum belongs to the upper half plane, because (2) and (7) are well defined for $\theta \in (-\pi/2, \pi/2)$. That is, assumption **A1** is necessary to avoid the singular points $\theta = \pm\pi/2$.
From the above discussion, we have:

Proposition 1. *Consider the partial linearization model of the IPC as described by (2), under assumption* **A1**, *in closed-loop with the controller:*

$$v = -\theta_2\cos^2\theta_1 - k\sigma_\alpha\left(\tfrac{q_3+\sigma_\beta(q_2+\sigma_\gamma(q_1))}{k}\right)\cos^2\theta_1 - 2\theta_2^2\tan\theta_1^2, \tag{38}$$

where $k = \sqrt{1/(14\times 2^{3/2})}$, q_1, q_2 and q_3 are given by

$$\begin{aligned} q_1 = z_1+3z_2+3w_1+w_2;\ q_2 = z_2+2w_1+w_2; \\ q_3 = w_1+w_2, \end{aligned} \tag{39}$$

with

$$\begin{aligned} w_1 &= \tan\theta_1; & w_2 &= \theta_2\sec^2\theta_1; \\ z_1 &= \log\left(\tfrac{1+\tan(\theta_1/2)}{1-\tan(\theta_1/2)}\right)+x_1; & z_2 &= \theta_2/\cos\theta_1+x_2. \end{aligned} \tag{40}$$

Then the closed-loop system is semi-globally stable and locally exponentially stable provided that the control parameters α, β and γ satisfy the inequalities (27).

4 Numerical Simulations

The efficiency of the proposed control strategy was tested by computer simulations. The experiments were implemented in Matlab program. The controller parameter values were set as $\alpha = 0.99$, $\beta = 0.49$ and $\gamma = 0.214$, and the initial conditions were set as $\theta_1(0) = 1.18$ [rad], $\theta_2(0) = -0.05$ [rad/ sec], $x_1(0) = -0.6$ and $x_2(0) = 0.5$.

Figure 2 and Figure 3 show the closed loop responses to the proposed controller (38), when it applied to the partial linearized model (2). As can be seen, the state x_1 converges very slowly to zero, in comparison with the state θ_1. This is because the cart position increases until the angle position of the pendulum approaches to zero. This event is expected since, firstly, the controller brings the pendulum into a small vicinity of zero, while, the cart position reaches its maximum, and secondly, the controller forces to move the cart slowly to the origin. Besides, it should be noticed that the control strategy was carried out with slowly movements. Finally, figure 4 shows the behavior of the control input v and the proposed energy function V, respectively. As can be seen, the control input v goes to zero and the Lyapunov function is decreasing after $t > 10$ and also converges to zero.

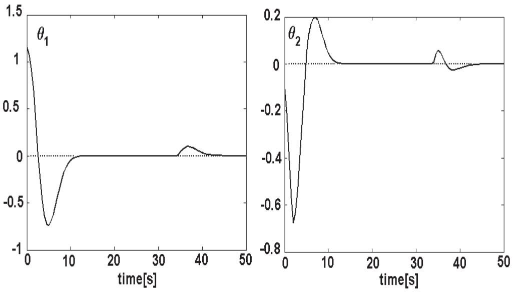


Fig. 2. Closed-loop response of the angle and the angular velocity to the proposed controller

5 Conclusions

A nested saturation-based controller for the stabilization of the **IPC** is presented, under assumptions that the initial value of the pendulum angle lies in the above horizontal plane. The fact that the **IPC** can be written (approximately) as a four cascaded integrators. Permits to use a nested saturation functions to design a stabilizing controller. The proposed controller makes the system be stable (under some restriction on the control parameters), and after some finite time assures that all states converge exponentially to zero. Physically, the control

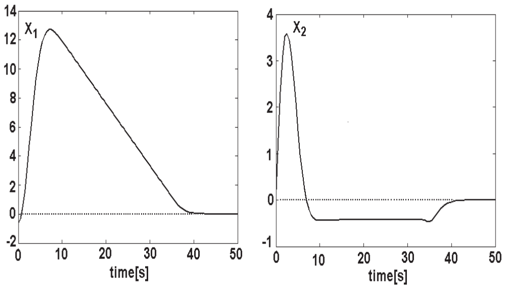


Fig. 3. Closed-loop response of the cart position and the cart velocity to the propose controller

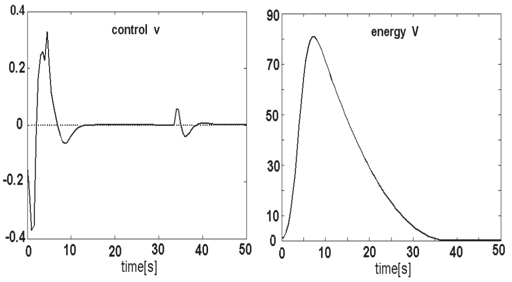


Fig. 4. Depict the behavior of the controller v and energy function V, respectively

strategy consist in bringing the pendulum close to the upper position, and then gradually the cart position is moving to the origin. The stability analysis is fairly simple, because it is carried out using Lyapunov's approach. Furthermore, some computer simulations have been performed in order to test the effectiveness of the proposed controller.

acknowledgements This research was supported by CIC-IPN, by the Secretaría de Investigación y Posgrado (SIP-IPN) under research Grant 20061128, and the Consejo Nacional de Ciencia y Tecnología (CONACYT-México), under Research Grant 46805, 105594. Octavio Gutiérrez F. is a scholarship holder of the CONACYT and he wants to thank for the received support.

References

1. Bloch, A. M., Chang,D.,Leonard, N. , and Marsden, J. E.: Controlled lagrangians and the stabilization of mechanical systems II: Potential shaping ,IEEE Trans. on Automatic Control. **46** (2001) 1556–1571.
2. Chang, D., Bloch, A., Leonard, N., Marsden, J.E., Woolsey, C.: Equivalence of Controlled Lagrangian and Controlled Hamiltonian Systems, Control and the Calculus of Variations **8** (2002) 393–422.
3. Chung, C.C. and Hauser, J.: Nonlinear control of a swinging pendulum, Automatica, **36** (2000) 287–295.
4. Fantoni, I. and Lozano, R.: Global Stabilization of the cart-pendulum system using saturation functions, In Proccedings of the 42nd IEEE Conference on Decision and Control, Maui Hawaii (Dec. 2002) 4393–4398.
5. Fradkov, A. L.: Swinging control of nonlinear oscillations, International Journal of Control, **31**(1995) 851–862.
6. Furuta, K., Yamakita, M. and S. Kobayash: Swing up control of inverted pendulum using pseudo-state feedback, Journal of System and Control Engineering. **206** (1992) 263–269.
7. Jakubczyk, B. , Respondek, W: On the linearization of control systems, Bull. Acad. Polon. Sci. Math. **28** (1980) 517–522.
8. Khalil, H. K.: Non-linear Systems, Prentice Hall, 2nd. Edition. N.J., 1996.
9. Lozano, R., Fantoni, I. and Block, D. J.: Stabilization ofthe inverted pendulum around its homoclinic orbit, Systems &Control Letters. **40** (2000) 197–204.
10. Lozano, R. and Dimogianopoulos, D.: Stabilization of a chain of integrators with nonlinear perturbations: Application to the inverted pendulum, in Proccedings of the 42nd IEEE Conference on Decision and Control, Maui Hawaii (Dec. 2002) 5191–5196.
11. Mazenc, F. and Praly, L.: Adding integrations, satured controls, and stabilization for feedforward system, IEEE Transactions on Automatic Control. **41** (1996) 1559–1578.
12. Aguilar-Ibañez., C. , Gutierrez F, O. and Suarez C, M. S.: Lyapunov based Control for the inverted pendulum cart system, Nonlinear Dynamics,Springer,**40**,**4**(2005) 367–374 .
13. Olfati-Saber, R.: Fixed Point Controllers and Stabilization of the Cart-Pole and Rotating Pendulum, In. Proc. of the 38th IEEE Conf. on Decision and Control, Phoenix Az. (Dec. 1999) 1174–1181.

14. Shiriaev, A.S., Pogromsky, A.,Ludvigsen, H. and Egeland, O.: On global propertiesof passivity-based control of an inverted pendulum, International Journal of Robust and Nonlinear Contol. **10** (2000) 283–300.
15. Sira-Ramirez, H. and Agrawal, S. K.: Differential flat system, Marcel Dekker, N Y, 2004.
16. Spong, M. V.: Energy Based Control of a class of Underactuated Mechanical Systems, IFAC World Congress, San Francisco CA. 1996.
17. Spong, M. W. and Praly, L.: Control of Underactuated mechanical systems using switching and saturation, In Proc. of Block Island Workshop on Control Using Logic Based Switching Springer-Verlag, 1996.
18. Teel, A.R.: Global stabilization and restricted tracking for multiple integrators with bounded controls, Systems &Control Letters(1992) 165–171.
19. Teel, A.R.: A nonlinear small gain theorem for the analysis of control system with saturation pendulum, IEEE Trans. on Automatic Control, **41**(1996) 1256–1270.
20. Castillo, P.,Lozano, R. and Dzul, A. ,Modelling and control of mini flying machines, Springer-Verlag, ISBN:1-85233-957. 2005.

High-Level Simulation of Chua's Circuit to Verify Frequency Scaling Behavior

R. Trejo-Guerra[1], E. Tlelo-Cuautle[1], J.M. Muñoz-Pacheco[1],
C. Cruz-Hernández[2], C. Sánchez-López[3]

[1] INAOE, Department of Electronics, México
{rodolfo13, etlelo, mpacheco}@inaoep.mx
[2] CICESE, Department of Electronics and Telecommunications, México
ccruz@cicese.mx
[3] UAT, Department of Electronics, México
cslopez@ingenieria.uatx.mx

Abstract. This work describes a high-level simulation technique to verify frequency-scaling behavior of Chua's circuit. Chua's diode is modeled by a piecewise-linear approximation which is simulated by using and combining Verilog and HSPICE. The frequency-scaling behavior is measured by both Verilog and HSPICE, and the last one is used to simulate the frequency spectrum to confirm the scaling.

1 Introduction

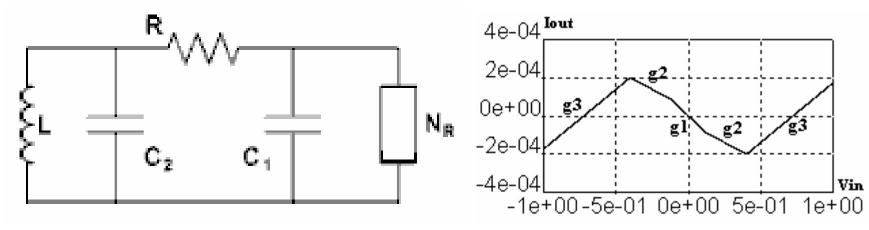
Nowadays, the electronic design automation (EDA) community is developing high-level simulation tools to help the designers to verify their design before the physical implementation. In this manner, hardware description languages (HDLs), such as Verilog [1], provides the environment to describe an electronic design at a level of abstraction higher than the transistor one, which is done normally in circuit design using HSPICE. Furthermore, it is possible to combine an HDL software (Verilog) with a transistor software (alike HSPICE) [2], in order to cover the gap between high-level to physical descriptions. That way, a designer can start the circuit description using equations within Verilog, and then gradually he can make a refinement process until obtaining a design at the transistor level using HSPICE. For instance, in [3] it is shown the high-level modeling of Chua's circuit using state variables and piecewise-linear (PWL) approximation [4], and in [5] is shown the implementation at the transistor level of abstraction.

Chaotic systems such as Chua's circuit can be described from high-level (Verilog) to circuit level (HSPICE). In this manner, the nonlinear element, i.e. Chua's diode, can be easily described by Verilog using PWL approximation in order to verify frequency-scaling behavior of the chaotic oscillator before its physical realization.

Chua's circuit is shown in Fig. 1(a). The PWL approximation of Chua's diode (N_R) is shown in section 2. Its Verilog description is derived in section 3. The HSPICE description is shown in section 4 along with the simulations using Verilog

Received 22/06/07
Accepted 19/10/07
Final version 24/10/07

and HSPICE. The frequency-scaling behavior and the frequency spectrum simulations are shown in section 5. Finally, the conclusions are summarized in section 6.



(a) (b)

Fig. 1. (a) Chua's chaotic oscillator, and (b) I-V characteristic of Chua's diode

2 PWL Approximation of Chua's Diode

According to [3], a third order chaotic oscillator can be described using only passive components except for N_R, as shown in Fig. 1. The system described by (1) depends on an initial condition, e.g. a voltage across a capacitor (Vc_1 or Vc_2) or a current through the inductor (I_L) to perform an oscillating behavior. The nonlinear component (N_R) enhances this oscillation and produces instability according with the I-V characteristic shown in Fig. 1(b). Basically, it performs the behavior of a two negative voltage to current slope device; the positive slopes have no effect on the chaotic phenomena but are generally presented by the nonlinear circuit. By using the values derived in [3], the slopes and break point values are: $g_1 = 1/1358$, $g_2 = 1/2464$, $g_3 = 1/1600$, $B_1 = \pm 0.114V$ and $B_2 = \pm 0.4V$. The behavior of Chua's diode is described by (2). By setting $C_1 = 450pF$, $C_2 = 1.5nF$, $L = 1mH$, $R = 1650\Omega$, these values maintain chaotic oscillation and scroll generation [3]. However, C_1, C_2, and L will be modified herein to show frequency-scaling behavior.

$$\begin{bmatrix} \frac{d}{dt}Vc_1 \\ \frac{d}{dt}Vc_2 \\ \frac{d}{dt}i_L \end{bmatrix} = \begin{bmatrix} -\frac{1}{RC_1} & \frac{1}{RC_1} & 0 \\ \frac{1}{RC_2} & -\frac{1}{RC_2} & \frac{1}{C_2} \\ 0 & -\frac{1}{L} & 0 \end{bmatrix} \begin{bmatrix} Vc_1 \\ Vc_2 \\ i_L \end{bmatrix} + \begin{bmatrix} \frac{i_{NR}}{C_1} \\ 0 \\ 0 \end{bmatrix} \tag{1}$$

$$i_{NR} = \begin{cases} g_2Vc_1 + (g_2 - g_1)B_1 & ....Vc_1 < -B_1 \\ g_1Vc_1 & -B_1 < Vc_1 < B_1 \\ g_2Vc_1 + (g_1 - g_2)B_1 & Vc_1 > B_1 \end{cases} \tag{2}$$

3 PWL Verilog-A Description

Chua's diode can be described by an I/V PWL function consisting of three different segments S_i parameterized by slopes g_i and breakpoints B_j which complete equations are given in (3) for a semi-plane X (positive/negative).

The I/V behavior is described by the Verilog sentence given in (4), as it is done in [2]. f(V) is the PWL function from Fig. 2, and its Verilog description is:

```
module fchua(x,y,g);
        input x;
        output y;
        electrical y,x,g;
        real res;
        analog begin
                if (V(x,g) < -0.4)
                res = V(x,g)/1600+4.500184e-4;
                else if ((V(x,g) > -0.4)&&(V(x,g) < -0.114))
                res = -V(x,g)/2464+3.76807e-5;
                else if ((V(x,g) > -0.114)&&(V(x,g)< 0.114))
                res = -V(x,g)/1358;
                else if ((V(x,g) > 0.114)&&(V(x,g) < 0.4))
                res = -V(x,g)/2464-3.76807e-5;
                else
                res = V(x,g)/1600-4.500184e-4;
                I(g,y) <+ -res;
        end
endmodule
```

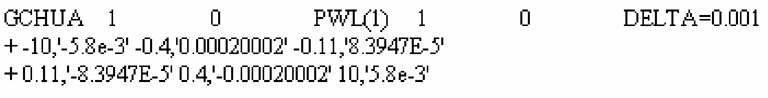
$$\begin{aligned} S_1(x) &= -\frac{1}{1358}x \\ S_2(x) &= -\frac{1}{2464}x \mu 37.6807 * 10^{-6} \\ S_3(x) &= \frac{1}{1600}x \mu 450.0184 * 10^{-6} \end{aligned} \qquad (3)$$

$$I(out,gnd) <+ f(V(in,gnd)) \qquad (4)$$

4 Verilog-A and HSPICE Simulations

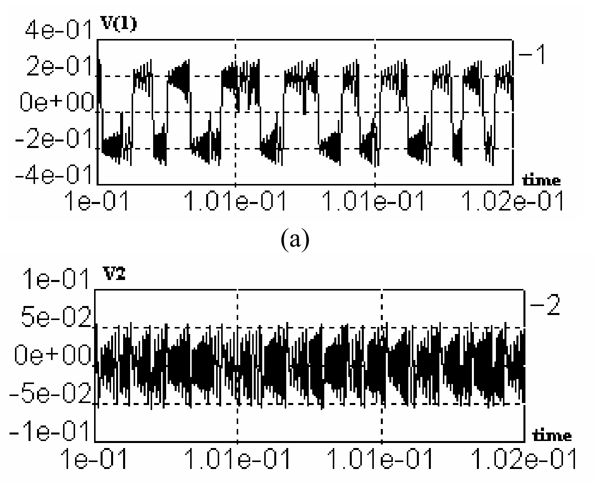
The HDL description, as the one shown in section 3, can be added to a more large circuit or system to prove its functionality using either or both Verilog and HSPICE. For instance, the comparison results between Verilog and HSPICE will confirm that this simple model has good approach with reality as concluded in [3],[5]. Furthermore, Chua's circuit was simulated by using Verilog-A to reproduce the chaotic phenomena shown in [5], by adjusting R. The voltage responses across the capacitors are shown in Fig. 2.

The HSPICE description of the I/V PWL behaviour of Chua's diode is done using a PWL polynomial voltage controlled current source, as follows:



```
GCHUA  1         0         PWL(1)  1         0         DELTA=0.001
+ -10,'-5.8e-3' -0.4,'0.00020002' -0.11,'8.3947E-5'
+ 0.11,'-8.3947E-5' 0.4,'-0.00020002' 10,'5.8e-3'
```

In Fig. 3 is shown the HSPICE simulation of Chua's circuit, so that one can conclude that the circuit keeps its behaviour when using either HSPICE or Verilog-A.



(a)

(b)

Fig. 2. Voltages across capacitors: (a) Vc_1 and (b) Vc_2

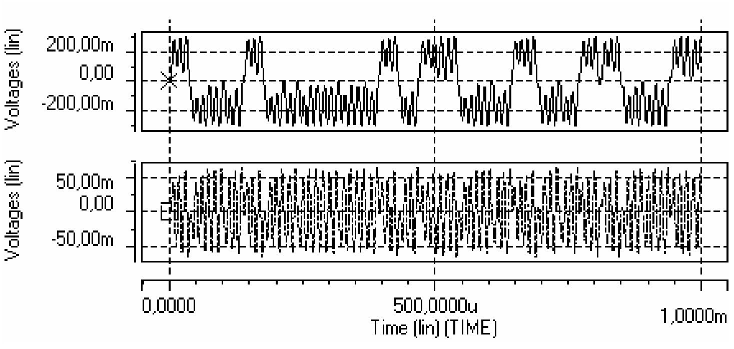


Fig. 3. Hspice response of the chaotic oscillator

The states trajectories (Vc_1 vs Vc_2) can be seen in Fig. 4.

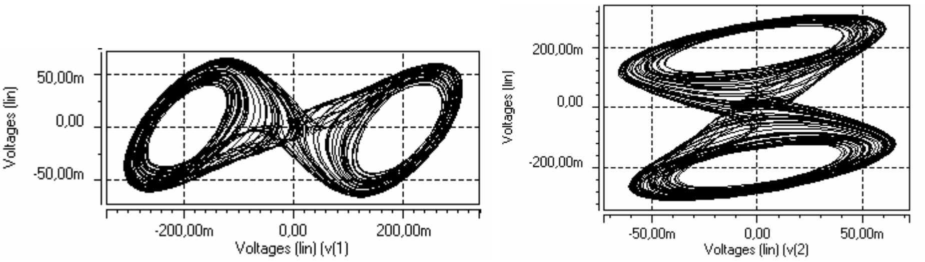


Fig. 4. State trajectories: Vc_1 vs Vc_2

5 Frequency Scaling Behavior

Chua's circuit can be used in secure communication systems as shown in [6]. Besides, it is quite convenient to verify its behavior at various ranges of frequency to cover wider applications. In [7] it is introduced an intelligent system to generate analog circuits which can be used to design Chua's diode and to reach higher frequencies. For instance, in this section is shown that by scaling the values of the capacitors and the inductor, Chua's circuit can scale its frequency spectrum.

From the results shown in Fig. 4(a), in Fig. 5 to Fig. 8 are shown chaotic oscillations which were scaled by 10, 1/2, 1/10 and 1/100, respectively. That is, each capacitor and the inductor are multiplied by the scaling factor. In each figure it is shown the behavior of the frequency spectrum, where x means mega Hertz on the horizontal axe.

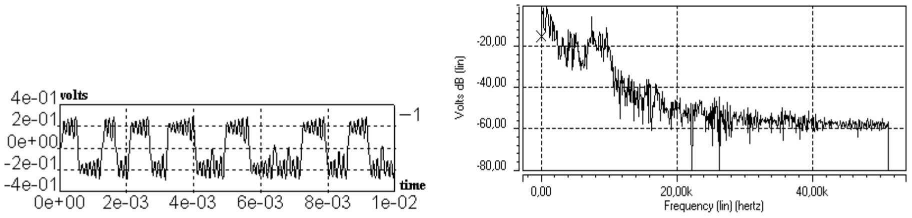


Fig. 5. Chaotic oscillation measured in Vc_1 at a scaling factor = 10, and its frequency spectrum

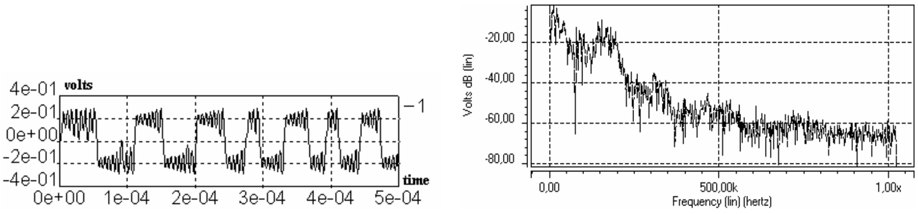


Fig. 6. Chaotic oscillation measured in Vc_1 at a scaling factor = 1/2, and its frequency spectrum

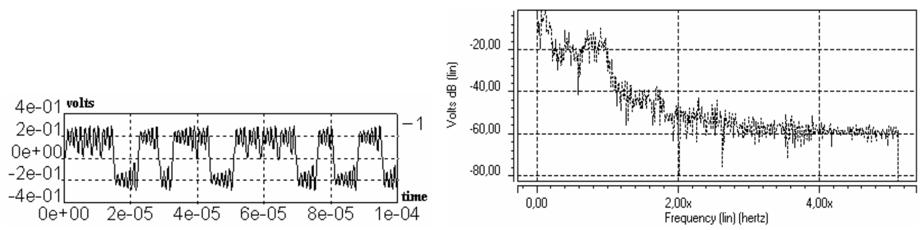


Fig. 7. Chaotic oscillation measured in Vc$_1$ at a scaling factor = 1/10, and its frequency spectrum

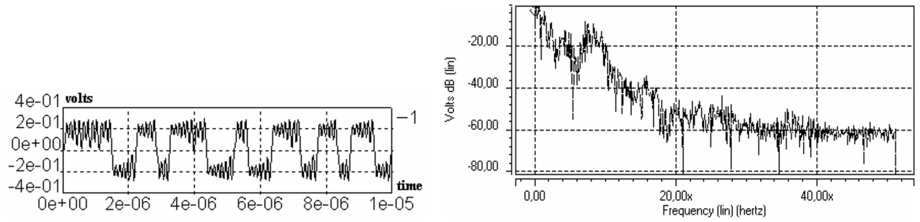


Fig. 8. Chaotic oscillation measured in Vc$_1$ at a scaling factor = 1/100, and its frequency spectrum

6 Conclusions

It has been shown the description of Chua's diode by using Verilog and HSPICE. Simulation results show that PWL approximation is suitable to verify the frequency scaling behaviour of Chua's circuit. Furthermore, it was shown that the frequency scaling presents a linear dependence with the component values. However, physical implementation will depend on the frequency limits of the analog circuits.

Equivalences between the Verilog-A models and Hspice models have been highlighted in order to prove sufficient concordance of both languages to describe circuits as macro-models giving hierarchy, simplicity and speed.

Acknowledgment

This work is supported by CONACyT/MEXICO under the project 48396-Y.

References

1. Kundert, K.S., Zinke, O.: The designer’s guide to Verilog AMS. Kluwer Academic Publishers, Boston (2004)
2. Tlelo-Cuautle, E., Duarte-Villaseñor, M.A., García-Ortega J.M., Sánchez-López, C.: Designing SRCOs by combining SPICE and Verilog-A. International Journal of Electronics 94(4) (2007) 373–379
3. Tlelo-Cuautle, E., Muñoz-Pacheco, J.M.: Numerical simulation of Chua´s circuit oriented to circuit synthesis. International Journal of Nonlinear Sciences and Numerical Simulation 8(2) (2007) 249–256
4. Vandewalle, J., Vandenberghe, L.: Piecewise-linear circuits and piecewise-linear analysis. In: Chen, W.K. (ed.): Circuits and Filters Handbook. CRC Press and IEEE Press (1995) 1034–1057
5. Tlelo-Cuautle, E., Gaona-Hernández, A., García-Delgado, J.: Implementation of a chaotic oscillator by designing Chua's diode with CMOS CFOAs. Analog Integrated Circuits and Signal Processing 48(2) (2006) 159–162
6. Cruz-Hernández, C., López, D., García, V., Serrano, H., Núñez, R.: Experimental realization of binary signals transmission using chaos. Journal of Circuits, Systems and Computers 14(3) (2005) 453–468
7. Tlelo-Cuautle, E. Duarte-Villaseñor, M.A.: Evolutionary electronics: automatic synthesis of analog circuits by GAs. In: Yang, A., Shan, Y., Bui, L. (eds.): Success in Evolutionary Computation, Series: Studies in Computational Intelligence , Vol. 92, Springer-Verlag, Berlin (2008)

Using Artificial Marks for Semi-Autonomous Wheelchair Localization

Marcel Yañez[1], Mireya García[1], Luis González[1],Alejandro Ramírez[2]

[1] Instituto Politécnico Nacional-CITEDI, Av. Del Parque No.1310, Tijuana BC,
marcel, lgonzal, mgarciav@citedi.mx
[2] Dpto R&D, PILIMTEC, Châteaugiron,Francia
alramirez10@yahoo.fr

Abstract. This paper describes a vision system to detect artificial landmarks for semi-autonomous Wheelchair (SAW) localization in structured indoor environments. The focus is placed on design and implementation of a computer vision based landmark detection system. The artificial landmark has been designed to be a low-cost, easy to implement and any user can be able to printing and placing this type of mark in the environment of its interest. Also, a practical advantage of the method is the computational time saving. It requires less processing resources than traditional methods. Experimental results have shown that the artificial landmark is suited for robot localization and navigation.

Keywords: Artificial landmarks, detection, vision system, wheelchair.

1 Introduction

Recent advances in mobile robotic technologies have already made enormous contributions in many areas. Robots are moving away from factories into environments such as private homes, in order to assist people in daily routines. One of the more researched areas in assistive technology is the development of intelligent or Autonomous Wheelchairs (AWs). By integrating intelligence into a powered wheelchair, a robotic wheelchair has the ability to safely transport a user to their desired destination.

Nevertheless, even though in the last couple of years, numerous research groups around the world have worked in the field of assistive technologies designing and testing Autonomous Wheelchairs [1,2,3,4], there still exists some restrictions in the implementation of these technologies, which affect the performance of AWs and their autonomy.

The restrictions that affect the performance of AWs according to J. Carlos Garcia [5] can be summed up as:

- It's a cost sensitive application, because the economic impact on possible users must be within acceptable terms.
- The system response must be fast, an AW user hopes to move at walking person's speeds.

Received 16/06/07
Accepted 31/08/07
Final version 24/10/07

- The vehicle displacement environment is highly structured.
- The on-board and the surroundings infrastructure cost must be minimum.

And finally, in the case of severely disabled, the security fixes very high limits on the global system reliability.

At present many of the previously mentioned restrictions focus directly on finding the location of the mobile robot, since the effectiveness of the autonomous navigation system of AW is obtained through its reliability and robustness.

In the following sections we will approach a solution for the AW vision system based on the detection of artificial marks placed within the navigation surroundings through a camera mounted on the mobile robot in order to find its own position. Then some of preliminary results using algorithms for finding the location and segmentation marks placed within the surroundings to ease AWs position calculation will be presented.

2 Artificial Mark

The artificial mark used to find the position of the AW is shown in figure 1. It consists fundamentally of 3 elements that altogether produce a precise estimate of the present position of the mobile robot with respect to the position of the mark within the navigation environment map. Each one of these elements can be described according to their function as follows:

1. A frame of known dimensions used for discriminating elements of a captured image.
2. Four circles of equal diameter and well-known position used for calculating the relative position of the mobile robot.
3. One pattern for all the marks that will act as a second discriminator for the false marks that satisfies the first condition (Barker Code).

The integration of these three elements allows us to find the relative position of the robot. Initially with the image captured by the camera mounted on the AW, a scan will be made with the purpose of detecting the first discriminator of a valid mark (the frame of well-known dimensions). This discriminator will allow us to reject those elements that are not part of a valid mark, resulting in a smaller processing time for detecting the valid mark within the image thus reducing the computational cost.

In regard to the election of the format used for the design of the artificial mark, it was designed to be a low-cost, easy to implement and any user can be able to printing and placing this type of mark in the environment of its interest.

3 General Description of the Position System

The block diagram in figure 2 shows the hierarchical organization of the search algorithm. In this diagram, it is included the elements that allow the vision system to

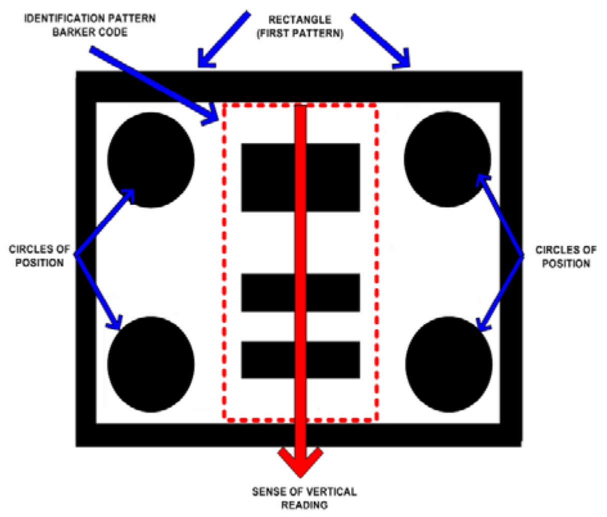


Fig. 1. Characteristics of the Artificial Mark.

obtain the relative position of the AW. The description of these elements is presented in later subsections:

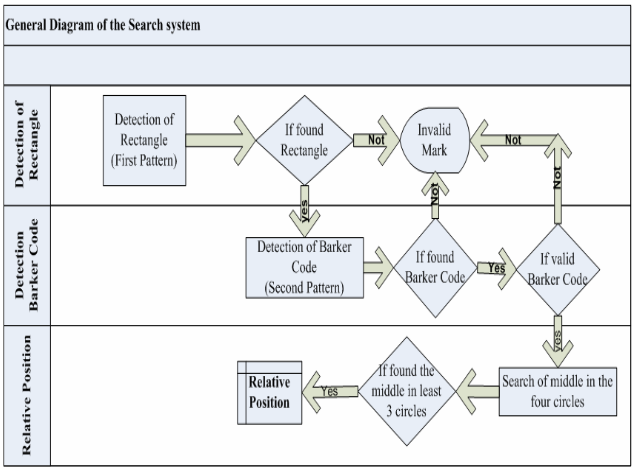

General Diagram of the Search system
Detection of Rectangle
Detection Barker Code
Relative Position
Detection of Rectangle (First Pattern)
If found Rectangle
Not
Invalid Mark
Not
Not
Yes
Not
Detection of Barker Code (Second Pattern)
If found Barker Code
Yes
If valid Barker Code
Yes
Search of middle in the four circles
If found the middle in least 3 circles
Yes
Relative Position

Fig. 2. General Diagram of the Search system.

3.1 Detection of Rectangle

In this stage the Search system scans through the captured image the first evidence (rectangle) that suggests the existence of a valid mark (see figure 3-left). In order to reduce the computational time of processing, the image (in binary scale) is subdivided into three parts for the detection of the rectangle.

3.2 Detection of Barker Code

The Barker code pattern chosen in this process is a Barker code with length of 7 bits. The structure is [1, 1, -1, -1, 1, -1, 1] which fulfills the property of autocorrelation that presents an accused peak when the sequence fits with a pattern. For simplicity, a method of direct codification of the image was adopted. Black stripe=1, white stripe = -1 (see figure 3- right).

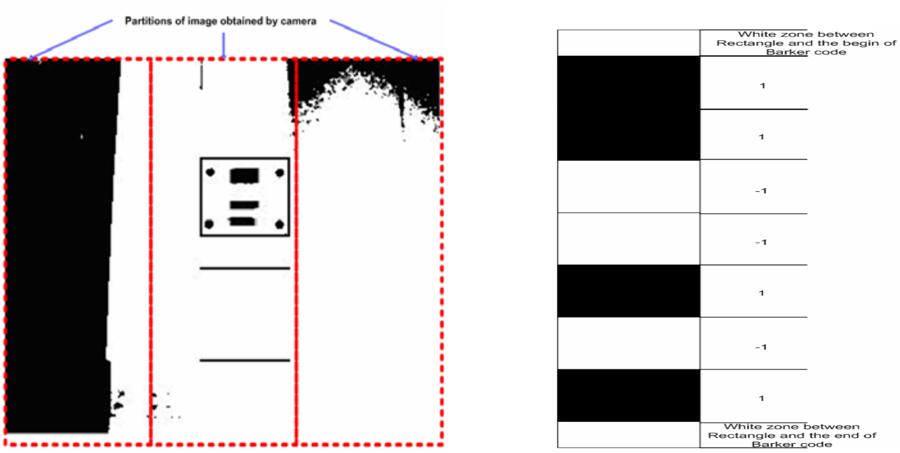

Partitions of image obtained by camera
White zone between Rectangle and the begin of Barker code
1
1
-1
-1
1
-1
1
White zone between Rectangle and the end of Barker code

Fig. 3. Left: Captured image from fixed camera. Right: Design of Barker code of 7 bits.

The detection algorithm consists of two main parts, the phase-search and the phase-confirmation.

In the phase-search, the entry of the algorithm is a column (gray levels) from the image to be processed. The scan starts from the initial X-Y coordinates of the found rectangle. Then, an important transition from white to black and vice versa is searched for each pair of columns of the image.

The procedure of search of abrupt changes in the levels of intensity of the image is implemented through a derivative filter of the form:

$$O(k) = I(k+1) - I(k-1) \qquad (1)$$

Where $I(k)$ is the nth row of entry of the scanned column and $O(k)$ is the nth row of the output image [5].

The output filtered image allows detecting remarkable changes in levels of intensity by localizing characteristic peaks of a Barker code that exceed a threshold given by:

$$O(p) = sg[O(k)] \bullet \left[\left(|O(k)| \phi U_p \right) \& \left(|O(k)| \geq |O(k-1)| \right) \& \left(|O(k)| \leq |O(k+1)| \right) \right] \qquad (2)$$

Where:

$$U_P = \sqrt{2 * \tfrac{1}{n} \sum_{1}^{N} O^2(n)}$$

The sign (*sg*) of peak in the intensity levels allows us to know which transition occurred (- 1 of white to black and 1 of black to white).

Once we have found the significant peaks in the levels of intensity of the image, the following step is to discard all those peaks that do not correspond in number and sequence of the transitions of the Barker code used. For it we used the knowledge of the correct sequence which must be $S(i) = [-1,+1,-1,+1,-1,+1]$ where $S(i)$ represents each possible group with six consecutive peaks.

If in the previous procedure has been detected a set of samples that contain a correct sequence in the transitions of a Barker code, then it is decoded as a possible valid Barker code. Later to reject the existence of a false code, the gray scale image is normalized to correlate it with a synthesized Barker code in the phase-confirmation. The normalized correlation between both elements is defined as:

$$C_n = \frac{\sum (I(x,y) - m_I)(h(x,y) - m_h)/N}{\sigma_I \sigma_h} \tag{3}$$

Where:

C_n =Result of normalized correlation.

$I(x,y)$ = Original image.

$h(x,y)$ = Synthesized Barker code pattern.

$m_I m_h$ = Image mean value of the original image and code pattern respectively.

$\sigma_I \sigma_h$ = Variation average of the pixels in relation to the image mean value.

N = Number of overlapped pixels between original image and code pattern.

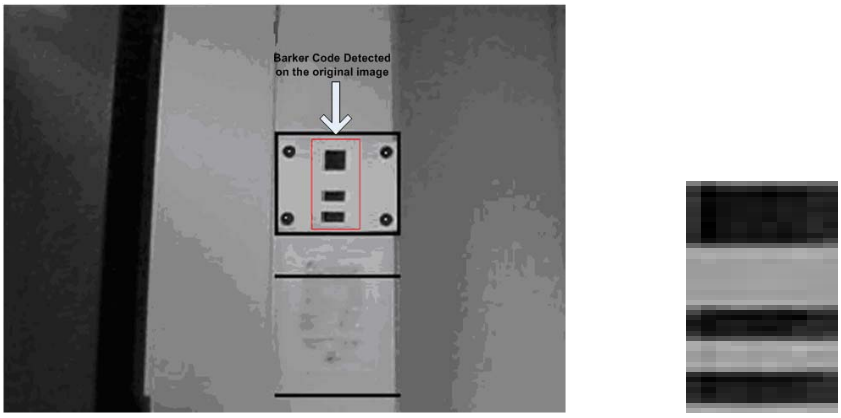


Fig 4. Left: Barker code detected on the image. Right: Synthesized Barker code.

If the value obtained by the normalized correlation process is $C = 1$ then exists a perfect correspondence between the image with the barker code detected (See figure 4-left) and the pattern compared (see figure 4-right).

In this experiment, it was considered as valid mark that one that exceeded a threshold of 0.72 in the correlation process. The algorithm used for the search of valid Barker code is based on the algorithms used in previous works [5]. The efficiency of the algorithm consists on carrying out the process of correlation not as a process of search of the barker code, but as a method of confirmation for those possible detected valid codes within the captured image.

3.3 Relative Position

This stage has been split in two main parts. Once that it has been detected and confirmed the existence of a code that indicates the presence of a valid mark, the following step is to find the center of at least three of the four circles placed in the corners of the artificial mark to find the relative position between camera and mark.

3.3.1 Segmentation of Circles and Determination of the Centroid Coordinates

The image is initially divided into two sub windows for the search of the circles: the upper sub window to cover the two high circles and the inferior sub window to cover the two low circles. The search of the circles is conducted using the a priori information (each circle was placed in each one of the corners of the rectangle in the artificial mark). We use this knowledge for saving computation time. For each of the four sub windows, the algorithm compute the vertical and horizontal diameter of each circle of the image using the same method of differentiation (Eq. 1) used in the detection of the Barker code. The scanning algorithm produce the segmented circle as is shown in figure 5. Once done this, the centroid coordinates of the each circle is computed by the following equations:

$$C_{XM} = C_{XYL} + \frac{C_{XYR} - C_{XYL}}{2} \tag{4}$$

$$C_{YM} = C_{XYT} + \frac{C_{XYD} - C_{XYT}}{2} \tag{5}$$

Where:
C_{XM} = Coordinate on X of circle centroid.
C_{YM} =Coordinate on Y of circle centroid.
C_{XYR} =Coordinate on axis XY of border right
C_{XYL} =Coordinate on axis XY of border left.
C_{XYT} =Coordinate on axis XY of border top.
C_{XYD} =Coordinate on axis XY of border down.

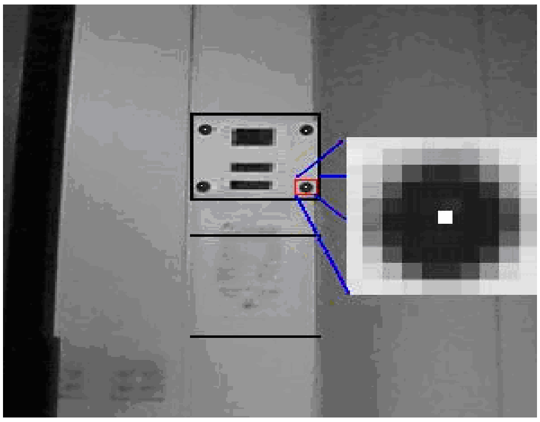


Fig. 5. Detection of circle centroid.

3.3.2 Algorithm for Position Computation

A basic scheme of the algorithm for the recovery of the position is show in the following figure:

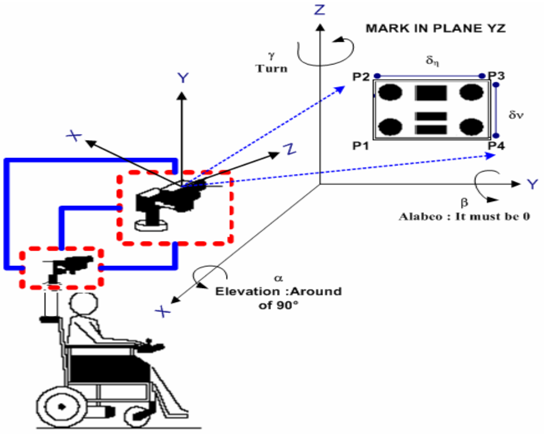


Fig. 6. Basic scheme of algorithm of search of relative position.

This algorithm is based fundamentally on the image shown in figure 6. From this we can recognize two triangular structures formed by two orthogonal oriented vectors referenced by the vector r_1 and r_3(see figure 7).

In both structures r_1 as r_3 are formed by the following sets of points and oriented segments:

$$r_1 \rightarrow [p_1, p_2, p_4, l_{12}, l_{14}] \quad (6)$$

$$r_3 \rightarrow [p_3, p_2, p_4, l_{32}, l_{34}] \quad (7)$$

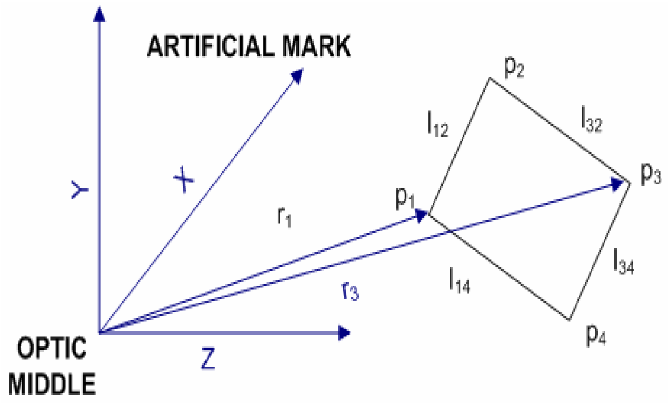


Fig. 7. Recovery of position.

Assuming that α and β it is well-known, and extracting the appropriate subgroup of the transformations of rotation and translation of the camera respect to the artificial mark we obtain [9]:

$$\begin{bmatrix} u_2 - u_1 \\ v_2 - v_1 \end{bmatrix} z_1 = \begin{bmatrix} -\delta_v u_2 \cos(\alpha) \\ \delta_v [\lambda \sin(\alpha) - v_2 \cos(\alpha)] \end{bmatrix}$$
$$\begin{bmatrix} u_4 - u_3 \\ v_4 - v_3 \end{bmatrix} z_3 = \begin{bmatrix} \delta_v u_4 \cos(\alpha) \\ -\delta_v [\lambda \sin(\alpha) - v_4 \cos(\alpha)] \end{bmatrix} \quad (8)$$

Where u_i and v_i are the coordinates horizontal and vertical of the projection of the centroids of pi on the plane image of the camera. λ is referred by the focal length; δ_v and δ_h are the lengths of the segments horizontal and vertical defined by the points π and α as the elevation angle of the camera.

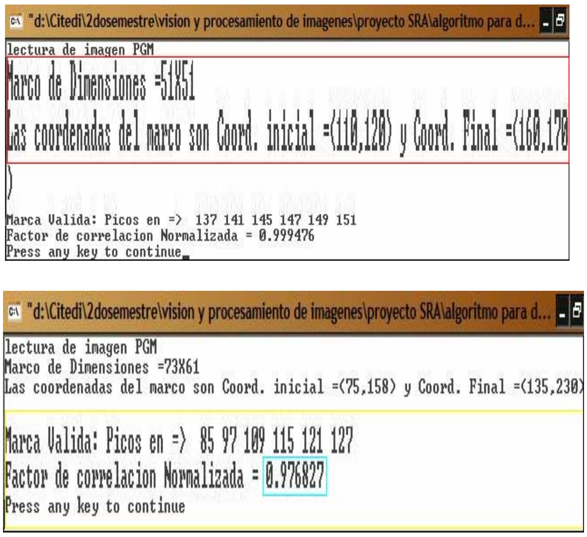
From the system preview (Eq. 8) are obtained the distances z_1 and z_3 to the points p_1 and p_3. Known these distances, the determination of the angle of turn is made by means of:

$$\begin{bmatrix} (u_4 - u_1)\frac{z_1}{\delta h} \\ (v_4 - v_1)\frac{z_1}{\delta_h} \\ (u_2 - u_3)\frac{z_3}{\delta_h} \\ (v_2 - v_3)\frac{z_3}{\delta_h} \end{bmatrix} = \begin{bmatrix} \lambda & u_4 \sin(\alpha) \\ 0 & [\lambda \cos(\alpha) + v_4 \sin(\alpha)] \\ -\lambda & -u_2 \sin(\alpha) \\ 0 & -[\lambda \cos(\alpha) + v_2 \sin(\alpha)] \end{bmatrix} \begin{bmatrix} Sin(\gamma) \\ Cos(\gamma) \end{bmatrix} \quad (9)$$

With this last equation we obtain the matrix of rotation $R = f(\alpha, \beta, \gamma)$. The recovery of the position-direction of the camera with respect to the position of the artificial mark is to carry out knowing the vectors r_1 and r_3 [5-8].

4 Results

The efficiency of the system was verified by a series of tests detecting diverse marks placed in strategic places of a room. Once located the rectangle that indicates the presence of a possible artificial mark, the next step was to confirm the validity of the mark, through the fast search algorithm of abrupt changes in the gray levels of image (Barker code). Such as it is indicated in the figure 8, the result shows a correlation factor of 0.98. It ratifies the legitimacy of the code found according to the criteria exposed in the previous sections. It's worth noting that of the number total of made tests were possible to detect and validate 60% of the placed marks.
The figure 9 shows the location of coordinates XY of the rectangle detected (see fig. 4 left) in the image obtained by the camera mounted over the AW in the first stage of system.


"d:\Citedi\2dosemestre\vision y procesamiento de imagenes\proyecto SRA\algoritmo para d...
"d:\Citedi\2dosemestre\vision y procesamiento de imagenes\proyecto SRA\algoritmo para d...

```
lectura de imagen PGM
Marco de Dimensiones =51X51
Las coordenadas del marco son Coord. inicial =(110,120) y Coord. Final =(160,170
)
Marca Valida: Picos en => 137 141 145 147 149 151
Factor de correlacion Normalizada = 0.999476
Press any key to continue_
```

```
lectura de imagen PGM
Marco de Dimensiones =73X61
Las coordenadas del marco son Coord. inicial =(75,158) y Coord. Final =(135,230)

Marca Valida: Picos en => 85 97 109 115 121 127
Factor de correlacion Normalizada = 0.976827
Press any key to continue
```

Fig. 8. Detection and validation of the artificial mark.

The figure 9 shows some preliminary experimental results of the localization estimation algorithm of the AW for distances between 0 to 3 m from the artificial mark. This figure shows that the algorithm with four landmark points defined and some previous knowledge about the α and β angles, the estimate position from the camera to landmark presents good results. Finally these experimental results have shown that the artificial landmark is suited for robot localization and n navigation.

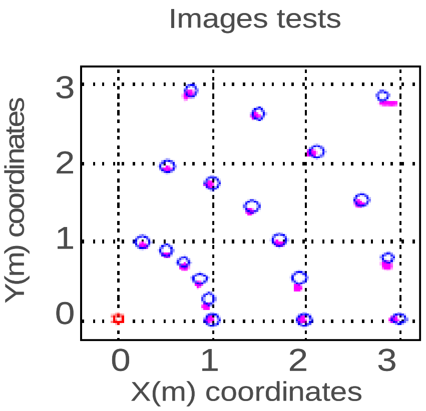


Fig. 9. Experimental results of the localization estimation algorithm of the AW.

Acknowledgements. The authors appreciate the support of SIP-IPN, for the support of this research project under grant 20071470 and 20070032.

References

1. Bernd Jähne, "Digital Image Processing" New York, Springer – Verlag, 1991
2. Kenneth R. Castleman, "Digital Image Processing", New Jersey, Prentice Hall, 1996
3. Rafael c. Gonzales, Richard E. Woods, "Digital Image Processing", New Jersey, Prentice Hall, 2002
4. M. Marron Romera, J. Carlos García, "Sistema de navegación autónoma en entornos interiores estructurados", Available: http://www.depeca.uah.es/personal/marta/TELEC-TFC.pdf
5. J. C. García García, M. Manzo, J. Ureña Ureña, M. Marron Romera, "Auto localización y posicionamiento mediante marcas artificiales codificadas",Available: http://www.depeca.uah.es/personal/jcarlos/SRAs/Tesis-pwd-JC-Sistema%20Posicionamiento.pdf
6. E. Santiso, M. Mazo, J. Ureña, J. A. Jiménez, J. C. Garcia, "Extracción de características para posicionamiento absoluto de robots móviles en interiores",Available: http://www.depeca.uah.es/personal/alvaro/parmei/docs_libres/PARMEI-trps-seguimiento_23Abril.pdf
7. V. Castello Martínez, "Localización y descodificación de códigos de barras en imágenes digitales",Available: http://www3.uji.es/~vtraver/e80/E80_Vicente_Castello.pdf
8. E. Aguirre, M Gómez, R. Muñoz, C. Ruiz, "Sistema Multi-agente que emplea visión active y ultrasonidos aplicados a navegación con comportamientosdifusos", Available: http://decsai.ugr.es/~salinas/publications/waf2003.pdf
9. R. M. Haralick and L. G. Shapiro, Computer and robot vision, vol II. Addison-Wesley Publishing company. 1993

Wavelet Domain RM L-Filter for Image Denoising

Jose Luis Varela-Benitez [1], Francisco Gallegos-Funes[1],
Volodymyr Ponomaryov[2], Oleksiy Pogrebnyak[3]

National Polytechnic Institute of Mexico
[1] Mechanical and Electrical Engineering Higher School, U.P. Zacatenco, fgallegosf@ipn.mx
[2] Mechanical and Electrical Engineering Higher School, U.P. Culhuacan,
vponomar@ipn.mx
[3] Center for Computing Research, U.P. Zacatenco,
olek@pollux.cic.ipn.mx

Abstract. In this paper we present the wavelet domain RM L-filter for the removal of impulsive and speckle noise in image processing applications. The proposed filter uses the robust RM-estimator in the filtering scheme of L-filter in the wavelet domain. Extensive simulation results have demonstrated that the proposed filter consistently outperforms the RM L-filter in the spatial domain by balancing the tradeoff between noise suppression and detail preservation.

1 Introduction

Noise suppression in digital images is a classical problem to the signal-processing community [1,2]. The corruption of images by noise is common during its acquisition or transmission. The aim of denoising is to remove the noise while keeping the signal features as much as possible. Traditional algorithms based on order statistics perform image denoising in the pixel domain [1-4].

In recent years, wavelet transform-based image denoising algorithms show a remarkable success [5-10]. For many image-processing applications, e.g. medical imagery, astronomical imagery, and remote-sensing imagery, the image denoising methods based on the overcomplete expansion of the wavelet transform show significant improvement in MSE than in the case of critically sampled wavelet transform [10].

In this paper we present the capability features of the wavelet domain RM L-filter for the removal of impulsive and speckle noise in image processing applications. The proposed filter uses the robust RM-estimator [11,12] with the Tukey biweight influence function [13] in the filtering scheme of *L*-filter [1] in the wavelet domain. Extensive simulation results have demonstrated that the proposed filter consistently outperforms the RM L-filter in spatial domain by balancing the tradeoff between noise suppression and detail preservation.

Received 23/06/07
Accepted 19/10/07
Final version 24/10/07

2 Proposed Wavelet Domain Filter

In recent works [11,12] we proposed the RM L-filters for image processing applications. We demonstrated that the RM L-filters consistently outperform other filters by balancing the tradeoff between noise suppression and detail preservation.

In this paper, we propose to use RM L-filter in the wavelet domain. Figure 1 shows a block diagram of proposed Wavelet domain Rank M-type L (WDRML) filter.

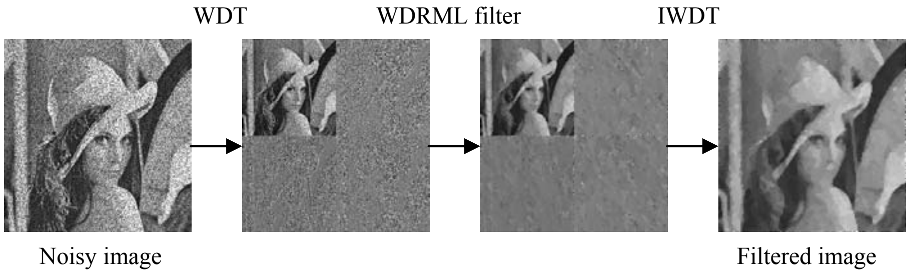


Fig. 1. Block diagram of proposed Wavelet domain Rank Median (WDRML) filter.

The WDRML filter is obtained by the combination of *L*-filter [1] and the RM-estimator [11,12] in the wavelet domain. The RM L-filter in the spatial domain can be writing as [11,12]

$$\theta_{RM\text{-}L} = \frac{\mathrm{MED}\{a_i \cdot [X_i \cdot \psi(X_i - \mathrm{MED}\{\vec{X}\})]\}}{a_{\mathrm{MED}}} \tag{1}$$

where $X_i \cdot \psi(X_i - MED\{\vec{X}\})$ are the selected pixels in accordance with the influence function in a sliding filter window, a_i are the weighted coefficients used into the proposed filter, and a_{MED} is the median of coefficients, and the influence function used here is the Tukey biweight [13],

$$\psi_{bi(r)}(X) = \begin{cases} X^2(r^2 - X^2), & |X| \le r \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The weighted coefficients of the RM L-filter were found using the uniform distribution function [1,2,13]. The coefficients are calculated by each sliding filter window due that the influence function selects whose pixels are used and then compute the weighted coefficients according with the number of pixels used into the filtering window. The RM L-filter is used in the wavelet domain by means of use the Daubechie wavelets [5,6]. Finally, we apply the proposed filter in the images of approaches and details obtained in the process of wavelet decomposition.

3 Results

We obtained from the simulation experiments the properties of proposed Wavelet Domain Rank M-type L (WDRML) filter and we compared it with its version in the spatial domain (RM L filter).

The criteria used to compare the performance of filters were the peak signal-to-noise ratio (PSNR) to evaluate the performance of noise suppression [1,2],

$$\mathrm{PSNR} = 10 \cdot \log\left[\frac{(255)^2}{\mathrm{MSE}}\right], \mathrm{dB} \tag{3}$$

and the mean absolute error (MAE) for evaluation of fine detail preservation [1,2],

$$\mathrm{MAE} = \frac{1}{M_0 N_0}\sum_{i=0}^{M_0\text{-}1}\sum_{j=0}^{N_0-1}\left|e(i,j) - \hat{e}(i,j)\right| \tag{4}$$

where $\mathrm{MSE} = \frac{1}{M_0 N_0}\sum_{i=0}^{M_0-1}\sum_{j=0}^{N_0-1}\left[e(i,j) - \hat{e}(i,j)\right]^2$ is the *mean square error*, $e(i,j)$ is the original image, $\hat{e}(i,j)$ is the restored image, and $M_0 \mathrm{x} N_0$ is the image size. In our experiments a 3x3 filter window is applied.

To determine the noise suppression properties of proposed filter the 256x256 standard test grayscale image “Lena” was corrupted by speckle noise. Table 1 shows the performance results in terms of PSNR in dB and MAE for the image “Lena” degraded with 0.1 of variance of speckle noise and free of noise by use the proposed filter in approaches (A) and details (D) with the wavelets db1, db2, db3, and db4 with one (1) and two (2) levels of decomposition. From this table one can see that the proposed WDRML filter provides better speckle noise suppression and detail preservation in comparison with the RM L-filter in the spatial domain in the most of cases.

Table 2 presents the performance results for the image “Mandrill” is degraded with 0.05 of variance of speckle noise and when is free of noise. This Table shows that the proposed WDRML filter provides better results in comparison with the RM L-filter in the spatial domain.

In Table 3 we show the performance results in the case of 5% of impulsive noise in the images “Lena” and “Peppers”. From this Table one can see that the proposed filter has poor performance in comparison with the RM L-filter.

Table 1. Performance results in the image "Lena" obtained by the use of proposed filter.

Filters	Free noise		σ^2=0.1	
	PSNR	MAE	PSNR	MAE
RM L	29.62	3.78	22.95	13.24
WDRML (db1,A,1)	27.84	5.09	23.35	12.24
WDRML (db1,D,1)	31.46	3.24	20.53	18.78
WDRML (db2,A,1)	27.90	5.11	23.60	12.15
WDRML (db2,D,1)	32.26	3.05	20.69	18.00
WDRML (db3,A,1)	27.92	5.24	24.02	11.77
WDRML (db3,D,1)	32.70	2.97	20.79	17.99
WDRML (db4,A,1)	27.87	5.27	24.33	11.28
WDRML (db4,D,1)	33.00	2.92	20.90	18.11
WDRML (db1,A,2)	24.83	8.32	22.46	13.57
WDRML (db1,D,2)	27.48	5.73	22.66	13.93
WDRML (db2,A,2)	25.40	7.61	22.94	12.85
WDRML (db2,D,2)	28.37	5.34	23.21	13.15
WDRML (db3,A,2)	25.24	7.89	23.14	12.59
WDRML (db3,D,2)	28.39	5.42	23.49	12.90
WDRML (db4,A,2)	25.06	8.21	23.38	12.47
WDRML (db4,D,2)	28.29	5.46	23.69	12.73

Table 2. Performance results in the image "Mandrill" obtained by the use of proposed filter.

Filters	Free noise		σ^2=0.05	
	PSNR	MAE	PSNR	MAE
RM L	23.22	10.38	21.31	15.42
WDRML (db1,A,1)	21.30	15.49	20.75	17.58
WDRML (db1,D,1)	23.63	11.49	21.60	16.24
WDRML (db2,A,1)	21.34	15.52	20.75	17.55
WDRML (db2,D,1)	24.56	10.36	22.15	15.17
WDRML (db3,A,1)	21.33	15.53	20.77	17.52
WDRML (db3,D,1)	25.53	9.31	22.66	14.37
WDRML (db4,A,1)	21.25	15.70	20.77	17.52
WDRML (db4,D,1)	26.06	8.81	22.96	13.89
WDRML (db1,A,2)	19.98	19.01	19.69	20.37
WDRML (db1,D,2)	21.69	14.69	21.11	16.78
WDRML (db2,A,2)	19.92	19.50	19.62	20.58
WDRML (db2,D,2)	21.79	15.01	21.20	16.77
WDRML (db3,A,2)	20.00	19.26	19.71	20.30
WDRML (db3,D,2)	21.94	14.74	21.31	16.59
WDRML (db4,A,2)	19.96	19.40	19.66	20.49
WDRML (db4,D,2)	22.28	14.06	21.64	15.84

From the experimental results presented in this paper we can say that the proposed WDRML filter can be suppress the speckle noise effectively but in the case of impulsive noise it has poor performance.

One can improve the performance of proposed filter for speckle noise suppression if we use up to five levels of decomposition in the wavelet analysis. In the case of impulsive noise we can improve the noise suppression by use an impulsive detector [11,12]

Table 3. Performance results in the images "Lena" and "Peppers" obtained by the use of proposed filter.

Filters	"Lena"		"Peppers"	
	PSNR	MAE	PSNR	MAE
RM L	28.29	5.76	29.03	5.21
WDRML (db1,A,1)	25.95	7.20	25.06	8.02
WDRML (db1,D,1)	23.28	10.37	23.18	10.12
WDRML (db2,A,1)	25.89	8.01	25.34	8.46
WDRML (db2,D,1)	23.57	10.74	23.43	10.63
WDRML (db3,A,1)	25.84	8.33	24.83	8.97
WDRML (db3,D,1)	23.60	11.11	23.62	10.81
WDRML (db4,A,1)	25.88	8.25	24.17	9.18
WDRML (db4,D,1)	23.65	11.06	23.78	10.61
WDRML (db1,A,2)	23.89	10.47	21.69	13.29
WDRML (db1,D,2)	24.81	9.74	23.59	10.62
WDRML (db2,A,2)	24.12	10.15	21.75	13.43
WDRML (db2,D,2)	25.45	9.38	24.51	10.21
WDRML (db3,A,2)	24.00	10.52	21.58	13.84
WDRML (db3,D,2)	25.61	9.37	25.02	10.03
WDRML (db4,A,2)	24.15	10.64	21.42	14.12
WDRML (db4,D,2)	25.58	9.51	24.71	10.34

Figure 2 presents the visual results to apply the proposed filter with one and two decomposition levels in the image "Lena", and in Figure 3 shows the processed images in the case of use the image "Peppers".

From Figures 2 and 3, one can see that the proposed filter outperforms the RM L-filter in the case of speckle noise but in the case of impulsive noise has poor performance. In the case when the images are free of noise the proposed filter can conserve the properties of detail preservation better in comparison with the RM L-filter.

4 Conclusions

We adapt the RM L-filter to work in the wavelet domain. The proposed WDRML filter has better properties of speckle noise suppression and detail preservation in comparison with the RM L-filter. It is expected that the performance of noise suppression and detail preservation can be increased if we use up to five decomposition levels in the proposed filter. Therefore we will use other influences functions and distribution function in the filtering scheme of proposed filter.

Acknowledgements. The authors thank the National Polytechnic Institute of Mexico for its support.

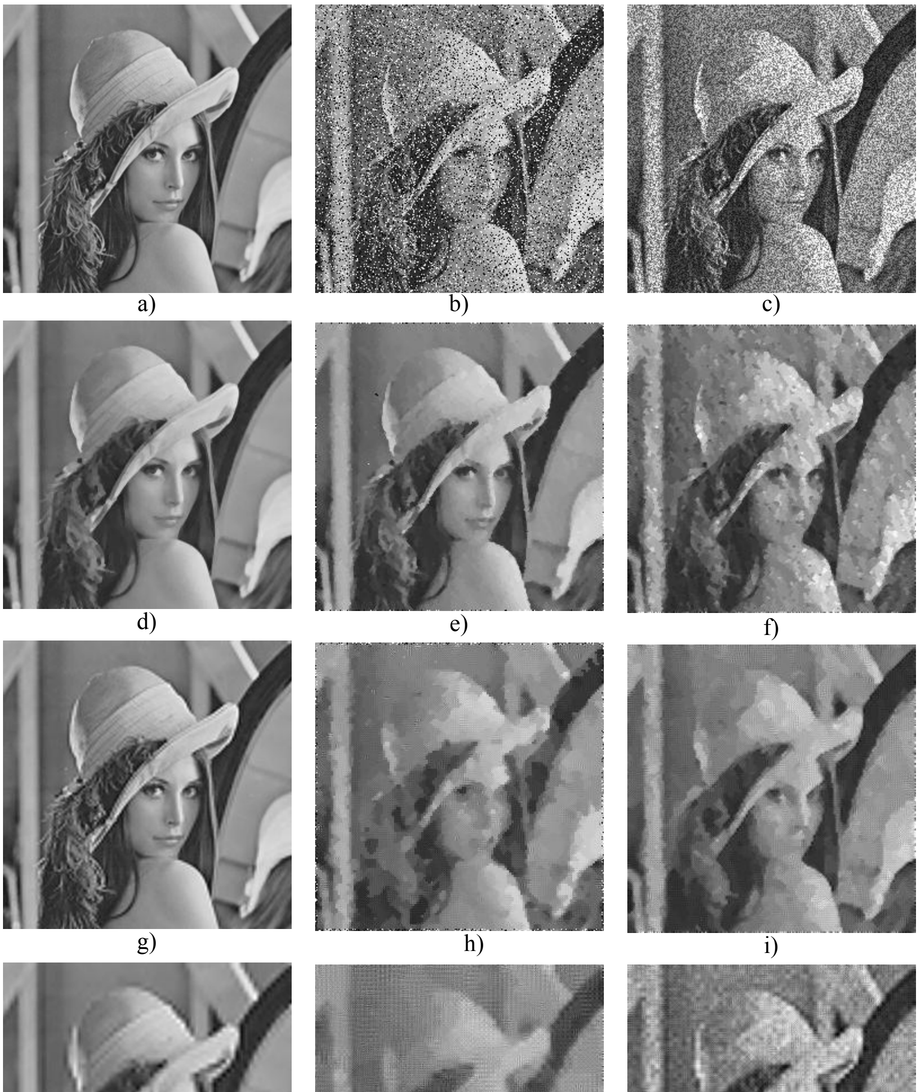


Fig. 2. Visual results in the image Lena, a) Original image, b) Degraded image with 20% of impulsive noise, c) Degraded image with 0.1 of variance of speckle noise, d) Restored image of a) with RM L-filter, e) Restored image of b) with RM L-filter, f) Restored image of c) with RM L-filter, g) Restored image of a) with WDRML filter (db4, D,1), h) Restored image of b) with WDRML filter (db1, A, 1), i) Restored image of c) with WDRML filter (db1, A, 1). j) Restored image of a) with WDRML filter (db3, D,2), k) Restored image of b) with WDRML filter (db3, A, 2), l) Restored image of c) with WDRML filter (db4, A, 2).

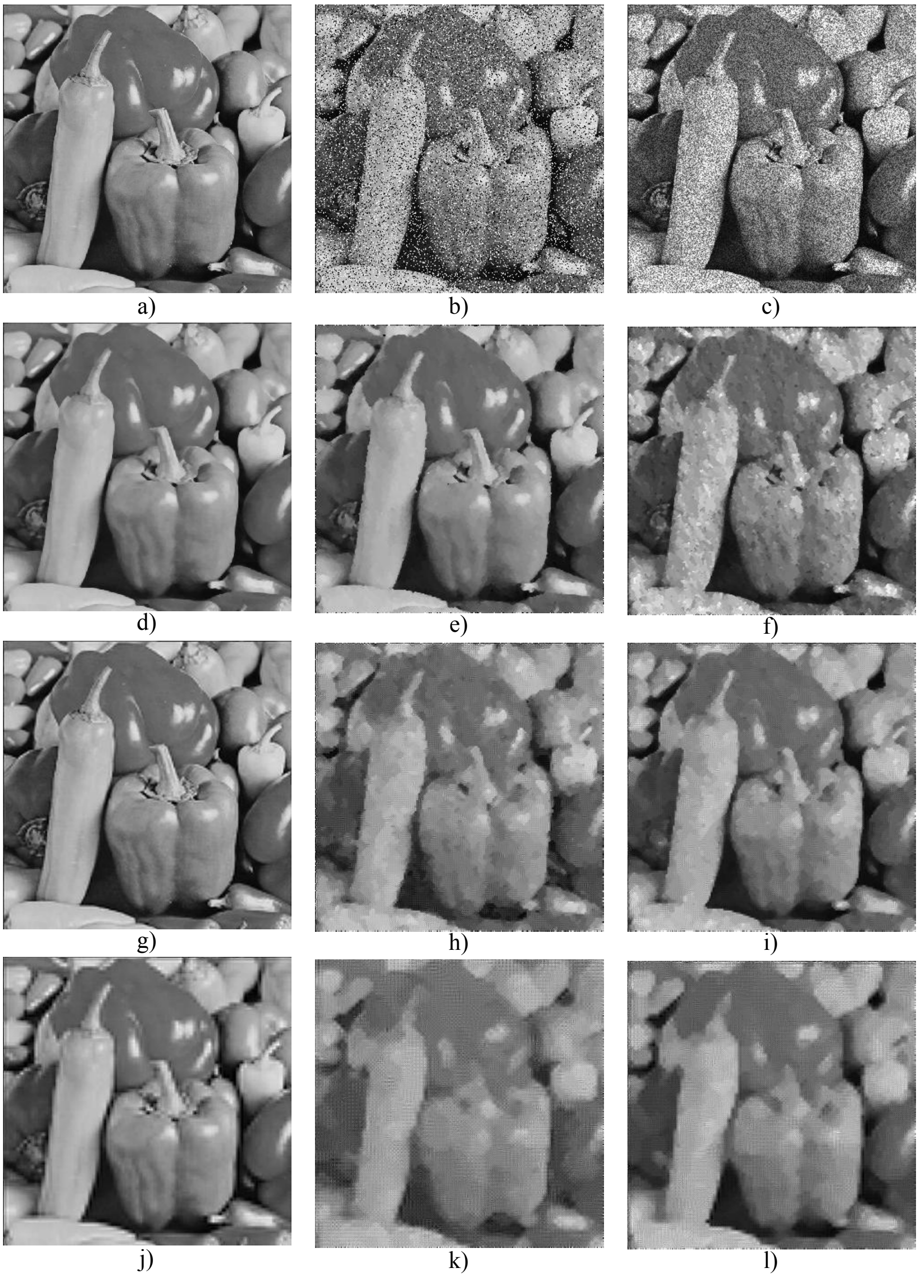


Fig. 3. Visual results in the image Peppers, a) Original image, b) Degraded image with 20% of impulsive noise, c) Degraded image with 0.1 of variance of speckle noise, d) Restored image of a) with RM L-filter, e) Restored image of b) with RM L-filter, f) Restored image of c) with RM L-filter, g) Restored image of a) with WDRML filter (db4, D,1), h) Restored image of b) with WDRML filter (db2, A, 1), i) Restored image of c) with WDRML filter (db2, A, 1), j) Restored image of a) with WDRML filter (db4, D, 2), k) Restored image of b) with WDRML filter (db2, A, 2), l) Restored image of c) with WDRML filter (db2, A, 2).

References

1. Pitas, I., Venetsanopoulos, A. N.: Nonlinear Digital Filters. Kluwer Academic Publishers, Boston. (1990)
2. Astola, J., Kuosmanen, P.: Fundamentals of Nonlinear Digital Filtering. CRC Press, Boca Raton-New York. (1997)
3. Peltonen, S., Kuosmanen, P.: Robustness of nonlinear filters for image processing. Journal Electronic Imaging. 10(3) (2001) 744-756
4. Öten, R., De Figueiredo, R. J. P.: Sampled-Function Weighted Order Filters. IEEE Trans. Circuits and Systems-II: Analog and Digital Processing. 49(1) (2002) 1-10
5. Coifman, R.R., Donoho, D.L.: Translation-invariant de-noising, in: A. Antoniadis, G. Oppenheim (Eds.), Wavelets and Statistics, Springer, Berlin, Germany. (1995)
6. Walker, J. S.: A Primer on Wavelets and their Scientific Applications. Chapman & Hall/CRC. (1999)
7. Chang, S.G., Yu, B., Vettereli, M.: Spatially adaptive wavelet thresholding with context modeling for image denoising. IEEE Trans. Image Process. 9(9) (2000) 1522–1531
8. Chang, S.G., Yu, B., Vettereli, M.: Adaptive wavelet thresholding for image denoising and compression. IEEE Trans. Image Process. 9(9) (2000) 1532–1546
9. Cai, Z., Cheng, T.H., Lu, C., Subramanium, K.R.: Efficient wavelet-based image denoising algorithm. Electron. Lett. 37(11) (2001) 683–685
10. Mahbubur Rahman, S. M., Kamrul Hasan, Md.: Wavelet-domain iterative center weighted median filter for image denoising. Signal Processing. 83 (2003) 1001-1012
11. Gallegos-Funes, F. J., Varela-Benitez, J. L., Ponomaryov, V. I.: Real-time image processing based on robust linear combinations of order statistics. Proc. SPIE 6063, Real-Time Image Processing 2006. San Jose, USA. (2006) 177-187
12. Varela-Benitez, J. L., Gallegos-Funes, F. J., Ponomaryov, V. I.: RM L-filters for Real Time Imaging. Proc. IEEE 15th International Conference on Computing, CIC 2006, Mexico City, Mexico. (2006) 43-48
13. Hampel, F. R., Ronchetti, E. M., Rouseew, P. J., Stahel, W. A.: Robust Statistics. The approach based on influence function. Wiley, New York. (1986)

Representing and Visualizing Vectorized Videos through the Extreme Vertices Model in the n-Dimensional Space (nD-EVM)

Ricardo Pérez-Aguila

Universidad Tecnológica de la Mixteca
Carretera Huajuapan-Acatlima Km. 2.5.
Huajuapan de León, Oaxaca 69000, México
ricardo.perez.aguila@gmail.com

Abstract. Several video compression methods were invented to be able to effectively store video data on common digital media. One method of compression we will explore in this work is oriented to vectorized video sequences. Each frame in a color video is initially represented as a set of orthogonal polygons whose displaying time depends on the temporal dimension. Moreover, one spatial dimension will be assigned to the color to apply to such polygons. Hence, a vectorized 2D color video sequence can be expressed as a Four-Dimensional Orthogonal Pseudo-Polytope which will be represented under the Extreme Vertices Model in the n-Dimensional Space (nD-EVM). The nD-EVM shares the representation of n-Dimensional Orthogonal Pseudo-Polytopes (nD-OPP's) by considering only a subset of their vertices: the Extreme Vertices. This work will describe how the source sequences can be converted in a vectorized video and then compressed, expressed, manipulated, and displayed in screen through the 4D-EVM. The results obtained from the conversion of two video sequences motivate some observations and properties of the proposed methodology.

Keywords: n-Dimensional Orthogonal Polytopes Modeling, Geometrical and Topological Representations, Color 2D-Videos Compression, Computational Geometry.

1 Introduction and Problem Statement

The Extreme Vertices Model (3D-EVM) was originally presented, and widely described in [1], for modeling 2-manifold Orthogonal Polyhedra and later considering both Orthogonal Polyhedra (3D-OP's) and Pseudo-Polyhedra (3D-OPP's) [2]. This model has enabled the development of simple and robust algorithms for performing the most usual and demanding tasks on solid modeling, such as closed and regularized Boolean operations, solid splitting, set membership classification ope-rations and measure operations on 3D-OPP's. It is natural to ask if the EVM can be extended for modeling n-Dimensional Orthogonal Pseudo-Polytopes (nD-OPPs). In this sense, some experiments were made, in [8], where the validity of the model was assumed true in order to represent 4D and

Received 07/09/07
Accepted 19/10/07
Final version 24/10/07

5D-OPPs. Finally, in [9] was formally proved that the nD-EVM is a complete scheme for the repre-sentation of nD-OPPs. The meaning of complete scheme was based in Requicha's set of formal crite-rions that every scheme must have rigorously defined: Domain, Completeness, Uniqueness and Vali-dity. Although the EVM of an nD-OPP has been defined as a subset of the nD-OPP’s vertices, there is much more information about the polytope hidden within this subset of vertices. In **Sections 2.5** and **2.6** we will describe basic procedures and algorithms in order to obtain some of this information.

It is well known that classical storage techniques like celluloid films or analogue video tapes carry various mechanical and physical degradations that significantly reduce their visual quality along time [4]. The sampling of analog signals and expressing them in digital form is used today to guarantee the quality of the information and make them media independent. As pointed out in [4], in order to achieve good fidelity it is often necessary to produce a large amount of data. Cartoon animations usually provide source video sequences to be vectorized. For example, Koloros & Zára [4] separate in first place the original animation frame into a set of regions using unsupervised image segmentation techniques. Then they use motion estimation in order to register parts of the background to stitch and store background layer as one big image. Shapes of homogeneous color regions in the foreground layer are converted from raster to vector representation and encoded separately. To search for frame duplicities and to store new frames they use a pool of already stored frames. During the playback standard graphics hardware is used to render the background layer as a textured rectangle and in front of it foreground layer as a set of flat colored polygons [4]. Another example of vectorization of cartoon animations is given by the work of Kwatra and Rossignac [6]. In their approach each region is first represented as a 3D volume by sweeping its 2D shape through the time. Then their Edgebreaker compression scheme is used to encode volume geometry. However, these authors ([4] & [6]) did not address the problem of vectorization for complex color and gray scale image sequences.

In this work, each frame in a color video is initially represented as a set of orthogonal polygons whose displaying time depends on the temporal dimension. Moreover, we will assign one spatial dimension to the color to apply to such polygons. Hence, we will express a vectorized 2D color video sequence as a Four-Dimensional Orthogonal Pseudo Polytope (4D-OPP) which will be represented under the 4D-EVM. In **Section 3** we will describe how the source sequences can be converted in a vectorized video and then compressed, manipulated, and displayed in screen through the 4D-EVM.

2 The Extreme Vertices Model in the n-Dimensional Space (nD-EVM)

2.1 Preliminary Background: n-Dimensional Orthogonal Pseudo-Polytopes

Definition 2.1: *A <u>Singular n-Dimensional Hyper-Box</u> in ¡^{n} is the continuous function*

$$\begin{array}{llll} I^n: & [0,1]^n & \rightarrow & [0,1]^n \\ & x & : & I^n(x) = x \end{array}$$

For a general singular kD hyper-box c we will define the boundary of c.

Definition 2.2: *For all i, 1 ≤ i ≤ n, the two singular (n-1)D hyper-boxes* $\underline{I^n_{(i,0)}}$ *and* $\underline{I^n_{(i,1)}}$ *are defined as follows: If* $x \in [0,1]^{n-1}$ *then* $I^n_{(i,0)}(x) = I^n(x_1,...,x_{i-1},0,x_i,...,x_{n-1}) = (x_1,...,x_{i-1},0,x_i,...,x_{n-1})$ and $I^n_{(i,1)}(x) = I^n(x_1,...,x_{i-1},1,x_i,...,x_{n-1}) = (x_1,...,x_{i-1},1,x_i,...,x_{n-1})$

Definition 2.3: *In a general singular nD hyper-box c we define the* <u>*(i,α)-cell*</u> *as* $c_{(i,\alpha)} = c \circ I^n_{(i,\alpha)}$

The next definitions indicate in precise way what we consider as the orientation of a (n-1)D cell.

Definition 2.4: *The* <u>*orientation*</u> *of an (n-1)D cell* $c \circ I^n_{(i,\alpha)}$ *is given by* $(-1)^{\alpha+i}$.

Definition 2.5: *An* <u>*(n-1)D oriented cell*</u> *is given by the scalar-function product* $(-1)^{i+\alpha} \cdot c \circ I^n_{(i,\alpha)}$

Definition 2.6: *A formal linear combination of singular general kD hyper-boxes, 1 ≤ k ≤ n, for a closed set A is called a* <u>*k-chain*</u>.

Definition 2.7 [11]: *Given a singular nD hyper-box* I^n *we define the (n-1)-chain, called the* <u>*boundary of* I^n</u>, *by* $\partial(I^n) = \sum_{i=1}^{n} \left(\sum_{\alpha=0,1} (-1)^{i+\alpha} \cdot I^n_{(i,\alpha)} \right)$

Definition 2.8 [11]: *Given a singular general nD hyper-box c we define the (n-1)-chain, called the* <u>*boundary of c*</u>, *by* $\partial(c) = \sum_{i=1}^{n} \left(\sum_{\alpha=0,1} (-1)^{i+\alpha} \cdot c \circ I^n_{(i,\alpha)} \right)$

Definition 2.9 [11]: *The* <u>*boundary of an n-chain*</u> $\sum c_i$, *where each* c_i *is a singular general nD hyper-box, is given by* $\partial\left(\sum c_i\right) = \sum \partial(c_i)$

Definition 2.10: *A collection* $c_1, c_2, ..., c_k$, $1 \le k \le 2^n$, *of general singular nD hyper-boxes is a* <u>*combination of nD hyper-boxes*</u> *if and only if*

$$\left[\bigcap_{\alpha=1}^{k} c_\alpha([0,1]^n) = (\underbrace{0,...,0}_{n}) \right] \wedge \left[\left(\forall i,j,\ i \ne j,\ 1 \le i,j \le k \right) \left(c_i([0,1]^n) \ne c_j([0,1]^n) \right) \right]$$

In the above definition the first part of the conjunction establishes that the intersection between all the nD general singular hyper-boxes is the origin, while the second part establishes that there are not overlapping nD hyper-boxes.

Definition 2.11: *We say that an* <u>*n-Dimensional Orthogonal Pseudo-Polytope*</u> *p, or just an* <u>*nD-OPP*</u> *p, will be an n-chain composed by nD hyper-boxes arranged in such way that by selecting a vertex, in any of these hyper-boxes, we have that such vertex describes a combination of nD hyper-boxes (**Definition 2.10**) composed up to* 2^n *hyper-boxes.*

Describing nD-OPP's as union of disjoint nD hyper-boxes in such way that by selecting a vertex, in any of these hyper-boxes, we have that such vertex is surrounded up to 2^n hyper-boxes, will be very useful because in the following propositions we consider geometrical and/or topological local analysis over such vertices and their respective incident hyper-boxes.

2.2 The nD-EVM: Foundations

Definition 2.12: *Let c be a combination of hyper-boxes in the n-Dimensional space. An Odd Edge will be an edge with an odd number of incident hyper-boxes of c.*

Definition 2.13: *A brink or extended edge is the maximal uninterrupted segment, built out of a sequence of collinear and contiguous* ***odd edges*** *of an nD-OPP.*

Definition 2.14: *We will call Extreme Vertices of an nD-OPP p to the ending vertices of all the brinks in p. EV(p) will denote to the set of Extreme Vertices of p.*

The brinks in an nD-OPP p can be classified according to the main axis to which they are parallel. Since the extreme vertices mark the end of brinks in the n orthogonal directions, is that any of the n possible sets of brinks parallel to X_i-axis, 1 ≤ i ≤ n, produce to the same set EV(p).

Definition 2.15: *Let p be an nD-OPP. $EV_i(p)$ will denote to the set of ending or extreme vertices of the brinks of p which are parallel to X_i-axis, $1 \leq i \leq n$.*

Theorem 2.1 [9]: *A vertex of an nD-OPP p, n≥1, when is locally described by a set of surrounding nD hyper-boxes, is an extreme vertex if and only if it is surrounded by an odd number of such nD hyper-boxes.*

Definition 2.16: *Let p be a nD-OPP. A kD couplet of p, 1<k<n, is the maximal set of kD cells of p that lies in a kD space, such that a kD cell e_0 belongs to a kD extended hypervolume if and only if e_0 belongs to a (n-1)D cell present in $\partial(p)$.*

Let Q be a finite set of points in ¡ ^{3}. In [2] was defined the ABC-sorted set of Q as the set resulting from sorting Q according to coordinate A, then to coordinate B, and then to coordinate C. For instance, a set Q can be ABC-sorted is six different ways: $X_1X_2X_3$, $X_1X_3X_2$, $X_2X_1X_3$, $X_2X_3X_1$, $X_3X_1X_2$ and $X_3X_2X_1$. Now, let p be a 3D-OPP. According to [2] the Extreme Vertices Model of p, EVM(p), denotes to the ABC-sorted set of the extreme vertices of p. Then EVM(p) = EV(p) except by the fact that EV(p) is not necessarily sorted. In this work we will assume that the coordinates of extreme vertices in the Extreme Vertices Model of an nD-OPP p, $EVM_n(p)$ are sorted according to coordinate X_1, then to coordinate X_2, and so on until coordinate X_n. That is, we are considering the only ordering $X_1 \ldots X_i \ldots X_n$ such that i–1 < i, 1 < i ≤ n.

Definition 2.17: *Let p be an nD-OPP. We will define the Extreme Vertices Model of p, denoted by $EVM_n(p)$, as the model as only stores to all the extreme vertices of p.*

2.3 Sections and Slices of nD-OPP's

Definition 2.18: *We define the Projection Operator for (n-1)D cells, points, and set of points respectively as follows:*

- *Let $c(I^n_{(i,\alpha)}(x)) = (x_1,..,x_n)$ be an (n-1)D cell embedded in the nD space. $\underline{\pi_j\left(c(I^n_{(i,\alpha)}(x))\right)}$ will denote the projection of the cell $c(I^n_{(i,\alpha)}(x))$ onto an (n-1)D space embedded in nD space whose supporting hyperplane is perpendicular to X_j-axis: $\pi_j\left(c(I^n_{(i,\alpha)}(x))\right) = (x_1,\ldots,\hat{x}_j,\ldots,x_n)$*

- *Let* $v=(x_1,...,x_n)$ *be a point in* ¡ ^{n}. *The projection of that point in the (n-1)D space, denoted by* $\underline{\pi_j(v)}$, *is given by:* $\pi_j(v)=(x_1,...,\hat{x}_j,...,x_n)$
- *Let Q be a set of points in* ¡ ^{n}. *We define the projection of the points in Q, denoted by* $\underline{\pi_j(Q)}$, *as the set of points in* ¡ $^{n-1}$ *such that* $\pi_j(Q)=\{p\in ¡^{n-1}: p=\pi_j(x),\ x\in Q\subset ¡^n\}$

In all the cases $\hat{x}_j$ *is the coordinate corresponding to* X_j*-axis to be suppressed.*

Definition 2.19: *Consider an nD-OPP p:*

- *Let* $\underline{np_i}$ *be the number of distinct coordinates present in the vertices of p along* X_i*-axis,* $1\le i\le n$.
- *Let* $\underline{\Phi^i_k(p)}$ *be the k-th (n-1)D couplet of p which is perpendicular to* X_i*-axis,* $1\le k\le np_i$.

Definition 2.20: *A* Section *is the (n-1)D-OPP, n > 1, resulting from the intersection between an nD-OPP p and a (n-1)D hyperplane perpendicular to the* X_i*-axis,* $1\le i\le n$, *which not coincide with any (n-1)D-couplet of p. A section will be called* external *or* internal *section of p if it is empty or not, respectively.* $\underline{S^i_k(p)}$ *will refer to the k-th section of p between* $\Phi^i_k(p)$ *and* $\Phi^i_{k+1}(p)$, $1\le k<np_i$.

2.4 Computing Couplets and Sections

Theorem 2.2 [9]: *The projection of the set of (n-1)D-couplets,* $\pi_i\left(\Phi^i_k(P)\right)$, *of an nD-OPP P, can be obtained by computing the regularized XOR (⊗) between the projections of its previous* $\pi_i\left(S^i_{k-1}(P)\right)$ *and next* $\pi_i\left(S^i_k(P)\right)$ *sections, i.e.,* $\pi_i\left(\Phi^i_k(P)\right)=\pi_i\left(S^i_{k-1}(P)\right)\ \otimes^*\ \pi_i\left(S^i_k(P)\right),\ \forall k\in[1,np_i]$

Theorem 2.3 [9]: *The projection of any section,* $\pi_i\left(S^i_k(p)\right)$, *of an nD-OPP p, can be obtained by computing the regularized XOR between the projection of its previous section,* $\pi_i\left(S^i_{k-1}(p)\right)$, *and the projection of its previous couplet* $\pi_i\left(\Phi^i_k(p)\right)$.

2.5 The Regularized XOR operation on the nD-EVM

Theorem 2.4 [2]: *Let p and q be two nD-OPP's having* $EVM_n(p)$ *and* $EVM_n(q)$ *as their respective EVM's in nD space, then* $EVM_n(p\otimes^* q)=EVM_n(p)\ \otimes\ EVM_n(q)$.

This result allows expressing a formula for computing nD-OPP's sections from couplets and vice-versa, by means of their corresponding Extreme Vertices Models. These formulae are obtained by combining **Theorem 2.4** with **Theorem 2.2**; and **Theorem 2.4** with **Theorem 2.3**, respectively:

Corollary 2.1 [2]: $EVM_{n-1}\left(\pi_i(\Phi^i_k(p))\right)=EVM_{n-1}\left(\pi_i(S^i_{k-1}(p))\right)\otimes EVM_{n-1}\left(\pi_i(S^i_k(p))\right)$

Corollary 2.2 [2]: $EVM_{n-1}\left(\pi_i(S^i_k(p))\right)=EVM_{n-1}\left(\pi_i(S^i_{k-1}(p))\right)\otimes EVM_{n-1}\left(\pi_i(\Phi^i_k(p))\right)$

Finally, the following corollary can be stated, which correspond to a specific situation of the XOR operands. It allows computing the union of two nD-OPP's when that specific situation is met.

Corollary 2.3 [2]: *Let p and q be two disjoint or quasi disjoint nD-OPP's having $EVM_n(p)$ and $EVM_n(q)$ as their respective Extreme Vertices Models, then* $EVM_n(p \cup q) = EVM_n(p) \otimes EVM_n(q)$.

2.6 Basic Algorithms for the nD-EVM

According to **Sections 2.2** to **2.4** we can define the following primitive operations which are based in the functions originally presented in [2]:

```
Output: An empty nD-EVM.
Procedure InitEVM( )
{ Returns the empty set.                 }
```

```
Input:  An nD-EVM p
Output: A Boolean.
Procedure EndEVM(EVM p)
{ Returns true if the end of p along
  $X_1$-axis has been reached.             }
```

```
Input:  An nD-EVM p
Output: An (n-1)D-EVM embedded in
(n-1)D space.
Procedure ReadHvl(EVM p)
{ Extracts next (n-1)D couplet
  perpendicular to $X_1$-axis from p.     }
```

```
Input:  An (n-1)D-EVM hvl embedded in
nD space.
Input/Output: An nD-EVM p
Procedure PutHvl(EVM hvl, EVM p)
{ Appends an (n-1)D couplet hvl, which
  is perpendicular to $X_1$-axis, to p.  }
```

```
Input:  An nD-EVM p
Output: An integer
Procedure GetN(EVM p)
{ Returns the number n of dimensions of
  the space where p is embedded.      }
```

```
Input:  An nD-EVM p
Output: A Boolean.
Procedure IsEmpty(EVM p)
{ Returns true if p is an empty set.  }
```

```
Input:  An nD-EVM p
Output: A coordinate of type CoordType
(the  chosen  type  for  the  vertex
coordinates: Integer or Real)
Procedure GetCurrentCoord(EVM p)
{ Returns the common $X_1$-coordinate
  of the next (n-1)D couplet to be
  extracted from p.                   }
```

```
Input/Output: An (n-1)D-EVM p embedded
in (n-1)D space.
Input:  A  coordinate  coord  of  type
CoordType  (the  chosen  type  for  the
vertex coordinates: Integer or Real)
Procedure SetCoord(EVM p,
CoordType coord)
{ Sets the $X_1$-coordinate to coord
  on every vertex of the (n-1)D
  couplet p. For coord = 0 it
  performs the projection $\pi_1(p)$.    }
```

```
Input:  Two nD-EVM's p and q.
Output: An nD-EVM
Procedure MergeXor(EVM p, EVM q)
{ Applies the Exclusive OR operation
  to the vertices of p and q and
  returns the resulting set.          }
```

Function MergeXor performs an XOR between two nD-EVM's, that is, it keeps all vertices belonging to either $EVM_n(p)$ or $EVM_n(q)$ and discards any vertex that belongs to both $EVM_n(p)$ and $EVM_n(q)$. Since the model is sorted, this function consists on a simple merging-like algorithm, and therefore, it runs on linear time [2]. Its complexity is given by $O(Card(EVM_n(p)) + Card(EVM_n(q))$ since each vertex from $EVM_n(p)$ and $EVM_n(q)$ needs to be processed just once. Moreover, according to **Theorem 2.4**, the resulting set corresponds to the regularized XOR operation between p and q.

From the above primitive operations, the **Algorithms 2.1** and **2.2** may be easily derived. The **Algorithm 2.3** computes the sequence of sections of an nD-OPP p from its nD-EVM using the previous functions [2]. It sequentially reads the projections of the (n-1)D couplets *hvl* of the polytope p. Then it computes the sequence of sections using function *GetSection*. Each pair of sections S_i and S_j (the previous and next sections about the current *hvl*) is processed by a generic processing procedure (called *Process*), which performs the desired actions upon S_i and S_j.

```
Input: An (n-1)D-EVM corresponding to
section S. An (n-1)D-EVM corresponding
to couplet hvl.
Output: An (n-1)D-EVM.
Procedure GetSection(EVM S, EVM hvl)
    // Returns the projection of the
    // next section of an nD-OPP
    // whose previous section is S.
    return MergeXor(S, plv)
end-of-procedure
```

Algorithm 2.1. Computing $EVM_{n-1}\left(\pi_1(S_k^i(p))\right)$ as $EVM_{n-1}\left(\pi_1(S_{k-1}^i(p))\right) \otimes EVM_{n-1}\left(\pi_1(\Phi_k^i(p))\right)$

```
Input: An (n-1)D-EVM corresponding to
section $S_i$. An (n-1)D-EVM corresponding
to section $S_j$.
Output: An (n-1)D-EVM.
Procedure GetHvl(EVM $S_i$, EVM $S_j$)
    // Returns the projection of the
    // couplet between consecutive
    // sections $S_i$ and $S_j$.
    return MergeXor($S_i$, $S_j$)
end-of-procedure
```

Algorithm 2.2. Computing $EVM_{n-1}\left(\pi_1(\Phi_k^i(p))\right)$ as $EVM_{n-1}\left(\pi_1(S_{k-1}^i(p))\right) \otimes EVM_{n-1}\left(\pi_1(S_k^i(p))\right)$

```
Input:  An nD-EVM p.
Procedure EVM_to_SectionSequence(EVM p)
        EVM hvl  // Current couplet.
        EVM $S_i$,$S_j$ // Previous and next sections about hvl.
        hvl = InitEVM( )
        $S_i$ = InitEVM( )
        $S_j$ = InitEVM( )
        hvl = ReadHvl(p)
        while(Not(EndEVM(p)))
                $S_j$ = GetSection($S_i$, hvl)
                Process($S_i$, $S_j$)
                $S_i$ = $S_j$
                hvl = ReadHvl(p) // Read next couplet.
        end-of-while
end-of-procedure
```

Algorithm 2.3. Computing the sequence of sections from an nD-OPP p represented through the nD-EVM.

3 Representing Color 2D Videos through 4D-OPP's and the EVM

The procedure described in [2] for processing black & white 2D animations can be directly extended to control colored frames through a 4D-OPP represented through the EVM. In the **Figure 1** an example of a simple color 2D-animation composed by four frames whose resolution is 9 × 9 pixels is shown. In each frame can be identified yellow, red, green and blue regions. We will use this simple animation to exemplify our procedure. We will label each colored frame in the animation as f_k and m will be the number of such frames.

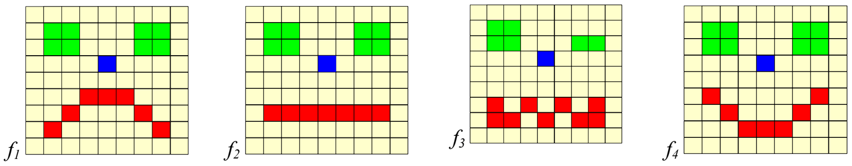


Figure 1. Example of a simple color 2D-animation.

A color animation can be handled as a 4D-OPP in the following way [9]:

a) The red-green-blue components of each pixel will be integrated into a single value. Such value represents the red-green-blue components as an integer with 32 bits. Bits 0-7 correspond to the blue value, bits 8-15 correspond to the green value, bits

16-23 correspond to the red value and bits 24-31 to the *alpha* (transparency) value. Each pixel will now be extruded towards the third dimension where the value integrating its red-green-blue components will now be considered as its X_3 coordinate (coordinates X_1 and X_2 correspond to the original pixels' coordinates). See **Figure 2**.

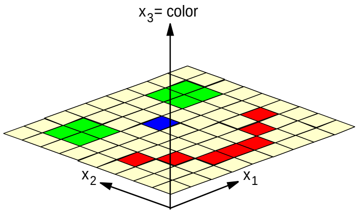
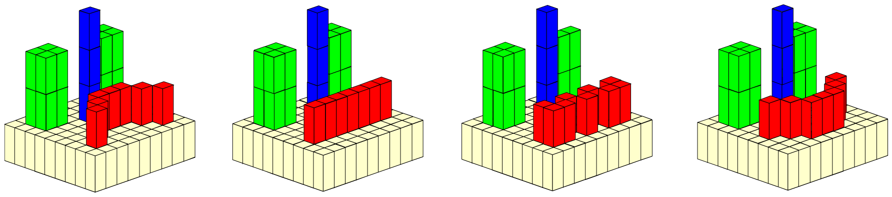


Figure 2. The 3D space defined for the extrusion of color 2D-pixels.

Let us call xf_k to the set composed by the rectangular prisms (the extruded pixels) of each extru-ded frame f_k. It is very important to avoid the zero value in the X_3 coordinate because a pixel could not be extruded and therefore its associated prism (a 3D-OPP) won't be obtained. See in **Figure 3** the sets of prisms xf_k which are the result of the extrusion of frames f_k of the animation from **Figure 1**.



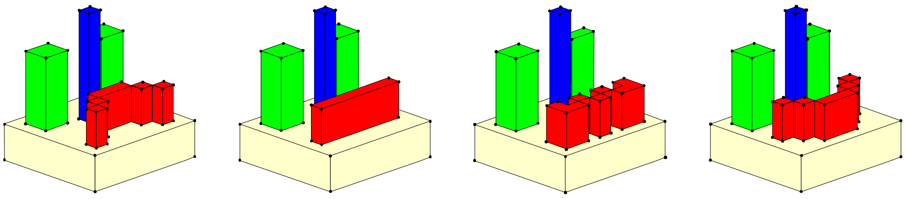
xf_1 xf_2 xf_3 xf_4

Figure 3. The sets of prisms which are the result of the extrusion of the frames of an animation (presented in Figure 1).

b) Let $prism_i$ be a prism in xf_k and npr the number of prisms in that set. Due to all the prisms in xf_k are quasi disjoint 3D-OPP's, we can easily obtain the final 3D-OPP and its respective 3D-EVM of the whole 3D frame by computing the regularized union of all the prisms in xf_k. Then, according to **Corollary 2.3**, we have to apply (all the vertices in a $prism_i$ are extreme):

$$EVM_3(F_k) = \bigotimes_{i=1}^{npr} EVM_3(prism_i \in xf_k)$$

where F_k is the 3D frame (a 3D-OPP) that represents the union of all the prisms in xf_k.



F_1 F_2 F_3 F_4

Figure 4. The 3D frames that represent a 2D colored animation (presented in Figure 1. Some of their extreme vertices are shown).

In the **Figure 4** are shown the 3D frames F_k from the animation presented in **Figure 1**.

c) Let us extrude F_k into the fourth dimension, and thus obtain a 4D hyperprism *hyperprism*$_k$ whose bases are F_k and its length is proportional to the time f_k is to be displayed. The new fourth dimension will measure and represent the time. See **Figure 5**.

d) Let $p = \bigcup_{k=1}^{m} hyperprism_k$, then p is a 4D-OPP that represents the given color 2D-animation. Due to all the m hyperprisms are quasi disjoint 4D-OPP's, then the 4D-EVM for p can be obtained by:

$$EVM_4(p) = \bigotimes_{k=1}^{m} EVM_4(hyperprism_k)$$

In the **Figure 6** are shown the couplets perpendicular to the axis that represent the time, of the 4D-OPP p that represents the animation from **Figure 1**. The **Algorithm 3.1** shows the procedure for converting a set of frames in an animation to a 4D-OPP that codifies it. Such 4D-OPP is represented through a 4D-EVM.

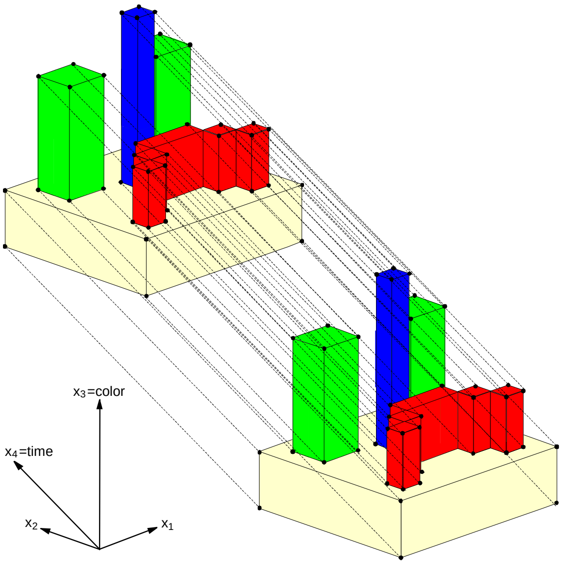
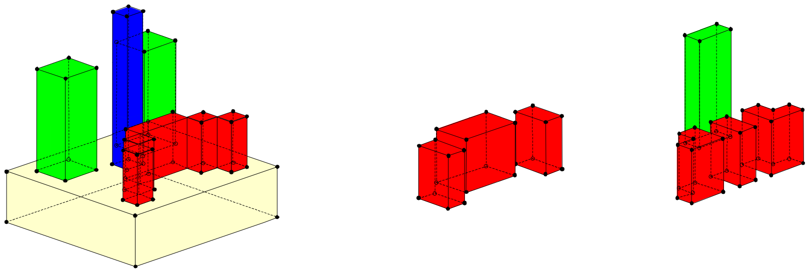


Figure 5. The process of extrusion of a 3D frame in order to obtain a *hyperprism* (some of its extreme vertices are shown).



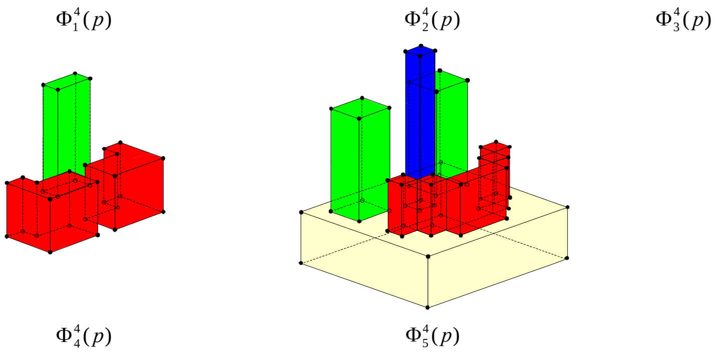


Figure 6. The 3D couplets of the 4D-OPP p that represents a color 2D-animation (from Figure 1. Their extreme vertices are shown).

```
Input:  A sequence of frames associated to a color 2D animation.
        The values xSize and ySize corresponding to the resolution of the input
        animation.
Output: The 4D-EVM corresponding to the polytope that codifies frames in the
        input animation.
Procedure GenerateEVM-movie(Movie animation, xSize, ySize)
   EVM evmMovie // The EVM that will store and codify the input animation.
   EVM hvl
   EVM Fcurr, Fprev      // Current and previous 3D frames being processed.
   real t                // The amount of time that current processed frame is
                         // displayed.
   Fprev = InitEVM( )
   for each frame in animation do
        Fcurr = InitEVM( )
        Frame f = animation.nextFrame( )
        t = animation.getDisplayingTime( )
        // Frame f is extruded towards 3rd dimension and its 3D-EVM is computed.
        for x = 0 until xSize – 1 do
            for y = 0 until ySize – 1 do
                rgb = getRGBComponents(x, y, f)
                // We obtain the EVM of the prism associated to (x, y, rgb).
                EVM prism = GetPrismEVM(x, y, rgb)
                Fcurr = MergeXor(prism, Fcurr)
            end-of-for
        end-of-for
        // We perform the Xor between the current and previous 3D frames.
        hvl = MergeXor(Fcurr, Fprev)
        // Amount of time t associated to frame Fcurr is attached to the current
        // 3D couplet.
        SetCoord(hvl, t)
        // A new 3D couplet is attached to the 4D polytope that codifies the
        // input animation.
        PutHvl(hvl, evmMovie)
        Fprev = Fcurr
    end-of-for
    return evmMovie
end-of-procedure
```

Algorithm 3.1. Codifying a Color 2D-animation through a 4D-OPP and the EVM.

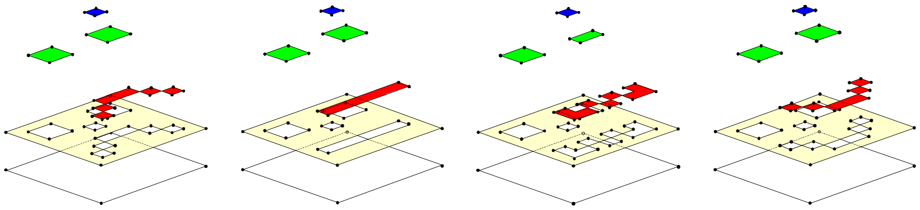
```
Input:  A 4D-EVM p that represents a color 2D-animation.
        The graphics context g where the animation is going to be displayed.
Procedure playEVM-movie(EVM p, g)
    EVM hvl              // Current 3D couplet in p.
    EVM Fprev, Fcurr     // Previous and current 3D frames in the animation.
    EVM hvlF             // Current 2D couplet in Fcurr. It contains polygons
                         // to display.
    int color            // The color to apply to the polygons to be displayed.
    Fprev = InitEVM( )
    hvl = ReadHvl(p)
    while(Not(EndEVM(p)))
         Fcurr = GetSection(Fprev, hvl)    // We get the next 3D frame.
         // We proceed to display the current frame in the animation.
         while(Not(EndEVM(Fcurr)))
                 // Get the common coordinate of the vertices in the next 2D
                 // couplet to be extracted from Fcurr.
                 color = GetCurrentCoord(Fcurr)
                 g.setColor(color)
                 hvlF = ReadHvl(Fcurr)
                 // Rectangles in the 2D couplet are displayed.
                 DisplayPolygons(hvlF, g)
         end-of-while
         Fprev = Fcurr
         hvl = ReadHvl(p)                  // Read next 3D couplet.
    end-of-while
end-of-procedure
```

Algorithm 3.2. Displaying a color 2D-animation represented through a 4D-OPP and the EVM.

By representing a given color 2D-animation using a 4D-OPP *p* and its 4D-EVM we have the following characteristics [9]:

- The sequence of the projections of sections in *p* corresponds to the sequence of 3D frames, i.e., $\pi_4\left(S_k^4(p)\right)=F_k$.
- Computation of 3D frames: Because p is expressed through the EVM then by **Corollary 2.2** the 3D-EVM of the frame F_k is computed by $EVM_3\left(F_k\right)=EVM_3\left(F_{k-1}\right)\otimes EVM_3\left(\pi_4\left(\Phi_k^4(p)\right)\right)$.
- Displaying the 2D colored animation: Each couplet perpendicular to the X_3 axis in each 3D frame F_k contains the polygons to display. The colors to apply to those polygons are referred through the X_3 coordinate that contains the integrated red-green-blue components.

In the **Figure 7** are presented the sequences of couplets of the 3D frames F_k for the 2D animation presented in **Figure 1**.



F_1's 2D couplets F_2's 2D couplets F_3's 2D couplets F_4's 2D couplets

Figure 7. The sequences of couplets of the 3D frames that represent a color 2D-animation.

The **Algorithm 3.2** applies the above ideas in order to extract animation colored 2D frames from a 4D-OPP and display them. Basically it extracts the 3D couplets perpendicular to X_4-axis and computes the sections that correspond to the extrusion to 3D space of the animation's 2D frames. When the extrusion of a frame is obtained then its 2D couplets perpendicular to X_3-axis are extracted. Such 2D couplets are the polygons to draw and their filling color is assigned according to their common X_3 coordinate in the 3D frame. A 2D couplet is processed through the procedure *DisplayPolygons* in the algorithm.

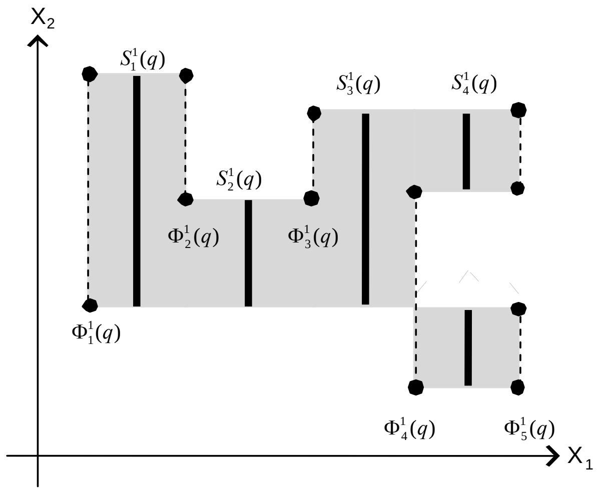


Figure 8. A 2D-OPP q whose composing rectangles are being computed. The coordinates of a rectangle in $Slice_i^1(q)$ are given by the coordinates of the projection of $S_i^1(q)$ and common coordinates of its bounding couplets $\Phi_i^1(q)$, Φ_{i+1}^1.

DisplayPolygons is implemented in **Algorithm 3.3**. In order to draw the rectangles that compose an input 2D-OPP we will consider the partition induced by its Slices. A Slice from a 2D-OPP can be seen as a set of one or more disjoint rectangles whose 1D base is the slice's section. The coordinates that define an specific rectangle in $Slice_k^1(p)$ can be determined through its respective section $S_k^1(p)$ (the 1D base of $Slice_k^1(p)$) and the common coordinates of $\Phi_k^1(p)$ and $\Phi_{k+1}^1(p)$, i.e., the common coordinates of the 1D-couplets that bound section $S_k^1(p)$. See **Figure 8**.

```
Input:  A 2D-EVM p and the graphics context g where p is going to be displayed.
Output: True if and only if the number of dimensions of p is 2.
        False if the number of dimensions of p is not 2, hence, no elements of p
        were displayed.
Procedure DisplayPolygons(EVM p, g)
    if (GetN(p) ≠ 2) then
        return False
    EVM hvl           // Current 1D couplet in p.
    EVM $S_i$, $S_j$        // Previous and next sections about hvl.
    $S_i$ = InitEVM( )
    int rectangleX[4] // Coordinates along $X_1$-axis of a rectangle to be displayed.
    int rectangleY[4] // Coordinates along $X_2$-axis of a rectangle to be displayed.
    int point1, point2   // Two consecutive points in 1D section $S_j$.
    if (Not(IsEmpty(p))) then
        double prevCoord = GetCurrentCoord(p)
        hvl = ReadHvl(p)
        while(Not(EndEVM(p)))
           $S_j$ = GetSection($S_i$, hvl)       // Current section is an 1D-OPP.
           // We extract the ordered sequence of points in 1D section $S_j$.
           int points[ ] = GetEVM($S_j$)
           k = 0
           while(k < points.size)
                // Each segment in the 1D current section is extruded and
                // displayed.
                point1 = points[k]
                point2 = points[k+1]
                // prevCoord and GetCurrentCoord(p) are the $X_1$-coordinates about
                // section $S_j$.
                rectangleX[0] = prevCoord
                rectangleY[0] = point1
                rectangleX[1] = prevCoord
                rectangleY[1] = point2
                rectangleX[2] = GetCurrentCoord(p)
                rectangleY[2] = point2
                rectangleX[3] = GetCurrentCoord(p)
                rectangleY[3] = point1
                // We display the rectangle.
                g.fillPolygon(rectangleX, rectangleY, 4)
                k = k + 2
            end-of-while
            prevCoord = GetCurrentCoord(p)
            S_i = S_j
            hvl = ReadHvl(p)
        end-of-while
    end-of-if
    return True
end-of-procedure
```

Algorithm 3.3. Displaying the rectangles that compose a 2D-OPP expressed through the EVM.

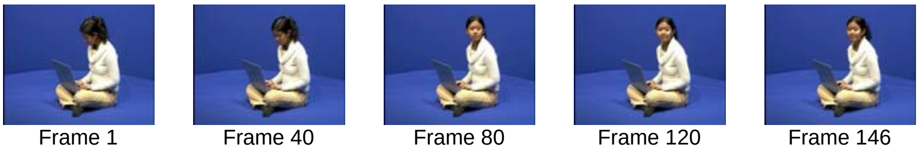
3.1 Experimental Results

We evaluated our procedure through two blue screen video sequences which were produced originally at a TV studio of the University of Arts in Bremen [3]. Such sequences are AVI XVID codified videos (720 × 576, 24 bits color). We converted such sequences, for our experiment, to videos with resolution of 320 × 240 pixels (standard TV) and 64 colors.

The first sequence was composed by 146 frames. The 4D-OPP that represented such set of selected frames has 848,598 extreme vertices. In another experimented case we considered a second movie sequence whose time length was 100 frames. The size of the 4D-EVM corresponding to its codification as a 4D-OPP required 1,472,174 extreme vertices.

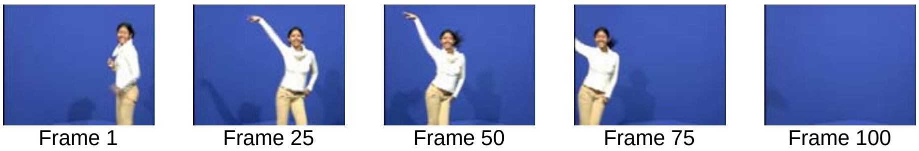
As can be noted, in the first referenced sequence we required 848,598 extreme vertices for representing 146 animation frames, while in the second sequence we required 1,472,174 extreme vertices for representing 100 frames. The reason behind this behavior was yet identified in [2]: $EVM_3\left(F_k\right) = EVM_3\left(F_{k-1}\right) \otimes EVM_3\left(\pi_4\left(\Phi_k^4(p)\right)\right)$, i.e., the regions at couplets $\Phi_k^4(p)$ represent the regions of a previous frame F_{k-1} that need to be modified in order to update it to the following frame F_k. In other words, a couplet perpendicular to X_4-axis $\Phi_k^4(p)$ only stores the differences between consecutive 3D frames F_{k-1} and F_k. The way the frames change through time has impact over the number of extreme vertices in the couplets associated to the 4D-OPP that represents the animation. The first animation contains a girl who is sat and working with a computer. As seen in **Figure 9**, the girl, along time, is practically quiet. Hence, we have a lot of redundancy between all frames in the animation. Therefore, only minimal differences are stored in the OPP's couplets, except the first and last couplets, whose visualization coincide with the first and last frames in the original animation. On the other hand, the second animation is a sequence where the girl is jumping and dancing along the screen from right to left (See **Figure 10**). In this case we have a level of redundancy that is minor than the one found in the first animation because we have more noticeable changes between consecutive frames.

According to this experiment we can conclude that the EVM's conciseness, respect to the representation of animations, depends of the degree of redundancy between the frames associated to the animations. As noted in [4], cartoon animations are a good example of animations with an elevated redundancy, but we are one step further by considering sequences with complex color and gray scale frames.



Frame 1 Frame 40 Frame 80 Frame 120 Frame 146

Figure 9. Five main frames taken from the first animation used for conversion to the 4D-EVM: 848,598 extreme vertices were required for encoding 146 frames (original sequence taken from [3]).



Frame 1 Frame 25 Frame 50 Frame 75 Frame 100

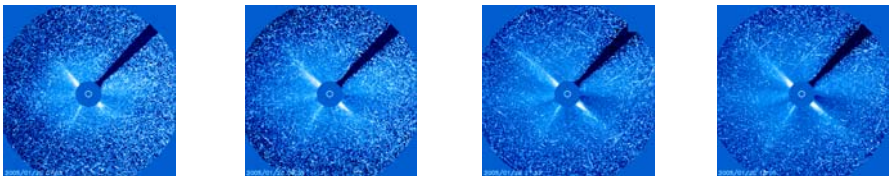
Figure 10. Five main frames taken from the second animation used for conversion to the 4D-EVM: There were required 1,472,174 extreme vertices for encoding 100 frames (original sequence taken from [3]).

4 Conclusions and Future Work

In this work we have described the **Extreme Vertices Model in the n-Dimensional Space** (**nD-EVM**). The Extreme Vertices Model allows representing nD-OPP's by means of a single subset of their vertices: the *Extreme Vertices*. **Section 2** is in fact a very brief description of the capabilities of the model because we have developed simple and robust algorithms, besides the ones presented in this work, for performing the most usual and demanding tasks on polytopes modeling such as closed and regularized Boolean operations, boundary extraction, and set membership classification operations (see [2] and [9] for more details). In this aspect we mention the development of other "real world" practical applications under the context of the nD-EVM, which are widely discussed and modeled in [9]. These practical applications, through we have showed the versatility of application of the nD-EVM, consider: (1) a method for comparing images oriented to the evaluation of volcanoes' activity; (2) the way the nD-EVM enhances Image Based Reasoning; (3) the manipulation and extraction of information from 3D datasets (see also [10]), and finally, (4) an application to collision detection between 3D objects through the nD-EVM.

There are many aspects related to the procedure we have described in this work that can be improved. An idea to consider is concerned to the non-supervised detection of polygons in the 2D video sequences to be vectorized. As commented in **Section 1**, Koloros & Zára [4] and Kwatra & Rossignac [6] consider the detection of polygons as a core step in their respective methodologies. In the case of [4] detected polygons in one frame are evaluated with the remaining frames in order to identify repeated or similar polygons with the objective to compress the final representation of the vectorized animation. In [6] detected polygons are considered 3D volumes whose third dimension is given by time. Finally, through their Edgebreaker compression the evolution across time of the polygons are encoded as volume geometry. We are open to consider the ideas given in [4] and [6] in order to provide much more compression

We commented in **Section 3.1** the results obtained from the vectorization of two video sequences. It is well known that Telecine process is used to transform motion picture film into digital video format [4]. For reproduction PAL or NTSC standard under the respective resolution of 720 × 576 or 720 × 480 pixels is used and the final data is commonly stored at betacam tapes or di-rectly in the computer. The use of MPEG-1 or MPEG-2 commonly results in lower quality with arti-facts that make the vectorization difficult [4], and in our case, impacts the cardinality of the obtained EVM's. According to the use of MPEG-4 codification we can expect better quality at reasonable data-rate. For our intention to vectorize video data we need the best possible quality. We will test our procedure with video sequences codified in MPEG-4 (yet available in the next generation DVD formats such as HD-DVD [7]) in order to prove that by obtaining better quality in our video sources we obtain better compression of the final vectorized video represented under the EVM.



Frame 1 Frame 3 Frame 5 Frame 7

Figure 11. A sequence of frames that presents a coronagraph image of a radiation storm (taken from [5]).

We will finalize by establishing a question to be addressed. According to our experiments and by the fact that a couplet perpendicular to X$_4$-axis $\Phi_k^4(p)$ only stores the differences between consecutive 3D frames F_{k-1} and F_k. we conclude that by more level of redundancy between consecutive frames we can expect more conciseness from the EVM. However, scientific sequences provide us with a huge set of examples where the degree of redundancy is very low or inexistent. Consider for example the sequence presented in **Figure 11**. Such sequence describes a coronagraph image of a radiation storm which took place in January 20, 2005 [5]. The main characteristic in this sequence shows that the value of a pixel (inside the main circle) in a frame is distinct from the value of that same pixel in the next frame. Because of the accuracy required when these sequences are analyzed, any kind of threshold, which could elevate the redundancy degree, is prohibited. Hence, the cardinality of the EVM's that represent these sequences is expected to be high. The procedures we have described consider the representation of a frame by considering the original two spatial dimensions for each one of its pixels. A possible solution to the addressed problem could consider the linearization of each frame. That is, a frame can be considered as a matrix but by stacking its columns on top of one another we obtain a vector. In this way we have that each pixel can be referenced by only one coordinate in the vector. Hence, we deal with only one spatial dimension instead of the original two dimensions. Obviously we have the way to recover the original position of a pixel given the original width and height of its frame. It can be observed that a 3D-OPP can represent our animation: X_1-axis will correspond to the position in the linearization, X_2-axis will refer to its red-green-blue integrated components, and X_3-axis will be associated to time. **Algorithms 3.1** to **3.3** should be modified to consider an OPP that represents a set of linearized frames. In the future we will discuss these described ideas and the obtained results.

References

1. Aguilera, Antonio. & Ayala, Dolors. Orthogonal Polyhedra as Geometric Bounds in Constructive Solid Geometry. Fourth ACM Siggraph Symposium on Solid Modeling and Applications SM'97, pp. 56-67. Atlanta, USA, 1997.
2. Aguilera, Antonio. Orthogonal Polyhedra: Study and Application. PhD Thesis. Universitat Politècnica de Catalunya, 1998.
3. Center for Computing Technologies, Digital Media/Image Processing, University of Bremen. Web site: http://www.tzi.de/tzikeyer/index.html
4. Koloros, Martin & Zára, Jirí. Coding of vectorized cartoon video data. Proceedings of Spring Conference on Computer Graphics 2006, pp. 177-183. Comenius University, Bratislava, 2006.
5. Koppeschaar, Carl. Astronet's Web Site: http://www.xs4all.nl/~carlkop/auralert.html
6. Kwatra, Vivek & Rossignac, Jarek. Space-Time surface simplification and Edgebreaker compression for 2D cel animations. International Journal of Shape Modeling, vol. 8, No. 2, December 2002.
7. Moeritz, S. & Diepold, K. Understanding MPEG 4: Technology and Business Insights. Focal Press, 2004.
8. Pérez-Aguila, Ricardo. The Extreme Vertices Model in the 4D space and its Applications in the Visualization and Analysis of Multidimensional Data Under the Context of a Geographical Information System. MSc Thesis. Universidad de las Américas, Puebla. Puebla, México, May 2003.
9. Pérez-Aguila, Ricardo. Orthogonal Polytopes: Study and Application. PhD Thesis. Universidad de las Américas - Puebla. Cholula, Puebla, México, November 13, 2006.
10. Pérez-Aguila, Ricardo. Modeling and manipulating 3D Datasets through the Extreme Vertices Model in the n-Dimensional Space (nD-EVM). Accepted and to appear in the First International Conference on Industrial Informatics, CICINDIN 2007. To be held in México City, México, November 5 to 9, 2007.
11. Spivak, M. Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus. HarperCollins Publishers, 1965.

Median M-Type Radial Basis Function Neural Network for Image Classification Applications

Jose Augusto Moreno-Escobar [1], Francisco Gallegos-Funes[1],
Volodymyr Ponomaryov[2], Oleksiy Pogrebnyak[3]

National Polytechnic Institute of Mexico
[1] Mechanical and Electrical Engineering Higher School, U.P. Zacatenco, fgallegosf@ipn.mx
[2] Mechanical and Electrical Engineering Higher School, U.P. Culhuacan,
vponomar@ipn.mx
[3] Center for Computing Research, U.P. Zacatenco,
olek@pollux.cic.ipn.mx

Abstract. We present the Median M-Type Radial Basis Function (MMRBF) neural network for image classification applications. The proposed neural network uses the Median M-type (MM) estimator in the scheme of radial basis function to train the neural network. Extensive simulation results have demonstrated that the proposed MMRBF neural network consistently outperforms other RBF algorithms in terms of classification capabilities.

1 Introduction

In recent years neural computing has emerged as a practical technology, with successful applications in several fields. These applications are concerned with problems in pattern recognition, and make use of feed-forward network architectures such as the multi-layer perceptron and the radial basis function network [1,2].

The Radial Basis Functions (RBF) have been used in several applications for pattern classification and functional modeling [3]. These functions have been found to have very good functional approximation capabilities [3]. The RBF have their fundamentals drawn from probability function estimation theory.

The RBF network involves three layers with entirely different roles [4-6]. The input layer is made up of source nodes that connect the network to its environment. The second layer is the only hidden layer in the network, applies a nonlinear transformation from the input space to the hidden space. The output layer is linear, supplying the response of the network to the activation signal or pattern applied to the input layer
In this paper, we present the use of the Median M-Type (MM) estimator with different influence functions [7] as statistic estimation in the Radial Basis Function network architecture for image classification purposes. Extensive simulation results have demonstrated that the proposed MMRBF neural network consistently outperforms other RBF algorithms in terms of classification capabilities.

Received 05/07/07
Accepted 19/10/07
Final version 24/10/07

2 Proposed MMRBF Neural Network

In the RBF neural network each of N_k components of the input vector **X** feeds forward to M basis functions whose outputs are linearly combined with weights $\{\lambda_j\}_{j=1}^{M}$ into the network output $Y_k(\mathbf{X})$ [4-6]. Figure 1 presents the structure of the RBF neural network.

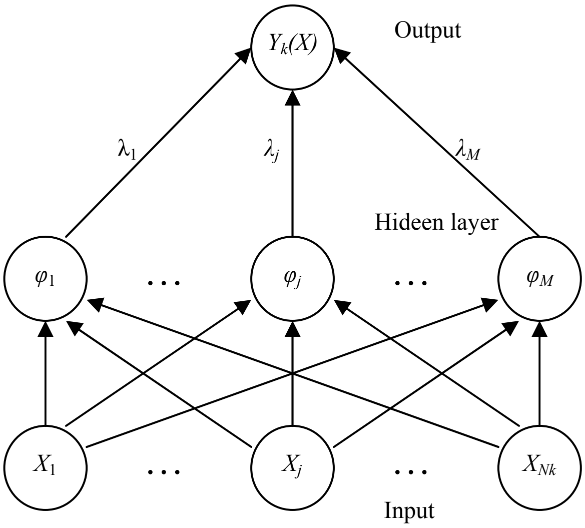


Fig. 1. Radial Basis Function Neural Network architecture.

The inverse multiquadratic function is used as activation function in the proposed MMRBF neural network [5],

$$\phi_j(\mathbf{X}) = \frac{1}{\sqrt{\mathbf{X}^2 + \beta_j^2}} \tag{1}$$

where **X** is the input feature vector, β_j is a real constant. In our simulation results β_j=1.

A combined unsupervised-supervised learning technique has been used in order to estimate the RBF parameters [5].

In the unsupervised stage, the k-means clustering algorithm is used to estimate the parameters of the MMRBF neural network [1,2]. The input feature vector **X** is classified in k different clusters. A new vector **x** is assigned to the cluster k whose centroid μ_k is the closest one to the vector. The centroids can be updated at the end of several iterations or after the test of each new vector, and they can be calculated with or without the new vector. The centroid vector is updated in the following way [8,9]

$$\mu_k = \mu_k + \frac{1}{N_k}(\mathbf{x} - \mu_k) \tag{2}$$

where N_k is the number of vectors already assigned to the k-cluster.

The Hard limit transfer function is used in the supervised stage to calculate the weights coefficients in the neural network [1,2].

$$\text{hardlim}\,(x) = \begin{cases} 1, & x \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{3}$$

The Median M-type (MM) estimator [7] is used in the proposal RBF neural network as robust statistics estimate of a cluster center,

$$\mu_k = \text{med}\{\mathbf{X}\tilde{\psi}(\mathbf{X} - \theta)\} \tag{4}$$

where $\mathbf{X}$ is the input data sample, $\theta = \text{med}\{X_k\}$ is the initial estimate, $\tilde{\psi}$ is the normalized influence function $\psi : \psi(\mathbf{X}) = \mathbf{X}\tilde{\psi}(\mathbf{X})$, k=1, 2,…,N_k, and the influence functions used are the following [7]:
the simple cut (skipped mean) influence function,

$$\psi_{\text{cut}(r)}(X) = \begin{cases} X, & |X| \leq r \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

and the Tukey biweight influence function,

$$\psi_{\text{bi}(r)}(X) = \begin{cases} X^2(r^2 - X^2), & |X| \leq r \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where X is a data sample and r is a real constant and depends of the data to process and can change for different influence functions.

3 Experimental Results

We obtained from the simulation experiments the properties of proposed Median M-Type Radial Basis Function (MMRBF) neural network with simple cut (sc) and Tukey biweight (tb) influence functions, and its performance has been compared with the Simple RBF (SRBF), Median RBF (ATMRBF) [8], and α-Trimmed Mean RBF (MRBF) neural networks [9].

To determine the classification properties of proposed Median M-Type Radial Basis Function (MMRBF) neural network and other RBF networks used as comparative we apply them to mammographic image analysis [10,11]. The images used to train and probe the RBF neural networks were obtained from the Mammographic Image Analysis Society (MIAS) web site [12]. Table 1 shows the number of images used in the stages of training and probe and the groups used to classify different mammographic images.

Table 1. Groups of mammographic images used to train and probe different RBF neural networks.

Group	Mammographic Images	Training	Probe
A	normal	8	40
	benign abnormalities	8	38
B	malign abnormalities	8	30
	benign microcalcifications	4	8
	malign microcalcifications	4	9

The criteria used to compare the performance of neural networks were the efficiency and error,

$$efficiency = \frac{\#\ of\, right\ \ probes}{total\ of\, i\, mages} \text{ x } 100\% \tag{7}$$

$$error = \frac{\#\ of\, errores}{total\ of\, images} \text{ x } 100\% \tag{8}$$

Therefore, to evaluate the performance of the neural networks in terms of medical purposes, we calculated two quantities [13]:
the *sensitivity* is the probability that a medical test delivers a positive result when a group of patients with certain illness is under study [13],

$$Sn = TP/(TP + FN) \tag{9}$$

and the *specificity* is the probability that a medical test delivers a negative result when a group of patients under study do not have certain illness [13],

$$Sp = TN/(TN + FP) \tag{10}$$

where Sn is the *sensitivity*, TP is the number of true positive that are correct, FN is the number of false negatives, that is, the negative results that are not correct, Sp is the *specificity*, TN is the number of negative results that are correct, and FP is the number of false positives, that is, the positive results that are not correct.

Figure 2 presents the performance results in terms of efficiency and error for the classification of mammographic images in the groups A and B (see Table 1). In this Figure one can see that the best results are obtained when we use the proposed MMRBF neural network.

Table 2 show the comparison between different RBF algorithms used in the mammographic image analysis. We observe from this Table that the proposed MMRBF neural network has the best efficiency in the probe stage in the most of the cases.

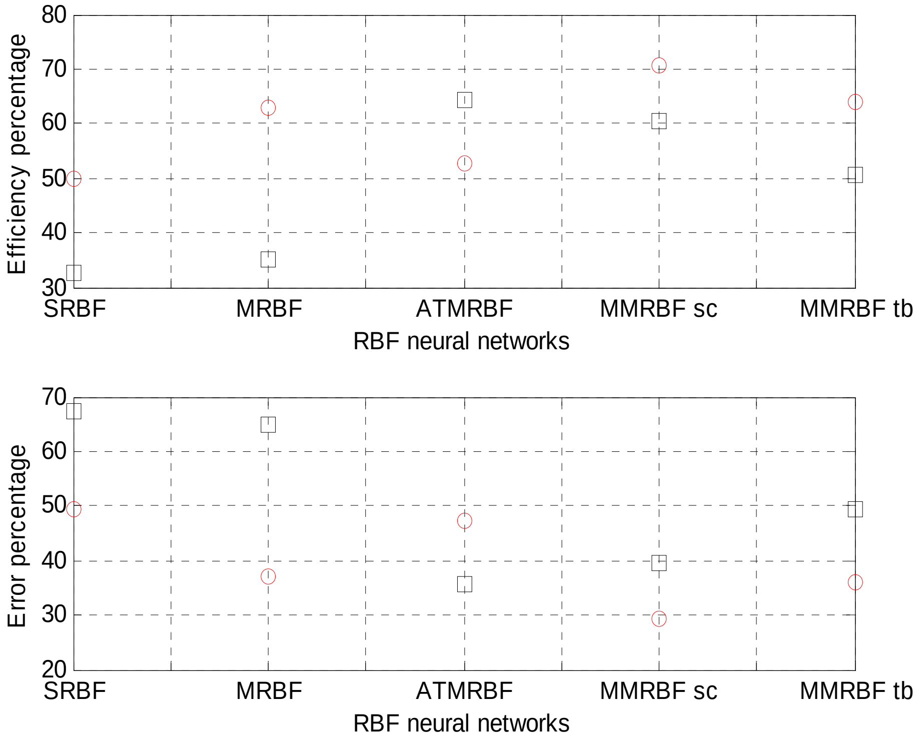


Fig. 2. Performance results of efficiency and error by group obtained by different neural networks, the red circle and black square indicate the group A and B, respectively.

Table 2. Efficiency results between the MMRBF and other algorithms used as comparative in the probe stage.

Neural Networks	SRBF	MRBF	ATMRBF
MMRBF sc	24.90%	18.22%	4.80%
MMRBF tb	16.38%	9.70%	-3.72%

Table 3 presents the *sensitivity* and *specificity* values obtained for different RBF neural networks. It can be appreciated that the specificity of the proposed MMRBF using simple cut influence function is the highest one, about a 20% above ATMRBF, but this last network has the best sensitivity, about 10% above the mentioned MMRBF sc.

Table 3. Sensitivity and specificity values for different RBF neural networks.

Neural Networks	Sensitivity	Specificity
SRBF	34.04%	50.00%
MRBF	31.91%	62.82%
ATMRBF	57.45%	52.56%
MMRBF simple cut	48.93%	70.51%
MMRBF Tukey	44.68%	64.10%

Figure 3 shows the visual results in the process of segmentation of mammographic images, with these images the proposed network realizes the classification of them into two groups (A and B). We notice that the error in the classification can be minimized by means of use other algorithms in the segmentation stage due that the mammography images are very irregular and it can cause false positive and false negative results. Figure 4 shows a mammographic image that corresponds to a proper result in terms of classification. In this case the proposed MMRBF neural network provides a better classification in comparison with other RBF networks. Figure 5 presents the case of an improper result due that some mammographic images are not regular.

In our experiments also was measured the time necessary for the system to deliver a result. We used a DELL ® Precision 380 PC, which has a Pentium 4 Intel® processor running at 3 GHz and 2GB RAM memory. The time to classify an image was measured in 65 images of 1024x1024 pixels and 8 bits per pixel. Table 4 shows the average processing time of the main stages of classification process.

Table 4. Average processing time (in minutes) for the main stages in the proposed method.

	Average time	% of total time
Segmentation and Feature Extraction	30.307	99.85
MMRBF Classifier	0.044	0.15

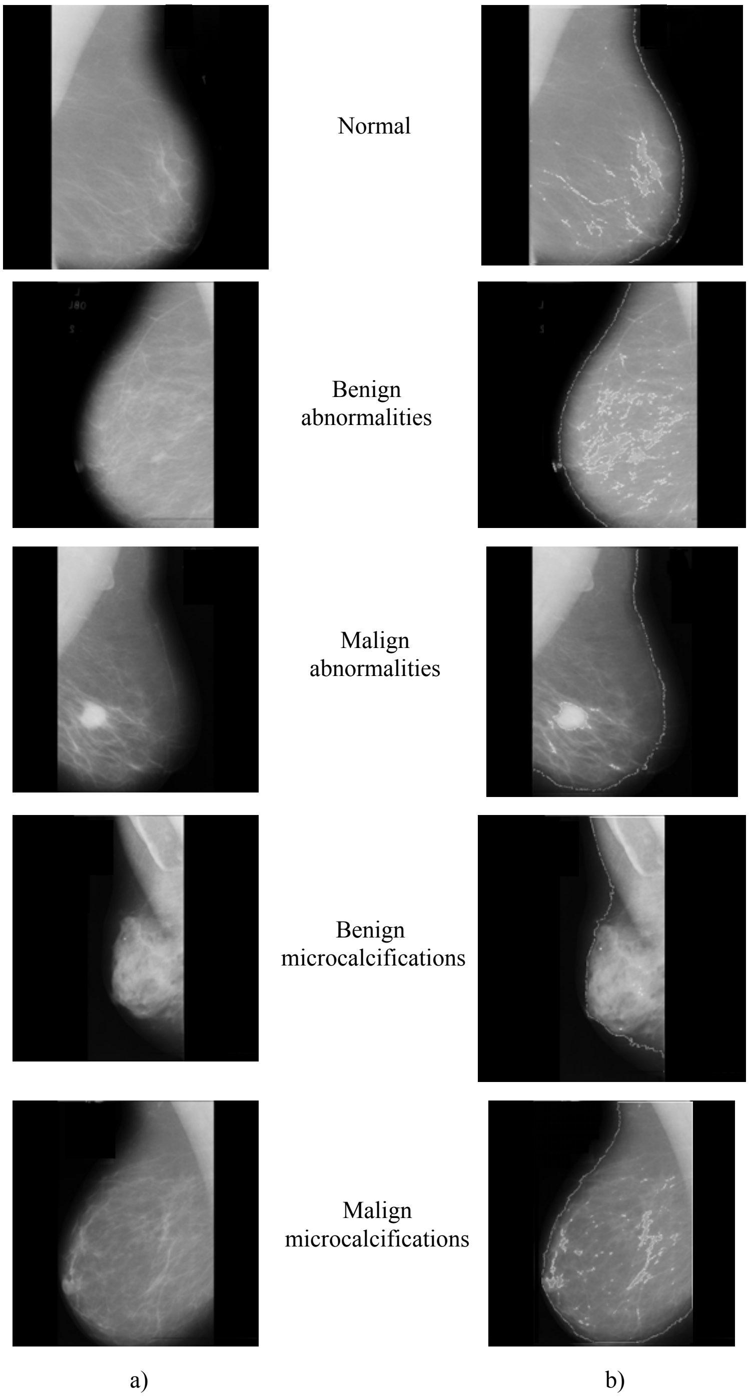
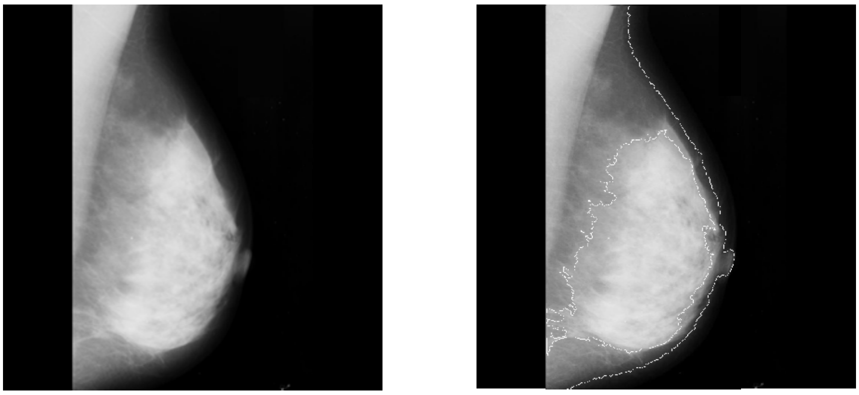
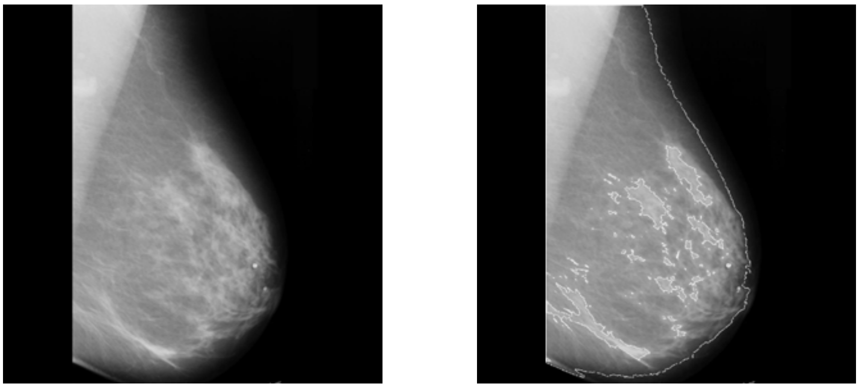


Fig. 3. Visual results in the classification of mammographic images, a) original images and b) segmented images.



a) Original image b) Segmented image

Fig. 4. Visual result of mammography image with a proper result in the classification of mammographic images.



a) Original image b) Segmented image

Fig. 5. Visual result of mammography image with an improper result in the classification of mammographic images.

4 Conclusions

In this paper we present the MMRBF neural network, it uses the MM-estimator with different influence functions in the scheme of radial basis function to train the proposed neural network.

Extensive simulation results in mammographic images have demonstrated that the proposed MMRBF neural network consistently outperforms other RBF algorithms in terms of classification capabilities.

Unfortunately the error is still big in the case of mammographic image analysis, it is due to simple segmentation algorithm used in this paper. The algorithm for segmentation used is based on morphology and thresholding. As future work we will probe with other segmentation algorithm to improve the classification of the regions of interest proposed in this paper.

Finally, in the case of use 256x256 mammographic images and digital signal processors, the processing time can be decreased for real-time applications.

Acknowledgements

The authors thank the National Polytechnic Institute of Mexico for its support.

References

1. Haykin, S.: Neural Networks, a Comprehensive Foundation. Prentice Hall, Upper Saddle River, NJ (1994)
2. Rojas, R.: Neural Networks: A Systematic Introduction. Springer-Verlag, Berlin. (1996)
3. Buhmann, M. D.: Radial Basis Functions: Theory and Implementations. Cambridge Monographs on Applied and Computational Mathematics. (2003)
4. Musavi, M.T., Ahmed, W., Chan, K.H., Faris, K.B., Hummels, D.M.: On the training of radial basis function classifiers. Neural Networks. vol. 5 (1992) 595-603
5. Karayiannis, N. B., Weiqun Mi, G.: Growing radial basis neural networks: merging supervised and unsupervised learning with network growth techniques. IEEE Trans. Neural Networks. 8(6) (1997) 1492-1506
6. Karayiannis, N. B., Randolph-Gips, M. M.: On the construction and training of reformulated radial basis function neural networks. IEEE Trans. Neural Networks. 14(4) (2003) 835-846
7. Gallegos, F., Ponomaryov, V.: Real-time image filtering scheme based on robust estimators in presence of impulsive noise. Real Time Imaging. 8(2) (2004) 78-90
8. Bors, A.G., Pitas, I.: Median radial basis function neural network. IEEE Trans. Neural Networks. 7(6) (1996) 1351-1364
9. Bors, A.G., Pitas, I.: Object classification in 3-D images using alpha-trimmed mean radial basis function network. IEEE Trans. Image Process. 8(12) (1999) 1744-1756
10. Webb, G.: Introduction to Biomedical Imaging. Wiley-IEEE Press, Hoboken, New Jersey. (2002)
11. Suri, J. S., Rangayyan, R. M.: Recent Advances in Breast Imaging, Mammography, and Computer-Aided Diagnosis of Breast Cancer. SPIE Press, Bellingham. (2006)
12. http://www.wiau.man.ac.uk/services/MIAS/MIAScom.html, Mammographic Image Analysis Society
13. http://www.cmh.edu/stats/definitions/

Algorithms for Cutting Surfaces Composed by Simplex Meshes

Luis Cruz and Luis G. de la Fraga

Cinvestav. Computer Science Departament.
Av. Instituto Politécnico Nacional 2508. 07300 México, D.F.
fraga@cs.cinvestav.mx

Abstract. In this work we develop three new operators to cut a volume surface composed by a simplex mesh that envelops such volume. These operators are based on remove the cut edges and reorganizing the mesh connectivity which, in turn, implies inserting new edges and reordering the faces' edges and edges' vertices. In order to test these operators we perform two simulations: in the first one a cylinder is cut, and in the second simulation a non-linear deformation with rupture is calculated over a sphere, thus the sphere is broken when are applied external forces. We also discuss the non-linear deformation model that we used in the second experiment.

Keywords: surface cutting, simplex meshes, surface deformation, non-elastic model, physically based simulation.

1 Introduction

The simplex meshes are a relatively new form to represent surfaces. Using simplex meshes allows some operators [1] for gluing other simplex meshes to construct more complicated models; such operators only change the links among the different elements that composes the mesh (vertices, edges and faces).

In this work three new operators are introduced, the *split operator*, that changes the genus of the mesh (i.e. the number of "holes" in the model) and outputs two distinct meshes; the *connectivity operator*, that checks if the mesh is already split; and the *cutting operator*, which, in turn, removes an edge from the mesh and re-orders its vertices and faces (and it possibly modifies the mesh genus). In order to implement these operators, we develop four algorithms.

Also, we apply external forces to deform the mesh using a non elastic deformation model over every edge in the mesh. In previous works [1,2] only was used a completely elastic model to deform a simplex mesh. We develop a non-elastic model with rupture based on a linear function and an exponential part which acts as the plastic region for the material, and with the rupture part itself.

To test our results we present two experiments: for the first one, a cylinder is built and some edges are removed until the model is split in two halves and enables the manipulation by the user of each part, one independently of the

Received 30/06/07
Accepted 19/10/07
Final version 24/10/07

other; the second experiment uses the non-elastic model in a sphere, an internal force is applied internally to deform it until the sphere is broken.

2 Previous Works

The representation of a volume's surface has been done with one of three kinds [3]: a triangular mesh [4], implicit representations (such as splines or hyperquadrics), and simplex meshes [1,5,6]. The research had been primarily concentrated on the first two, leaving the simplex meshes to a side method.

Principally the research with simplex meshes had been developed by Delingette [7], fundamentally to model some human [8,2] and mouse [9] organs. Delingette proposed some operators to modify the vertex or the edges connectivity to obtain a new mesh gluing basic forms.

Some research was performed about the problem of cutting the volume tetrahedrization, but not had used the simplex mesh from its surface [10]. Another work had been conducted towards triangle meshes, modeling the 3D surface and its properties and manipulation [9].

3 Description of the new operators

A simplex mesh has a constact connectivity. In this work we use a 2-simplex mesh in which every vertex has only three neighboring vertices.

Now, we will describe the algorithms which conforms the developed operators, the algorithms 1 and 3 are used by the edge remove operator (and its corresponding faces), the algorithm 2 represents the connectivity operator, and the algorithm 4 represents the split operator.

The algorithm 1 shows the steps taken to remove one edge from the mesh. In Fig. 1 we can see the used notation.

Algorithm 1 Edge remove

```
Require: A simplex mesh ($S$), and the edge to remove ($A$)
Ensure: Removed adjacent faces to the removed edge
  Find out the $E$'s adjacent vertices ($V_1$ and $V_2$)
  Find out the $V_1$ and $V_2$'s adjacent vertices ($V_3$, $V_4$, $V_5$ and $V_6$)
  Remove adjacent mesh faces to $E$ ($F_1$ and $F_2$) using algorithm 2
  Verify if there is some special case, if so, proceed to normalize the edges, so there is
  not a triangle adjacent to the edge $E$
  Remove $E$
  Add the two new edges ($E_1$ and $E_2$)
  Modify the opposite faces to $E$ ($F_3$ and $F_4$)
  Remove $V_1$, $V_2$, $E_3$, $E_4$, $E_5$ and $E_6$
```

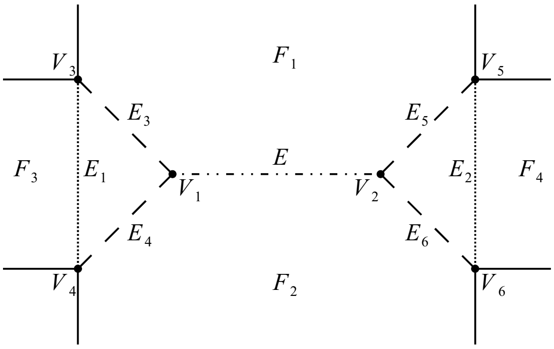


Fig. 1: Notation used in algorithm 1, F_1 to F_4 are the faces involved in the process, E the edge to be removed, E_1 and E_2 the new edges to be added, E_3 to E_6 the edges that need to be removed, V_1 and V_2 the vertices to be removed and V_3 to V_6 the vertices involved in the process but not removed

Algorithm 2 checks if the connectivity is preserved between two mesh's edges. This algorithm is used whenever an edge is removed from the mesh (it cuts from the geometrical model) to verify if the iterative process stops. A model who has been split in two parts will generate two different meshes and each one will be independently iterated.

Fig. 2 shows the edges where it is necessary to check the connectivity (these edges are E_1 and E_2). If both edges are conected then there is a path between them. For cheking this path one vertex from the edge E_1, say V_2, is chosen. Then the algorithm runs going in the next order: to the edge E_3, vertex V_3 is chosen, go to the edge E_4, to vertex V_4, to the edge E_5, to vertex V_5 and, finally, to the edge E_2, therefore a path exists and E_1 and E_2 are connectted.

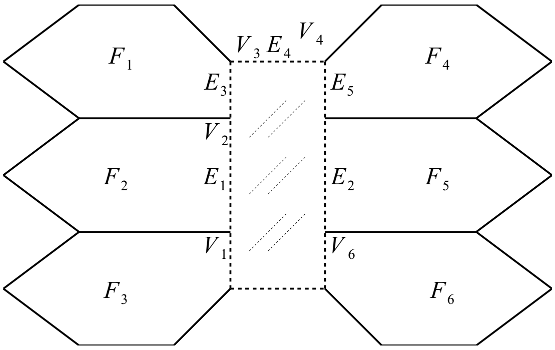


Fig. 2: Following edges to know if the mesh is already split

Algorithm 2 Verify the connectivity between edges

```
Require: Simplex mesh's edges ($S^E$) and the two edges to test ($E_1$ and $E_2$)
Ensure: Verify if the mesh was split in two parts
  Set $E$ to edge $E_1$
  Choose a vertex, $V_1$, from $E$
  repeat
    Find the opposite vertex from $E$, $V_2$
    Find the adjacent edge to $V_2$, which does not have a face and is not $E$
    Set this edge to $E$
    Set $V_2$ to vertex $V_1$
  until Reach the edge $E_1$ or $E_2$
  if Reached the edge $E_2$ then
    The mesh was not split
  else
    The mesh was split
  end if
```

The algorithm 3 shows how the face between two meshes is deleted. Fig. 3 shows the notation used, and Fig 4 shows the mesh after the face has been removed, the mesh could be split in two separate meshes, at that point we use algorithm 2 to check this fact. Finally, if the mesh is already disconnected, we use algorithm 4 to compute the two new meshes.

Algorithm 3 Remove jointed face

```
Require: Simplex mesh ($S$, faces –$S^F$–, edges –$S^E$– and vertices –$S^V$–) and the face
    to remove ($F$)
Ensure: Removed face and one, or two disjointed, meshes
  Remove the face edges ($E_1$ and $E_2$) which are adjacent to empty faces (faces already
  removed before before)
  Use algorithm 2 to verify if the connectivity is preserved, using the new created edges
  if Connectivity was lost then
    Use algorithm 4 to get the new two meshes
  end if
```

The algorithm 4 shows how we get two disconnected meshes from a single one. This algorithm takes a complexity order of $O(nm)$, where n is the number of faces in the original mesh (S^F) and m is the mean edges in each face (once every time approximately 6) because it processes each face and edge.

Three special cases must be treated separately for the case when a edge is deleted. These cases always involve triangles in the mesh (see Fig. 5), and the algorithm collapses in triangles until it finds a configuration without them, or get an empty mesh.

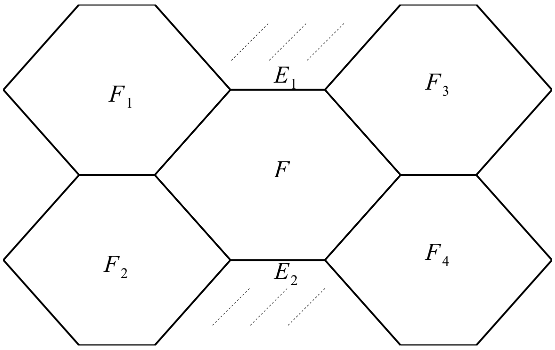


Fig. 3: The mesh with the face to be removed (edges E_1 and E_2 will be deleted also)

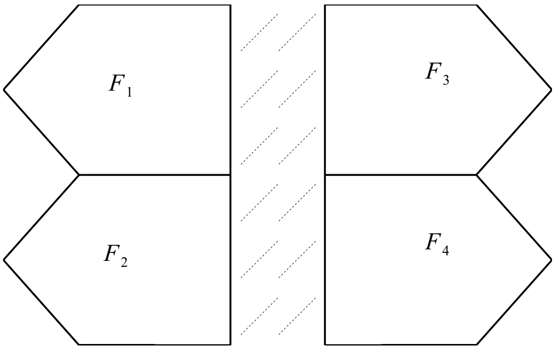


Fig. 4: The mesh without the deleted face

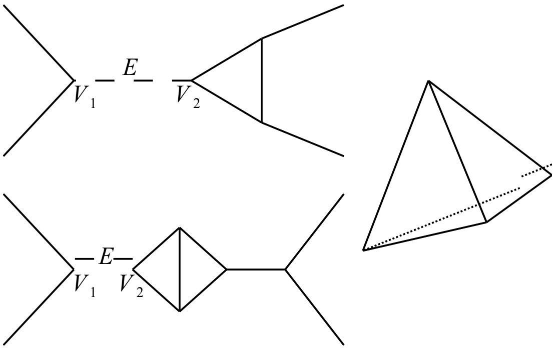


Fig. 5: The three special cases in the algorithm to remove an edge

Algorithm 4 How to get two disjoint meshes

```
Require: Simplex mesh ($S$, faces –$S^F$–, edges –$S^E$– and vertices –$S^V$–)
Ensure: Compute one or more disjoint meshes
  Let $L^F$ be a face's empty set, $L^E$ an edge's empty set, and $L^V$ a vertex's empty set
  Set all faces of $S^F$ as non-processed
  while There are non-processed faces do
    Borrow the next non-processed face, $F_i$, from $S^F$
    Put face in a new face list, $L^F_{F_i}$, do the same with its edges, $L^E_{E_i}$, and its vertices, $L^V_{V_i}$
    Put face in the processing queue, $P_f$
    while The processing queue, $P_f$, is not empty do
      Get the next face, $F_j$, from the processing queue, $P_f$
      for all Adjacent face to $F_j$, that are not in any list ($F_{j,k}$, edges, $E_{j,k}$ and vertices, $V_{j,k}$) do
        Put $F_{j,k}$ in the $F_j$ list ($L^F_{F_j}$)
        Put $E_{j,k}$ in the $E_j$ list ($L^E_{E_j}$)
        Put $V_{j,k}$ in the $V_j$ list ($L^V_{V_j}$)
        Put $F_{j,k}$ in the processing queue, $P_f$
      end for
    end while
  end while
```

4 Experiment cutting a cylinder

We used a cylinder to test our new operators, this cylinder is show in Fig. 6. The cutting process starts by removing the first edge from the cylinder's central strip (see the upper left of Fig. 7), the algorithm 1 is applied to delete the edge and refresh the vertex coordinates and the configuration of the affected faces.

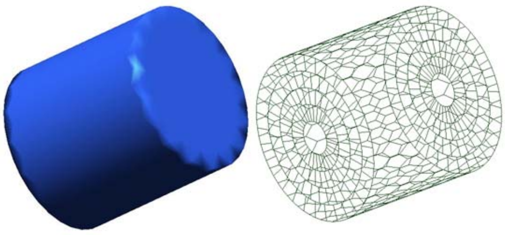


Fig. 6: Original cylinder

Once the first edge has been removed, we proceed to remove the second edge (see upper right of Fig. 7), here two faces were removed and one of them is adjacent to the intended edge, so this fact must be taken into account to avoid delete the non-existent face. Also, the corresponding vertices have one less adjacent face and it is necessary to take care of that, indicating that the face is no more in its list of neighbor faces. In Fig. 7 we can see one thin central section produces by the cutting process, this is a consequence of the process itself and, also, of the vertex moving to the barycenter of its three neighbors.

The process goes ahead, deleting edges until the last one from the central strip must be removed; then the mesh clearly must be split in two parts, as can be seen in lower left Fig. 7.

Once the mesh has been cut in two meshes, it is possible to independently manipulate both (see lower right of Fig. 7).

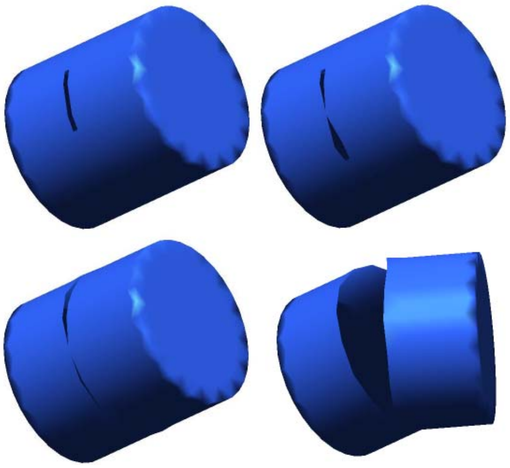


Fig. 7: The cutting process

5 Deformation model with rupture and second experiment

To deform a simplex mesh we attach a simple mechanical system composed by a mass, a spring, and a dashpot, to each mesh edge. This system is represented by Eq. ((1)):

$$m\ddot{x} + b\dot{x} + kx = f_{\text{ext}} \tag{1}$$

where m is the mass, b is called damping coefficient, and k is the spring stiffness coefficient. Eq. ((1)) is solved numerically to obtain the elongation x by using the finite differences method, because it takes less operations and its result is enough for our purposes. The system is in a steady stable at rest, so there are not external (f_{ext}) or internal forces applied to it.

This elastic model represented by Eq. (1) models a perfectly elastic material, i.e., a material whose graph of elasticity is linear. We develop a non-linear model using a completely inelastic material, with one segment linear, another exponential, and if eventually the force is too much, the material breaks. The linear model is represented $f_{\text{int}} = k \cdot x$, where f_{int} is the internal force stored in the spring. The non-elastic model is:

$$f_{\text{int}}(x) = \begin{cases} k \cdot x & \text{if } x \leq l_1, \\ k \cdot l_1 + e^{\gamma(x-l_1)} - 1.0 & \text{if } l_1 < x \leq l_2, \\ 0 & \text{if } l_2 < x. \end{cases} \tag{2}$$

where γ is a parametric value (if we want a soft transition between the linear and exponential parts γ must be equal to k, so the first derivative in the point where $x = l_1$ would be equal), l_1 is the desired elongation limit (which depends on γ) and l_2 is the elongation at which the edge breaks.

At Fig. 8 we show, when the value of $\gamma = 0.05$, how the l_1 affects the maximum elongation the element can take (the model parameters are $m = 0.01$, $d = 0.01$, $k = 0.01$, $f_{\text{ext}} = 0.01$), as we can see with this value of γ the maximum elongation is effectively limited, but, is greater than l_1, if we increase γ this value would be closer to l_1.

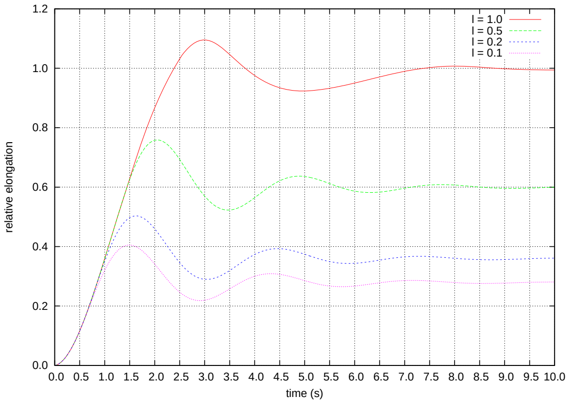


Fig. 8: Comparing the deforming limits at different values of l_1

The break limit (l_2) is defined as a factor that represents the maximum length that an element can have without rupture, at that length the element breaks; that factor represents the maximum stress the element can sustain in the point of rupture, above which the internal force is zero.

The second experiment is apply an internal force, as an internal pressure, to a deformable sphere until it breaks. The sphere used is shown in Fig. 9. In Fig.e 10 we can see the sphere when it breaks and the deforming process is stopped. We can see when the applied force deforms the model more or less in a symmetrical way, except at six rectangles located in the middle strip of the sphere.

The external force in each vertex is computed based on the influence area of it, as shown in the Eq. ((3)), where ρ is a constant which represents the inside

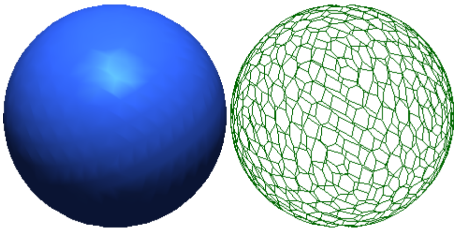


Fig. 9: Original sphere model

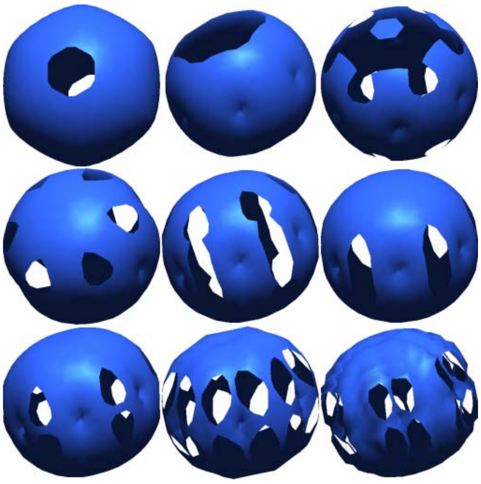


Fig. 10: The model when the internal pressure breaks it.

pressure, $\mathcal{A}_v$ is the influence area of the vertex v and $\mathcal{A}_T = \sum_{\forall v} \mathcal{A}_v$. Each value of ρ yields a different breaking pattern, as shown in Fig. 10, where the values are 8.7, 15.2, 16.2, 16.4, 17.3, 17.4, 23.0, 50.0 and 150.0, respectively.

$$f_{ext} = \rho \frac{\mathcal{A}_v}{\mathcal{A}_T} \tag{3}$$

6 Conclusions and future work

We have developed three new operators that allow cut volumes built with a simplex mesh. The first operator computes if a simplex mesh has lost its connectivity, the second operator split a simplex mesh in two halves, and, the third operator deletes an edge from the mesh and their corresponding faces.

The way we compute the simplex mesh connectivity has an acceptable computational complexity level, because we only check if it is keep between the two new added edges, so it only has to check a small set of edges. The split operator case needs to operate over all the simplex mesh, because we have no information about the faces connectivity, only its adjacency, so we could not optimize it well, but we are trying to develop a new algorithm to directly split the mesh when removing the edge, without travel through all the faces.

The described operators have deficiencies (the cutting operator needs to take care of the face's and vertices' ordering, the connectivity operator needs to operate in edges with faces removed, and the split operator needs only a performance boost), over which we are already working and there is hope we get a complete definition of them, so they could operate in whatever simplex mesh we throw at it.

Our non–elastic deformation model has a good behavior as we expected, when it is applied to a surface simplex mesh.

As future work, we will extend operators to cut volumetric models based also on simplex meshes, so they can operate in this new domain, and, apply this same elastic model to it to see if it fits nicely in such structure or needs considering some other features to shows a well behavior.

We are thinking in apply the developed work to visualize and to simulate human organs in surgery processes.

References

1. H. Delingette. General object reconstruction based on simplex meshes. *International journal of computer vision*, 32(2):111–146, September 1999.
2. H. Delingette and N. Ayache. Hepatic surgery simulation. *Communications of the ACM*, 48(2):31–36, February 2005.
3. J. Montagnat, H. Delingette, and N. Ayache. A review of deformable surfaces: topology, geometry and deformation. *Image and vision computing*, 19(14):1023–1040, December 2001.
4. Jiuxiang Hu, Anshuman Razdan, Gregory M. Nielson, and Gerald E. Farin. Improved geometric constraints on deformable surface model for volumetric segmentation. *IEEE Geometric modeling and processing*, 04:237–248, 2004.
5. H. Delingette and J. Montagnat. General deformable model approach for model based reconstruction. *IEEE international workshop on model-based 3D image analysis*, 98, January 1998.
6. H. Delingette and J. Montagnat. Shape and topology constraints on parametric active contours. *Computer vision and image understanding*, 83(2):140–171, September 2001.
7. J. Montagnat and H. Delingette. Volumetric medical image segmentation using shape constrained deformable models. *Proceedings of computer vision, virtual reality and robotics in medicine*, 97:13–22, March 1997.
8. J. Montagnat and H. Delingette. Globally constrained deformable models for 3d object reconstruction. *Signal Processing*, 71(2):173–186, December 1998.
9. G. Hamarneh, H. Delingette, and M. Henkelman. 3d segmentation of mouse organs from mr images using deformable simples mesh models. *Proceedings of the International Society of Magnetic Resonance Medical*, page 779, 2003.
10. Stéphane Cotin, Hervé Delingette, and Nicholas Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. *Visual Computer journal*, 16(8):437–452, 2000.

Skull Fractures Detection by Finite Element Method

Victor Ortiz[1], Cornelio Yáñez[1] and Isaac Chairez[2]

[1] Center of Computing Research, Av. Juan de Dios Batiz, esq. Miguel Othon de Mendizabal
s/n Col. Nueva Industrial Vallejo, Mexico D.F., 07738, Mexico
vh.ortiz@ieee.org, cyanez@cic.ipn.mx
[2] UPIBI, Av. Acueducto de Guadalupe, s/n, Col. Barrio la Laguna Ticoman, 07340, Mexico
D.F., Mexico
jchairez@ctrl.cinvestav.mx

Abstract. The opportune detection of skull fractures may be determinant to save a patient life when he or she is affected by any head traumatism. In this paper, an algorithm to detect skull fractures using the method of finite element is developed. The suggested method was applied to divide the image in specific regions which define a particular structure in the skull. In this case, the proposed technique is available to determine small irregularities along the skull structure, using just the information provided by the mathematical support of the finite element structure. Likewise, it should be noticed the importance to acquire some particular information on certain areas of the image segmentation. The suggested method has been successfully applied to detect small fractures and to solve the hard-task to avoid any incorrect diagnosis.

1 Introduction

Usually, strong blows in the head provoke structural changes on the skull structure at the impact point. In several occasions, small objects can penetrate the skull and producing a local laceration of encephalon. If big objects hit with great force can introduce some pieces or bone fragments into encephalon in the impact site. These reasons lead to develop an adequate method to realize an opportune detection of those types of fractures, particularly because if they are not adequately diagnosed, the patient could dye [1].

1.1. Fracture types.

In general, the fractures can be classified in:

Linear fracture. This kind of fractures is characterized by an elastic deformation in the skull and represents the 80% for all of them. In general, these fractures do not require a specific treatment; however, some special cares must be taken depending on the intensity of skull traumatism.

Fracture with collapse. In this class of fractures, there is a depression in the skull structure.

Received 03/07/07
Accepted 19/10/07
Final version 20/10/07

Simple or closed. This appears when the hairy leather that covers the fracture remains intact.

Composed or opened. This class of fractures happened when the hairy leather is lacerate and represents 80% of the collapsed fractures. According to its cause and aspect, it type could be sub-classified in: armor-piercing, penetrating associated to linear fractures or comminutes. Also, they can be associated to lacerations of duramatter that constitutes a front door for the infection. Mostly of cases require debrillation and surgical elevation.

The progressive fractures or, badly called, leptomeníngeos cysts (better pseudomeningoceles), take place by a duramatter's defect under a cranial fracture, with the consequent risk of cephalon-raquideum liquid droop out and, sometimes, cerebral weave. They are not common, around 0.6% to 1% of the fractures, being more frequent while the patient is young. Habitually, they require surgical attention.

Fractures of frontal bone. - They take place by severe traumatism on the frontal skull region. The frontal sine can be jeopardized, and if the later wall of the sine is fractured, injury of duramatter and of nasofrontal conduit can also exist [2].

1.2. Segmentation of Images.

The segmentation is the process where an image is spread out in regions, components, parts or objects. Whatever adopted segmentation definition, the main idea behind this process, is to isolate different objects in the image that should be recognized. It is clear the result of this process compromise the performance of the analyzing system. A bad segmentation will cause bad object recognition within the image bounds. On the other hand, a good segmentation will provoke that any possible automatic system for objects recognition (ASOR) gives good results. A formal way to define the segmentation process is the following one [3]: The segmentation is the image $f(x,y)$ dividing process. Each one of these subdivisions is called regions, which are obtained in such a way each one of these sub-images represents a complete object within the image limits, that is, there exists *n* regions $R_1, R_2, \ldots, R_n$ such that the following conditions are fulfilled:

a) $\bigcup_{i=1}^{n} R_i \subseteq f(x,y)$

b) $R_1, i = 1, 2, \ldots n$ is connected

c) $Ri \bigcap Rj = \varnothing, \forall i, j, i \neq j.$

d) $P(R_i) = TRUE, i = 1, 2, \ldots, n.$ Each R_i satisfies a predicate with some set properties. All elements of each R_i share a given properties set.

e) $P\left(R_i \cup R_j\right) = FALSE$ for $i \neq j$. This means that a specific pixel belonging to adjacent regions can not be at the edge. Otherwise these points will be considered like an independent region.

1.3. Finite Element Method.

This method is based on dividing the body, structures or dominion (average continuous) on which are defined certain integral equations characterizing the physical behavior for a selected problem in a series of non-intersecting sub-dominions to each other, denominated finite elements. The set of finite elements forms a partition of the dominion also denominated discretization [4]. Within each element, a series of representative points called nodes is distinguished. Two nodes are adjacent if they belong to itself finite element, in addition a node on the border of a specific finite element can belong to several elements. The set of nodes considering their relations of adjacencies is called mesh. The calculations are made on a mesh or discretization created from the dominion generating some special meshes distributions (this is done in a previous stage to the calculations that denominate pre-process). In agreement with these relations of adjacencies or connectivity, the value of a set of unknown variables defined in each node is related and denominated degrees of freedom. The set of relations between the values of variable determining between the nodes can be written in form of system of linear equations (or linearized), where the matrix of this system of equations is called matrix of elasticity of the system. The number of equations of this system is proportional to the number of nodes [5].

2 Methodology

The mathematical description of the procedure to be followed can be summarized as follows: The vectors of image characteristics used in this work are profiles of derivative functions. They are vectors whose components are directional derived which are calculated in points of the image where those points are on a straight line segment. In order to reduce the effect of the noise in the images, it is advisable to apply them a filter. Thus, E represents the function of intensities of the image and G_σ is a Gaussian filter. The directional derived from the filtered image $G_\sigma * E$ in the point $\left(x, y, z\right)$, in the direction of the unitary vector n, is given by:

$$D_n\left(G_\sigma * E\right)\left(x, y, z\right) = n'\nabla\left(G_\sigma * E\right)\left(x, y, z\right) \tag{1}$$

The Gaussian filter is described like:

$$G_\sigma\left(i, j, k\right) = c \cdot e^{-\frac{\left(i^3 + j^3 + k^3\right)}{3\sigma^3}} \tag{2}$$

Where:

σ = standard deviation. The constant c one calculates so that the coefficients add. The greater σ, the smoothness property is increased. In the vertex v_i of a mesh, the normal unitary outside calculates n_i to the mesh and other unitary vectors are selected conformer a same angle with n_i. This creates an uniformly distribution around n_i, in order to obtain a set N_v of directions.

Secondly the image is segmented. In this case we chose the method of Canny. The operator of Canny edges is derived from the Gaussian filter. This operator approximates the segmentation operator strongly, optimizing the product of the quotients signal to noise and location [6].

Thirdly we began the Method of the Finite Element. Considering a closed enclosure the steps for the resolution are [7]:

- Divide the enclosure in Finite Elements: Triangles (3 nodes), Tetrahedrons (4 nodes), etc.
- Deduce the equation describing the potential *f* within an EF.
- Raise the equations giving the adjustments conditions for solutions in the EF borders.
- Calculate the potentials in the nodes of each EF by means of some of the methods that will be introduced below.
- Solve the raised algebraic equations. Generation of the finite Elements.
- Contours can be irregular
- EF will be as small as the programmer considers.
- If the potential varies a lot, the EF will consider a mesh with small "holes". It means the nodes are closer.

We propose the Energetic Function.

$$E = \underbrace{\sum_{i,j}\left[p_{i,j,k} - x_{i,j,k}\right]^2}_{\text{data fidelity term}} + \underbrace{\lambda\left(\sum_{i,j,k}\sum_{i_s,j_s,k_s}\left(x_{i,j,k} - x_{i_s,j_s,k_s}\right)^2\right)}_{\text{enforces smoothness}} \tag{3}$$

Where:

$p_{i,j,k}$ is a voxel value from the original image.

$x_{i,j,k}$ is the classification of that voxel.

$i_s j_s k_s$ represent the neighborhood of the voxel.

The energy is a numerical value representing a weighted summation of two main distances to the voxels taking part in the segmentation function of intensities [8]. In this study, it is desirable to make a detailed registry for all images and considering the derivates form each image structure. This can be solved by the definition of vector field that can vary point to point in the images. In this case, the energetic function can be redefined as [9]

$$E=\sum_{i,j}\left[p_{i,j,k}-x_{i,j,k}\right]^2+\lambda\left(\sum_{i,j,k}\sum_{i_s,j_s,k_s}\left(x_{i,j,k}-x_{i_s,j_s,k_s}\right)^2\right)$$

$$+\left[\lambda_x\ \lambda_y\ \lambda_z\right]\begin{bmatrix}\int_\Omega\left|\nabla u(p)\right|^2 d\Omega\\ \int_\Omega\left|\nabla v(p)\right|^2 d\Omega\\ \int_\Omega\left|\nabla w(p)\right|^2 d\Omega\end{bmatrix} \tag{4}$$

And where the updated coordinates in the resulting image are:

$$E=\begin{bmatrix}\hat{x}\\ \hat{y}\\ \hat{z}\end{bmatrix}=\begin{bmatrix}x+u(x,y,z)\\ y+v(x,y,z)\\ z+w(x,y,z)\end{bmatrix}$$

$$=\begin{bmatrix}x+\sum_{i=1,j=1}^{m}N_{i,j,k}(x,y,z)u_{i,j.k}\\ y+\sum_{i=1,j=1}^{m}N_{i,j,k}(x,y,z)u_{i,j.k}\\ z+\sum_{i=1,j=1}^{m}N_{i,j,k}(x,y,z)u_{i,j,k}\end{bmatrix} \tag{5}$$

Where the interpolation or form functions $\left(N_{i,j.k}(x,y,z)\right)$ are traditionally used by the method of finite elements for rectangular meshes [10]. In general, *m* is the number of total nodes depending on the number of nodes forming each element and the total number of elements which defines the mesh. Since the dominions in the images are generally of parallel sides, we have used regular lagrangians elements of 8 nodes for 3 dimensions and 4 nodes for 2 dimensions. The interpolation functions can be written easily, based on the image space. By simplicity we are going to define the system coordinate in 2 dimensions within an element anyone as it is in Figure 1 [11].

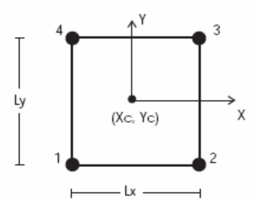


Fig. 1. - Definition of the system of coordinates.

To avoid the separate computation of the forces *E*, the elastic deformation, and the matching criterion, we propose to directly compute a deformation field that readily satisfies both aspects, the elasticity constraint and a local image similarity constraint between the images to be matched (I_1 and I_2). Hence, the total energy to be minimized is expressed as:

$$E = \int_{\Omega} \lambda^{t} d\Omega + \int_{\Omega} \left(I_1\left(x + u\left(x, y, z \right) \right) - I_2\left(x, y, z \right) \right)^2 d\Omega \qquad (6)$$

3 Results

The Figure 2 shows 3 different images of people who present anomalies and that can present fractures that cannot be, so easily detected.

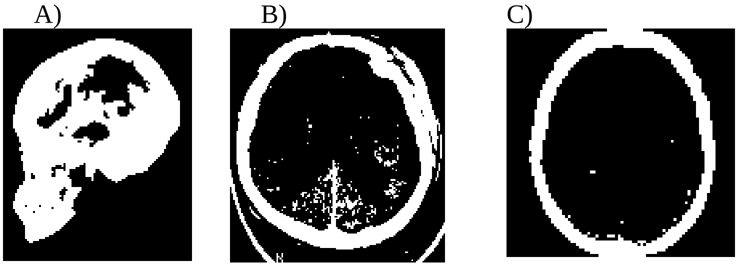


Fig. 2 Three types of encephalon analyzed by Computed Axial Tomography

In order to eliminate the image noise generated by the acquisition, in addition to initialize with the vertices of the finite element, a Gaussian filter was applied with $\sigma = 10$ (figure 3).

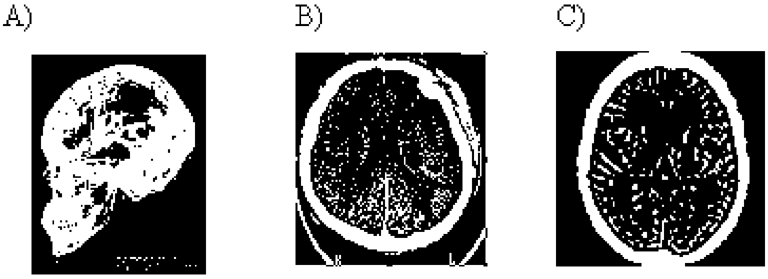


Fig. 3 Images of Skull, with Gaussian filter

To observe some details of the filtered images, an edge canny detection was used as it was stated above. Under this analysis, some relevant aspects for each figure were observed. For example, lines, among others (Figure 4).

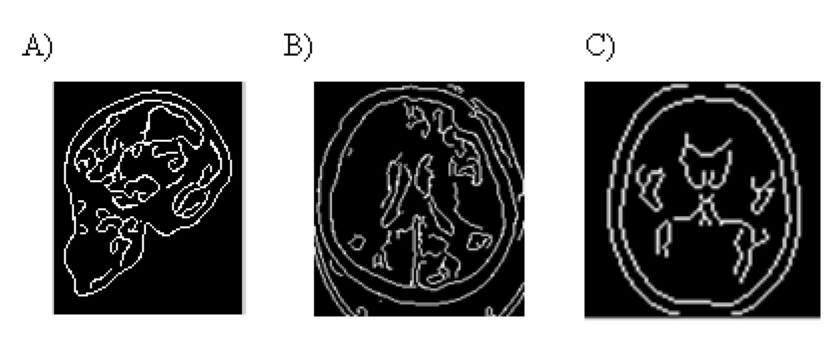


Fig. 4 Images of Skull segmented.

Also, it was applied an alternative method to show the vectors at the edge of the images (equation 1). These vectors can be helpful to design an adequate mesh to initialize the Finite Element procedure [9]. (Figures 5 y 6)

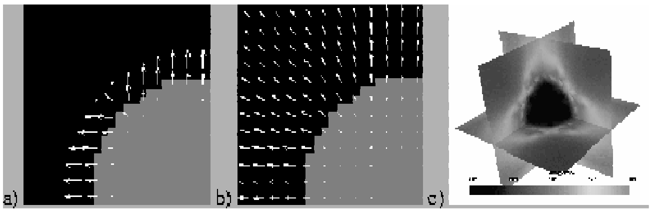


Fig. 5: Growing sphere. a) and b): close-ups of 2D cuts through 3D image with a) classical OF, and b) FE matching deformation fields overlaid, c) and 3D orthogonal cuts through the FE mesh with intensity coding of the displacement field. The displacement field is mainly located at the boundaries of the sphere and is propagated through the surrounding elastic medium.

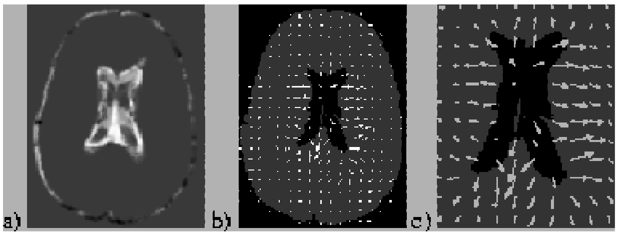


Fig. 6: Enlarging ventricles. a) slice of difference between segmented images at both time points (gray means no difference), b) deformation field superimposed on same image at the first time point. c) close-up

In the following figures (7 and 8), it is showed the surfaces of the finite element solution. These surfaces are composed by 8 nodes and 9 elements of nodes, for the first and 16 nodes and 18 elements for the second one representing a particular zone where a fracture is located.

.

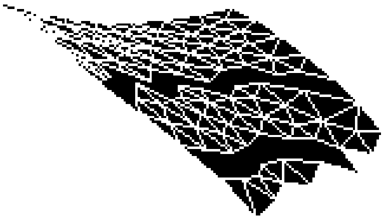


Fig. 7. Surface 1 of the Finite Element

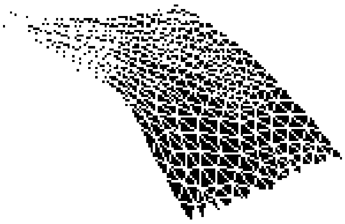


Fig. 8. Surface 2 of the Finite Element.

Once the adequate mesh (by trial and error method) has been applied to solve the finite element, it is possible to observe small details in the images that can be considered fractures (confirmed by a-priory knowledge of where the fracture is). This could be an interesting method to help physicians without experience that are not experts to detect the fractures form the magnetic resonance or computed axial tomography (Figure 9).

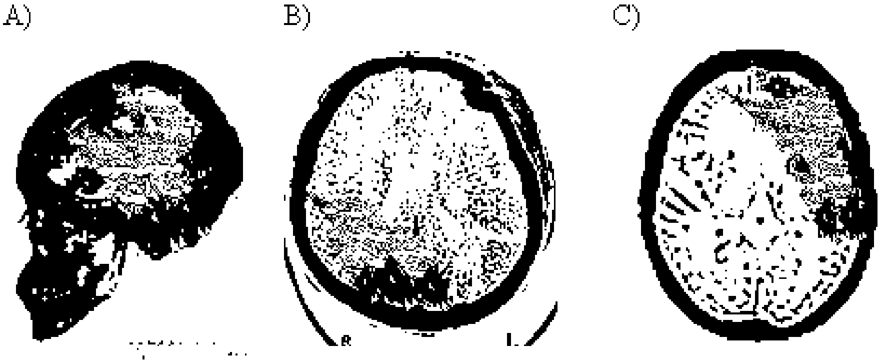


Fig. 9. Images with surface 1 of the Finite Element

With the second surface that has a large number of nodes and therefore a “better” mesh, it is possible to appreciate similar fractures (Figure 10).

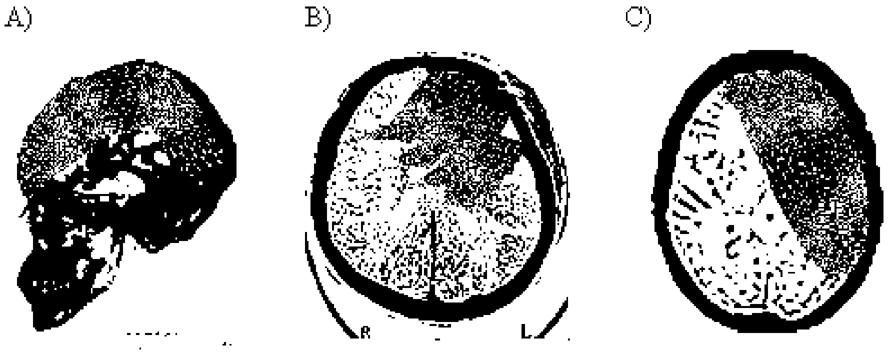


Fig. 10. Images with surface 1 of the Finite Element.

Here it is clear that a method to determine the adequate amount of nodes and the elements form must be designed. Some interesting approaches can be used to realize this aspect: the neural networks theory, the genetic algorithms, etc.

3 Conclusion

This paper presents a new, physics-based deformable model for tracking physical deformations (Fractures) using image matching. The model results from the minimization of a deformation field simultaneously satisfying the constraints of an elastic body and a local image similarity measure. The model provides a physically realistic deformation field and also allows us to inspect the characteristics of the deformed objects. This can be very useful for the inspection of stresses induced by the deformation of certain objects on their surroundings, in our case the detection of skull fractures.

In the experiments, the objects were considered to be homogeneous elastic bodies. Further improvements of the algorithm include the assignment of different elasticity's to the different objects represented in the image. This will require a preliminary segmentation of the objects to be deformed so as to be able to set appropriate elasticity coefficients to every cell of the mesh. Also, the anisotropy of certain skull fractures could be included into the model by modifying the elasticity matrix *D* appropriately.

We would lack to design an algorithm that assures that the meshes are the best possible one (in the sense of the image representation), also to include one better area within the image, providing an improvement in the diagnostic supplied by the method suggested in this manuscript.

References

1. Snell, R.. Clinical Neuroanatomy for Medical Students, Lippincott-Raven Publishers, 4th Edition (2001)
2. Hauser, K. Longo, B., Jameson, F.. Harrison's Principles of Internal Medicine, Mc Graw Hill. 16th Edition (2006)
3. Gonzalez, R. C., Woods, R.E.. Digital Image Processing. Prentice Hall, Inc.. New Jersey (2002)
4. Felippa, C.A.. Introduction to finite element methods. Department of Scientist, Engine Aerospace Colorado University. Boulder, http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html.
5. Ledesma, A. The Method of Finite Element Foundations for Elliptical Problems. UNAM (2006)
6. Sonka, M., Hlavac, V., Boyle, R.. Image Processing, Analysis, and Machine Vision. THOMSON, USA (2006)
7. Passaro, A., Junior, J., Abe, N., Sasaki, M., Santos, J.. Finite-Element and Genetic Algorithm Design of Multi-segmented Electro-optic Sensor for Pulsed High-Voltage Measurement, 6th World Congresses of Structural and Multidisciplinary Optimization (2005)
8. Bribiesca, E., Measuring 3D shape similarity using progressive transformations. Pattern Recognition, vol. 4128, pp. 1117--1129 (1996)
9. Yushkevich, P., Piven, J., Cody, H., Ho, S., Gerig, G.. User-Guided Level Set Segmentation of Anatomical Structures with ITK-SNAP. MICCAI Workshop, (2005)
10. Brook,s R.. Model Based 3-D Interpretation of 2-D Images. IEEE Transaction Pattern Analysis Machine Intelligence, vol. 5, no.2, pp 140--150 (1983)
11. Galvez, J.M., Canton, M.. Normalization and shape Recognition of three-dimensional objects by 3-D Moments. Pattern Recognition, vol.26, no.5, pp.667--681 (1993)
12. Smith, S.. BET: Brain Extraction Tool, Technical Report. UK (2001)
13. Hu, M.K.. Visual Pattern Recognition by Moment Invariants. IEEE Transaction in Information Theory, vol.8, pp.179--187 (1962)

Knowledge Engineering Issues for Footprinting Feature Extraction

Rosales Peña Alfaro, Eric Manuel

Graduate Section
UPIICSA, IPN
Mexico City, Mexico
emrosales@ipn.mx

Abstract. This paper addresses knowledge engineering methodological issues concerning the integration of image processing techniques into a framework module for a foot classification expert system. The main problem is not just to build-up a module based on existing techniques but to test them on the new domain application so to produce a reliable and robust system. The paper introduces general issues concerning the process of knowledge engineering; it then presents results from the application of the methodology to implement a foot classification expert system based on image processing techniques. Results and experiments are described to demonstrate those issues mentioned.

1 Introduction

Research on image processing has developed a lot of techniques and algorithms, from image filtering process to image understanding methodologies [1–5]. However, when trying to integrate them into a single knowledge-based architecture some insight about knowledge domain has to be taken into consideration so to produce a reliable and robust system [6]. An example of an integration problem is the main-features extraction from footprinting. This paper addresses the main issues concerning the methodological aspects concerning the knowledge engineering process to deploy a foot classification system for the National Polytechnic Institute.

At the National Polytechnic Institute, a general medical diagnosis is carried out each year for incoming students at high school and superior levels. This diagnosis includes not only general health but also the determination of the foot type so to recommend proper therapies. Especially for students, who want to dedicate part of their studies to practice some sport, or those, who once finished their educational formation, will be dedicated to activities that require them to walk or to be stood most of the time, it is important to determine if they are physiological fit to perform those activities. Thus, an initial research was carried out to determine the best way to perform the foot classification considering the high volume of cases to be diagnosed. The conclusion was to build-up an automated system able to proper classify the footprint of approximately 40 thousand students. The next step was to propose a formal solution, which was the development of a computerized processing system to record, manage and perform the classification process with a minimum time response. The system was divided into two sub-systems:

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 29, 2007, pp. 110-119

Received 29/06/07
Accepted 19/10/07
Final version 24/10/07

the management sub-system, which enables recording of digitalized footprints and the foot classification sub-system. The first one was treated as a normal information system facility applying conventional software engineering techniques while the second sub-system motivate a more in-depth research in which knowledge engineering techniques were applied [7].

The determination of the foot type implies a three-step process: firstly a podography or footprint is produced by means of walking through an inked paper; secondly the specialist looks at the footprint and mentally marks feature points on the podography, and finally those feature points are mentally measured and compared against foot-type parameters [8–11]. As a consequence, the research focused on devising a knowledge-based system, which extracts the feature points from the digitalized podography and performs the comparison against the foot-type parameters.

The paper summarizes the knowledge engineering process to determine the values of the foot-type parameters and the process to extract feature points on the podography. Some image processing techniques are described showing the suitability of use for the described problem. The next section describes the overall architecture of the foot classification system, explaining in detailed the image processing module. Some experiments are then described to show the reliability of the image processing module, which is expected to functionally work on the overall system in a near future. Finally some conclusions regarding future work are drawn.

2 Knowledge Engineering Methodology

The process of building-up a knowledge-based system implies two main phases: the knowledge acquisition and the knowledge modeling phase. Literature presents these two processes as part of the **KADS** methodology, which guides the deployment of knowledge-based systems [6, 12, 13]. Modern approaches differ on the description of the methodology in a way that the design phase is interpolated from the knowledge acquisition process to the knowledge validation one [6]. The present work was carried out following the second approach due to its flexibility to build not only expert systems but generally knowledge-based systems and figure 1 shows the knowledge engineering methodology.

2.1 Knowledge acquisition

As a result of reading the literature on orthopedics and following a series of interviews with a human expert, the knowledge elicitation phase produced an understanding that there are mainly three types of foot among students aged 15 to 18: normal, cave and flat foot [8–11]. The knowledge acquisition phase concluded with the steps to ”scientifically” classify the foot into one of the three types. The steps are described as follow:

1. Identify the fifth metatarsial contact point (external contact), call it A,
2. Identify an external heel contact point, call it B,
3. Join points A and B to generate the line $\bar{AB}$,
4. Identify the narrowest arch zone, and mark a point C over the lateral side,
5. Trace a line from C to $\bar{AB}$, obtaining the longitudinal lateral arch,

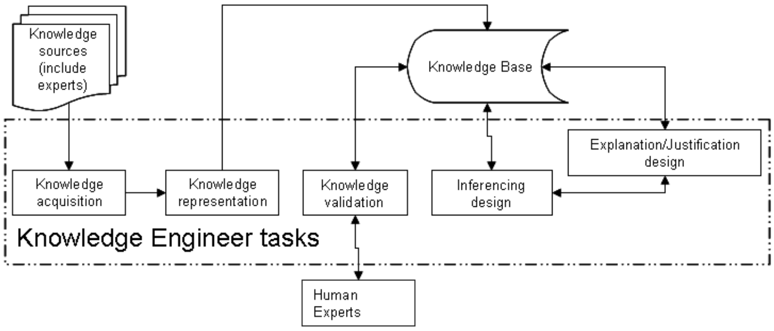


Fig. 1: Knowledge Engineering Methodology adapted from [6]

6. Identify the first metatarsial contact point (internal contact), call it D,
7. Identify an internal heel contact point, call it E,
8. Join points C and D to generate the line $\bar{CD}$,
9. Mark a point F over the internal side of the narrowest arch zone,
10. Trace a line from F to $\bar{CD}$, obtaining the longitudinal medial arch,
11. Measure the longitudinal lateral and longitudinal medial archs, and
12. Compare the measures against the foot-type parameters

The points and arcs are depicted in figure 2. The foot-type parameters were defined by the human expert as shown un table 1:

Table 1: Foot-type Parameters.

PARAMETERS	NORMAL FOOT	FLAT FOOT	CAVE FOOT
longitudinal lateral arch	5 mm.	>5 mm.	<5 mm.
longitudinal medial arch	$[15, 18]$ mm.	<15 mm.	>18 mm.

As a conclusion of the knowledge acquisition phase the foot classification method can be divided into two main process: a) the feature extraction, and b) the comparison process.

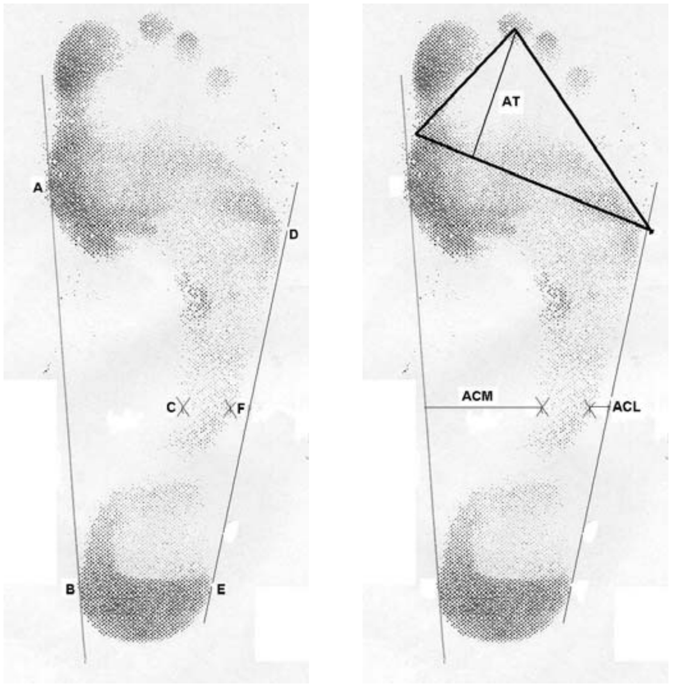
2.2 Knowledge representation

For the second process knowledge can be represented by a production system containing three simple rules:

1. If $LLA < 5$ and $18 < LMA$ then foot is a cave foot
2. If $LLA = 5$ and $15 \leq LMA \leq 18$ then foot is a normal foot
3. If $LLA > 5$ and $LMA < 15$ then foot is a flat foot

Where: LLA is the Longitudinal Lateral Arch and LMA is the Longitudinal Medial Arch.

Because the division of the method into process, some image processing techniques need to be applied. The next section revises some of the image processing techniques available for this problem.



(a) Characteristic points

(b) Arches

Fig. 2: Points needed to determine arches

3 Image Processing Techniques

Image processing comprises the set of algorithms intended to transform the input image in an understandable image. Several processes and techniques have arisen as a result of research and most of them have shown efficacy on getting good results. A common image process can be composed by the following tasks: a) the detection of a threshold value to differentiate the foreground from the background or to segment different regions of the image; b) edge detection to take away foreground pixels leaving only the contour of the objects; and c) feature extraction from edges, which depends on the domain knowledge and the application to be deployed.

3.1 Threshold Detection

This process is used when a thresholding operator needs to be applied and no threshold value is known. The thresholding operator selects pixels that have a particular value. It is useful for finding objects within an image, when the separation brightness is known. The thresholding operator can also be applied for region growing [1], and for segmentation, which results in a set of disjoints regions uniquely corresponding to objects in the input image [4]. When the latter is not known the value is determined by some the following methods: a) p-tile thresholding; b) methods based on histogram shape analysis; c) histogram concavity analysis; d) entropic; e) relaxation methods; f) mutithresholding [4]; g) optimal thresholding [5]; and h) iterative threshold detection [4], which was applied in the present work. The method was implemented following Sonkas algorithm [4] and was used together with a gray-level to binary conversion.

3.2 Edge Detection

According to [1] boundaries of objects tend to show up as intensity discontinuities in an image providing the principal motivation for representing objects by their boundaries. These boundaries are usually called edges and they can be located by applying an edge operator. Among the most applied edge operators, the canny operator is one of the most useful [3–5] because it can be applied under noisy conditions. This edge operator optimally operates by complying with three criteria: a) detection, b) localization; and c) one-response [3–5]. It is a gradient-based operator and it was implemented following Parkers algorithm as described in [3].

It was reported in [7] that under certain images conditions, several edge-points were not being closer enough as to form a solid region, the operator just produced a blurred image. As this result was not useful for the purpose of just determining the edge of the foot, other approaches were considered. The solution to the blurred image problem was found on the work of Rothwell et al about topological geometry, in which edge operators are defined following three properties: a) geometric information of the image to be processed; b) curve fitting to optimally guarantee a good edge; and c) recover of scene topology [14]. These findings made the work to focus on a more practical approach called digital morphology, which is a way to describe or analyse the shape of a digital object [3].

The main binary operator used was the dilator one followed by images subtraction. The dilation operator works over an image by enlarging objects belonging to the foreground as a result of applying a morphological structural element. The dilation operation can be formally defined, following [3] as: A dilation of a set A by the set B is $A \oplus B = \{c/c = a + b, a \in A \wedge b \in B\}$, assuming that A is the set of points defining an object on the image and B is the morphological structuring element. A more detailed description of the operator can be found on [3]. The implementation of the binary dilation follows [3] description.

From above follows the application of the subtraction operator to a couple of images: the result of dilating an image, images1 is subtracted from a double-dilated image, image2. Results in [7] showed that it was possible to produce an image in which only the edge of the objects appears.

It is shown that the iterative threshold detection, the conversion from gray level to binary and the edge detection using digital morphology provided a good means to foot image processing. Therefore, those were chosen to compose the architecture of the feature extraction module described in next section.

3.3 Feature Extraction

Feature extraction, in this case, is totally domain dependable; therefore, the procedure described before was implemented to find: firstly the characteristics points (A to G) and then three main arches. A first attempt was done by dividing the image in six uniform sections in which points might be found. After a careful revision, it was found that this approach did not considered all types of foot and thus, a second approach was attempted by dividing the image in eight uniform sections, which can be enumerated from 1 to 8 depending on the foot. This means if the foot being analyzed is the right one, section 1 will be allocated in the upper left square, and section 8 will be the lower right square; whereas, analyzing a left foot would implied that section 1 will be the upper right square and section 8, lower left one. Then, some assumptions need to be mentioned so the aforementioned procedure would be implemented: (i) Point A is located in section 1, at the nearest column from column 1; (ii) Point B is located in section 7, at the nearest column from column 1; (iii) Point C is located in section 5, at the column far away from column 1; (iv) Point D is located in section 2, at the nearest column from the maximum column of the image being analyzed; (v) Point E is located in section 8, at the nearest column from the maximum column of the image being analyzed; (vi) Point F is located in section, at the column far away from the maximum column of the image being analyzed.

It should be mentioned that the previous assumptions would work well on images clearly define and which comply with them. Experiments showed that the second approach improved the location of characteristics points. Measures of arches were computed applying basic concepts of analytical geometry such as distance of one point to a line.

4 Knowledge-based System Architecture

It was mentioned that knowledge based system give rise to not necessarily symbolic knowledge bases but usually to process of automatic data acquisition and small knowledge bases such as the one described earlier to classify digitalized foot images. In this way, an automatic foot classification system is composed by the following modules [7]:

1. A register keeping module which aims to register the digitalized images and serves to lunch the image processing module.
2. The image processing module aiming to extract the main features: the longitudinal medial arch and the longitudinal lateral arch. This module is composed by the following processes:
   (a) a format conversion from jpg to pgm, mainly to produce an image that can be processed faster;
   (b) the iterative threshold detector to find out the differential value between the foreground and the background;
   (c) an image conversion from gray level image to binary image as a means of decrementing the processing time;
   (d) the edge detection phase, which was further divided into double dilation and subtraction operations;

(e) the feature extraction process, which results in a file containing the two main foot parameters.

3. The classification module in which the file with the two foot parameters is read and then a searching process is triggered to activate one of the three knowledge rules.

Figure 3 depicts the foot classification knowledge-based system, which works as follows: All the available foot images are registered into the system; after a period of time or a number of registered cases, each case includes both, the right and left, images; lunch the image processing module for each foot image, i.e suppose there are one hundred cases registered, the image processing module will be executed 200 times; as mentioned above, for each foot image the image processing module extract the longitudinal medial arch and the longitudinal lateral arch measures, which then are registered as part of the record of each case. After all the foot images have been processed, a flag is turn on and an alarm is switched on to announce end of this process. Finally, the operator or the user runs the classification module, in which parameters are red and compared against the knowledge base. This final process registers the foot types in the data base. After the completion of process, the user can print a report showing the results of the classification.

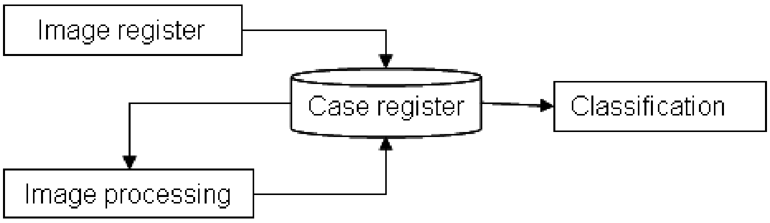


Fig. 3: Dilated image of a right foot

The image processing module, object of study on this paper, is shown in figure 4.

The decompression process was implemented following the graphics utilities installed with unix-like operating systems. The rest of the processes: threshold detect to feature extraction; were implemented in ISO C++. They were functionally and individually tested on WINDOWS.

5 Experiments

The overall image processing module was tested on a PC under SOLARIS system. Average time processing and parameters-extracted values were the main focus of experimentation. The number of digital podographies tested was 200, then; foot parameters were measured manually for comparison purposes and a manual classification was made in order to compare the data extracted against the knowledge base. The average time processing for each case was less than 1 second, so using this as a time processing estimation for 40 thousand images the worst processing time would be 20,000 seconds

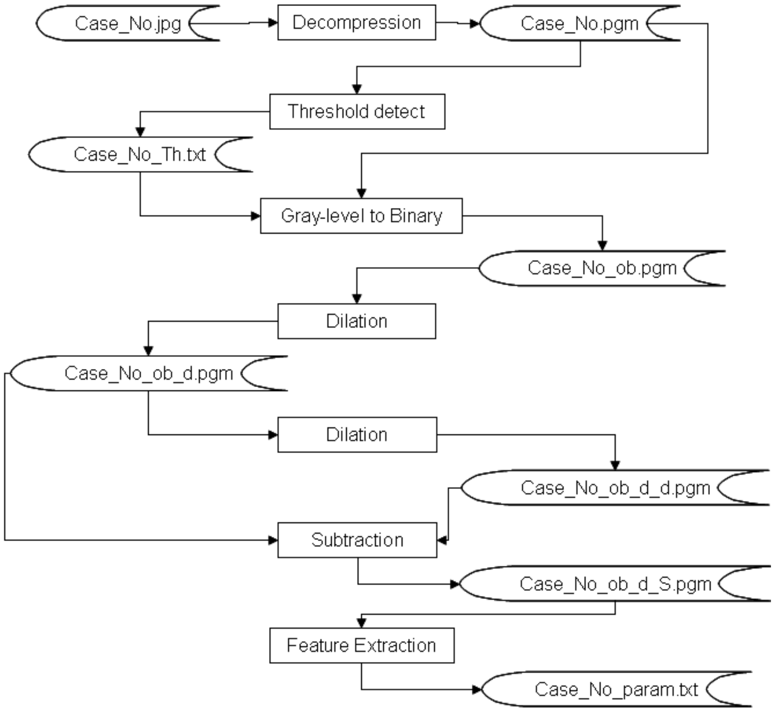


Fig. 4: Dilated image of a right foot

(or 5 hours) considering that the image process module runs sequentially for each image. The best case would be when the processes are run in parallel.

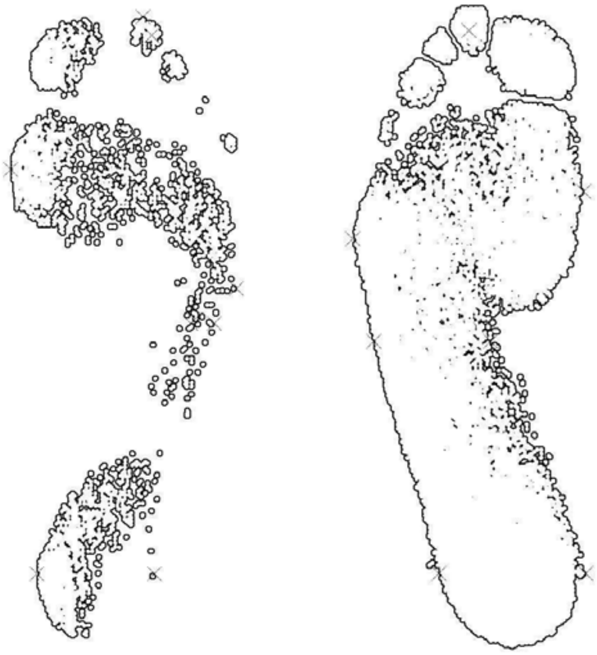
A comparison of values extracted automatically against the manual extraction showed up that the image processing module performed efficiently due to the computation of 95% of correct values. The remaining incorrect 5% was due to images do not complying with the underlying assumptions.

Figures 5 shows two digitalized podographies in which the process was applied:

Figure 5a shows the foot of case number 69, which has been classified as cave foot, whereas figure 5b shows left foot number 1597 classified as normal. Cross in all images mark important points to further extract the three main parameters. When comparing these measures against the knowledge base elicited from the human expert as explained above, the classification system asserts correctly their classification.

6 Conclusions

It was shown that image processing techniques need to be experimented before committing to one or some of them. Part of the knowledge engineering process also implied the refinement of knowledge, which also applied to refining the feature extraction process and consequently the image processing module. This refining stage started with the



(a) No 69: Right (b) No 1597: Left

Fig. 5: Two feet cases after the image processing module showing some marks

definition of having as a main part of the module an edge detection process based on conventional edge operators; however, as experimentation proofed, image morphology demonstrated its efficiency when combining simple operators like image dilation and addition. Further more, as the feature extraction process was based on an edge following algorithm, the process only focused on finding the important points (crossed points on figures 2 and 3) and computing both measurements using simple analytical geometry properties.

The knowledge engineering methodology shows that by assembling together the available and proper techniques, good results can be produced as shown in the experiments section. Also, it was shown that useful image processing techniques can be applied to images when they comply with special conditions and that further preprocessing need to be done in order to guarantee the proper extraction of features. However, at the moment of writing this paper, more testing was being carried out to guarantee the functionality of the system.

7 Acknowledgment

This project was sponsored by the IPN through the research projects program with register No 20060649. We also give thanks to the orthopedist, Rafael Ángel Matos y Santos as he was the human expert from whom knowledge was acquired.

References

1. Ballard, D., Brown, C.H.: Computer Vision. Prentice Hall Inc., New Jersey, USA (1982)
2. Pajares, G., de la Cruz, J., Molina, J., Cuadrado, J., López, A.: Imágenes Digitales: Procesamiento Práctico con Java. RA-MA, Spain (2004)
3. Parker, J.: Algorithms for Image Processing and Computer Vision. John Wiley & Sons, Inc., Canada (1969)
4. Sonka, M., Hlavac, V.: Image Processing, Analysis and Machine Vision. Chapman & Hall, Cambridge, UK (1994)
5. Nixon, M., Aguado, A.: Feature Extraction & Image Processing. Elsevier, Oxford, UK (2002)
6. Turban, E.: Expert Systems and Applied Artificial Inteligence. MacMillan Publishing Company, New Yor, USA (1992)
7. Rosales Peña Alfaro, E. M.: Determinacin Automtica del Tipo de Pie en Alumnos de Nuevo Ingreso del IPN, Technical Report, Research Project 20060649. Technical report, UPIICSA-IPN, Mexico (2006)
8. Pitzen, P., Rössler, H.: Manual de Ortopeda. Ediciones Doyma, Spain (1993)
9. Sponseller, P.: The 5-Miute Orthopaedic Consult. Lippincott Williams & Wilkins, USA (2002)
10. Rao, U., Joseph, B.: The influence of Footwear on the Prevalence of Flat Foot: A survey of 2300 children. J. of Bone and Joint Surgery **74-B** (1992) 525–527
11. Cailliet, R.: Síndromes Dolorosos: Tobillo y Pie. Manual Moderno, Mexico (1998)
12. Ignizio, J.: Introduction to Expert Systems: The Development and Implementation of Rule-Based Expert Systems. Mc GrawHill, USA (1991)
13. Rhem, A.: UML for Developing Knowledge Management Systems. Auerbach Publications, USA (2006)
14. Rothwell, C., Mundy, J., Hoffman, B., Nguyen, V.: Driving vision by topology, technical report no 2444. Technical report, Institut National de Recherche en Informatique et en Automatique (1994)

Image Transmission over Internet

Mireya García[1] and Alejandro Ramírez[2]

[1] Instituto Politécnico Nacional-CITEDI, Av. Del Parque No.1310, Tijuana BC,
mgarciav@citedi.mx
[2] Dpto R&D, PILIMTEC, Châteaugiron,Francia
alramirez10@yahoo.fr

Abstract. Due to the explosive growth of the internet and increasing demand for multimedia information on the web, streaming video over the Internet has received tremendous attention from academia and industry. Video streaming over Internet represents big challenges due to the fact that this network offers generally the best-effort service. This means that this type of service does not provide a guarantee for the bandwidth, delay (jitter) and losses. These characteristics are uncontrollable and dynamic. The purpose of this article is to give a general overview of video over IP and to examine the challenges that make simultaneous delivery and playback, or streaming, of video difficult over packet networks such as the Internet.

Keywords: Streaming, video, IP, compression.

1 Introduction

Video has been an important media for communications and entertainment for many decades. Initially video was captured and transmitted in analog form, but the advent of digital integrated circuits and computers led to the digitization of video [1]. Digital video enabled a revolution in the compression and communication of video. Video compression became an important area of research in the late 1980's and 1990's and enabled a variety of applications including video storage on DVD's and Video-CD's, video broadcast over digital cable, satellite and terrestrial (over-the-air) digital television (DTV), and video conferencing and videophone over circuit switched networks. The growth and popularity of the Internet in the mid 1990's motivated video communication over best-effort packet networks. Video over best-effort packet networks is complicated by a number of factors including unknown and time-varying bandwidth, delay, and losses, as well as many additional issues such as how to fairly share the network resources amongst many flows and how to efficiently perform one-to-many communication for popular content [2]. The purpose of this article is to give a general overview of video over IP and to examine the challenges that make simultaneous delivery and playback, or streaming, of video difficult over packet networks such as the Internet.

Received 07/07/07
Accepted 19/10/07
Final version 24/10/07

Each of the following sections provides a brief description of important factors inside the scheme of video on Internet. The section 2 describes a perspective of the type of applications that they can find inside the scheme of video on IP. The section 3 summarizes the functioning of the compression and its importance in video streaming. Section 4 identifies the three fundamental challenges in video streaming. The section 5 presents the principal current problems of the video on IP. The most important standards used in the video streaming are described in section 6. Standardized media streaming are described in section 7. The article concludes with our vision of the video on IP.

2 Framework Aplications

There exist a very diverse range of different video communications and streaming applications, which have very different operating conditions or properties. Examples of this reality: the video communication application may be for point to point communication or for multicast or broadcast communication, and video may be pre-encoded (stored) or may be encoded in real-time (e.g. interactive videophone or video conferencing). The video channels for communication may also be static or dynamic, packet-switched or circuit-switched, may support a constant or variable bit rate transmission, and may support some form of Quality of Service (QoS) or may only provide best effort support. The specific properties of a video communication application strongly influence the design of the system. Therefore, we continue by briefly discussing some of these properties and their effects on video communication system design.

3 Video Sreaming and Compression

A streaming video system is one in which a source encodes video content and transmits the encoded video stream over a data network (wired or wireless) where one or more receivers can access, decode, and display the video to users in real-time. The presence of the network, which allows the source to be physically distant from the receivers, differentiates streaming video from pre-recorded video used in consumer electronic devices such as DVD players.

Given that uncompressed video has very large bandwidth demands, the need for efficient video compression is paramount in this type of applications.

Video compression is achieved by exploiting the similarities or redundancies that exist in a typical video signal [3]. For example, consecutive frames in a video sequence exhibit temporal redundancy since they typically contain the same objects, perhaps undergoing some movement between frames. Within a single frame there is spatial redundancy as the amplitudes of nearby pixels are often correlated. Similarly, the Red, Green, and Blue color components of a given pixel are often correlated. Another goal of video compression is to reduce the irrelevancy in the video signal that is to only code video features that are perceptually important and not to waste valuable bits of information that is not perceptually important or irrelevant.

The compression of still images, based on the standard JPEG [4], consists of exploiting the spatial and color redundancy that exists in a single still image. Neighboring pixels in an image are often highly similar, and natural images often have most of their energies concentrated in the low frequencies. JPEG exploits these features by partitioning an image into 8x8 pixel blocks and computing the 2-D Discrete Cosine Transform (DCT) for each block. The motivation for splitting an image into small blocks is that the pixels within a small block are generally more similar to each other than the pixels within a larger block. The DCT compacts most of the signal energy in the block into only a small fraction of the DCT coefficients, where this small fraction of the coefficients are sufficient to reconstruct an accurate version of the image. Each 8x8 block of DCT coefficients is then quantized and processed using a number of techniques known as zigzag scanning, run length coding, and Huffman coding to produce a compressed bitstream.

A video sequence consists of a sequence of video frames or images. Each frame may be coded as a separate image, for example by independently applying JPEG-like coding to each frame. However, since neighboring video frames are typically very similar much higher compression can be achieved by exploiting the similarity between frames.

Currently, the most effective approach to exploit the similarity between frames is by coding a given frame by (1) first predicting it based on a previously coded frame, and then (2) coding the error in this prediction. Consecutive video frames typically contain the same imagery, however possibly at different spatial locations because of motion. Therefore, to improve the predictability it is important to estimate the motion between the frames and then to form an appropriate prediction that compensates for the motion.

The process of estimating the motion between frames is known as *motion estimation (ME)*, and the process of forming a prediction while compensating for the relative motion between two frames is referred to as *motion-compensated prediction (MC-P)*. Block-based ME and MC-prediction is currently the most popular form of ME and MC-prediction: the current frame to be coded is partitioned into 16x16 pixel blocks, and for each block a prediction is formed by finding the best matching block in the previously coded reference frame. The relative motion for the best matching block is referred to as the *motion vector* [5].

There are three basic common types of coded frames: (1) intra-coded frames, or I-frames, where the frames are coded independently of all other frames, (2) predictively coded, or P-frames, where the frame is coded based on a previously coded frame, and (3) bi-directionally predicted frames, or B frames, where the frame is coded using both previous and future coded frames [5]. Figure 1 illustrates the different coded frames and prediction dependencies for an example MPEG Group of Pictures (GOP). It is important to indicate that the selection of prediction dependencies between frames can have a significant effect on video streaming performance, e.g. in terms of compression efficiency and error resilience [6].

It is of relevancy to indicate that the current compression standards achieve compression by applying the same basic principles previously presented.

Summarizing, the temporary redundancy is exploited by applying MC prediction; the spatial redundancy is exploited by applying the DCT. The resulting DCT coefficients are quantized and the nonzero quantized DCT coefficients are run length and Huffman coded to produce the compressed bitstream.

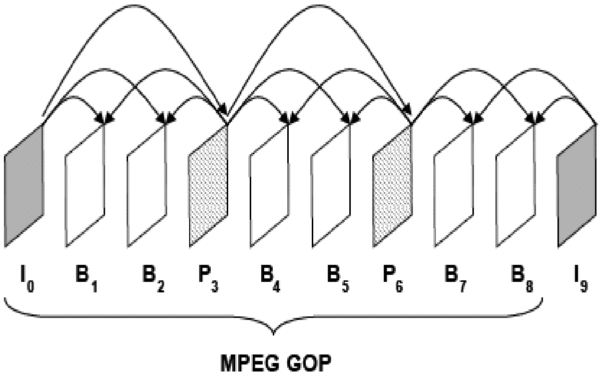


Fig. 1. Example of the prediction dependencies between frames.

At present, the standards of compression that are basically used for communications by video and video streaming are H.263, MPEG-4 and MPEG-4 AVC/H.264 [7].

4 Challenges in Video Streaming

Three fundamental problems exist in video streaming:

4.1 Video Delivery via File Download

Probably the most straightforward approach for video delivery over the Internet is the download, but we refer to it as video download to keep in mind that it is a video and not a generic file. Specifically, video download is similar to a file download, but it is a large file. This approach allows the use of established delivery mechanisms, for example TCP as the transport layer or FTP or HTTP at the higher layers according to OSI model [7].

However, it has a number of disadvantages. Since videos generally correspond to very large files, the download approach usually requires long download times and large storage spaces. These are important practical constraints. In addition, the entire video must be downloaded before viewing can begin. This requires patience on the viewers part and also reduces flexibility in certain circumstances, e.g. if the viewer is unsure of whether he/she wants to view the video, he must still download the entire video before viewing it and making a decision.

4.2 Video Delivery via Streaming

Video delivery by video streaming attempts to overcome the problems associated with file download, and also provides a significant amount of additional capabilities. The basic idea of video streaming is to split the video into parts, transmit these parts successively, and enable the receiver to decode and playback the video as these parts are received, without having to wait for the entire video to be delivered.

Video streaming can conceptually be thought to consist in the following steps:

1. Partition the compressed video into packets
2. Start delivery of these packets
3. Begin decoding and playback at the receiver while the video is still being delivered

Video streaming enables simultaneous delivery and playback of the video. This is in contrast to file download where the entire video must be delivered before playback can begin. In video streaming there is usually a short delay (usually on the order of 1-15 seconds) between the start of delivery and the beginning of playback at the client. This delay, referred to as the pre-roll delay, provides a number of benefits to smoothly balance any network alteration.

Video streaming provides a number of benefits including low delay before viewing starts and low storage requirements since only a small portion of the video is stored at the client at any point in time. The length of the delay is given by the time duration of the pre-roll buffer, and the required storage is approximately given by the amount of data in the pre-roll buffer.

4.3 Expressing Video Streaming as a Sequence of Constraints

Consider the time interval between displayed frames to be denoted by Δ (Δ is 33ms for 30 frames/s video and 100ms for 10 frames/s video).

Each frame must be delivered and decoded by its playback time; therefore the sequence of frames has an associated sequence of deliver/decode/display deadlines:

Frame N must be delivered and decoded by time T_N

Frame N+1 must be delivered and decoded by time $T_N + \Delta$.

Frame N+2 must be delivered and decoded by time $T_N + 2\Delta$.

Any data that is lost in transmission cannot be used at the receiver. Furthermore, any data that arrives too late is also useless. Specifically, any data that arrives after its decoding and display deadline is too late to be displayed.

Note that data may still be useful even if it arrives after its display time, for example if subsequent data depends on this “late” data. Therefore, an important goal of video streaming is to perform the streaming in a manner so that this sequence of constraints is met.

5 Basic Problems in Video Streaming

There are a number of basic problems that afflict video streaming. For example, video streaming over the Internet is difficult because the Internet only offers best effort service. That is, it provides no guarantees on bandwidth, delay jitter, or loss rate. These characteristics are unknown and dynamic. Therefore, a key goal of video streaming is to design a system to reliably deliver high-quality video over the Internet when dealing with unknown and dynamic.

Since it was mentioned in the previous paragraph, the bandwidth available between two points in the Internet is generally unknown and time-varying. If the sender transmits faster than the available bandwidth then congestion occurs, packets are lost, and there is a severe drop in video quality. If the sender transmits slower than the available bandwidth then the receiver produces suboptimal video quality. The goal to overcome the bandwidth problem is to estimate the available bandwidth and than match the transmitted video bit rate to the available bandwidth.

Additional considerations that make the bandwidth problem very challenging include accurately estimating the available bandwidth, matching the preencoded video to the estimated channel bandwidth, transmitting at a rate that is fair to other concurrent flows in the Internet, and solving this problem in a multicast situation where a single sender streams data to multiple receivers where each may have a different available bandwidth.

The end-to-end delay that a packet experiences may fluctuate from packet to packet. This variation in end-to-end delay is referred to as the delay jitter. Delay jitter is a problem because the receiver must receive/decode/display frames at a constant rate, and any late frames resulting from the delay jitter can produce problems in the reconstructed video, e.g. jerks in the video. This problem is typically addressed by including a playout buffer at the receiver. While the playout buffer can compensate for the delay jitter, it also introduces additional delay.

The third fundamental problem is losses. A number of different types of losses may occur, depending on the particular network under consideration.

For example, wired packet networks such as the Internet are afflicted by packet loss, where an entire packet is erased (lost). On the other hand, the wireless channels are typically afflicted by bit erros or burst errors. Losses can have a very destructive effect on the reconstructed video quality.

To combat the effect of losses, a video streaming system is designed with error control. Approaches for error control can be roughly grouped into four classes: (1) forward error correction (FEC), (2) retransmissions, (3) error concealment, and (4) error-resilient video coding [8].

6 Protocols for Video Streaming

The Internet was developed to connect a heterogeneous mix of networks that employ different packet switching technologies. The Internet Protocol (IP) provides baseline best-effort network delivery for all hosts in the network: providing addressing, best-effort routing, and a global format that can be interpreted by everyone.

On top of IP are the end-to-end transport protocols, where Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are the most important [7]. TCP provides reliable byte-stream services. It guarantees delivery via retransmissions and acknowledgements. On the other hand, UDP is simply a user interface to IP, and is therefore unreliable and connectionless. Additional services provided by UDP include checksum and port-numbering for demultiplexing traffic sent to the same destination.

Some of the differences between TCP and UDP that affects streaming applications are:

- TCP operates on a byte stream while UDP is packet oriented.
- TCP guarantees delivery via retransmissions, but because of the retransmissions its delay is unbounded. UDP does not guarantee delivery, but for those packets delivered their delay is more predictable (i.e. one-way delay) and smaller.
- TCP provides flow control and congestion control. UDP provides neither. This provides more flexibility for the application to determine the appropriate flow control and congestion control procedures.
- TCP requires a back channel for the acknowledgements. UDP does not require a back channel.

Web and data traffic are delivered with TCP/IP because guaranteed delivery is far more important than delay or delay jitter. For media streaming the uncontrollable delay of TCP is unacceptable and compressed media data is usually transmitted via UDP/IP despite control information is usually transmitted via TCP/IP.

The entity that specifies the protocols for media delivery, control and description over Internet is IETF (Internet Engineering Task Force) [9].

7 Video Streaming Standards

Standard-based media streaming systems, as specified by 3GPP (3rd Generation Partnership Project) for media over 3G cellular and by ISMA (Internet Streaming Media Alliance) for streaming over the Internet, employ the following protocols [10]:

Media encoding
    MPEG-4 video and audio (AMR for 3GPP), H.263.
Media transport
    RTP for data, usually over UDP/IP
    RTCP for control messages, usually over UDP/IP
Media session control
    RTSP
Media description and announcement
    SDP

The streaming standards do not specify the storage format for the compressed media, but the MP4 file format has been widely used [11].

One advantage of MP4 file format is the ability to include “hint tracks” that simplify various aspects of streaming by providing hints such as packetization boundaries, RTP headers and transmission times.

8 Conclusions

Video communication over packet networks has witnessed much progress in the past few years, from download-and-play to various technologies.

This work presented a general overview of video over IP and examined the challenges that make simultaneous delivery and playback, or streaming, of video difficult over packet networks such as the Internet.

Given this integral vision of the video streaming where big challenges exist to resolving, we believe that video streaming will continue to be a compelling area for exploration, development, and deployment in the future.

Acknowledgments. The authors appreciate the support of SIP-IPN, for the support of this research under grant 20070032.

References

1. B. Girod, J. Chakareski, M. Kalman, Y. Liang, E. Setton and R. Zhang. “Advances in Network-Adaptive Video Streaming” Tyrrhenian Inter. Workshop on Digital Communications, Sept 2002.
2. M.-T. Sun and A. Reibman, eds, Compressed Video over Networks, Marcel Dekker, New York, 2001.
3. Y. Wang, J. Ostermann, and Y.-Q. Zhang, Video Processing and Communications, New Jersey, Prentice-Hall, 2002.
4. G. K. Wallace, “The JPEG Still Picture Compression Standard,” Communications of the ACM, April, 1991.
5. V. Bhaskaran and K. Konstantinides, Image and Video Compression standards: Algorithms and Architectures, Boston, Massachusetts, Kluwer Academic Publishers, 1997.
6. D. Wu, Y. Hou, W. Zhu, Y.-Q. Zhang, and J. Peha. “Streaming Video over the Internet: Approaches and Directions”, IEEE Transactions on Circuits and Systems for Video Technology, March 2001.
7. W. Tan and A. Zakhor, “Real-time Internet Video using Error- Resilient Scalable Compression and TCP-friendly Transport Protocol,” IEEE Trans. on Multimedia, June 1999.
8. Y. Wang and Q. Zhu, “Error control and concealment for video communications: A review”, Proceedings of the IEEE, May 1998.
9. http://www.ietf.org/
10. http://www.isma.tv/.
11. http://www.ww.sinobiometrics.com.

Database of the Mayan Script[*]

Obdulia Pichardo-Lagunas and Grigori Sidorov

Natural Language and Text Processing Laboratory,
Center for Research in Computer Science,
National Polytechnic Institute,
Av. Juan Dios Batiz, s/n, Zacatenco, 07738, Mexico City,
Mexico
ayilina@hotmail.com, sidorov@cic.ipn.mx

Abstract. The deciphering of Mayan script is an intricate but interesting problem. During years, the community of Mayan researchers was not open to the usage of computer tools. Still, the progress of the computer science and the current state of Mayan research proves the necessity of this type of software. We present the project related to the development of Mayan script database, which is the first necessary step in development of computer representation of Mayan script data. The database contains several tables and allows for various queries. The main idea of the project is the development of the system that would allow managing Mayan script data for specialists and as well for persons without any previous knowledge of Maya. This includes structural visual description of glyph images, expert system facilities, and, in future, calculation of glyphs similarity and development of digital corpus for analysis of similarity of the contexts on the fly. Another possible direction of further investigations is confirmation of deciphering results using large corpus data.

1 Introduction

Mayan hieroglyphic writing contains more than thousand of different glyph signs. Many of them are variations of one sign (allographs), others are different signs with the same reading (homophones). Some other signs are glyph variants, – as the mayanist Tatiana Proskouriakoff describes them, – that were used during a certain period of time or in a certain area.

Mayan writing system can be described as a logosyllabic system ([3], [4], [5], [7]), based on signs that represent words (logograms) and syllables that also can be used as phonetic signs. There are approximately 200 different syllabic (i.e., purely phonetics) signs, of which about a 60 percent are homophones. Namely, there are about 80 syllables in the classic Mayan language (according to its phonetics); still more than 200 glyphs signs were used in the phonetic writing. The Mayans used a system of writing capable to register complete oral manifestations of their language.

[*] Work done under partial support of Mexican Government (CONACyT, SNI) and National Polytechnic Institute, Mexico (CGPI, COFAA).

Received 20/0707
Accepted 19/10/07
Final version 24/10/07

The discoveries of the last decades in Mayan epigraphy field allowed deciphering of almost all documents and known inscriptions, according to the dominant theories of deciphering. Nevertheless, during interviews with epigraphists of the Center for Philological Studies, National Autonomous University of Mexico (UNAM), we realized that there is no kind of computational tool that can be used by the Mayanists in their investigations. These investigations are normally carried out manually on the basis of facsimiles of documents and Mayan inscriptions ([1], [6], [8]).

On the other hand, the public in general should also have possibilities of using some specific computer tools if they are interested in reading Mayan glyphs. Though they are not specialists, the subject is interesting from the general cultural perspective.

Even though the specialists are aware of the existence of various dictionaries in the Internet, they are not frequently used. Even more: during years, the community of Mayan researchers was very skeptic as far as the usage of any type of computer tools. Still, the progress of computer science and the current state of Mayan research proves the necessity of this type of software, because it makes the work of a researcher more fast, reliable and productive.

We present the project related to the development of Mayan script database, which is the first necessary step in development of computer representation of Mayan script data.

We believe that developing of this application would be impossible without support of investigators of Maya and without considering their needs as end users. Our purpose is development of the application in which investigators of Maya would participate in design and implementation. Also, this application should be able to resolve specific problems that can solely be raised by a specialist.

This software will help to diminish the burden of many rather complex procedures that are performed manually by now, like context comparison or glyph identification. In addition, this software offers to the investigators a tool that facilitated the process of search and classification of the Mayan glyphs and can serve as a didactic tool that helps in learning of the glyph signs that compose Mayan writing system.

2 Database Description

2.1 Database

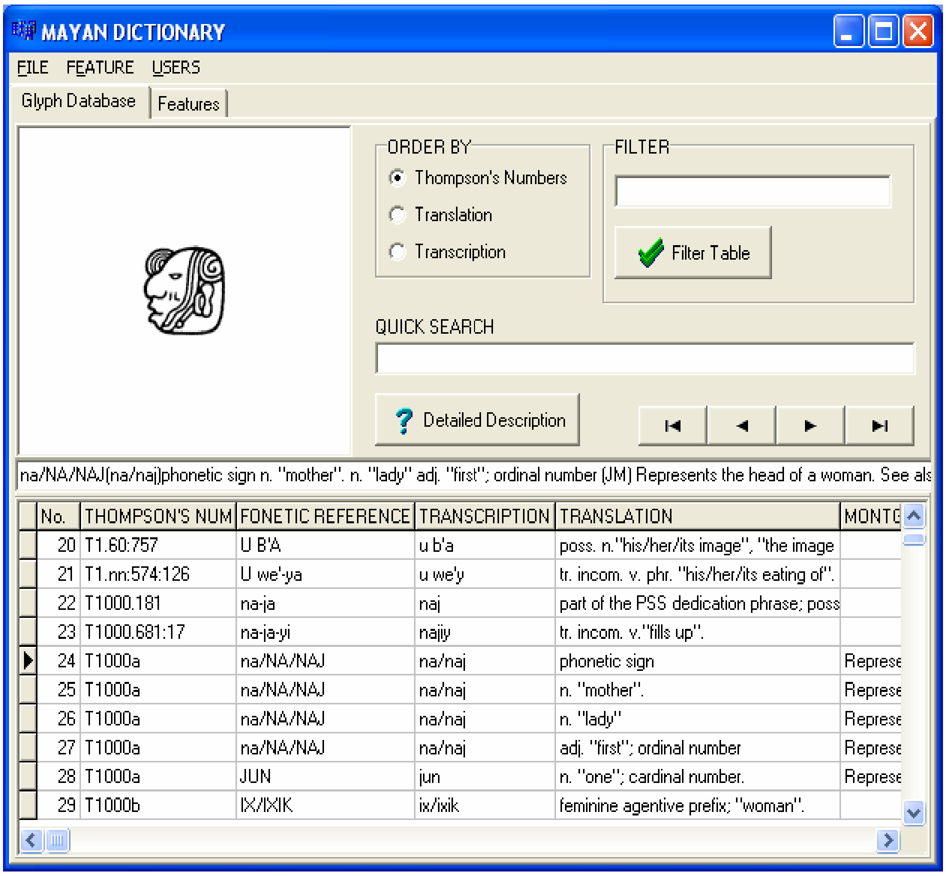
We based our development on John Montgomery's dictionary [2]. It contains hieroglyphs of Mayan classic writing, organized alphabetically by phonetic writing of words, phrases or syllables, and also includes appendices that list signs using additional classification categories.

The developed application is a relational database – and the corresponding interface – that stores general information about glyphs: glyph image, translation, transcription, Thompson numbers (these can be considered as glyphs identifiers (ID), they were assigned in arbitrary manner by E. Thompson, an outstanding Mayan investigator), phonetic reference and descriptive notes corresponding to each glyph that forms part of John Montgomery's dictionary. Total size of the dictionary is 1,241 entries. The database is ordered according to Thompson numbers, but the system offers two more options for its ordering using the fields containing transcription or translation.

Due to normalization of the database, in the cases that a glyph has more than one meaning according to the dictionary, these are separated and marked with ID that corresponds to the sign and as well with the IDs that correspond to the number of meaning (translation); thus, different meanings are stored as separate records in the database.

The database can also store information provided by the user related to the structural visual graphical image description. This is based on a feature set created by each user if he wants to describe the glyphs using his own feature set, see Section 2.3. Besides, the system has the predefined feature sets. The user has a possibility to use any feature set: the predefined one or his own set.

Fig. 1. Interface for the database (main dictionary view).

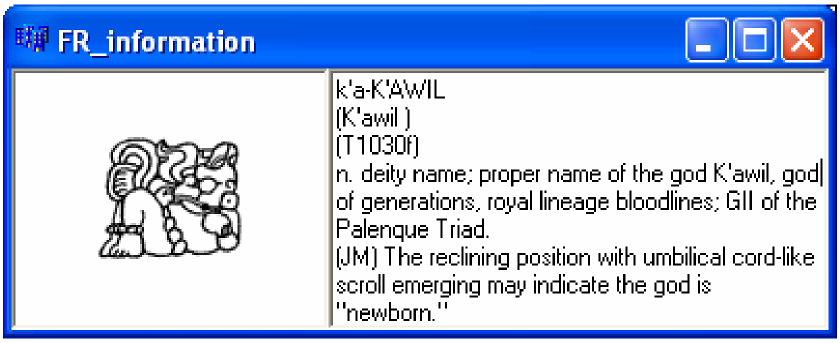


ORDER BY
● Thompson's Numbers
○ Translation
○ Transcription

na/NA/NAJ(na/naj)phonetic sign n. "mother". n. "lady" adj. "first"; ordinal number (JM) Represents the head of a woman. See als

No.	THOMPSON'S NUM	FONETIC REFERENCE	TRANSCRIPTION	TRANSLATION	MONTG
20	T1.60:757	U B'A	u b'a	poss. n."his/her/its image", "the image	
21	T1.nn:574:126	U we'-ya	u we'y	tr. incom. v. phr. "his/her/its eating of".	
22	T1000.181	na-ja	naj	part of the PSS dedication phrase; poss	
23	T1000.681:17	na-ja-yi	najiy	tr. incom. v."fills up".	
24	T1000a	na/NA/NAJ	na/naj	phonetic sign	Represe
25	T1000a	na/NA/NAJ	na/naj	n. "mother".	Represe
26	T1000a	na/NA/NAJ	na/naj	n. "lady"	Represe
27	T1000a	na/NA/NAJ	na/naj	adj. "first"; ordinal number	Represe
28	T1000a	JUN	jun	n. "one"; cardinal number.	Represe
29	T1000b	IX/IXIK	ix/ixik	feminine agentive prefix; "woman".	

Fig. 2. Complete information about a glyph.



k'a-K'AWIL
(K'awil)
(T1030f)
n. deity name; proper name of the god K'awil, god of generations, royal lineage bloodlines; GII of the Palenque Triad.
(JM) The reclining position with umbilical cord-like scroll emerging may indicate the god is "newborn."

2.2 Dictionary Representation

The information about glyphs is visualized as records in a standard tabular way, i.e., each visual field value corresponds to a specific database field. The following fields are used: glyph image, Thompson number, phonetic reading, meaning (translation from Maya), and comments.

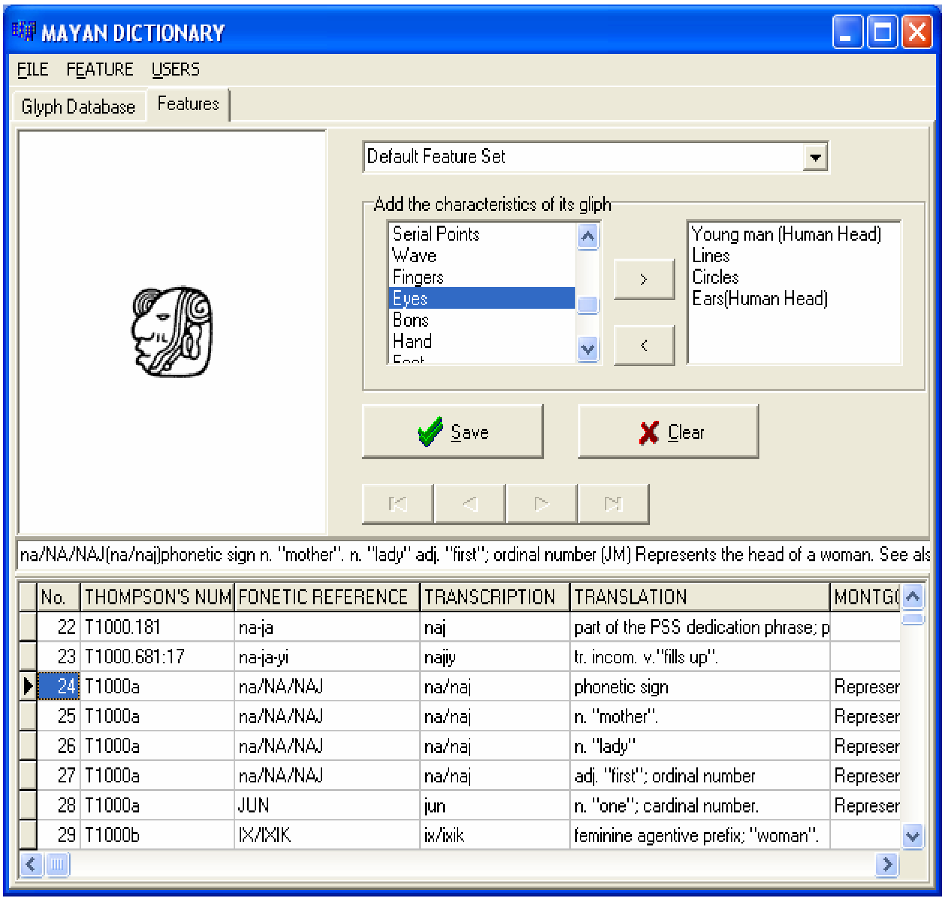
Different meanings are stored as separate records (Fig. 1), thus, duplicating other field values. We chose this option for enabling searches in the field “meaning”. Still, we have only one glyph image visible at a time and we present additionally the concatenated information of values referring the meaning (from several records) just below the glyph. This is justified by the fact that otherwise this information is presented in separate records and, thus, it is not clearly related to the same glyph.

Also, the complete information about glyphs can be visualized within separate dialog window, see Fig. 2.

The user has an option to choose the ordering by marking the corresponding field in the RadioButton “Order by”: Thompson numbers, translation or transcription.

This interface also allows for two types of searches: fast search and filtering. Both of them are performed in accordance with the selected ordering. The fast search moves the table cursor to the desired value according to the ordering. The filtering restricts the records that are shown to the user according to the data present in the field “Filter” and the current ordering.

Fig. 3. Feature set assignment.



Default Feature Set

Add the characteristics of its gliph

Serial Points
Wave
Fingers
Eyes
Bons
Hand
Foot

Young man (Human Head)
Lines
Circles
Ears(Human Head)

na/NA/NAJ(na/naj)phonetic sign n. "mother". n. "lady" adj. "first"; ordinal number (JM) Represents the head of a woman. See als

No.	THOMPSON'S NUM	FONETIC REFERENCE	TRANSCRIPTION	TRANSLATION	MONTG(
22	T1000.181	na-ja	naj	part of the PSS dedication phrase; p	
23	T1000.681:17	na-ja-yi	najiy	tr. incom. v."fills up".	
24	T1000a	na/NA/NAJ	na/naj	phonetic sign	Represer
25	T1000a	na/NA/NAJ	na/naj	n. "mother".	Represer
26	T1000a	na/NA/NAJ	na/naj	n. "lady"	Represer
27	T1000a	na/NA/NAJ	na/naj	adj. "first"; ordinal number	Represer
28	T1000a	JUN	jun	n. "one"; cardinal number.	Represer
29	T1000b	IX/IXIK	ix/ixik	feminine agentive prefix; "woman".	

2.3 Glyph's Characterizing Facilities

The system allows for structural description of glyph images using various feature sets. This facility permits a user who is not familiar with Maya glyphs to make searches and identify the glyphs.

The user can use predefined feature sets or create his set and describe glyphs according to this set.

The systems shows the window, where provides general information about a glyph. Also, a list of possible graphical features (characteristics) is presented, see Fig. 3.

The user can add and remove these features while describing an individual glyph. This makes possible assigning characteristics that correspond to each one of the glyphs. Each user (professional or amateur) can use his judgments for this characterizing.

The system stores each one of assigned feature values and allows visualizing of the characteristics that have been assigned previously to the glyph. Each set of features has its name, thus, changing the set is just picking up a different set name. This allows for different users applying different feature sets prepared by them or by other users.

This information allows for development of an expert system capable to recognize the glyphs on the basis of their structural visual graphical characteristics, asking a user several questions about the glyph image.

3 Conclusions and Future Work

We present the project related to the development of Mayan script database, which is the first necessary step in development of computer representation of Mayan script data. The database contains several tables and allows for various queries. The main idea of the project is the development of the system that would allow managing Mayan script data for specialists and for persons without any previous knowledge of Maya. This includes structural description of glyph images and expert system facilities.

We hope that the experts in the field will reconsider their position towards the usage of computer tools starting from the usage of the described system that will help them to identify glyphs and reduce the time spent for search and classification of the glyphs

The developed application will serve also as a didactic tool that helps not only the professional investigators, but it will also serve to any person interested in the Mayan writing system.

We plan to implement a determinist expert system based on the structural characteristics of glyph images. This will allow for performing automatic glyph classification, so that when the database contains the characteristics that correspond to each glyph according to the selected feature set, it will be possible to make reasoning procedures for quick glyph searches.

When the glyphs are classified, it will be easier to locate them within documents and Mayan inscriptions. As further perspective, we plan to develop a corpus when the glyphs are represented by their identifiers, for example, Thompson numbers. Next interesting step in the future is related with applying statistical methods of corpus linguistics to confirm the results of the deciphering procedures.

4 References

[1] Foundation for the advancement of Mesoamerican studies.inc. http://www.famsi.org/spanish/mayawriting/dictionary/montgomery/index.html.1/10/2007

[2] John Montgomery and Christophe Helmke. Dictionary of Mayan hieroglyphs. Hippocrene Books Inc., 2007.

[3] Michael D Coe and Mark Van Stone. Reading the Maya Glyphs. Thames & Hudson, 2001.

[4] Inga E. Calvin. Maya Hieroglyphs Study Guide. www.famsi.org/**maya**writing/calvin/index.html

[5] John Montgomery. How to Read Maya Hieroglyphs. Hippocrene Books Inc., 2003.

[6] Alexander Wolfang Voss y Hans Juergen Kremer. Estudio epigráfico sobre las inscripciones jeroglíficas y estudio icnográfico de la Fachada del Palacio de los Estucos en Acanceh Yucatán México.

[7] Yuri V. Knorosov. La antigua escritura de los pueblos de America Central. Mexico City: Biblioteca Obrera, 1954.

[8] Linda Shele. Notebook for the Maya Hieroglyphic Writing Workshop at Texas Austin: Institute of Latin American Studies, University of Texas, 1978.

[9] Tatiana Proskuriakoff. Historical data in the inscriptions of Yaxchilan (part II). Estudios de cultura maya, 4, pp. 177-202.

The Digital Music Rest e-Service

Tatiana Tambouratzis[1]

[1] Department of Industrial Management and Technology, University of Piraeus,
107 Deligiorgi St, Piraeus 185 34, Greece
tatiana@unipi.gr
http://www.tex.unipi.gr/dep/tambouratzis/main.htm

Abstract. The digital music rest music-on-demand e-service is presented. Following an e-request from the user for a music score (or part thereof), the digital music rest presents the selected musical manuscript on the user's computer screen. The digital music rest incorporates a variety of viewing options concerning manuscript presentation (such as size and color of the measures, notes and accompanying notation; background color; contrast; page configuration; digital metronome; static/dynamic viewing or scrolling) that allows custom-made screen layout and transition; these can be modified at any time. The initial realization of the digital music rest is put forward as a superior alternative to the traditional music-rest/musical-manuscript arrangement for professional, amateur as well as visually impaired musicians during musical instruction and therapy. Future research aims at developing the digital music rest in order to accommodate for more music scores, users as well as (synchronized) ensembles of digital music rests during performances of orchestras.

1 Introduction

Any musician - be (s)he a music student, an amateur player, a member of an orchestra or a soloist – uses a musical manuscript placed upon a music rest in order to learn and perform a piece of music. At the end of each page of the musical manuscript, the musician uses one hand in order to turn the page and proceed playing. By repetition, the music score is memorized and page turning becomes progressively less imperative, although music-rest/musical-manuscript arrangements are still used during performances: it is important to have a written source to refer to as a prompt or initial stimulus or simply so as to minimize memorization (given that only soloists play by heart during performances).

In this piece of research, the digital music rest (DMR) music-on-demand e-service is put forward as an alternative to the traditional music-rest/musical-manuscript arrangement. Following an e-request from the user for a given music score (or part thereof), the DMR presents the selected musical manuscript on the computer screen of the user. The DMR incorporates a number of viewing options concerning manuscript presentation (such as size and color of the measures, notes and accompanying notation; background color; contrast; page configuration; digital

Received 21/06/07
Accepted 19/10/07
Final version 24/10/07

metronome; static/dynamic viewing or scrolling etc.) that allow custom-made screen layout and transition. Additionally, the user can modify the viewing options at any time. The DMR e-service allows on-line access to (potentially) any musical manuscript. Moreover, no wear-and-tear of the musical manuscript occurs, while it becomes straightforward and economical for groups of musicians to have access to the same music score at the same time.

No similar e-service for acquiring music scores has been reported to date. The display options of the DMR render it a superior alternative to the professional, amateur as well as visually impaired musician during musical instruction, rehearsal and performance.

This paper is organized as follows: section 2 describes the profile of the potential DMR users, focusing upon the needs of each group (especially of visually impaired musicians); section 3 describes the viewing options afforded by the DMR, while section 4 details the implementation characteristics of the initial DMR realization; section 5 summarizes the advantages of the DMR over the existing music reading options and points towards future extensions; finally, section 6 concludes the paper.

2 Potential DMR Users

The DMR music-on-demand e-service is aimed at amateur and professional musicians (either with normal or corrected-to-normal vision) as well as at visually impaired musicians (persons with vision problems that cannot be completely corrected with an ordinary pair of glasses or contact lenses); the latter may employ music instruction as a means of education, social interaction and the ensuing pleasure.

As far as visually impaired musicians are concerned, the existing options for learning a music score are restricted to either reading the music from Braille [1-4] or using a tape recorder/CD player in order to memorize the music prior to playing it. In Braille, musical information is presented consecutively rather than graphically, whereby appearance is not necessarily related to sound; furthermore, the lack of vertical alignment obscures the relationship between notes in chords, while a single misread dot may render the corresponding measure meaningless. Finally, there is no way of playing with both hands while reading. All of the above, combined with the limited number of Braille music scores available and the variety of formats employed, severely restrict the usefulness of Braille music scores. Unfortunately, it is even harder to accomplish aural memorization of a music score at a level that allows the visually impaired musician to play the score.

In any case, and assuming that learning has somehow been accomplished, large-print music (either printed directly in large format, enlarged while photocopied, or scanned and subsequently enlarged and printed) has been proposed as a memorization aid. While, however, the direct print option is not available for all musical instruments and/or music scores, the other two options are expensive and not easily adaptable to the individual needs of the visually impaired musician: the larger the note size the smaller the amount of music that can be seen at once, and thus the more page turning and head movement become necessary; the latter calls, in turn, for frequent adjustments of the amount and direction of lighting as well as of the position of the music rest relative to the visually impaired musician.

Recently, a number of software programs providing aid in the musical instruction of visually impaired musicians have been developed:

- Goodfeel by Dancing Dots converts the music files used to produce the music score directly into Braille music notation. The transcription is displayed via pins which are raised/lowered (e.g. ALVA Braille Terminal), accommodating for most Western languages; both the tune and the accompanying notation (e.g. *allegro ma non troppo*) are transcribed.
- Opus Dots Lite by Opus Technologies accomplishes transcription into Braille music notation directly from the scanned musical manuscript.
- SharpEye by Music Reader, Midiscan and Pianoscan by Musitek, Toccata convert the scanned musical manuscript into a MIDI file or some similar format that supports direct transcription into Braille music notation. Furthermore, Magni-Ccata incorporates the option of exposing the melody/musical manuscript correspondence, by highlighting on-screen (and thus assisting the musician to focus upon) the notes currently played.

PortaNum, a software program that acts as a general viewing tool for the visually impaired individual, is also worth mentioning. In PortaNum, a camera connected to a computer takes frames (or a film) of what the individual cannot see. Subsequently, the frames are submitted to user-controlled image processing (zoom-in and navigation within the zoomed window, contrast and brightness adjustment, smoothing, color selection and inversion, edge extraction and enhancement, edge widening etc.) in order to render the desired details visible to the visually impaired individual.

3 DMR Viewing Options

In order to be able to satisfy the requirements and needs of all the potential music-on-demand e-service users and to directly generate a custom-made presentation of the musical manuscript on screen, the DMR offers a number of on-line interactive actions concerning the viewing options of the music score.

The musical manuscript layout and the transition between pages (screens) can be selected as on of the following:

- Static viewing. The user determines the desired number of staffs to fit on the screen as well as the desired number of measures to appear side by side on each staff (single staffs are used for music notation of musical instruments such as the violin, while grand/double staffs are used for musical instruments such as the piano; in the latter case, the grand staff is composed of the upper and lower staff, pertaining to the right and left hand, respectively). Provided that the selected layout is feasible, the first screen is presented to the user with the initial (and appropriately configured) measures of the manuscript; by pressing a certain key (e.g. PgDn/PgUp) the transition to next/previous screens of similarly configured measures is achieved. In order to avoid distortion, the measures are proportionally scaled (by equal amounts along the vertical and horizontal direction); hence, in order to be feasible, the user's request must result in an easily readable configuration of measures and staffs on the screen. Static viewing is appropriate during musical instruction, as it allows the user custom-made layout of the musical manuscript as well as entire control over the transition to subsequent/previous screens.

- Automatic viewing. Further to the desired number of staffs and measures per staff (as in static viewing), the user determines the tempo with which the music score is to be played. Provided that the requested layout is feasible, the first screen of appropriately configured measures is presented to the user. The digital metronome is activated by pressing a certain key (e.g. Enter), whereby – and depending on the specified tempo - automatic transition to the next screen of similarly configured measures is made as soon as the time required for the on-screen measures to be played has elapsed. Automatic viewing is appropriate during musical rehearsal, as it not only allows custom-made layout of the musical manuscript, but it also provides the use of a metronome as well as automatic advancement to next screens according to the desired tempo, thus allowing the musician free use of his/her hands.
- Scrolling. The user determines the desired size of the measures on screen as well as the tempo with which the music is to be played. The digital metronome is activated by pressing a certain key, whereby consecutive measures of the selected manuscript slide smoothly on the center of the screen, concatenated and with horizontal direction from right to left; the speed of sliding is given by the tempo, as kept by the metronome. Scrolling can significantly aid visually impaired musicians by reducing the amount of head-turning and re-focusing necessary; tempo tuning allows instruction at any level of virtuosity and/or familiarity with the selected music score.

Additional services that enhance the aforementioned viewing options include:

- ♦ Selecting the color of the measures, notes and accompanying notation; picking the background color; adjusting the contrast and luminance. These options resemble the image processing capabilities of PortaNum.
- ♦ Pausing and resuming, simply by pressing a certain key (e.g. P).
- ♦ Adjusting the volume of the metronome.
- ♦ Opting for the appearance of a number over each measure. Numbering indicates the count of each measure relative to the first measure of the selected score (or part thereof) and enables the user to directly return/proceed to a given measure.
- ♦ Directly moving to a given measure of the score (via its number), to the beginning or end of the score, to the next/previous screen etc.
- ♦ Highlighting and/or providing musical accompaniment to the groups of notes currently played; this service is only available for automatic viewing and scrolling since it presupposes the use of the metronome.

The viewing options can be adjusted at any time during DMR operation.

4 DMR Implementation

The initial realization of the DMR is detailed next. The implementation characteristics are focused - on the one hand - upon pre-processing and storing the musical manuscript and – on the other hand – upon ensuring prompt communication with the user and satisfactory music-on-demand e-service.

4.1 Preprocessing and Storage

In order to store a music score, the musical manuscript corresponding to each musical instrument must be isolated. Subsequently, each musical manuscript is scanned and converted into digital form, one page at a time. Digitization is followed by segmentation of the staffs within each page and segmentation of the measures within each staff, keeping together upper and lower staffs in the case of grand staffs. The accompanying notation relating to the clef, time signature and accidentals is retained separately during staff segmentation; the same applies to the notation relating to dynamics. Each measure is stored separately in such a way that the order of measures within the staff as well as the order of staffs in the manuscript are preserved. This format allows custom-made reconstruction of the manuscript by concatenating successive files according to the user's viewing options. Figure 1 illustrates the aforementioned steps; (a) shows the digitized page of a musical manuscript, while (b) presents the result of segmenting the page into grand staffs for the fifth (last) grand staff; (c) demonstrates the result of segmenting the fifth grand staff of the page into measures for the second measure of the grand staff as well as the components that may also be necessary for on-screen viewing. If, due to the user's viewing options, the segmented measure is to appear first on a staff, the accompanying notation must be concatenated to the left of the measure. Furthermore, any dynamics notation must appear above the measure.

Some additional points concerning realistic on-screen presentation are worth mentioning:

- Measure segmentation must be accurately performed so that the bar line appears as a one-pixel-wide vertical line at either end of the measure, with no white space at either end. Double bar lines must be processed differently for the two measures involved: the double bar line must appear with a one-pixel wide vertical line at the left/right end of the following/preceding measure, again with no space at either end of the measure. Dotted bar lines must be ignored during segmentation.
- A certain amount of scaling may be necessary in order to unify the height of measures from different staffs. Scaling can be either proportional or along the vertical dimension only.
- During presentation, the clef, time signature and accidentals corresponding to a staff must appear to the left of the first measure of the staff; the same applies to the notation relating to dynamics, which must be shown on top of the implicated measure(s) (Fig. 1(c)).

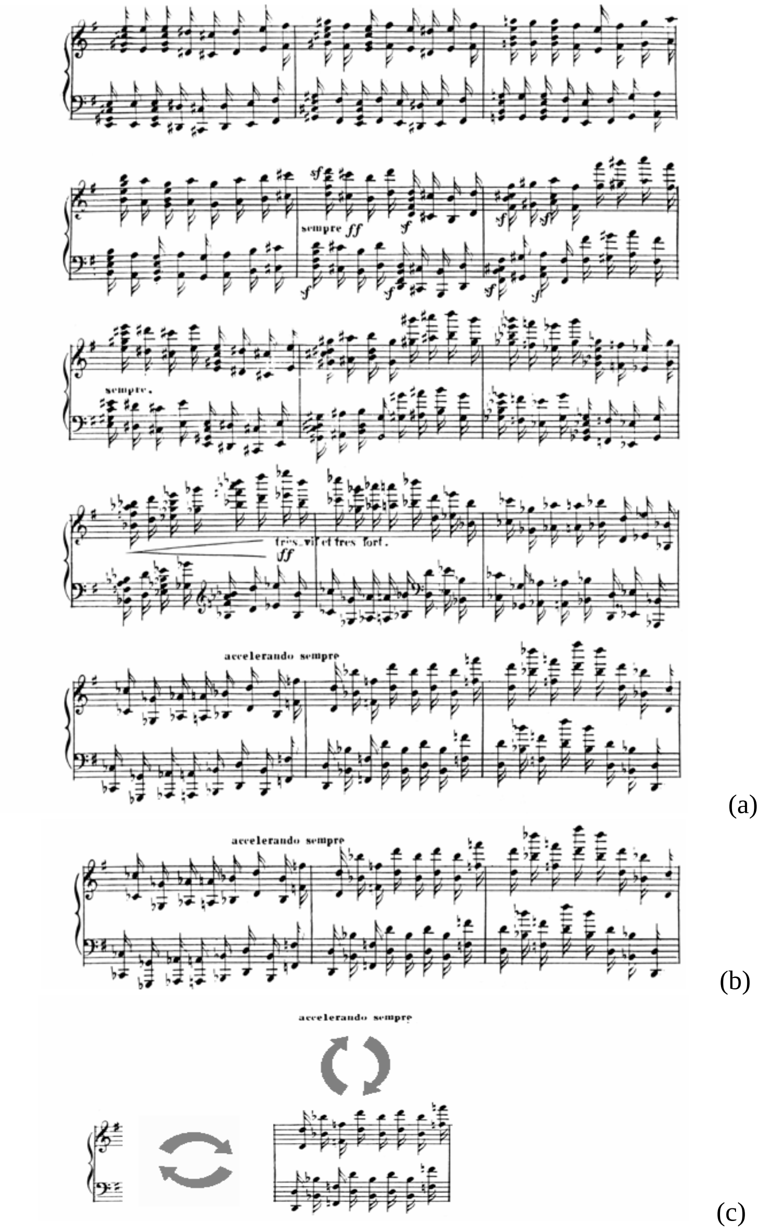


Fig. 1. Preprocessing and storage; digitized page (a), segmentation per grand staff (b); segmentation per measure, segmentation of accompanying notation and dynamics notation, desired reconstruction (c). Excerpt from “Douze Etudes – For Solo Piano”, Op. 35, C.V.Alkan

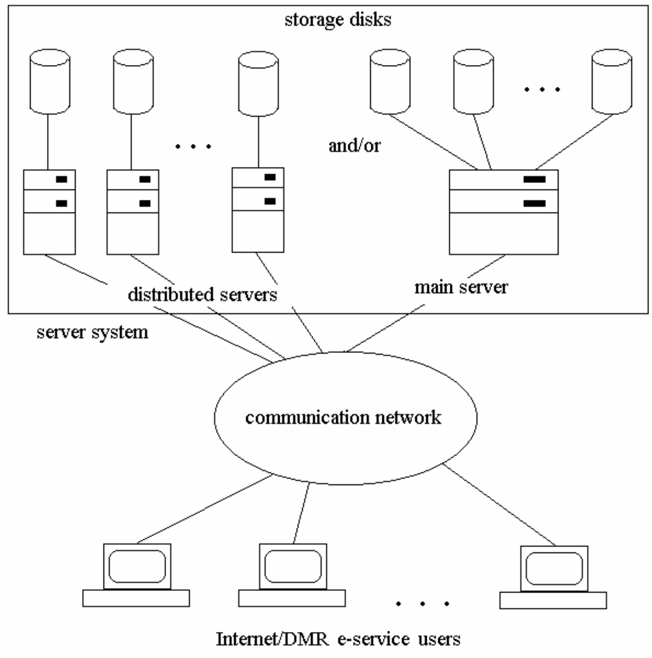


Fig. 2. DMR architecture

4.2 User Interface with the e-Service

Fig. 2 illustrates the implementation characteristics of the DMR music-on-demand e-service; clearly, the architecture is similar to that employed for video-on-demand [5-6] systems. A user (one of the computers at the bottom of Fig. 2) requests a given music score from the communication network (center of Fig. 2)), also specifying the musical instrument as well as the part of the manuscript in which (s)he is interested. The communication network relays the request to the server system (top part of Fig. 2) and the musical manuscript is retrieved from the appropriate server(s). Once configured according to the user's specifications by the server system, the manuscript is delivered – again through the communication network - to the user. The server system can comprise either distributed server configuration (leftmost top part of Fig. 2) or a a main (centralized, single) server (rightmost top part of Fig. 2). For the initial realization of the DMR, a single server makes up the communication network: such a configuration has been found sufficient for the small number of manuscripts and users involved at present.

5 Advantages – Future Directions

The DMR constitutes an economical and efficient means of handling musical manuscripts: a single scan/storage cycle of the music score suffices and there is no need to print, recopy or otherwise handle the music-sheet; additionally there is no wear-and-tear of the music score even after repeated playing.

Furthermore, the DMR facilitates musical practice of all groups of musicians (amateur and professional musicians as well as visually impaired musicians): the musician can directly acquire the desired music score, specify the passage to be viewed, isolate it and have it displayed on screen in a custom-made form; (s)he is allowed extra freedom during practice by not being obliged to use one hand in order to turn pages (screens); the musician can directly employ different configurations at different parts of the manuscript according to his/her current needs. Especially concerning the visually impaired musician, the problem of shadows is totally obliterated since illumination of the screen comes from within, while the requirement of employing a high degree of enlargement is easily accommodated for and can be instantly adjusted.

In terms of realization, and in order to ensure widespread acceptance, the DMR must increase its communication and disk capacity as well as provide low cost and quality of service to a multitude of concurrent users. The use of a distributed server configuration must be adopted, thereby offering:

- superior scalability, i.e. the availability of more music scores, the ability to serve concurrently more users requesting the same or different music scores under a variety of continually changing viewing options,
- robustness, i.e. no disruption in service (either in the speed of service or in the musical manuscripts that are available) when some of the servers fail or are under maintenance,
- access-time minimization, i.e. reduction in network delay and congestion.

Future research shall expand upon the development of accompanying software/hardware similar to the “set-up box” of Internet video-on-demand systems. This will be used not only as a buffer, signal decoder and screen-presentation timer concerning the information relayed from the communication network, but also for setting and adjusting the user’s viewing options on his/her computer (rather than on the server system), thus reducing the traffic load imposed on the DMR.

Such a DMR realization will also render possible the synchronized presentation of the various musical manuscripts of a given music score, thereby showing the way to custom-made manuscript presentation during rehearsal and performance of orchestras. Since a full-sized orchestra (a) consists of more than one hundred musicians playing anywhere between eighteen to twenty-five different kinds of instruments, where (b) woodwind, brass, and percussion players play separate parts, whereby they need one music rest each while (c) string musicians sit in pairs and it is the duty of the musician farther from the audience to turn the pages of the shared music rest, the use of an ensemble of DMRs (supplemented with the aforementioned appropriately developed software/hardware) will:

- ease the burden of the musicians assigned with page turning, thus allowing them to have their hands free most (or all) of the time as well as obliterating the risk of incorrectly turning, mixing or dropping the pages,
- eliminate the problem of the wind blowing away or in any way mixing up the pages of the musical manuscripts during open-air concerts,
- give an elegant hi-tech touch to the proscenium of opera houses, concert halls etc.

In order to successfully synchronize the DMR ensemble, and rather than having all DMRs share the same digital metronome, it would be necessary to directly follow the conductor's tempo-setting as well as tempo variations; image processing may prove a viable tool for that.

6 Conclusions

The digital music rest music-on-demand e-service has been put forward as a novel and potentially superior means of reading music scores that can be used by any musician, for any music score and during any kind of musical activity. The digital-music-rest aims at totally replacing the music-rest/musical manuscript combination that constitutes the universal means of reading and playing music.

References

1. Music Braille Code (compiled by the Braille Authority of North America). American Printing House for the Blind, Louisville (1997)
2. New International Manual of Braille Music Notation (ed. Krolick, B.). Braille Music Subcommittee, World Blind Union, Amsterdam (1997)
3. Krolick, B.: How to Read Braille Music. Opus Technologies, San Diego (1998)
4. Smaligo, A.: Resources for Helping Blind Music Students: a Variety of Resources is Available to Help Educators Teach Blind Students how to Read Music and Become Part of the Music Classroom. Music Educators Journal (September 1998) 24-27
5. Leung, Y.-W., Hou, R.Y.-T.: Assignment of Movies to Heterogeneous Video Servers.. IEEE Trans. Syst., Man, and Cybern.(A): Syst. and Hum. 35 (2005) 665-681
6. Chen, L., Veervalli, B.: Multiple-Server Movie-Retrieval Strategies for Distributed Multimedia applications: a Play-While-Retrieve Approach. IEEE Trans. Syst., Man, and Cybern. (A): Syst. and Hum. 36 (2006) 786-803

Using Team Automata to Specify Software Architectures

Mehran Sharafi [1], Fereidon Shams Aliee [2], Ali Movaghar [3]

[1] Faculty of Engineering, Azad Islamic University of Najafabad
[2]Faculty of Computer Engineering, Shahid Beheshti University, Tehran, Iran
[3] Computer Engineering Department, Sharif University of Technology, Tehran 11365, Iran

Mehran_sharafi@iaun.ac.ir, F_shams@sbu.ac.ir,
Movaghar@sharif.edu

Abstract. Using formal methods to specify software architectures make it possible to form a rigorous foundation to automatically verify different aspects of software architectures. In this paper we introduce a framework to formally specify and evaluate software architectures. The framework includes an algorithm to transform software architecture described in UML to a powerful and formal model, called Team Automata. The framework also proposes a performance model over the obtained formal descriptions. This model is used to specify, evaluate and enhance the architecture of a Web-Service software under flash-crowd condition and the results of analyses and experiments are presented.

Keywords: Software Architecture, Team Automata, Component Interaction, Performance.

1 Introduction

Software Architecture (SA in short) in early development phases represents models which contain basic structural components of software and their interactions; on the other hand, it contains both static structure and dynamics of the system behavior. Despite very high level of abstraction of architectural models, they comprise important design features which could be used to anticipate functional and non-functional attributes (like performance, security, etc.) of software. In the past several years, many methods have been proposed to specify and evaluate SA and their primary goal is to facilitate architectural decision-makings; for example, in order to choose a suitable architecture among several architectural alternatives, one that best fits functional and non-functional requirements of relevant software [2, 3, 4, 5].

In this work, we introduce a formal framework to specify and evaluate software architectures and try to overcome the usual limitations of common formal models. Within the framework, we have proposed an algorithm to transform SA behaviors described in UML 2.0 to an automata-based model called Team Automata [8]. Along by

Received 30/08/07
Accepted 19/10/07
Final version 24/10/07

the formal descriptions, we proposed a performance model which is used to evaluate performance aspects of software architecture. Thus, our framework could be used by software architects to choose suitable architecture from among many alternatives and/or help them to make changes to architecture to fit desired performance requirements. This paper organized as follows: After Introduction, in Section 2 a comparison is made between some extended automata-based models, and their capabilities and weaknesses to specify components interaction. In this section also Team Automata, as a selected model, has been introduced and some definitions applied in our algorithm have been explained. Section 3 introduces overall framework. In Section 4,the proposed framework, have been applied on two alternative architectures of a web-service software as a case study, and the results have been presented. Section 5 refers to conclusions and future work.

2 Using Automata-Based Models to Specify SA

As we mentioned before, automata-based models have been used in the literature to specify dynamics of software architectures. However some of extended automata are more consistent for this issue because they have been designed for modeling the interaction among loosely coupled components in systems. For example, Input/Output Automata (IOA in short) [9] as a labeled transition system provide an appropriate model for discrete event systems consisting of concurrently operating components with different input, output and internal actions. IOA can be composed to form a higher-level I/O automaton thus forming a hierarchy of components of the system. Interface Automata (IA) [10, 11] are another extended automata model suitable for specifying component-based systems, which also support incremental design. Finally, Team Automata [8] is a complex model designed for modeling both the conceptual and architectural level of groupware systems.

The common feature of these automata models is that "actions" are classified in 'inputs', 'outputs' and 'internals', so that internal actions cannot participate in components interaction. This feature has made them powerful to specify interaction among loosely coupled and cooperating components. It is clear that there are many similarities between application domains of the mentioned models and the literature of Software Architectures. Thus, applying these models in SA area must be greatly taken into consideration by software engineers. In [12] we also made a detailed comparison among these models and described why we have selected Team Automata for our framework.

2-1. Team Automata

Team Automata model was first introduced in [13] by C.A.Ellis. This complex model is primarily designed for modeling groupware systems with communicating teams but can also be used for modeling component-based systems [14]. In this section, some definitions of TA literature are briefly described. These definitions have been used in the algorithm proposed in this paper. Readers are referred to [8] for more complete and detailed definitions.

Let Ι $\subseteq$ N be a nonempty, possibly infinite, countable set of indices. Assume that Ι is given by Ι = $\{i_1, i_2 \ldots\}$, with $i_j < i_k$ if $j < k$. For a collection of sets Vi , with $i \in$ Ι, we denote by $\Pi_{i\in\Gamma} V_i$ the Cartesian product consisting of the elements $(v_{i1}, v_{i2}, \ldots)$ with $v_i \in Vi$ for each $i \in$ Ι. If $v_i \in Vi$ for each $i \in$ Ι, then $\Pi_{i\in\Gamma} vi$ denotes the element $(v_{i1}, v_{i2}, \ldots)$ of $\Pi_{i\in\Gamma} V_i$. For each $j \in$ Ι and $(v_{i1}, v_{i2}, \ldots) \in \Pi_{i\in\Gamma} V_i$, we define proj$_j((v_{i1}, v_{i2}, \ldots)) = v_j$. If $\phi \neq \zeta \subseteq$ Ι, then proj$_\zeta((v_{i1}, v_{i2}, \ldots)) = \Pi_{j\in\zeta} v_j$.

For the sequel, we let $S = \{C_i \mid i \in$ Ι$\}$ with Ι $\subseteq$ N be a fixed nonempty, indexed set of component automata, in which each C_i is specified as $\left(Q_i, \left(\Sigma_{i,inp}, \Sigma_{i,out}, \Sigma_{i,\text{int}}\right), \delta^i, I_i\right)$, with $\Sigma_i = \Sigma_{i,inp} \cup \Sigma_{i,out} \cup \Sigma_{i,\text{int}}$ as set of actions and $\Sigma_{i,ext} = \Sigma_{i,inp} \cup \Sigma_{i,out}$ is the set of external actions of C$_i$. $\Sigma = \cup_{i\in\Gamma} \Sigma_i$ is the set of actions of S; also we have $Q = \Pi_{i\in\Gamma} Q_i$ as the state space of S.

Component automata interact by synchronizing on common actions. Not all automata sharing an action have to participate in each synchronization on that action. This leads to the notion of a complete transition space consisting of all possible combinations of identically labeled transitions.

Definition 1. A transition $(q, a, q') \in Q \times \Sigma \times Q$ is a *synchronization* on a in S if for all $i \in$ Ι, (proj$_i(q)$, a, proj$_i(q')) \in \delta^i$ or proj$_i(q)$ = proj$_i(q')$, and there exists $i \in$ Ι such that (proj$_i(q)$, a, proj$_i(q')) \in \delta^i$.

For $a \in \Sigma$, $\Delta_a(S)$ is the set of all synchronizations on a in S. Finally $\Delta(S) = \cup_{a\in\Sigma} \Delta_a(S)$ is the set of all synchronizations of S.

Given a set of component automata, different synchronizations can be chosen for the set of transitions of a composed automaton. Such an automaton has the Cartesian product of the states of the components as its set of states. To allow hierarchically constructed systems within the setup of team automata, a composed automaton also has internal, input, and output actions. It is assumed that internal actions are not externally observable and thus not available for synchronizations. This is not imposed by a restriction on the synchronizations allowed, but rather by the syntactical requirement that each internal action must belong to a unique component: S is *composable* if $\Sigma_{i,\text{int}} \cap \bigcup_{j\in\Gamma\setminus\{i\}} \Sigma_j = \phi$ for all $i \in$ Ι.

Moreover, within a team automaton each internal action can be executed from a global state whenever it can be executed by its component at the current local state. All this is formalized as follows.

Definition 2. Let S be a composable set of component automata. Then a *team automaton* over S is a transition system $\mathrm{T} = \left(Q, \left(\Sigma_{inp}, \Sigma_{out}, \Sigma_{\text{int}}\right), \delta, I\right)$, with set of states $Q = \Pi_{i\in\Gamma} Q_i$ and set of initial states $I = \Pi_{i\in\Gamma} I_i$, actions $\Sigma = \cup_{i\in\Gamma} \Sigma_i$ specified by $\Sigma_{\text{int}} = \cup_{i\in\Gamma} \Sigma_{i,\text{int}}$, $\Sigma_{out} = \cup_{i\in\Gamma} \Sigma_{i,out}$, $\Sigma_{inp} = (\cup_{i\in\Gamma} \Sigma_{i,inp}) \setminus \Sigma_{out}$ and transitions $\delta \subseteq Q \times \Sigma \times Q$ such that $\delta \subseteq \Delta(S)$ and moreover $\delta_a = \Delta_a(S)$ for all $a \in \Sigma_{\text{int}}$.

As definition 2 implies, one of the important and useful properties of TA compared to other models is that there is no unique Team automata composed over a set of component automata, but a whole range of Team Automata distinguishable only by their synchronizations can be composed over this set of component automata. This feature enables Team automata to be architecture and synchronization configurable, moreover, it makes possible to define a wide variety of protocols for the interaction among components of a system.

Two other definitions effectively used in our algorithm are "subteams" and "communicational actions" that we briefly introduce. Reference,[8] supports detailed definitions.

Definition3. A pair Ci,Cj with i,j$\in \Gamma$, of component automata is *communicating* (in S) if there exists an $a \in (\Sigma_{i,ext} \mathrm{Y} \Sigma_{j,ext})$ such that $a \in (\Sigma_{i,inp} \mathrm{I} \Sigma_{j,out}) \mathrm{Y} (\Sigma_{j,inp} \mathrm{I} \Sigma_{i,out})$.

Such an *a* is called a *communicating* action (in S). By Σ_{com} we denote the set of all *communicating actions* (in S).

Definition 4. Let $\mathrm{T} = (\Pi_{i\in\Gamma} Q_i, (\Sigma_{inp}, \Sigma_{out}, \Sigma_{\mathrm{int}}), \delta, \Pi_{i\in\Gamma} I_i)$ be a team automaton over the composable system S, and let $J \subseteq \Gamma$. Then the *subteam* of *T* determined by *J* is denoted by $SUB_J(T)$ and is defined as $SUB_J(T) = (\Pi_{j\in J} Q_j, (\Sigma_{j,inp}, \Sigma_{j,out}, \Sigma_{j,\mathrm{int}}), \delta_J, \Pi_{j\in J} I_j)$, where:

$\Sigma_{J,\mathrm{int}} = \mathrm{Y}_{j\in J} \Sigma_{j,\mathrm{int}}, \Sigma_{J,out} = \mathrm{Y}_{j\in J} \Sigma_{j,out}, \Sigma_{J,inp} = (\mathrm{Y}_{j\in J} \Sigma_{j,inp}) \setminus \Sigma_{J,out}$

and for all $a \in \Sigma_J = \mathrm{Y}_{j\in J} \Sigma_j$, $(\delta_J)_a = proj_J^{[2]}(\delta_a) \mathrm{I} \Delta_a(\{C_j | j \in J\})$.

The transition relation of a subteam of T determined by some $J \subseteq \Gamma$ is obtained by restricting the transition relation of T to synchronizations among the components in $\{C_j | j \in J\}$. Hence, in each transition of the subteam, at least one of the component automata is actively involved. This is formalized by the intersection of $(\delta_J)_a = proj_J^{[2]}(\delta_a)$ with $\Delta_a(\{C_j | j \in J\})$, for each action *a*, as in each transition in this complete transition space, at least one component from $\{C_j | j \in J\}$ is active.

3 Proposed Framework

In this section, we describe an extension made to UML to become consistent, and could be used as our input model. Then we introduce an algorithm to transform extended UML models of software architecture to formal descriptions of Team Automata. We called this algorithm UML2TA. Finally, a performance model is described over TA, to evaluate performance aspects of software architecture. Fig.1. shows the input models and the overall steps of our framework.

3-1. UML2TA: An algorithm for transforming software architecture to Team Automata.

UML diagrams are highly understandable and are widely used by software developers. New versions of UML (UML 2.X) have enhanced notations for specifying component-based development and software architectures. [1, 15]

Since our target model-TA, is highly formal, direct translation of UML to TA is problematic. Therefore, we first provided formal definitions of UML model elements to create a consistent input model. Static structure of software architecture is described with UML 2 Component Diagram, while the interaction among components is described by Sequence Diagrams. Because of space limitation, we ignore describing details of the algorithm (UML2TA) and formal descriptions which we added to initial UML models. Readers are referred to [20] for a complete explanation of our framework. However, in this paper a comprehensive example of applying our framework on a casestudy will be described.

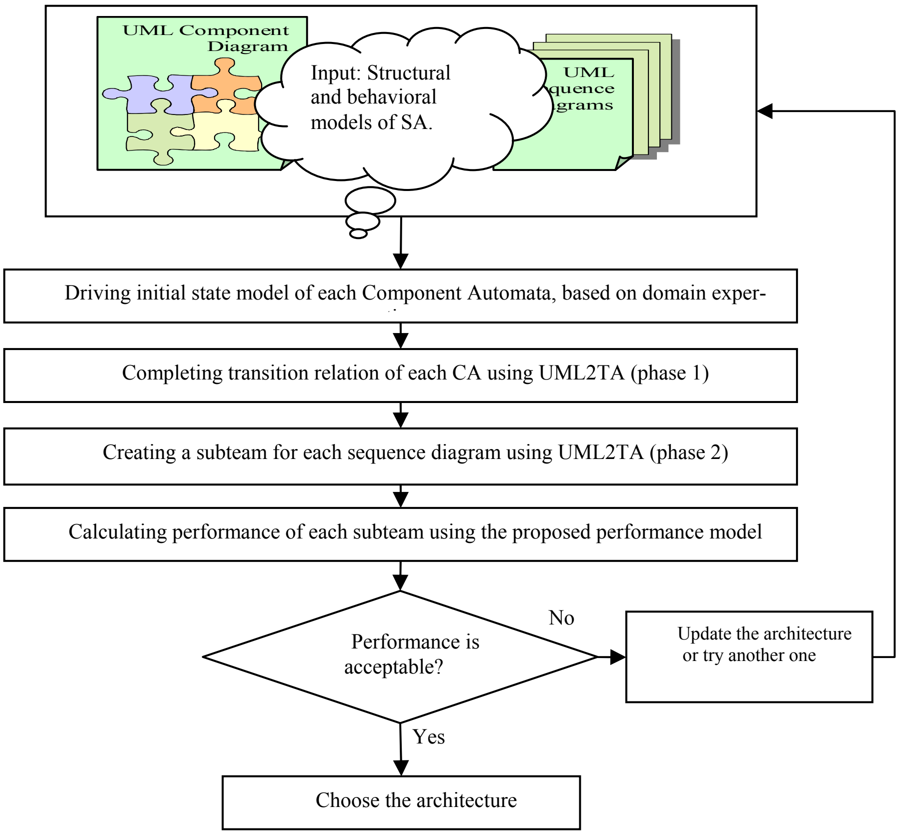

UML Component Diagram
Input: Structural and behavioral models of SA.
UML quence grams
Driving initial state model of each Component Automata, based on domain exper-
Completing transition relation of each CA using UML2TA (phase 1)
Creating a subteam for each sequence diagram using UML2TA (phase 2)
Calculating performance of each subteam using the proposed performance model
Performance is acceptable?
No
Update the architecture or try another one
Yes
Choose the architecture

Fig. 1. Overall steps in the framework to formally specify and evaluate software architecture.

3-2. A Performance Model over TA Specifications

TA model achieved by UML2TA, is a formal foundation for software architecture which can be used for evaluating several attributes (For example in [6], [7] TA has been used for security analysis of groupware systems). In this section we introduce a model to evaluate performance of software architecture described by team automata. In this way, two features have been considered for evaluating performance:

a) Performance specifications of components communication. In our performance model, we have considered a delay for each synchronization within a subteam.

b) The granularity of the performance analysis. Performance can be analyzed as either behavior-dependent or behavior-independent. For example, performance can be defined by processing time of the entire component or processing time of each service invocation in the component. In our model, performance is considered at the service level. Since service requests to a software component are assumed to be input actions to corresponding component automata, we assign a processing time to each input action (These data are again obtained from existing similar systems). According to suggestions a and b, we can extend Team automata models to include performance information as follows:

For each Component Automata a processing-time function P and a delay function P' is defined as follows:

P= $\{ (a,r) | a \in \Sigma_{i,inp}$, r is the processing time corresponding to action $a\}$

P$'$= $\{ (\theta,d) | \theta \in \delta_i$, d is the delay corresponding to transition θ }

We now model each Component Automata in the architecture with the extension of performance model as follows:

$$CP_i = ((Q_i, (\Sigma_{i,inp}, \Sigma_{i,out}, \Sigma_{i,\text{int}}), \delta_i, I_i), P_i, P_i') \qquad \textbf{(1)}$$

Delays of transition within a component could be ignored (comparing with communication delay between components, especially for distributed components). If we assume components interactions synchrony and sequential, then we can consider a whole subteam as a complex server [19] whose mean service time is equal to summation of service time of input actions (those which are synchronized) plus all synchronization delay in the subteam. Thus, if $\theta_i \in \delta_{J_k}$ be the ith synchronization in $SUB_{J_k}(\tau)$ and $\Sigma_{J_k,com}$ be the set of all communicating actions in $SUB_{J_k}(\tau)$ and $A \subseteq \Sigma_{J_k,com}$, $A = \{a_1, a_2, ..., a_m\}$, $m = |\delta_{J_k}|$) be the set of communication actions which are synchronized within $SUB_{J_k}(\tau)$, then we have:

$$\frac{1}{\mu_k} = \sum_{i=1}^{m} (P'(\theta_i) + P(a_i)) \qquad \textbf{(2)}$$

, Where $\frac{1}{\mu_k}$ is mean service time of scenario k (corresponding to SD$_k$) which has been modeled by subteam, $SUB_{J_k}(\tau)$.

Now suppose that software has k independent scenario whose probability of request by users is f$_k$ and suppose, λ is the total input rate of requests to the system (When a request for a scenario arrives while a previous one has not been answered, the new request will be queued). The system response time corresponding to architecture under evaluation is equal to R=1/(λ-μ); where μ is total service rate and is calculated by the following formulas:

$$\frac{1}{\mu} = \sum_{i=1}^{k} \frac{f_i}{\mu_i} \qquad (3)$$

4 An Application System Example

We evaluated UML2TA method on a part of a web-service software architecture. In this example, we have a component diagram describing major components and connectors (Fig 2), and a sequence diagram (Fig 3) describing components interaction corresponding to a scenario where some end user requests the web content available from /ping URL (This system has been used as a case-study in [17] in a different scope). We use extension defined in [18] for sequence diagrams.

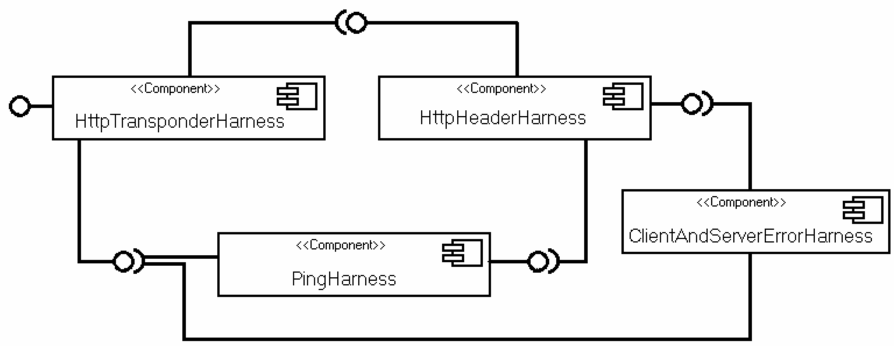


Fig. 2. Component Diagram of a part of Web-Service Software

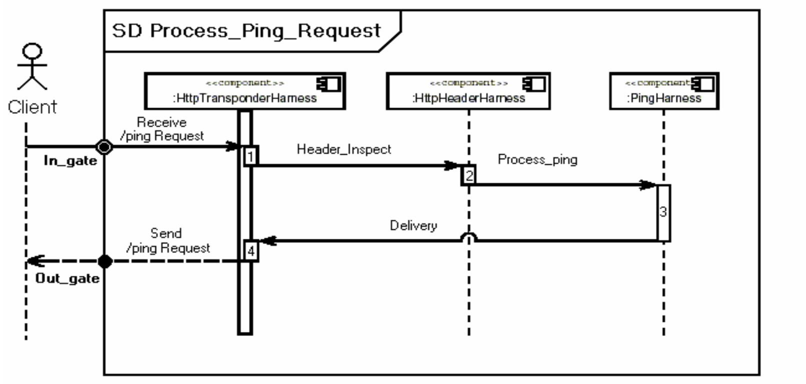


Fig. 3. Sequence Diagram specifying components interaction for '/ping' Scenario.

According to UML2TA, first, we manually model each software component with a Component Automata from informal behavioral descriptions which has been briefly mentioned in Table 1.

Table 1. CA models of Web-service Software components.

Component Automata model of HttpTransponderHarness **:**	Component Automata model of HttpHeaderHarness	Component Automata model of PingHarness

Actions:
Input actions : /ping_req , delivery.
Output actions: /ping_resp , header_inspect.
Internal action: new_thread_allocation.
 	
State Variables:
Process_Input :{0,1}
Prepare_resp: {0,1}
 	
Transitions(per actions):
 	
/ping_request:
Effect: process_inp = 1;
delivery:
Effect: prepare_resp = 1;
/ping_resp:
Preconditions: prepare_resp=1;
Effects: prepare_resp=0;
/header_inspect:
Preconditions: process_inp=1;
Effects: process_inp:= 1;
	Actions:
	
	Input actions: header_inspect;
	Output actions: proc_ping;
	Internal action: none;
	
	State Variables:
	Identify_request_type : {0,1};
	
	Transitions: (per actions)
	
	header_inspect:
	Effects: Identify_request_type := 1;
	
	proc_ping:
	Preconditions: Identify_request_type = 1;
	Effects: Identify_request_type = 0;
		Actions:
		Input action: proc_ping;
		Output action: delivery;
		Internal action: None;
		
		State Variables:
		Generate_response:{0,1};
		
		Transitions (per actions):
		
		Proc_ping:
		Effects: generate_response = 1;
		
		delivery:
		Preconditions: generate_response = 1;
		Effects: generate_response = 0;

If we have all scenarios of the system, then we can model TA of the overall system; However according to algorithm UML2TA, for each scenario we can create a sub-team; therefore if components HTTPTransponderHarness, HttpHeaderHarness and PingHarness correspond to component automata C_1, C_2 and C_3, respectively, then we have:

$$SUB_J(\tau)=\left(Q_J,(\Sigma_{inp},\Sigma_{out},\Sigma_{\mathrm{int}}),\delta_J,\prod_{j\in J} I_j\right) where\ J=\{1,2,3\},\ Q_J=\prod_{j\in J} Q_j\ ,$$

$$\Sigma_{inp}=\{/\,ping\_req\}\ ,$$

$$\Sigma_{out}=\{/\,ping\_resp, Header\_insp, proc\_ping, delivery\},$$

$$\Sigma_{\text{int}} = \{new\_thread\},$$

$$Q_J = \{(w,w',w''),(w,w',gr),(w,I,w''),(w,I,w),(pi,w',w''),(pi,w',gr),$$
$$(pi,I,w''),(pi,I,gr),(po,w',w''),(po,w',gr),(po,I,w''),(po,I,w'')\}$$

and briefly we have:

$$\delta_J = \{((w,w',w''),/\,ping\_req,(pi,w',w'')),((pi,w',w''),/\,Header\_insp,(w,I,w'')),$$
$$\ldots,((po,w',w''),/\,ping\_resp,(w,w',w''))\}$$

4-2. Performance Evaluation and Architectural Changes

In Section 4-1, UML2TA was applied on Web-Service Software Architecture and relevant component automata and subteam were generated. In this section, we represent results of applying UML2TA on a different version of previous architecture, and show how an architect can choose more suitable architecture regarding overload condition using our framework. Before that, we briefly explain overload and flash crowd conditions in systems especially in web.

In web service provision, it is possible for the unexpected arrival of massive number of service requests in a short period; this situation is referred to as a flash crowd. This is often beyond the control of the service provider and has the potential to severely degrade service quality and, in the worst case, to deny service to all clients completely. It is not reasonable to increase the system resources for short-time flash crowd events. Therefore, if Web-Service Software could detect flash crowds at runtime and change its own behavior proportional to occurred situation, then it can resolve this bottleneck. In the new architecture, a component has been added to the previous one, i.e. PingFactoryHarness; it controls response time of each request, detects the flash crowd situation and directs PingHarness to change its behavior proportional to occurred condition. At the end of this section, results of analysis of both architectures are presented and it is shown how the new architecture is more effective than the old one facing flash crowds. Thanks to Lindsey Bradford for giving us the initial performance data of the system.

Fig.4. shows component diagram along with performance data and the new component PingFactoryHarness. We have used notations defined in [15] by OMG Group.

In new architecture (sequence diagram of the new scenario has been ignored) HttpTransponderHarness takes a snapshot of the system time just after the request text has been received and just before that text is sent to the client. This snapshot data is used to calculate an elapsed time for responding to the request later in sequence and finally to detect abnormal conditions (e.g. flash crowd. The component PingHarness is an updated component; it has the ability to change its behavior when it receives relevant message from PingFactoryHarness. PingFactoryHarness receives the elapse time from HttpTransponderHarness and decides if change is needed in the behavior of PingHarness. PingHarness then receives the direction to change behavior.

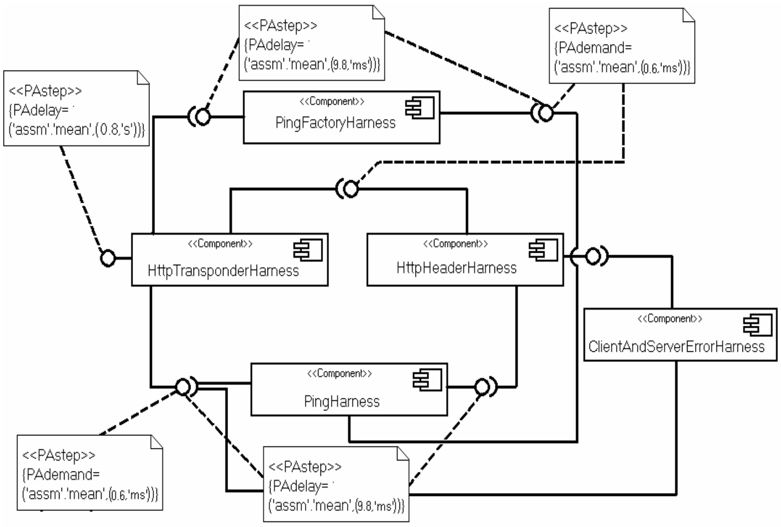


Fig. 4. Extended Component Diagram of new Web-Service Software architecture.

In experiments performed on both architecture models, in an overload condition, we observed that service times are not stable. It is because of sudden increase in requests for the system resources. This situation does not follow the flow balancing condition in usual queuing models [16], thus formulating an analytic approach covering the situation is problematic. Hence, we use simulation for this part of work and the results of the simulation were used to calibrate analytic model introduced in Section 3-2. We summarized the results of our hybrid method to Tables 2 and 3 for the original and updated architecture, respectively.

Table 2. Performance data of the first architecture.

Request per Sec.	Response time(ms)			*Average number of responses per Sec.*
	Avg.	Min.	Max.	
2	285.9	284.8	373.9	*2*
3	1906.3	305.5	7843.5	*0.5*
5	2877.8	428.8	7744.6	*0.2*
10	1180.2	1011.2	1397.5	*0.0*

Table 3. Performance data of updated architecture.

Request per Sec.	Response time(ms)			*Average number of responses per Sec.*
	Avg.	Min.	Max.	
2	223.2	222.2	270.8	*2*
3	229.9	222.3	241.2	*3.1*
5	7478.1	239.1	10673	*3*
10	8683.4	255.7	10706	*3.4*

The difference between the two architectures at the request rate of 10 per second is interesting. At first glance, it seems that the first architecture response times are much better than the second, however, comparing throughput between both architectures indicates that first architecture delivered almost no responses at request rate higher than 5. In contrast, the second architecture continued to deliver responses, despite the worse response time.

5 Conclusion and Future Work

In this paper, a framework was introduced to formally specify and evaluate Software Architectures. SA specification is initially described in UML2.0 which is the input model for a transformation algorithm called UML2TA introduced within our framework. UML2TA transforms SA descriptions in UML2.0 to a formal model called Team Automata (TA). TA is inspired by Input/Output Automata and has been used in the literature for modeling components interaction in groupware systems. It has also a great generality and flexibility to specify different aspects of components interaction, so it could be best fit to model dynamics of SA. By modeling software architectures with a powerful model such as TA, we have suggested a rigorous basis to evaluate (and also verify) functional and non-functional attributes of SA. Furthermore, we extended usual TA model to include performance aspects which could be involved in UML2.0 diagrams. We also proposed a performance evaluation model over TA specifications. Finally we applied our framework to the architecture of a web-service software and showed how the framework could be used practically to anticipate performance aspects of an architecture.

In future work, we decide to firstly, promote our performance model to support a wide variety of interactions such as asynchronous, anonymous in distributed environments. Secondly, we are going to enhance our framework to include other non-functional attributes e.g. security; this issue will facilitate simultaneous evaluation of several attributes regarding their conflicting natures.

References

1. Ivers, P. Clements, D. Garlan, R Nord, B. Schmerl, J. R. Oviedo Silva. Documenting Component and Connector Views with UML2.0. Technical report, CMU/SEI, TR-008 ESC-TR-2004-008, 2004.
2. L. Bass, P. Clements, R. Kazman, Analyzing development qualities at the architectural level, in: Software Architectures in Practice, SEI Series in Software Engineering, Addison-Wesley, Reading, MA, 1998.
3. K. Cooper, L. Dai, Y. Deng, Performance modeling and analysis of software architectures: An aspect-oriented UML based approach. Science of Computer Programming, Elsevier ,2005.
4. J.J.Li , J.R. Horgan , Applying formal description techniques to software architectural design, The journal of Computer Communications, 23,1169-1178, 2000.
5. M. Shaw, D. Garlan, Software Architecture—Perspectives on an Emerging Discipline, Prentice Hall, Englewood cliffs, NJ, 1996.

6. Maurice H. ter Beek, Gabriele Lenzini, Marinella Petrocchi, Team Automata for Security– A Survey –,Electronic Notes in Theoretical Computer Science, 128 (2005) 105–119.
7. L. Egidi, M. Petrocchi, Modelling a Secure Agent with Team Automata, The Journal of Electronic Notes in Theoretical Computer Science 142 (2006) 111–127.
8. M. Beek, C. Ellis, J. Kleijn, and G. Rozenberg. Synchronizations in Team Automata for Groupware Systems. Computer Supported Cooperative Work—The Journal of Collaborative Computing, 12(1):21–69, 2003.
9. N. A. Lynch and M. R. Tuttle. An introduction to input/output automata. CWI Quarterly, 2(3):219–246, September 1989.
10. Luca de Alfaro and Thomas A. Henzinger. Interface Automata. In Volker Gruhn, editor, Proceedings of the Joint 8th European Software Engeneering Conference and 9th ACM SIGSOFT Symposium on the Foundation of Software Engeneering (ESEC/FSE-01), volume 26, 5 of Software Engineering Notes, pages 109–120. ACM Press, September 10–14 2001.
11. Luca de Alfaro and Thomas A. Henzinger. Interface-Based Design. In Proceedings of the Marktoberdorf Summer School, Kluwer, Engineering Theories of Software Intensive Systems, 2004.
12. M. Sharafi, F Shams Aliee, A. Movaghar. A Review on Specifying Software Architectures Using Extended Automata-Based Models, FSEN07, LNCS 4767,423-431, Springer-Verlag Heidelberg, 2007.
13. C. Ellis. Team Automata for Groupware Systems. In Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work: The Integration Challenge (GROUP'97), pages 415–424. ACM Press, New York, 1997.
14. Lubo¡s Brim, Ivana Cern , Pavl´ýna Va¡rekov, Barbora Zimmerova , ComponentInteraction Automata as a Verification Oriented Component-Based System Specification, 2005.
15. Object Management Group. UML Profile, for Schedulability, Performance, and Time. OMG document ptc/2002-03-02, http://www.omg.org/cgi-bin/doc?ptc/2002-03-02.
16. K. Kant and M.M. Sirinivasan. Introduction to Computer Performance Evaluation, McGrawhill Inc. 1992
17. Lindsay William Bradford. Unanticipated Evolution of Web Service Provision Software using Generative Object Communication. Final report of PhD thesis , Faculty of Information Technology Queensland University of Technology, GPO Box 2434, Brisbane Old 4001, Australia, 10 May, 2006.
18. A. Di Marco, P. Inverardi. Compositional Generation of Software Architecture Performance QN Models Dipartimento di Informatica University of L'Aquila Via Vetoio 1, 67010 Coppito, L'Aquila, Italy, 2004.
19. Federica Aquilani, Simonetta Balsamo , Paola Inverardi, Performance analysis at the software architectural design level, Performance Evaluation 45, Elsevier, (2001) 147–178.
20. M.Sharafi, Developing a Framework to Formal Specification and Evaluation of Software Architectures, Final Report of PhD Thesis, Faculty of Computer Engineering, Azad University of Tehran (Science and research branch of), August 2007.

Double Life Cycle

Alejandro Canales[1], Rubén Peredo[1] and Ivan Peredo[1]

[1] Computer Science Research Center of National Polytechnic Institute, Col. Nueva Industrial
Vallejo, Del. Gustavo A, Madero, D.F. 07738, Mexico City, Mexico
{acc, peredo}@cic.ipn.mx

Abstract. In this paper a new life cycle called "double" for the development of Web-Based Education Systems (WBES) is presented. These WBES are centered in the student and adapted to their personal necessities in intelligent form. The double life cycle assembles to the patterns of software development and instructional design. On the one hand, the software development pattern is supported under the methods and technical of the Domain Engineering (components), Learning Technology System Architecture of IEEE 1484, Sharable Content Object Reference Model (SCORM) of ADL, a Multi-Agents System and Service-Oriented Architecture for the reusability, accessibility and interoperability of the services. On the other hand, the instructional design pattern incorporates a mental model as the Conceptual Maps to transmit, build and generate appropriate knowledge to this educational environment type.

Keywords: WBES, SiDeC, IRLCOO, Evaluation System.

1 Introduction

Student-centered education pursues that sequencing, authoring content, pedagogic patterns, assessment, and evaluation processes meet the learning goals of the students. Also, the content and evaluation repositories must be suitable for the particular requirements of each individual. But at the same time, they have to be flexible and available for being tailored and used by a wide community of developers and students, respectively. It is necessary the development of a WBES that considers a whole diversity of requirements (technological and pedagogic) and provides the needed functionalities based on the facilities of the Web.

As well, the Domain Engineering has to consider the particular specifications claimed by the WBES in order to tailor solutions. Furthermore, authoring reusable components for content and evaluation tasks have to be fulfilled automatically as well as possible. The components help us to reduce the complexity, managing change, and reuse [1]. Thus, in order to deliver teaching-learning experiences, it is necessary the use of mental model as Conceptual Maps (CM).

Wherefore, the purpose of this paper is to show a new life cycle for the WBES development. In order to achieve this goal, this paper is organized as follows: In Section 2, the double life cycle is presented; whereas in Section 3, the instructional design pattern are analyzed. In Section 4 the software development pattern is

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 29, 2007, pp. 161-170

Received 01/07/07
Accepted 19/10/07
Final version 24/10/07

described. Afterwards, in Section 5 the SiDeC (authoring content) and Evaluation System are respectively depicted. Finally, in Section 6 the conclusions and future work are discussed.

2 Double Life Cycle

If the actual WBES are reviewed, we can note that the pedagogic aspect is considered but not in significant form. We believe in the incorporation of pedagogical aspects or an instructional design pattern inside the applications development process. Because the instructional design is a process that allows translating pedagogic principles of learning, in an action plan to develop learning content and activities as well as information and evaluation resources. It is a systemic process that allows developing the plan step by step and the results of each one serve as input to the consequent steps [2].

In conclusion, this suggests the incorporation of an instructional design pattern inside the software development process for WBES, and consequently we need a life cycle with this characteristic. But if the international standards as ISO/IEC 9001, 9003, 12207 and 15504 of Software Engineering are reviewed [3], they only contemplate the software development from a technological perspective basically and they do not deepen inside the requirements of the software application field, such as in this case is the education. In all life cycles referenced, only in the requirements step is possible to include the pedagogical requirements. We consider that the pedagogical aspects must intervene in a direct way in all processes inside the software development.

In Fig. 1 a double life cycle pattern is shown. It joins the software development pattern and the instructional design pattern, with the purpose of not only developing educational software based on technological or computational questions, but by adding, also, cognitive elements that collaborate in the acquisition of the students' knowledge.

The double life cycle provides to the developer facilities to go and come from a pedagogic extreme to other technological for the application design. This is because the software has an educational purpose; therefore the pedagogic principles guide the software technological development. Also, the double life cycle helps to the development and delivery fasts of software parts.

In the first phase of the double life cycle the software objectives are defined. For the second phase the software requirements so much pedagogic as technological are obtained. In the third phase the learning activities and the global design of the system are developed. It can be observed that the double life cycle allows in these three phases to go back and to advance among them during the software development with the purpose of incorporating the necessary information to design the software until reaching the goals traced for each phase.

In phase 4, a software version is developed that later in fifth phase is shown to the user. The result is refined in function of the user's feedback. This cycle is repeated until reaching appropriate software, since the characteristic of the double life cycle is in fact to be evolutionary. This characteristic is very important, because it provides the possibility to change the product direction to halfway in response to the user's petitions (learner and tutors' developers). If it is used the evolutionary delivery carefully, it can improve the product quality, reduce the code size and produce a more uniform distribution of the development and test resources [4]. Finally, during sixth phase the final version to the user is delivered.

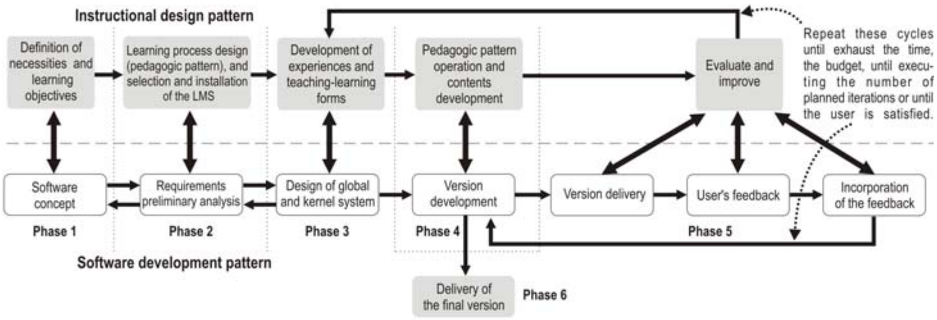

Instructional design pattern
Definition of necessities and learning objectives
Learning process design (pedagogic pattern), and selection and installation of the LMS
Development of experiences and teaching-learning forms
Pedagogic pattern operation and contents development
Evaluate and improve
Repeat these cycles until exhaust the time, the budget, until executing the number of planned iterations or until the user is satisfied.
Software concept
Requirements preliminary analysis
Design of global and kernel system
Version development
Version delivery
User's feedback
Incorporation of the feedback
Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Software development pattern
Delivery of the final version
Phase 6

Fig. 1. Double life cycle.

3 Instructional Design Pattern

The instructional design pattern represents a substantial advance in the teaching-learning process (see Fig. 1). On the one hand, it incorporates cognitive elements (as CM) with the purpose of helping the students to maximize the knowledge acquisition. On the other hand, it is specifically designed to the Web environment.

The instructional design pattern allows generate a process where pedagogic principles of learning can be translated in an action plan for development of activities and learning contents, as well as evaluation and information resources [5].

A CM is a graphical technique used during the teaching-learning process. CM as instructional and learning strategy and as schematic resource or navigation map is used [6].

4 Software Development Pattern

The software development pattern is based on the Domain Engineering (see Fig. 1). The Domain Engineering aims to identify, build, classify and divulge software components [7]. While the traditional Software Engineering aims to the development of software system or application to satisfy some specific requirement for a particular necessity [8].

A software component is "a unit of composition with contractually specified interfaces and explicit context dependencies. A software component can be deployed independently and is object to composition by third parties" [9]. Between the key issues of Domain Engineering is the aim for developing reusable software. Thus, components are widely seen by software engineers as a main technology to address the ‘‘software crisis’’. The Industrial Software Revolution is based upon component software engineering. Between the reasons that explain the relevance of the component-oriented programming are: the high level of abstraction offered by this paradigm, and the current trends for authoring reusable component libraries, which support the development of applications for different domains. In addition, the three major goals pursued by Component-Oriented Programming are considered: Conquering complexity, managing change, and reusability [1].

4.1 IRLCOO Profile

With regards to the SCORM terminology, Intelligent Reusable Learning Components Object Oriented (IRLCOO [10]) are a special type of Sharable Content Object (SCO) that represent an alternative approach to content development, which is based on the Reusable Learning Object Strategy Definition stated by Advanced Distributed Learning (ADL) [11], IRLCOO are self-contained learning components that are organized as learning resources, which are accessed independently. IRLCOO are digital resources that can be reused to support WBES thru: live, streaming and prerecorded video and audio, a course module, animations, graphics, applications, Web pages, PDF and Office documents, and other pieces devoted to deliver complete experiences.

IRLCOO were developed with Flash 8. Flash is an integrator of media and have a powerful programming language denominated ActionScript 2.0 [12]. This language is completely Object Oriented and enables the design of client components that allows multimedia content. In addition, this IRLCOO development platform owns certain communication functionalities inside the Application Programming Interface (API) of the LMS, Multi-Agent System (MAS), and different frameworks, as AJAX [13], Hibernate [14], Struts [15], etc.), and dynamic load of Assets in Run-Time.

IRLCOO are meta-labeled with the purpose of complete a similar function as the product bar codes, which are used to identify the products and to determine certain characteristics specify of themselves. This contrast is made with the meta-labeled Resource Description Framework (RDF-XML) [16].

From a pedagogical perspective, each IRLCOO might play a specific role within an instructional design pattern. IRLCOO can be re-assembled to create new courses or sequenced to tailor individual learning paths. The use of IRLCOO deals with the following key issues: (1) The IRLCOO must be able to communicate with learning management systems (LMS) using a standardized method that does not depend on the system. (2) The sequence system, that usually is a module of the LMS, defines the navigation rules that a learner uses to move between IRLCOO. (3) IRLCOO own a description that enables designers to seek and find the appropriate IRLCOO for the right job. These considerations offer clear benefits, such as: IRLCOO enable mass-customization of learning with more personalized and content ‘just for the learner’, and for authors, there is the opportunity to seek existing IRLCOO within the organization or from external providers in order to reuse them, save time and money.

Furthermore, ActionScript 2.0 adds the component WebServiceConnector to connect to Web Services (WS) from the IRLCOO. The WebServiceConnector component enables the access to remote methods offered by a LMS through SOAP protocol. This gives to a WS the ability to accept parameters and return a result to the script, in other words, it is possible to access and join data between public or own WS and the IRLCOO.

5 SiDeC

In order to facilitate the development of learning content, it was built an authoring tool called eCourses Development System (SiDeC - Sistema de Desarrollo de eCursos) [17]. SiDeC is a tool based on components, which facilities the authoring content by tutors who are not willing for handling multimedia applications. In addition, the Structure and Package of content multimedia is achieved by the use of IRLCOO, as the lowest level of content granularity.

According to the IEEE 1484 LTSA specification [18], SiDeC is used to construct Web-based courseware from the stored IRLCOO (Learning Resources) though of the coach in the way illustrated in Fig. 2.

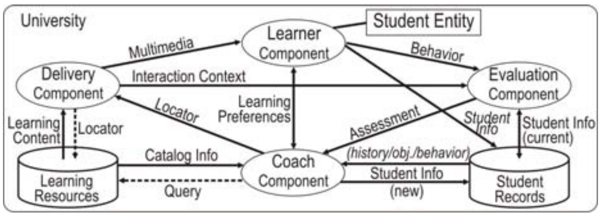
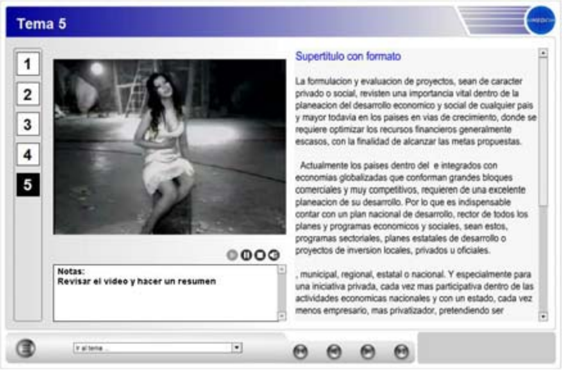


Fig. 2. LTSA of IEEE 1484 (layer 3).

SiDeC has a metadata tool for the generation of IRLCOO and on-line courses (see Fig. 3). This courseware estimates learners' metrics with the purpose to tailor their learning experiences. These deliverables are compliance with the specifications of the IRLCOO and with learning items of SCORM 1.2 Models (Content Aggregation, Sequencing and Navigation, and Run Time Environment) [11]. Metadata represent the specific description of the component and its contents, such as: title, description, keywords, learning objectives, item type, and rights of use. The metadata tool provides templates for entering metadata and storing each component in the SiDeC or another IMS/IEEE standard repository.

At this moment, the SiDeC lesson templates are based on the cognitive theory of CM [6], but in the future we will consider others theories such as: Based-problems learning (BPL), the cases method, and the project method.

In Fig. 4 the SiDeC implements the conceptual map as a navigation map, allowing to the learner interacts with content objects along the learning experiences. These experiences follow an instructional-teaching strategy. There kind of strategies carry out modifications of the learning content structure. Such modifications are done by the learning experience designer with the objective of provide significant learning, and to teach the learners how to think [2].



Tema 5

1
2
3
4
5

Notas:
Revisar el video y hacer un resumen

Supertitulo con formato

La formulacion y evaluacion de proyectos, sean de caracter privado o social, revisten una importancia vital dentro de la planeacion del desarrollo economico y social de cualquier pais y mayor todavia en los paises en vias de crecimiento, donde se requiere optimizar los recursos financieros generalmente escasos, con la finalidad de alcanzar las metas propuestas.

Actualmente los paises dentro del e integrados con economias globalizadas que conforman grandes bloques comerciales y muy competitivos, requieren de una excelente planeacion de su desarrollo. Por lo que es indispensable contar con un plan nacional de desarrollo, rector de todos los planes y programas economicos y sociales, sean estos, programas sectoriales, planes estatales de desarrollo o proyectos de inversion locales, privados u oficiales.

, municipal, regional, estatal o nacional. Y especialmente para una iniciativa privada, cada vez mas participativa dentro de las actividades economicas nacionales y con un estado, cada vez menos empresario, mas privatizador, pretendiendo ser

Fig. 3. Learning content generated for the SiDeC.

Based on a CM the SiDeC represents the course structure that the student follows. The delivery process identifies a learning content for the student. The learning content owns IRLCOO associated with it. Afterwards, the delivery process launches (see Fig. 2) the IRLCOO and presents them to the student. Fig. 4 depicts how the course structure is organized as a manifest, and the learning content can be interpreted in a Learning Content Tree. A Learning Content Tree is a conceptual structure of learning activities managed by the delivery process for each learner. The Tree representation is just a different way for presenting content structure and navigation. This information is found in the manifest that is defined into the SCORM Content Aggregation Model (CAM) [11].

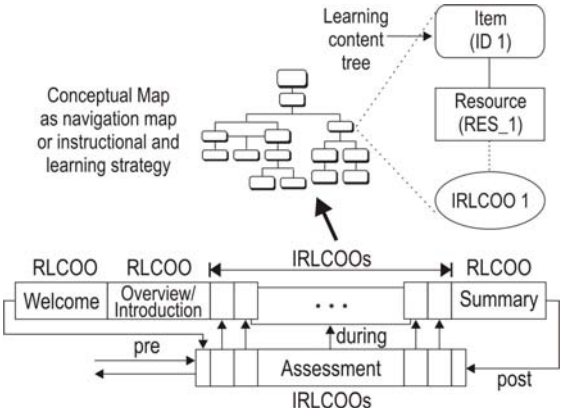


Fig. 4. The course structure.

According with Fig. 4, the next fragment code shows how the course structure is organized as *imsmanifest.xml*.

```
<manifest>
  <organizations>
     <organization>
        <item>
          <item identifier=”ID1” identifierref=”RES_1”>
             <adlnav:presentation>
                <adlnav:navigationInterface>
                   <adlnav:hideLMSUI>previous
                      </adlnav:hideLMSUI>
                   <adlnav:hideLMSUI>continue
                      </adlnav:hideLMSUI>
                </adlnav:navigationInterface>
             </adlnav:presentation>
             <imsss:sequencing>
                <imsss:controlMode choice="false"
                   flow="true"/>
                <imsss:rollupRules
                   rollupObjectiveSatisfied="false"/>
             </imsss:sequencing>
          </item>
        <item>   ...   </item>
    </organization>
  </organizations>
   <resources>
      <resource identifier=”RES_1”>   ...   </resource>
       ...
</manifest>
```

6.1 Evaluation System

The Evaluation System is designed under the same philosophy used for the SiDeC. The functionality of the Evaluation System lays on the analysis of the learner’s profile, which is built during the teaching-learning experiences. The profile is based on metrics that elicited from the learner’s behavior at Run-Time. These measures are stored into the learner records that compose the profile (see Fig. 2). The generation of new sequences of courses is in function of the results obtained, besides the account of the adaptation level.

The Evaluation System combines IRLCOO, additional meta-labels, and a Java Agent platform. Also, some technologies of the Intelligence Artificial field are considered in order to recreate a Semantic Web environment. Semantic Web aims for assisting human users to achieve their online activities. Semantic Web offers plenty of advantages, such as: reduction of the complexity for potential developers, standardization of functionalities and attributes, definition of a set of specialized APIs, and deployment of a SWP.

All resources have a Universal Resource Identifier (URI). An URI can be a Unified Resource Locator (URL) or some other type of unique identifier. An identifier does not necessarily enable access to a resource. The XML layer is used to define the

SCORM metadata of IRLCOO that are used to interchange data over the Web. XML Schema tier corresponds to the language used to define the structure of metadata [19]. The RDF level is represented by the language used for describing all information and metadata sorts [16]. The Meta-ontology tier is devoted to define the semantic for establishing the usage of words and terms in the context of the vocabulary. Logical level corresponds to the reasoning used to establish consistency and correctness of data sets and to infer conclusions that are not explicitly stated [20].

In resume, the components and operation of the SiDeC and Evaluation System are outlined in Fig. 5. Basically the Evaluation System is fulfilled through two phases. The first phase is supported by the LMS, and is devoted to present the course and its structure. All the actions are registered and the presentation of the contents is realized with IRLCOO content. The evaluations are done by evaluating IRLCOO and in some cases by simulators based on IRLCOO. These processes are deployed by the Framework of Servlets/Java Server Pages/JavaBeans.

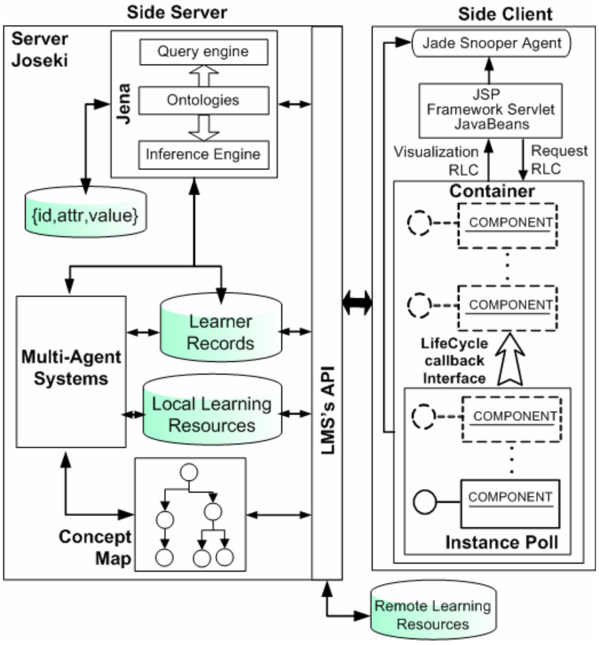


Fig. 5. Semantic Web Platform for WBE.

The second phase analyzes the learner's records carried out by the Server based on JADE MAS. This agent platform owns seven agents: Snooper, Buffer, Learner, Evaluation, Delivering, Coach, and Info. The fundamental idea is to automate the learner's analysis through the coach, and to give partial results that can be useful for the learner's final instruction. These agents are implemented as Java-Beans programs, which are embedded in the applications running both at the client and server sides. These agents employ the dynamic sequencing to change the course or assessment sequence. The sequencing is defined for the instructional strategy based on CM and it employs the SCORM Sequencing/Navigation. Once the necessary information is received (sequence, kind of IRLCOO and localization, etc.), this is represented as a string dynamically constructed by the rule-based inference engine known as JENA [21] and JOSEKI server [22], to generate dynamic feedback.

6.2 Semantic Web Platform

The overall architecture of Semantic Web Platform, which includes three basic engine representing different aspects, is provided in Fig. 5.

1. The query engine receives queries and answers them by checking the content of the databases that were filled by info agent and inference engine.

2. The database manager is the backbone of the entire systems. It receives facts from the info agent, exchanges facts as input and output with the inference engine, and provide facts to the query engine.

3. The inference engine use facts and Meta-ontologies to derive additional factual knowledge that is only provided implicated. It frees knowledge providers from the bur-den of specifying each fact explicitly.

Again, Meta-ontologies are the overall structuring principle. The info agent uses them to extracts facts, the inference engine to infer facts, the database manager to structure the database, and query engine to provide help in formulating queries.

JENA was selected as the inference engine. It is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine [21].

While JOSEKI was selected as Web API and server. It is an HTTP and SOAP engine supports the SPARQL Protocol and the SPARQL RDF Query language. SPARQL is developed by the W3C RDF Data Access Working Group [22].

7 Conclusions

This paper has introduced an instance of an adaptive and intelligent WBES. Our approach focus on: reusability, accessibility, and, interoperability of the learning contents, which are built as IRLCOO, as the main component for delivering teaching and evaluation content.

IRLCOO offer a common interface and functionality that makes easy the authoring of learning content that is delivered by dynamic sequencing. The IRLCOO accept feedback via assessment based upon MAS platform. The information provided is considered as a rough data because it is based on parameters elicited from the behavior of the student.

References

1. Wang, A., Qian, K.: Component-oriented programming. John Wiley & Sons, Inc., Publication (pp. 3–5), USA (2005)
2. Díaz-Barriga, F.: Educational strategies for a significant learning (Estrategias docentes para un aprendizaje significativo). 2nd ed. DF, Mc Graw Hill Publication, México (2002)
3. Internationals Standards Organization, ISO 9001, 9003, 12207 and 15504, http://www.iso.org
4. Mc Connel, S.: Informatics Projects - Develop and Management (Desarrollo y Gestión de Proyectos Informáticos). Mc-Graw Hill Editorial, Spain (1997)
5. Díaz, J., Ramírez, V.: Instructional Design pattern (Modelo de diseño instruccional), http://www.uv.mx/jdiaz/DisenoInstrucc/ModeloDisenoInstruccional2.htm
6. Novak, J., Gowin, B.: Learning How to Learn, Cambridge University Press, Cambridge, USA (1984)
7. Simos, M., Klingler, C., Levine, L., Allemang, D.: Organization Domain Modeling (ODM), Guidebook –Version 2.0. Technical Report, Lockheed Martin Tactical Defense Systems, USA (1996)
8. Pressman, R.: Software Engineering – A practical vision (Ingeniería de Software - Un Enfoque Práctico). McGraw-Hill Editorial, Spain (2002)
9. Szyperski, C.: Component Software. Beyond OPP. Addison-Wesley Editorial, USA (1998)
10. Peredo, V. R., Balladares, O. L., Sheremetov, L.: Development of intelligent reusable learning objects for web-based education systems. Expert Systems with Applications, 28(2), 273–283 (2005)
11. Advanced Distributed Learning Consortium, http://www.adlnet.org
12. Macromedia, Inc., http://www.macromedia.com
13. Grane. D., Pascarello, E., James, D.: Ajax in Action. Manning Publications, Greenwich, USA (2006)
14. Peak, P., Heudecker, N.: Hibernate Quickly. Manning Publications, Greenwich, USA (2006)
15. Holmes, J.: Struts: The Complete Reference. Mc Graw Hill – Osborne Publications, Edited by Herbert Schild, California, USA (2004)
16. RDF specification, http://www.w3.org/RDF/default.htm
17. Canales, A., Peredo, R., Fabela, O., Sossa. H.: Architecture for development of WBES based on components and agents. Internarional Conference on Computing, 223-228 (2006)
18. IEEE 1484.1, Draft Standard for Learning Technology - Learning Technology Systems Architecture (LTSA), http://ieee.ltsc.org/wg1
19. XML specifications, http://www.w3.org/XML/
20. Antoniou, G., Van Harmelen, F.: A Semantic Web Primer, The MIT Press, USA (2004)
21. JENA, http://jena.sourceforge.net/
22. JOSEKI server, http://www.joseki.org/

A Secure Electronic Election System for the Mexican Presidential Election

López García M.L., Leon Chávez M.A. and Rodríguez Henríquez F.

Centro de Investigación y de Estudios Avanzados del IPN
Departamento de Computación
mlopez@computacion.cs.cinvestav.mx
francisco@cs.civestav.mx
Benemerita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación
mleon@cs.buap.mx

Abstract. In 2006, the Mexican presidential election offered the opportunity for those Mexican citizens with residency in foreign countries to cast their vote through certified postal mail. The present work proposes an Internet Electronic System (SEVI after its name in Spanish, "Sistema Electrónico de Voto por Internet") that emulates the election of the President of the Mexican United States. SEVI automatizes the voting process by certified postal mail. SEVI was developed using the Unified Software Development Process on a client/server environment, it offers sound security services while it takes into account the electoral laws to guarantee the credibility of the vote.

1 Introduction

Recent advances in communication networks and cryptographic techniques have made possible to consider online voting systems as a feasible alternative to conventional elections [1].

Independently of the electronic device used for sending votes, getting started with a public election includes the creation of electoral laws that define the operation of the system and the exact procedures to be followed in the event of any contingency.

An electronic voting scheme can be defined as an election system that generates electronic ballots which allow registered citizens to cast their votes from an electronic device and to transmit them via Internet towards an electronic electoral urn, where they will be stored and counted at the end of the electoral day.

This type of systems offers a quick and comfortable form of casting a vote; however, these factors are not a substitute for the accuracy of the results and the trust in the electoral process [2]. Therefore, developing an Electronic System of Voting for Internet (SEVI) implies to comply with the following requirements:

- Since the Internet is an insecure means of communication, SEVI should provide sound security services in order to avoid either passive or active

Received 04/07/07
Accepted 19/10/07
Final version 24/10/07

attacks, that is to say that the vote should not be intercepted, observed, amended, erased or fabricated.

- SEVI design should strictly comply with the electoral laws and it should cover all functional requirements that depend from the electoral process to be implemented.
- The basic properties of an electronic voting system must be fulfilled, namely, [3]:
  1. Authentication: only authorized voters should be able to vote.
  2. Fairness: no voter should be able to vote more than one time.
  3. Accuracy: election systems should record and count the votes correctly.
  4. Integrity: it should not be possible to modify, forge, or delete votes without detection.
  5. Anonimity: no one should be able to link any vote with the individual that casted it and voters should not be able to prove how they voted.
  6. Transparency: the voters should understand and know the voting process.
  7. Verification and Accountability: it should be possible to verify that the votes have been correctly counted at the end of the election and therefore, to prove their authenticity.
- Guarantee the dependability, scalability, flexibility and accessibility of the system [4].

In an electronic election system, privacy and security are mandatory features. However, it is not always obvious how to achieve these characteristics at a reasonable price, due to the fact that when an election process takes place, mechanisms that assure both, security and privacy may be too expensive for system administrators on one side, and inconvenient for users on the other. If the election system is at a national level, it implies that millions of voters and thousands of officials will be interacting, thus the reliability and precision of those systems become crucial [5].

Although in many countries only conventional elections have been instrumented, others have adopted information technology novelties, such as Brazil that captured 115 million votes through voting machines with the Windows NT operating system and touch screen monitors in 2002, and India in 2004 where 670 million electronic votes were issued [6]. In United States, during the primary and secondary elections of 2004, the system SERVE (Secure Electronic Registration and Voting Experiment) was utilized [7]. In that system, firstly voters were asked to perform a pre-registration step and later they were allowed to vote from any Internet-connected computer by establishing a secure session with the server. In Estonia, electronic voting was made by Internet in 2005 with an enormous citizen participation, probably due to the fact that Estonia's citizens are heavy Internet consumers [8].

In Mexico several electronic systems have been built. For example, the Extraterritorial Vote [9] and SELES [10]. Extraterritorial Vote was developed in the state of Coahuila during the presidential election of 2006, by free-lance developers. In that system, a citizen should first register to obtain a secret code that was then sent to him/her by postal mail. This password allowed him/her to vote through any Internet connected computer. SELES was developed and implemented in the Computer Science Department of CINVESTAV-IPN for medium scale on-line elections with less than five thousand voters. It requires a pre-registration and it guarantees security services by performing three phases: Voter authentication, vote cast and vote counting.

This paper presents the development of an Internet Electronic System (SEVI), especially designed for the Mexican presidential election of 2006. SEVI automatizes the voting process by certified postal mail of Mexican citizens with residence in other countries as it was legislated in the sixth book of the Federal Code of Institutions and Electoral Procedures (COFIPE) [11]. SEVI was developed using The Unified Software Development Process [12] and the Unified Modeling Language.

The rest of this paper is organized as follows: Section 2 describes the legal procedure of voting by Mexican citizens with residence abroad. Section 3 presents the models of cases of use, analysis, design, implementation and testing. Finally concluding remarks and perspectives of this work are presented in Section 4.

2 Vote of Mexican Citizens with Residence Abroad

COFIPE articles 273 to 300 describe the voting procedure for Mexican citizens with residence abroad, this procedure is summarized next.

The citizen should request his/her participation by filling out a registration form that she/he can obtain in Internet or diplomatic embassies close to where she/he resides. The form should be presented along with a copy of her/his voter credential and a proof of her/his real address. Then, that form should be sent by certified postal mail to the Electoral Federal Institute (IFE) in Mexico.

IFE should accept the documentation if the registration is properly filled. Afterwards, if the application fulfills all the requirements, the citizen request is granted and the citizen is included in the Electoral Nominal List (LNEE) while she/he is temporarily removed from the Voter's Nominal List (LNE) in Mexico.

By using citizen's postal address, IFE can inform the citizen the result of his/her request, either indicating the cause of its rejection or sending his/her the voting documentation.

In case of being accepted, the citizen should get ready to vote according to the following instructions: she/he should take the envelope with her/his elector key and then enclosed there the ballot with his/her vote. After that, the citizen must seal the envelope and place it in a second envelope labeled with her/his data. The citizen's final step is to send the second envelope to the IFE and to wait till the electoral date for verifying the result of the voting process.

In Mexico, IFE functionaries receive the envelopes, they annotate the arrival date and register them labeling the citizen's name with the legend "Vote" in the LNEE.

The election day, the president of each table of scrutiny will check whether the sum among the marked voters with the word "Vote" and the sum of the envelopes that contain the electoral ballots match or not. After having confirmed the previous point, the ballot is extracted from the inner envelope and it is placed in the electoral urn, this way guaranteeing the citizen's anonymity.

At the end of the electoral day, the votes are counted by the tellers, in presence of the political parties' representatives, and the result is registered in the scrutiny act corresponding to each scrutiny table. The acts are grouped according to the corresponding electoral districts, where the sum of the votes is computed.

When the scrutiny process has totally concluded, the General Secretary, will let the General Council know the official results of all the votes collected from abroad in the President election of the Republic of the Mexican United States.

Using this voting procedure, the IFE accepted little more than 40,000 request and 32,621 votes were obtained [13] for the Mexican presidential election of 2006. After having assigned a considerable budget for this task, analysts perception was in the sense that the whole process was slow and quite expensive. Also, since the entire communication between the IFE and the citizens with residence abroad was made through the certified postal mail, at least two major problems were reported. The first one is that IFE election process obligated citizens with residence abroad to inform about their exact addresses, which caused that many citizens refrained from carrying out their application (due to privacy reasons and the fear that this information could be used against them by local authorities such as the ones in United States). The second problem was caused by the inefficiency that characterizes the Mexican Postal Service.

3 Electronic System of Voting for Internet (SEVI)

This work proposes an Electronic System of Voting for Internet (SEVI) that automates in a secure way the voting procedure of Mexican citizens with residence abroad as it was specified in COFIPE sixth book. SEVI will provide the services of Registration, Voting, Results and Audit as outlined in Figure 1.

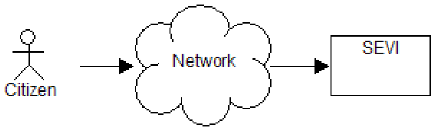


Fig. 1. Internet Electronic Voting System (SEVI)

For the sake of developing SEVI in a modular fashion, the object oriented programming was used, reason for which The Unified Software Development Process and the Unified Modeling Language were adopted [12]. Accordingly, the following models are reported next: Cases of use, Analysis, Design and Implementation.

Case of Use model

The user's requirements for the software system, whether functional or not, are captured in the diagram of cases of use shown of Figure 2.

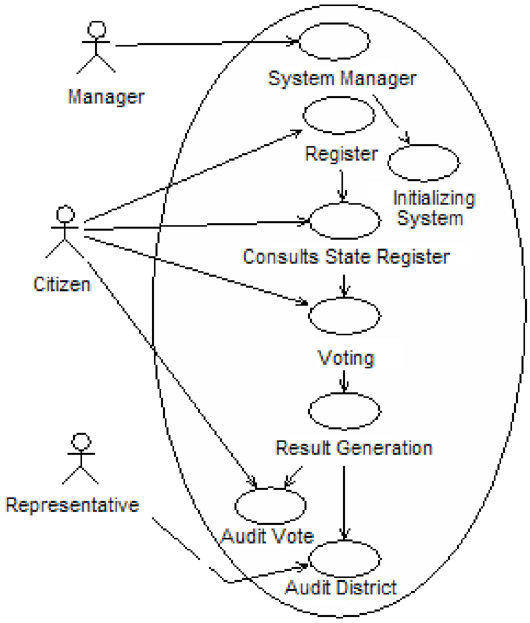


Fig. 2. SEVI Cases of Use Diagram

.

- Register: the actor Citizen requests her registration to the LNEE providing the information of his voter credential and attaching a copy of her credential and a proof of her/his actual postal address.
- State of Register Query: the Citizen queries the state of her request, which can be: pending, accepted or rejected. SEVI will validate the information provided by the Citizen against the LNE. If it matches, SEVI will temporary remove the Citizen from the LNE database and the Citizen will be included in the LNEE database.
- Voting: the Citizen, provided that a positive authentication has been accomplished, casts her/his vote using an electronic electoral ticket (ballot). SEVI receives the votes, it classifies them according to the section and electoral district and it stores them in the urn of the district.
- Result generation: SEVI counts the votes stored in each electronic urn by electoral district.

- Auditing the district: the Representative actor, which can be an electoral official of the district table or a representative of the political parties, can verify the validity of each vote stored in the district urn.
- Auditing the vote: The Citizen actor can verify that his vote was valid.

It is worth remarking that SEVI should enable each of the cases of use previously described during an interval of time. In accordance with the COFIPE, the registration is open from October first of the year previous to the election up to January 15 of the presidential election year. Votes are received by certified postal mail from May 21 up to twenty four hours before the electoral day.

Analysis Model: It is the detailed specification of the user's requirements. The oriented object approach; along with UML diagrams of classes/collaboration represent this model.

The Diagram of Classes of SEVI is shown in Figure 3. The classes Module IFE, Council and Stall are responsible of accomplishing the functional requirements of the system, maintaining communication with the databases for the registration of the citizens, the voting and the counting servers. The classes Secure Station and Secure Server are responsible of the execution of the implemented security protocol, which is discussed in the design model. For the sake of clarity, in the diagram of classes only the classes of the register and counting servers, and of the classes that are implemented through the personal computers used by the Citizens to interact with SEVI are shown. Due to space constraints we do not show the attributes or the class methods.

Design Model: It is the realization of the cases of use and it is a sketch of the implementation. This model consists of the refined diagram of classes and the diagrams of sequence and collaboration.

SEVI is a distributed system that was implemented in an architecture client/server, therefore, its classes are grouped in software components to be distributed among the computational nodes. The communication between the client and server is instrumented in a secure way using the SSL secure protocol [14].

SEVI consists of five phases: Register, Authentication, Voting, Counting and Auditing, which correspond to the cases of uses specified before. SEVI utilizes four databases administered in the same number of servers: Registration Server (RS), Authentication Server (AS), Voting Server (VS) and Counting Server (CS).

During the Registration phase, the Citizen connected from any computer, accesses remotely to the RS and fills out the registration format, which in our application is shown in a pop-up window. She/he must attach the files corresponding to the copies of her voter's credential and her address proof. The tasks of SEVI in this phase are: to verify the match between the information provided by the Citizen and the one stored in the Nominal List of Voters (LNE), to store the copy files for a post-processing and to accept or to reject the request. If the request is accepted, SEVI temporarily removes the Citizen from the LNE database and it includes her in the Nominal List of Voters Abroad (LNEE) generating the corresponding certificate and public/private keys; the Citizen will

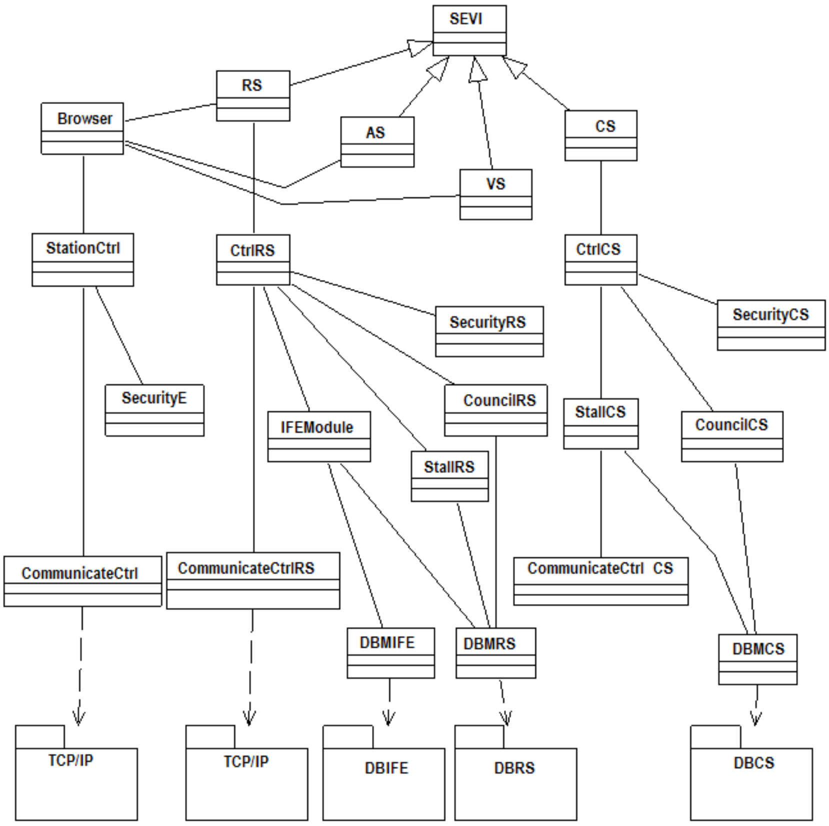


Fig. 3. SEVI Class Diagram

discharge this information within the case of use: query of transaction state, in order to continue with the following phases.

The following three phases will be carried out through the security scheme proposed by Lin-Hwang-Chang in [15] which is briefly described next.

Security scheme by Lin-Hwang-Chang: It is a protocol based on blind signatures. It protects the privacy of the voters and it is able to detect duplicity on votes. The last modification made to this scheme was reported in the SELES electronic voting system [10], where the digital signature ElGamal was substituted by the digital signature DSA.

SEVI implements the scheme with the aforementioned modification adding in the exchange of messages between the voter and the servers, the information of the voter's district (*de*), with the purpose of adjusting it to the electoral law that is trying to mimic. This process is shown in Figure 4. In the Authentication phase, the Citizen sends the message of equation (1) to the AS, which contains the following parameters: Citizen's name (V), citizen's Digital Certificate ($Cert_v$), two blind signatures z_1 y z_2 (computed based on RSA with two blind factors and two random numbers (k_1, a), which have been previously chosen by the voter, plus the DSA parameters), a timestamp (t) and the voter's digital signature (f_v).

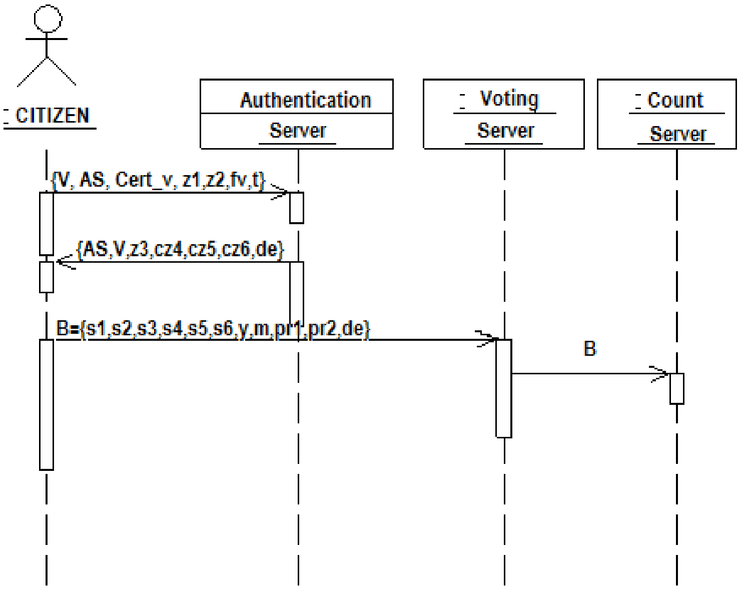


Fig. 4. Security Scheme adjusted to the Electoral Law

.

SA receives the message and verifies the digital signature by recovering the citizen's public key from her certificate $Cert_v$. If it is valid, it generates and assigns a unique identifier k_2 for the voter, identifying the electoral district where the citizen is registered and performing the calculations required for sending the message of Equation (2).

$$\{V, SA, Cert_v, z_1, z_2, t, f_v\} \tag{1}$$

$$\{SA, V, z_3, ((z_4 + t)^{e_v} \, mod \, n_v), ((z_5 + t)^{e_v} \, mod \, n_v), ((z_6 + t)^{e_v} \, mod \, n_v), d_e\} \tag{2}$$

In z_3 k_2 gets encrypted, whereas z_4, z_5 and z_6 are the SA's encrypted signatures, which have been encrypted separetedly using the voter's public key.

The citizen receives (3), it decrypts it and it obtains k_2, z_4, z_5 y z_6. This phase is accomplished by removing the blind factors in z_4, z_5 y z_6 thus obtaining the SA's signatures s_1, s_2 y s_3, with which the validity of the vote could be proved by the Voting Server (SV).

In the voting phase, the citizen casts his/her vote, signs it using DSA and it sends it to the SV. For that computation the private and public keys x and r, respectively are utilized. As a unique identifier k_2 and k_1 chosen during the authentication phase. The vote is signed using the operations indicated in Equations (3) and (4), where m is the vote contents, q is a DSA parameter and a is a random number.

$$s_4 = x_1^{-1}(m + ar_1)\, mod\, q \tag{3}$$

$$s_5 = x_2^{-1}(m + ar_2)\, mod\, q \tag{4}$$

Finally, the values pr_1 and pr_2 are computed and encapsulated taking advantage of the Chinese Remainder Theorem. This is done with the purpose that the SV performs the corresponding verifications by using the n_{sa} modules (SA public key) and q (DSA parameter). The message that the voter sends to the SV is:

$$B = \{s_1, s_2, s_3, s_4, s_5, y, pr_1, pr_2, m, de\} \tag{5}$$

Where y is a DSA parameter and m is the vote data.

The SV task is to verify the five signatures in order to validate the vote and store it in the electoral district database specified by the parameter de. Three of the signatures are performed modulus $n_s a$ (corresponding to the AS public key), and two are performed in arithmetic modulo q. If the five signatures are correctly verified, the vote is stores and after the end of the election day, the VS sends all received votes to the Counting server (CS).

In the counting phase, the CS receives the valid votes and it counts them and finally, it publishes the results. A voter trying to cheat the system by voting more than once can be easily detected by comparing the two votes shown in Equations (5) and (6) and by obtaining k_2 from equations (7) and (8). Let us recall that k_2 is the citizen's unique identifier.

$$\hat{B} = \{s_1, s_2, s_3, \hat{s_4}, \hat{s_5}, y, pr_1, pr_2, \hat{m}, de\} \tag{6}$$

$$x_1 = \frac{\hat{m} - m}{\hat{s_4} - s_4}\, mod\, q \qquad x_2 = \frac{\hat{m} - m}{\hat{s_5} - s_5}\, mod\, q \tag{7}$$

$$k_1 = x_2 - x_1 \qquad k_2 = x_1 - k_1 \tag{8}$$

With this computation the security protocol is accomplished. However, the auditing phase must still be calculated. The auditing phase is divided into two steps: vote auditing and district auditing.

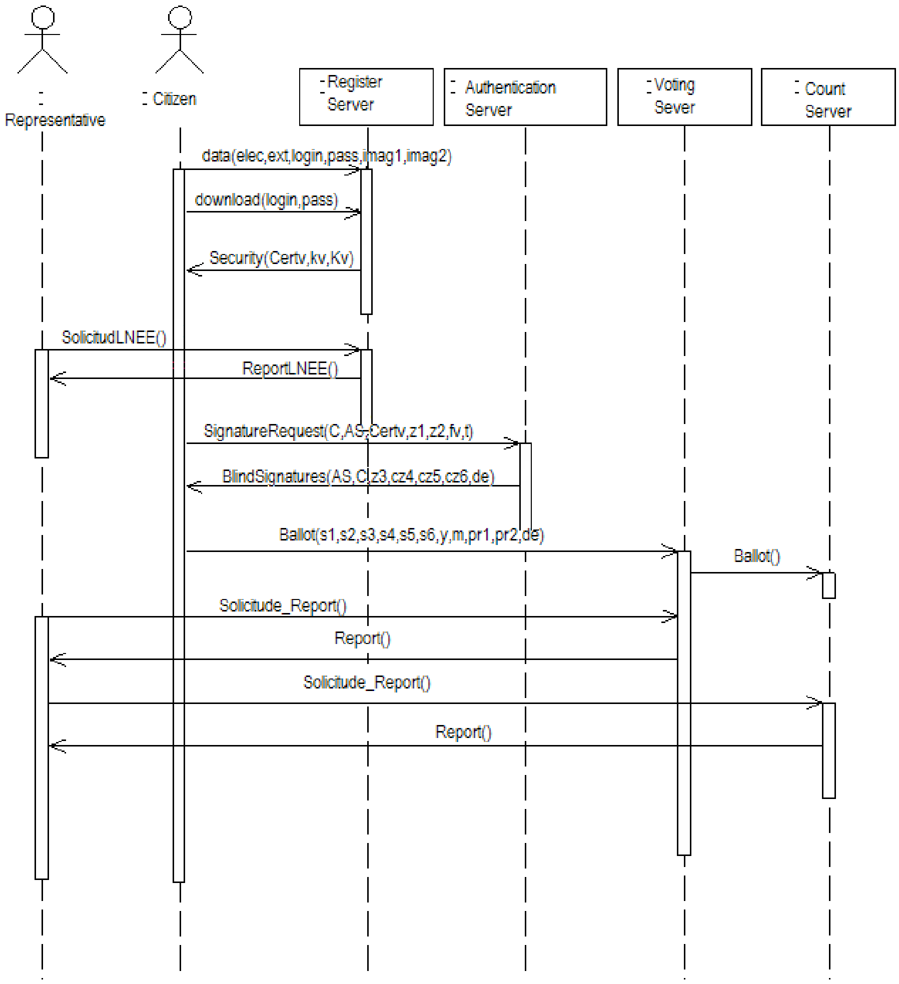
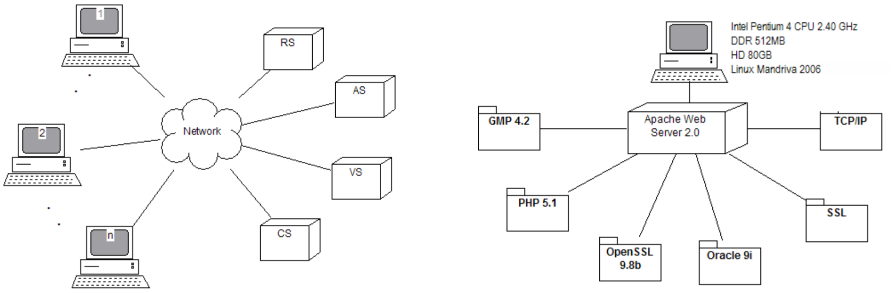


Fig. 5. Registration, Authentication, Voting and Counting SEVI Processes

Vote auditing refers to the accounting process performed in order to verify that the citizen's vote was indeed counted and included in the election's final tally. District auditing refers to the analysis, verification and validation of the register process perform by the political parties representatives according to the Articles 281-282 [11]. During this step also the members of the scrutiny table must validate the registration process as specified in Articles 291 through 293 [11]. The sequence diagram of Figure 5 shows the message exchange among actors and SEVI servers.

Implementation Model: SEVI is a design that strives for automating the election process of Mexican citizens with residency abroad. The security services are guaranteed by the Lin-Hwang-Chang protocol as it was implemented and tested in SELES [10]. In our case, it was decided that SEVI was going to utilize the Oracle 9i database modulo. Also, in order to guarantee the process fairness and the information security, the four SEVI servers must be physically independent. The client/server architecture is shown in Figure 6.



Client/Server Model

Architecture

Fig. 6. Implementation Model

Each server has a Web Apache 2.0 modulo implemented under the operating system Linux Mandriva 2006. We used PHP 5.1 as the programming language along with the cryptographic libraries OpenSSL and GMP 4.2.1, respectively. Figure 7 shows the server organization.

Testing Model: After the presidential elections, the IFE institute reported that more than 40000 votes were received from citizens with residence abroad. Out of them, little more than 30000 were accepted and counted with the rest of the votes emitted in Mexico. To test this situation in our system we simulated 40000 registers in our database and we simulated the voting process in a sample of 123 citizens with valid results.

4 Conclusions

Due to the problems identified in the presidential election process of 2005-2006 for the Mexican citizens with residency abroad because of the usage of the certified postal mail and taking advantage of the information technology tools, this paper has presented an electronic voting system (SEVI), which represents an efficient, reliable and secure option for accomplishing the same process in an automated way.

References

1. Xenakis, A., Macintosh, A.: E-electoral administration: organizational lessons learned from the deployment of e-voting in the uk. ACM International Conference Proceeding Series **89** (2005) 191–197
2. Grove, J.: Acm statement on voting systems. Communications of the ACM (2004) 69–70
3. Foundation, N.S.: Report on the national workshop on internet voting: Issues and research agenda. ACM International Conference Proceeding Series (2001)
4. Carroll, T., Grosu, D.: A secure and efficient voter-controlled anonymous election scheme. Information Technology: Coding and Computing, IEEE 2005 **1** (2005) 721–726
5. Bryans, J., Littlewood, B., Strigini, L.: E-voting: dependability requirements and design for dependability. Availability, Reliability and Security, ARES, IEEE (2006)
6. Kaminski, H., Kari, L., Perry, M.: Who counts your votes? (vev electronic voting systems). e-Technology, e-Commerce and e-Service, IEEE (2005) 598–603
7. Jefferson, D., Rubin, A., Simons, B., Wagner, D.: A security analysis of the secure electronic registration and voting (serve). New York Times Article (2004)
8. Trechsel, A., Breuer, F.: Voting: E-voting in the 2005 local elections in estonia and the broader impact for future e-voting projects. ACM International Conference Proceedings Series **151** (2006) 40–41
9. y de Participación Ciudadana de Coahuila IEPCC, I.F.E.: Voto extraterritorial, http://www.iepcc.org.mx/ademocracia/a01.html (2004)
10. García, C., Rodríguez, F., Ortiz, D.: Seles: an e-voting system for medium scale online election. Computer Science, ENC 2005. (2005) 50–57
11. Electoral, I.F.: Codigo federal de instituciones y procedimientos electorales cofipe, http://ife.org.mx (2005)
12. Jacobson, I., Booch, G., Rumbaugh, J. In: The Unified Software Development Process. Addison Wesley (1999)
13. Electoral, I.F.: Resultados de la votacion extranjera, http://mxvote06.ife.org.mx//pdf/resultados_03_06.pdf (2006)
14. Hirsch, F.: Introducing ssl and certificate using ssleay. Web Security: A Matter or Trust, World Wide Web Journal **2** (1997)
15. Lin, I., Hwang, M., Chang, C.: Security enhancement for anonymous secure e-voting over a network. Computer Standards and Interfaces **25** (2003) 131–139

Resource and Precedence Constraints in a Hard Real-time System

Luis Gutiérrez, Raúl Jacinto, Julio Martínez, José Barrios

Universidad Autónoma de Aguascalientes, Av. Universidad s/n,
Aguascalientes, Ags., 20100 México
lgutierr@cencar.udg.mx, rjacinto@cucea.udg.mx, jucemaro@yahoo.com,
jmbarrio@hotmail.com

Abstract. This work addresses the problem of resource allocation and precedence constraints in a Hard Real-time system with Earliest Deadline First policy (EDF): the concurrency control and the precedence constraints in periodic tasks. It is an improved variant of Stack Resource Policy (SRP) [1], which strictly bounds priority inversion using dynamic scheduling policies. The task deadlines are reduced to manage precedence constraints; a modified SRP-based resource access control approach is proposed. This paper uses resource constraints as in the SRP policy along EDF; the proposed algorithm produces a well-formed policy to handle both precedence-constrained periodic tasks and shared resource access in a Hard Real-Time System.

Keywords: deadline, precedence-constraints, real-time, resource-constraints.

1 Introduction

Real-time system activities are called process, tasks or jobs. A real-time system is characterized by a timing constraint in its tasks called deadline, which represents the time before a process should complete its execution without causing a catastrophic result to the system. A system is considered to be a hard real-time system, if a failure caused by a missed deadline may lead to disastrous effects.

Handling tasks with precedence relationships and resources contention is an issue which has not yet fully explored. In this paper this issue is addressed taking basis on the proposed solution in the SRP algorithm [1] where task priorities define the order of resource access using a stack mechanism. In designing a Real Time System restrictions (other than those inherent to the operating environment) need to be considered. Three main inherent restrictions of a real-time task can be usually found [2]: Time restrictions, determined by its deadline and its execution frequency or period, precedence restrictions and resource access restriction, meaning resource access management to guarantee the fair use of available resources.

The unpredictability caused by shared resource access in conflict with other concurrent tasks needs to be controlled and it is the main subject of this paper. Determining the feasibility of scheduling a set of time-restricted tasks with shared-resource access and with precedence constraints is a NP-hard problem [3]. In order to

Received 01/07/07
Accepted 19/10/07
Final version 24/10/07

reduce the complexity, this paper adopts a preemptive model with assumptions which are defined below.

The paper presents a variant of the Stack Resource Policy (SRP) proposed by Baker of a policy to include handling of precedence constraints. The restrictions mentioned above are solved by the conjunction of Earliest Deadline First policy [4] and SRP.

In section 2 the main concepts on Real-time Systems and concurrency control are presented. In section 3 the proposal's model and the proposed resource allocation policy are discussed. Section 4 is devoted to conclusions and to sketch future work.

2 Concurrency Control and the Stack Resource Policy

Concurrency control is the management of concurrent conflicting task operations that access shared resources or data [5] to avoid conflicts among them. Real-time concept adds time requirements satisfaction – deadlines – to this definition [6]. A conflict occurs when two tasks try to access the same resource at the same time and, at least, one of the tasks operations being a write operation.

Concurrency control problem can be summarized in two main functions: conflict detection and conflict resolution [7]. The standard conflict detection is accomplished by using locks (usually mutex semaphores) over resources. When a resource is unlocked, the resource is free; otherwise is busy. Usually the lock is considered to be exclusive, but in more advanced systems, the type of operation determines the type of lock – either read (non exclusive or weak) or writes (exclusive or strong) locks. Conflict resolution often is achieved by choosing one of the conflicting tasks for abortion, forcing it to release its locks.

In real-time systems, the concurrency problem is even harder than in conventional systems. Priority inversions, blocking, and deadlocks, are problems that should be considered to keep the schedulability in the processes involved. To deal with concurrency, a natural approach consists in conflict avoidance rather than the conventional detection and resolution approach.

If the problem is avoided, the time used in detection and resolution (time expended in maintaining wait-for graphs, task election for abortion and resource releases) can be dismissed allowing for a more efficient system execution. However none of possible solutions, conflict avoidance or conflict detection and resolution can avoid the Blocking problem. Specialized locks for reading and for writing can reduce this problem but at the end, there is a need for locks to avoid inconsistencies in the system due to concurrency problems.

Blocking is a source of unpredictability in real-time systems in several forms: Undefined waiting, priority inversion, deadlock, etc. When undefined waiting is potentially caused by priorities, this can be avoided by *priority inheritance*: when a higher priority task is blocked by a lower priority task, the lower priority task inherits priority from the higher one so it cannot be blocked by medium priority tasks.

Unhappily this technique can lead to *priority inversion*, however, when properly used, reduces blocking time and chained blocking generation [8]. Even if it is not clear whether to use conflict detection and resolution or conflict avoidance, the work in concurrency control in real time systems seems to be biased to avoidance mechanisms. Among this work, Stack Resource Policy (SRP) proposed by Baker in [1] offers improvements over others strategies. SRP can be applied directly to some dynamic scheduling policies like EDF, which can support stronger schedulability test.

SRP reduces the number of context switches in the execution jobs.

2.1 Stack Resource Policy

The Stack Resource Policy (SRP) is a technique proposed by Baker, It was proposed for accessing shared resources. SRP works with priority and preemption level in each task. The priority indicates the importance of a task with respect to another in the system, and it can be assigned either statically or dynamically.

In SRP each tasks τ_i was characterized by a preemption level π_i, which is a static parameter assigned to each task before running the system. With the preemption level, a task τ_a can preempt to another task τ_b only if $\pi_a > \pi_b$. A fixed parameter makes easier the prediction of potential blocking in spite of dynamic priority schemes like EDF.

In SRP, each resource is required to have a current ceiling C_R which is a dynamic value computed as a function of the units of R (the system resource set) that are currently available, n_R denotes the number of units of R that are currently available; $\mu_R(J)$ indicates the maximum requirement of job J for R. The current ceiling of R is defined by

$C_R(n_R) = \max[\{0\} \text{ U } \{\pi(J): n_R < \mu_R(J)\}]$

The system ceiling π_s is defined as the maximum of the current ceilings of all resources, it is $\pi_S = \max(C_{Ri}: i=1,\ldots,m)$. When a job needs a resource that is not available, it is blocked at the time it attempts to preempt, rather tan later. In SRP, a job is not allowed to begin until the resources currently available are sufficient to meet the maximum requirement of every job that could preempt it, so SRP prevents multiple priority inversions. A job could preempt only if its priority is the highest among all the tasks in the ready state, and its preemption level is higher than the system ceiling.

3. System Model and a Resource Allocation Policy

The system model consists of a periodic task set τ. For τ the restrictions considered are: time restrictions (execution time, deadline and period), precedence restrictions and resource restrictions (resource requests by each task).

3.1 Symbols

τ_i A generic periodic task
r_i The release time of an instance of a generic task

C_i	The computation time of a task (periodic or aperiodic)
d_i	Absolute deadline of a task
D_i	Relative deadline of a task
s_i	Start time of a task
Φ_i	The phase of a periodic task.
f_i	Finish time
T_i	Period of τ_I
U_p	Processor utilization factor
U_s	Processor available factor
τ	A periodic tasks set
G_i	A directed acyclic graph. It describes the precedence constraints between a task subset with causal relations.
R	A system resources set
R_i	A generic resource

3.2 Assumptions

- A single-processor system
- A set of periodic task τ.
- Each periodic task τ_i has a period T_i, a computation time C_i, and a relative deadline D_i.
- Each Deadline D_i may be different to the period $T_{i.}$
- Periodic tasks are scheduled using dynamic-priority assignment, namely Earliest Deadline First (EDF);
- Periodic tasks can start at any time and not only at time t=0.
- Tasks are preemptive (i.e. they may be suspended and inserted into the ready queue to service ready tasks with major priority).
- All periodic tasks have hard deadlines.
- The precedence relations are described by directed acyclic graphs.
- The whole solution is based on EDF, mainly because it allows a maximum utilization of the available computing resources but also because it allows a dynamic behavior in arriving tasks.

3.3 Problem Definition

The particular problem in this paper is to create a policy with EDF which assigns resources from R to the tasks from τ and keeps the precedence constraints in the tasks which are defined by directed acyclic graph. The goal is to create a feasible schedule that adheres to the policy.

3.4 Proposed Solution

The proposal considers two objectives: Solving the precedence constraints and after that, solving the resource contention problems in the tasks. The final solution ensures schedulability at same time it keeps resource and precedence restrictions. The proposal works in three steps: defining an execution order considering the precedence constraints, setting the execution order, and assigning preemption levels.

Defining an Execution Order considering the Precedence Constraints. For each graph (fig. 1) is necessary to generate a serialized schedule which determines the execution order of the task with precedence constraints of G_i. This schedule is obtained by applying the preorder algorithm on the tree graph. An example is showed in fig. 2.

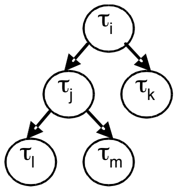


Fig. 1. Periodic tasks with precedence.

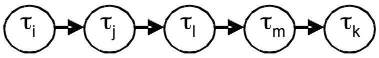


Fig. 2. An execution serialized schedule.

Setting the Execution Order. After the execution order has been established, the next step is to adjust the parameters for the tasks in order to guarantee precedence constraints. The parameters to be modified are the relative deadlines and the phase in periodic tasks; the phase Φ_i is the first activation time of τ_i. Clearly the root must be activated first. So, the tasks phases must be set in the growing order taking as reference the execution order of the first step of the algorithm. Considering the example in fig. 2, the phases must be $\Phi_i<\Phi_j<\Phi_l<\Phi_m<\Phi_k$, however it can lead to a priority inversion anomaly due to the fact that not necessarily the deadlines are also in a decreasing order; taking into account the example of the established order in fig. 1 and fig. 2, here is showed an example of the precedence anomaly in fig. 3 where the task τ_l starts the execution before the conclusion of τ_j.

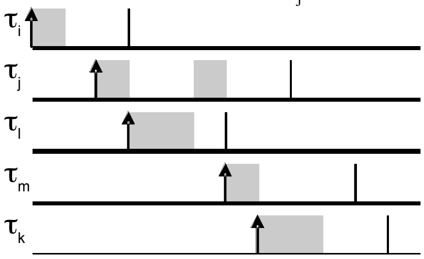


Fig. 3. A violation in the precedence constraints occurs with the tasks τ_2 and τ_3.

In order to cope with this problem, the relative deadlines need to be modified to create a schedule with increasing deadlines. To handle the modification, the "execution breadth" is defined as the maximum time that a task could wait to start its execution in order to meet the deadline.

The execution breadth in a periodic task τ_i is defined as the difference between the relative deadline and the computation time, it is expressed by: $H_{j,i}$=D_i-C_i, where H_j is the set of all execution breadth in the tasks that belongs to G_j. For all periodic tasks in G_i, in order to keep the precedence constraints assignment of the same execution breadth is needed. The new execution breadth is determined by: $H'_{j,i}$=min (H_j).

In this way, the new relative deadlines of the task in G_i are defined by $D_{j,i}$= $H'_{j,i}$ +C_i. With this modification the relative deadlines produce a schedule with increasing deadlines which leads to the generation of a schedule which keeps the precedence constraints. This is proved by the next theorem.

Theorem 1. Let τ be a set with n periodic task: τ_1, τ_2, τ_3,.., ,τ_n. Assume for all τ_i : $C_i \leq D_i \leq T_{i.}$ Now, the precedence constraints are represented by a directed acyclic graph called G. H is the set of all execution breadth in the tasks that belongs to G_j, H>0. σ is a schedule of the entire task in τ with an order keeping the precedence relations.

If it is applied the modification in the parameters as above is defined then the real execution with EDF will be executed in the order established by σ.

Proof. It is proceed by Induction and is based on demonstrating that in all time the deadlines will be increasing congruently with the serialization; this proof is enough to keep the causal or precedence relations.

For τ_1: The task τ_1 keeps the order, because it is the first task in execution and is activated in r_1^*=0

For τ_2: The task τ_2 also keeps the order, because it is activated C_1 time units after the activation of τ_1 and d_2>d_1, because $\forall C_i$:C_i>0 and $d_2 = C_1+C_2+ H > d_1 = C_1+ H$. Now, suppose the theorem is maintained for τ_k, this is:

$$d_k>d_{k-1} \ . \qquad (1)$$

Now, it is needed to prove that the theorem is maintained for $\tau_{k+1.}$ Taking (1) and adding C_k+H in the left part and C_{k+1}+H in the right part, the relation is kept and produces

$$d_k + C_{k+1}+H > d_{k-1}+ C_k +H \ . \qquad (2)$$

It is kept because $d_{k-1}+ C_k +H = d_k$ and $H+C_{k+1}>0$, and $\forall_i$:C_i>0 and $H \geq 0$

Now, substituting d_{k+1} by d_k+C_{k+1}+H and d_k by $d_{k-1}+C_k$+H in (2), the result is $d_{k+1} > d_k$ □

With this modification in the deadlines, a schedule is obtained that keeps the precedence relationship and shown in the fig. 4.

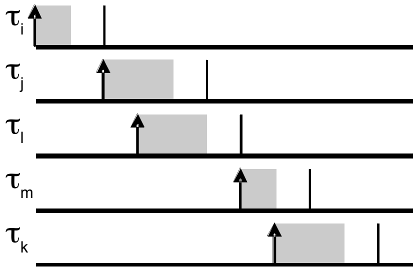


Fig. 4. Schedule that keeps the precedence relationship and time restriction (deadline)

Assigning Preemption Levels for Considering the Integration between Resource Allocations and the Precedence Handling. The next and last step is considering the resource request in tasks. The original job from [1] does not consider precedence constraints. This step modifies the Baker proposal to consider the precedence constraints.

Actually, the simple integration of EDF + the actual proposal + Baker proposal, may lead to the wrong impression that the problem is solved because time restriction is considered by EDF, precedence relations are covered by the first part of this proposal, while resource handling is covered via SRP.

This is a mistake because direct application of SRP may lead to situations where the precedence relationships are not respected as it is shown later.

To show the problem lets include a new independent task τ_v which requires a resource R in order to be executed, and τ_j request the same resource in the first computation time, then the scenario showed in fig. 5 can arise, where τ_v is executed before than τ_i although $d_v>d_i$.

This is because τ_v executes a lock (R) that produces a blocking of the task τ_i, after which τ_l arrives and preempts τ_v because $d_v>d_l$.

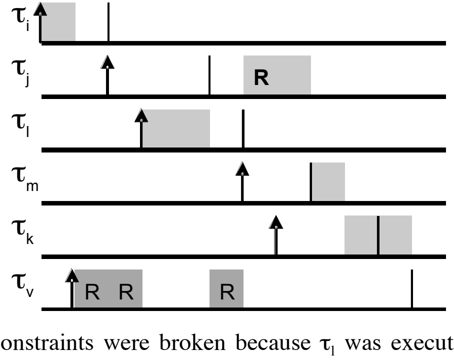


Fig. 5. The precedence constraints were broken because τ_l was executed before than τ_j, and three tasks failed to meet the deadlines.

SRP could work with EDF if preemption levels are ordered inversely with respect to the order of relative deadlines; that is $\pi_i>\pi_j \Leftrightarrow D_i<D_j$. This is because of the general definition in order to keep all the properties is required that if J_a arrives after J_b and J_a has higher priority than J_b, then J_a must have a higher preemption level than J_b. SRP

by itself does not consider the precedence constraints. To achieve this, the proposal consists in changing the preemption level assignation to integrate SRP to this work.

Each graph Gi generates an execution serialized schedule σ_i. Before assigning the preemption level, for each G_i, should be considered that the preemption level assignation must be in an inverse order with respect to the deadline order presented above. From the example presented in fig. 1 and fig. 2, it is if $\Phi_i<\Phi_j<\Phi_l<\Phi_m<\Phi_k$ and $d_i<d_j<d_l<d_m<d_k$ then must be $\pi_i> \pi_j>\pi_l>\pi_m>\pi_k$ in order to keep the precedence constraints. To consider two or more directed acyclic graphs, the preemption level assignation must be combined in competetion with the best element of each graph. This is in congruence with the original strategy of SRP because it considers the relatives deadlines.

Because schedule σ_i is ordered in congruence with the internal preemption levels, the first element of σ_i is the element with the smallest preemption level. Initially a counter called pl and with a value in "1" is set, it assign its value as the preemption level to the selected element (task). Let get(σ_1,σ_2,..,σ_{n-1},σ_n) be an instruction which compares the first element in each schedule, takes the element or task with the smallest relative deadline, assigns the pl as preemption level in the selected task (this task won't be considered in the next steps) and increments pl in "1". The get instruction must be executed until that all task in τ have received a preemption level. It can be seen like a general schedule construction as is showed in fig. 6. This way of preemption levels assignment integrates SRP with the precedence management policy presented in this paper. So the execution with SRP keeps the established order in the tasks, as proved in the Theorem 2.

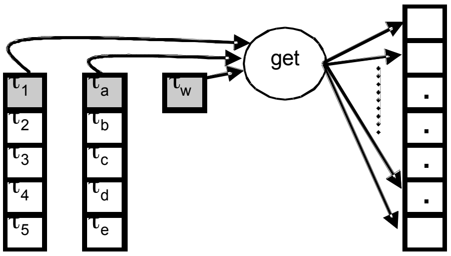


Fig. 6. The get instruction: functionality.

Lemma 1. Let τ be a set with n periodic task: τ_1, τ_2, τ_3,.., ,τ_n. Assume for all τ_i : $C_i \leq D_i \leq T_{i.}$ Now, the precedence constraints are represented by a directed acyclic graph called G. Let be σ an ordered schedule of the entire task (τ_a-> τ_b-> τ_c-> τ_d-> τ_e) in τ with an order keeping the precedence relations, τ_a is predecessor of τ_b, τ_b is predecessor of τ_c, etc. If a decreasing order of preemption levels with respect to the graph is assigned, where the time parameters were modified to ensure precedence relationships, then a feasible schedule is produced, which considers precedence constraints and resource allocations.

Proof. In order to prove this lemma, is enough to demonstrate that the modification in the preemption levels keeps the precedence constraints. The other aspects do not change the properties of SRP. It is proceed by contradiction. Let be two tasks with precedence constraints, being τ_a the task that must be executed first and τ_b the second one, and $\pi_a > \pi_b$. Suppose that the assignation produces an execution inversion, it is τ_b is executed before than τ_a, it means that τ_a was blocked by another task, it is $\pi_a < \pi_s$. So, τ_b was executed before than τ_a, then it means than $\pi_b > \pi_a$, which leads to a contradiction.

Theorem 2. Let G_1, G_2, ..,G_n be the directed acyclic graphs that represents the precedence constraints between all tasks in τ. Each G_i is defined by at least one task and for all i.j: 1..n, $G_i \cap G_j$ = {}. The get instruction defined above produces an assignation of preemption levels to all task of τ which produces a feasible schedule considering SRP and the proposal for managing precedence constraints.

Proof. In order to prove this theorem, it is only necessary to exhibit that the assigned preemption levels maintain the precedence rules established. The get instruction in order to construct the general assignment, for each σ_i always considers the element with the time more distant to be activated as candidate, it for assigning the preemption level; which is incremented after an assignment. The candidates of σ_1, σ_2 ,.., σ_{n-1}, σ_n competes as in SRP, by considering the relative deadline, which was proved in [1] that maintains the properties for SRP with independent tasks. This is true also in this work, because the tasks between two graphs are independent, $G_i \cap G_j$ = {}. Now, the last thing is to demonstrate that the order in each graph keeps the precedence constraints, which was proved in Lemma 1. So, this theorem remains.

4. Conclusions and Future Work

In this paper, a policy for considering precedence constraints and resource allocation in a system with EDF has been proposed, it works in three steps: defining an execution order considering the precedence constraints, setting the execution order with modifications in the deadlines and phases of the tasks, and finally, assigning preemption levels. The last part is a refinement of SRP with a modification in the form that preemption levels are assigned. The modification consists in considering a serialized execution, which keeps the causal relationships. The proposed algorithm correctness has been demonstrated with analytical proofs; in the sense that modifying tasks deadlines and phases is enough to keep the precedence constraints with EDF. It was proved that with the proposed assignment in the preemption levels, SRP could consider precedence constraints after that a serialized schedule has been defined. This work is part of a local scheduler in a distributed scheduling mechanism. Actually a distributed simulator is in place; in this simulator the main parts of the architecture are implemented trying to simulate real conditions. The proposed algorithm is being compared with a proposal that uses resource reservation [9], [10]; the results will be presented in a future work.

References

1. Baker, T.P.: Stack-based scheduling of real-time processes. Journal of Real-Time Systems, 3 (1991)
2. Liu, J.W.S.: Issues in distributed real-time systems: Workshop on Large, distributed, parallel architecture real-time systems (1993)
3. Bernstein, P.A., Hadzilacos, V., Goodman, N.: Concurrency control and recovery in database systems, Addison Wesley Publishing (1987)
4. Horn, W.: Some simple scheduling algorithms. Naval Research Logistics Quarterly, 21, (1974)
5. DiPippo, L., Wolfe, V.F.: Real-time databases, University of Rhode Island (1995)
6. Ramamritham, K.: Real-time databases. International Journal of Distributed and Parallel Databases (1996)
7. Yu, P.S., Wu, K.L., Lin, K.J., Son, S.H.: On real-time databases: concurrency control and scheduling. IBM Watson research center, University of Illinois, University of Virginia (1994)
8. Sha, L., Rajkumar, R., Lehoczky, J.P.: Priority inheritance protocols: An approach to real-time synchronization. IEEE Transactions on Computers (1990)
9. Abeni, L., Lipari, G., Buttazzo, G.: Constant bandwidth vs proportional share resource allocation. In Proceedings of the IEEE International Conference on Multimedia Computing and Systems, Florence, Italy, (1999)
10. Saowanee, S., Rajkumar, R.: Hierarchical reservation in resource kernels. Technical report, Electrical and Computer Engineering CM, (2001)

Fast and Accurate Signature-Generator for Detecting P2P Traffic

Cruz Alonso Bejarano, Luis A. Villa Vargas, Marco A. Ramírez, Oscar Camacho

Center for Computing Research of the National Polytechnic Institute, México

e-mails: {alonsob, lvilla, mramirez, oscarc}@cic.ipn.mx
Phone: +52-55-57-29-6000, Ext.56519

Abstract. Over the last few years, sharing information using P2P systems is widely used. Music, videos, software and documents contents are some of the information more requested by P2P users. However, the traffic generated for these applications have increased considerably. Moreover, pornography, no legal software distribution and virus propagation, have found in this systems an ideal platform for propagation. Detect P2P traffic is difficult. Current applications uses camouflaging techniques to avoid be detected: dynamic port numbers, port hopping and encrypted payload among others. In this paper we present a new approach to detect P2P traffic. We focus our work on the automating generation of signatures.

1 Introduction

Over the last few years Peer-to-Peer (P2P) application has come up as one of the main traffic generator over Internet bandwidth. Some studies revels that much of the ISP's traffic is caused for P2P application [10, 11]. Most of the traffic content related to P2P is Music, videos, software and documents. Moreover, it is likely that P2P is used as a favorite communication route for pornography. Because of this fact, proposing network infrastructure techniques to identify P2P traffic is attracting much attention within research circles. The accurate P2P traffic identification is indispensable for traffic controlling and blocking. The methods more used to identify P2P traffic are based on: port-based analysis, signature matching and heuristics methods. The port-based studies focus their strategy in identify the well-known port numbers [1,8,9]. It is due to most application use a very know port numbers for communications, and in this way traffic classification it is a trivial task. However, ports assignment in new P2P applications is dynamic, that is, using arbitrary ports. These new mechanism enable the possibility of avoid firewall-blocking mechanism. Heuristic methods use rules combinations created with information extracted from the transport layers: ports numbers, IP address, protocols, among others.

Signature mashing techniques extract strings for payload that can be used for identification [6,7]. Two major drawbacks are attached to these techniques: these do not work without payload information; either without signatures (does not work if the traffic was not previously analyzed). In this paper we present a new and accurate

Received 11/07/07
Accepted 19/10/07
Final version 20/10/07

approach for P2P traffic-detection. We have developed a platform which dynamically can extract signatures from packets, the SiGeSy *Signature-Generarator-System*. This fast platform creates signatures from network traffic traces that can be synthesized in an FPGA-based system. The rest of the paper is organized as follows. Section 2 discusses the related work in P2P traffic detection. In section 3 a brief description of our selected P2P applications set are presented. The design and description of our signatures-generator platform is presented in section 4, and finally in section 5 our conclusions are presented.

2 Related Work

The P2P traffic detection has engaged the attention within research circles. Transport layer information like port numbers, IP address, payload-data, packet inspection among others are the mostly used in related works to identify P2P traffic [1,2,3,4,5]. Multiprotocol detection system has been analyzed also for P2P [6]. In this work the port usage, connection life times churn rates and overlay technology is analyzed as well. Detect P2P traffic based in the hosts behavior is analyzed in [8]. Motivated by the slow process of manual signature generation, some researchers have recently given attention to automating the generation of signatures mainly for virus and worms detection [7].

3 P2P Applications Analyzed

In order to prove the accurately of our Signatures detector platform, this work analyzes three different P2P applications: *eMule*, *Ares* and *Kazaa*. We have selected these applications due to these are the most popular in the University Campus. The Figure 1 shows the results survey of the P2P applications most used in our campus. More than 20% of people involved in the opinion poll use *Ares* and *Kazaa*, and 14.2% use *Limewire* and *eMule*.

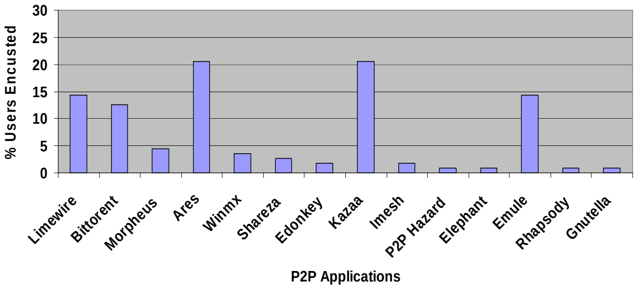


Fig. 1. Opinions pull of the most P2P application used in the University Campus.

The *eMule* system is maintained as an open source application. It is connected mainly to the *eDonkey* P2P network, although it is also connected to other networks achieving a large user base. Its main drawback is the slow download files speed. A faster P2P application is *Ares*. This application is attractive due to its simplicity and also for the fast connection offered. However, the most popular P2P program has been *Kazza*. Although seemingly at the present time *Kazza* popularity is declining.

4 SiGeSy - Signature Generator System

4.1. Architecture Overview

The Architecture is organized in four main structures: trace-creation, string-generator, string-matching and signatures-filtering.

4.2. Trace-Creation

We have created and analyzed different packet traces (showed in Table 1), grouped by P2P application. A total of 30 traces were created using the Ethereal ver. 0.99, running with WinPcap version 3.1. Our traces-set were generated in a controlled fashion and in isolated way, launching instances of each of the desired P2P application and collecting all the packets exchanged. Three different activities were monitored: during the application start-up, File-searching and file-downloading. Clean configuration are traces that were created when any P2P application were running.

Table 1. The set of traces used to generate signatures

Appli.	Start-Up	File-Search	File-Download
Ares	A1,A2,A3	As1,As2,As3	Ad1,Ad2,Ad3
Kazaa	K1,K2,K3	Ks1,Ks2,Ks3	Kd1,Kd2,Kd3
eMule	e1,e2,e3	es1,es2,es3	ed1,ed2,ed3
Clean	C1, C2, C3		

4.3. SPG - String Process Generator

Because the fact that the location of strings in the packets-payload are not deterministic, is important to have a system with the ability to recognize strings of different lengths, where these strings can be situated in different locations into the packet. The figure 2 shows the string-creating system. Using our pre-generated *start-up* traces (we use *Ares* for the system description) as an input (Table 1), the system reads a data stream packet at rate of one byte at once. The SPG system showed in the figure 2, has a string-data granularity from 2 up to 16 bytes (W_{min}=2 and $W_{max=}$16). In this way, fifteen sub-strings between W_{min} and W_{max} are generated having the same 16 bytes in the window; and after that data stream can be advanced byte for byte.

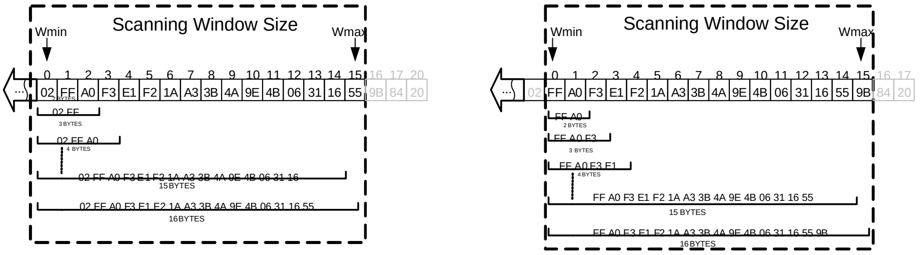

Scanning Window Size
Wmin
Wmax
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 20
02 FF A0 F3 E1 F2 1A A3 3B 4A 9E 4B 06 31 16 55 9B 84 20
2 BYTES
02 FF
3 BYTES
02 FF A0
4 BYTES
02 FF A0 F3 E1 F2 1A A3 3B 4A 9E 4B 06 31 16
15BYTES
02 FF A0 F3 E1 F2 1A A3 3B 4A 9E 4B 06 31 16 55
16BYTES
Scanning Window Size
Wmin
Wmax
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
02 FF A0 F3 E1 F2 1A A3 3B 4A 9E 4B 06 31 16 55 9B 84 20
FF A0
2 BYTES
FF A0 F3
3 BYTES
FF A0 F3 E1
4 BYTES
FF A0 F3 E1 F2 1A A3 3B 4A 9E 4B 06 31 16 55
15 BYTES
FF A0 F3 E1 F2 1A A3 3B 4A 9E 4B 06 31 16 55 9B
16BYTES

Fig. 2. The SPG produce fifteen strings using 16 bytes window size.

By generating signatures in this way, eventually all the possible strings of length from W_{min} byes to W_{max} bytes in every packet are scanned. This procedure is followed with each group of traces. The strings created with the SPG are written in a MySQL database file. Because of the fact that we have three start-up traces for each P2P application, we have also three SQL-files. Once the three files with strings are created, all the information is processed in the SMP (String Matching Process) as showed in the figure 3.

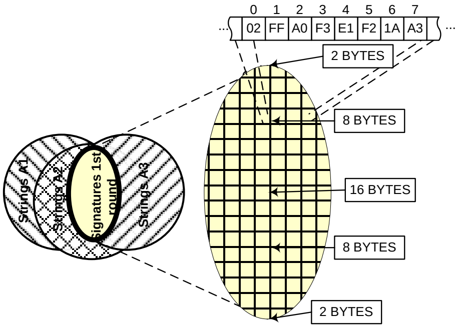


Fig. 3. The String Matching process.

The intersection between strings from A1, A2, and A3, create the first round signatures information file (***Signatures_1st***). Most of these signatures can be used for detect *Ares*, however, there are many signatures that are not *Ares*-exclusive information: NTF access, firewall access, IDS server access, among others. In order to have a more *Ares*-exclusive signatures file, we most filter the ***Signatures_1st***.

4.4. The SFP-Signatures Filtering Process

The strategy used to filter the *Ares* ***Signatures_1st*** is showed in the figure 4. We use the 27 traces no used to create the ***Signatures_1st***. The idea behind the SFP is to remove all signatures that matching with *Kazaa*, *eMule* and *Clean* strings generated with same SPG process. As we can see, in figure 4, all the signatures that not match with the ***Signatures_1st*** (*Ares*), make up the ***Signatures_2nd*** which is a more pure *Ares* signature file.

After repeat the whole process for *Ares*, *Kazaa* and *eMule*, three signatures-files were produced (one for each application). The content of these files is presented in the next section.

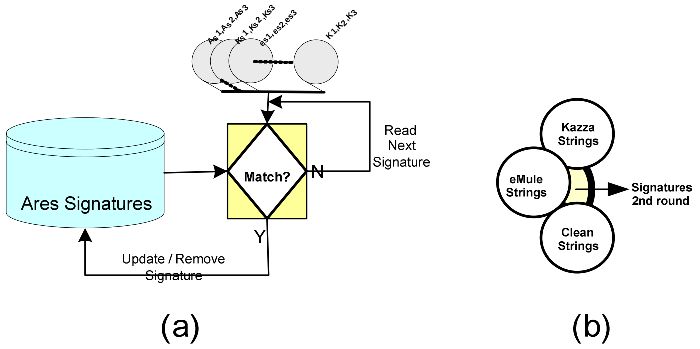


Fig. 4. (a) The SFP - Signatures Filtering Process. (b) A *signature_2nd* file generated after the SFP conclusion.

5 SiGeSy Results

In this section we present the signatures generated with SiGeSy. The Table 2 shows the packets reduction achieved. It is very important because for *Ares* the initial trace has 935 packets, which could contain signatures and on the other hand, SiGeSy shows that the signatures which can be used to detect *Ares* are only located in eleven packets.

Table 2. Packets reduction achieved by SiGeSy

P2P Application	Ini-Packets	End-Packets
Ares	935	11
Kazaa	632	18
eMule	729	5

It is important to note that, *Kazaa* payloads are encrypted in ten of the eighteen packets. That means that the other eight packets include the information useful to extract signatures. Although applications as *Kazza* use more advanced camouflaging techniques, we ha observed that this type of mechanism are not applied until the initial connection has been setup and therefore the SiGeSy get to find the packets with the adequate information.

The Table 3 shows a selection of signatures-set extracted from the **signatures_2nd** files. Although each file contains from 90 to 150 signatures, a visual analysis shows that some of them are clearly identified as the packets transmitted during the initial connection.

Table 3. A selection of signatures extracted with the SiGeSy

P2P Applications	Signatures Selected
Ares	GET /ares/hme2.php http://www.youtube.com
Kazaa	1.kazaa.com1%0#.* Host: images.kazaa.com Hermannet-works.com1%0#
eMule	http://emule-project.net DonkeyServer N01

6 FPGA Based IDS Design

In this section we present the signatures generated with SiGeSy. The Table 2 shows the packets reduction achieved. It is very important because for *Ares* the initial trace has 935 packets, which could contain signatures and on the other hand, SiGeSy shows that the signatures which can be used to detect *Ares* are only located in eleven packets.

7 Conclusions

This paper focused on the automating generation of signatures to detect P2P traffic. We have developed a platform which dynamically can extract signatures from packets. This fast platform creates signatures from network traffic traces. Our results shows that our system reduce the number of packets that most be analyzed to find an accurate set of signatures. For example, the relation for packets reduction is from 925 packets to 11 for *Ares* application. We have demonstrated that SiGeSy can extract signatures even when more sophisticated techniques are used for the P2P applications like *Kazaa* (payload encryption). Finally we have observed that these sophisticated mechanisms are not applied until the initial connection has been setup and therefore the SiGeSy get to find signatures. Finally we have reported a prototype experiment to demonstrate the portability of our platform. The Prototype was synthesized in a Field Programmable Gate Array- FPGA.

References

1. Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang, "Accurate, Scalable In-network identification of P2P Traffic Using Application Signatures", 3th International World Wide Web Conference, New York City, May 2004, pp. 512 – 521.

2. Sarang Dharmapurikar, Praveen Krishnamurthy; Sproull, T.S.; Lockwood, J.W., "Deep packet inspection using parallel bloom filters", Micro, IEEE, Volume 24, Issue 1, Jan.-Feb. 2004 Page(s): 52 – 61
3. Sen, S.; Jia Wang, "Analyzing peer-to-peer traffic across large networks", IEEE/ACM Transactions on Networking, Volume 12, Issue 2, April 2004, pp. 219 – 232.

4. Madhukar, A. Williamson, C., "A Longitudinal Study of P2P Traffic Classification", 4th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Sept. 2006, pp. 179- 188.

5. Sun-Myung Hwang, "P2P Protocol Analysis and Blocking Algorithm", Springer Berlin / Heidelberg Lecture Notes in Computer Science, Volume 3481/2005, pp. 21-30.

6. Holger Bleul Erwin P. Rathgeb Stefan Zilling, "Advanced P2P Multiprotocol Traffic Analysis Based on Application Level Signature Detection", Telecommunications Network Strategy and Planning Symposium, 2006. NETWORKS 2006. 12th International, Nov. 2006, pp. 1-6.

7. Newsome, J. Karp, B. Song, D., "Polygraph: automatically generating signatures for polymorphic worms", IEEE Symposium on Security and Privacy, May 2005, p.p 226-241.

8. Fivos Constantinou, Panayiotis Mavrommatis, "Identifying Known and Unknown Peer-to-Peer Traffic", Proceedings of the Fifth IEEE International Symposium on Network Computing and Applications, 2006, pp. 93 - 102.

9. T. Karagiannis, A. Broido, N. Brownlee, Kc Claffy, and M. Faloutsos. Is P2P dying or just hiding? In Globecom, Dallas, TX, USa, November 2004.

10. Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, John Zahorjan, "Measurement, modeling, and analysis of a peer-to-peer file-sharing workload", Proceedings of the nineteenth ACM symposium on Operating systems principles, Bolton Landing, NY, USA, October 2003, pp. 314 – 329.

11. Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble, "Measurement Study of Peer-to-Peer File Sharing Systems", Proceedings of the Multimedia Computing and Networking (MMCN), San Jose, January, 2002, pp. 156-170.

12. www.xilinx.com. Spartan-3 Starter Kit Board User Guide. UG130 (v1.1) May 13, 2005

13. Julien Lamoureux , Steve Wilton, HDL Conversion Tools. University of British Columbia, November 11, 2005.

A Platform for the Development of Parallel Applications using Java

Erick Pinacho Rodríguez, Darnes Vilariño Ayala

Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación
epinacho@gmail.com, darnes@cs.buap.mx

Abstract. The search for solutions for real problems generally involves a lot of calculations. This brings itself a great amount of time consumed when generating a response. The only solution for this problem is offered by Parallelism. By now, the hardware is not a trouble anymore. It has become more important the development of software tools that take advantage of the hardware possibilities.
One of the most widely used function library for developing parallel applications is Message Passing Interface (MPI) [1, 2]. This library has limits about memory handling, so the developer must handle complex synchronization mechanisms for using shared memory.
Java language offers upgrades about concurrency mechanisms and technologies for developing distributed applications [3, 4]. Then it is a good language to create a platform to develop parallel applications. Even today, there are no major efforts in the development of utilities to create parallel software on this language.
Our developed Platform is oriented for clusters or Local Area Networks (LAN). The design applied has considered slaves and a master process, which is in charge of controlling the tasks executed by each of the slaves. Both the master and slave processes have their own architecture. The master architecture is composed by three layers, while the slave architecture is composed by five layers.
For the communication between master and slaves inside the platform, a multicast User Datagram Protocol socket (UDP) is used. The communication is transparent to developer, who disposes of functions to send and receive messages to develop his applications.
This Platform offers a shared memory. This one is paged, so the amount of memory used is not a problem.
Remote Method Invocation (RMI), has permitted to distribute the classes between all of the nodes. This aspect is controlled by the master process, in order to guarantee an adequate work balance.

Keywords: Parallel Computing, Distributed Computing.

1 Introduction

The performance of the computers has increased exponentially from 1945 until now, on an average factor of 10 every 5 years [5]. The performance of a computer is

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 29, 2007, pp. 202-210

Received 01/07/07
Accepted 19/10/07
Final version 24/10/07

directly dependant to the time required to perform a basic operation, and the number of basic operations that can be performed concurrently. It is not about faster processors at all to improve the computer performance.

Two important features are time and space, it means the number of steps that an algorithm requires -in the worst case scenario- to generate a result, and the memory required to solve the current problem [6].

The calculations over a network of computers -also known as “distributed computing”- are not only a subfield of the parallel computing. Distributed computing is deeply related with troubles such as reliability, security and heterogeneity that are considered as ___ in the parallel computing research.

A distributed system is where all of the computers that belong to the network must work as one. Once one of the computers that form part of the system fails, even when the user does not know which computers form this 'system', an application running may not be performed correctly [5].

The concurrency theory has been an active research field on the computer sciences, since the results presented by Carl Adam about the Petri Webs in 1960 [7]. Since then, a wide variery of theoretical and logical models and tools have been developed to understand the concurrent systems.

The clusters are gaining a wide area of the current applications, due to they offer acceptable performance levels for a relative low cost.

Regarding the software, there are a wide variety of tools that help to the programmer to develop distributed and/or parallel tools. Some of these tools are currently being standarized, while some others are gaining certain acceptance from the developer community.

A great part of the dedicated effort on the development of parallel or distributed applications is put over (a) the handling of structures or communication channels offered by the current tools; (b) the management of the platform resources; and (c) the control of events of the system, amongst others.

Particularly, and regarding the Java Language, those efforts to develop parallel or distributed tools are minimal. Some of these tools are on a mature stage, and the others are still under development.

Thanks to the creation of the Forum "Java Grande", the interest and effort has been focused on the standarization and development of more tools over the Java Language.

Nevertheless, already disposing of these tools, there are missing higher abstraction levels that ease the programming task to the developer.

More tools and applications are required; these tools must help to the developer to focus their efforts into solving the initial problem. These tools must also help to decrease the time dedicated to solve communication details, event control and management or resources.

The platform presented in this paper is executed over a cluster of computers with Linux as OS. The migrative nature of java allows to use any OS with the minimal adequations needed.

This platform is directed to the development of Java parallel applications that require the use of shared or distributed memory, with transparency to the user. In the parallel applications, the use of shared memory becomes an important problem. Another important advantage offered by this platform is the message passing, because the programmer can forget completely about the protocols for this matter.

The platform supports and manages the following tasks:

1. Support to the mpiJava implementation. The MPI libraries are not originally designed to support concurrence. The platform makes the programmer free of the concurrent handling of the MPI libraries.
2. Shared memory. The developer can use shared memory. The platform is in charge of maintaining the memory consistent and of the automatic delivery and refreshment of the data.
3. Support for a single model of pool threads for each of the execution nodes, considering also mechanisms that guarantee the good use and performance of the model.
4. Support for the delivery of packages -not related to the MPI- in a transparent manner to the user.
5. Support for the execution of N parallel tasks using M nodes of a cluster.
6. Management tasks for the nodes that form the network of the platform.

The paper is organized as follows: Section 2 describes the considerations made to the design and construction of the platform. Section 3 shows the architecture of the platform, particularly the master and slave daemons, and how they comunícate. Section 4 presents the implementation of the platform, the programmed structure of the messages sent on it, how to handle memory block and how to synchronize them. Section 5 shows the conclusions obtained during the development of this research project.

2 Design Considerations

The platform is executed over a network interconnected by a TCP/IP protocol. Each node has its own IP address on the same address range, in other words they form the same subnetwork. On execution time, the platform is formed by an instance of a class named MasterDaemon, and one or more instances of a class named SlaveDaemon.

When the MasterDaemon is uploaded, it opens a multicast socket, which is an UDP socket with the ability of joining to socket groups which are also multicast. In this kind of socket, when any of the participants sends a package, all the group - including the emitter- receives the same package. This is important and is done over again many times in the platform, specially on Shared Memory mode.

Once the remaining daemons are uploading, they will be joining to the multicast group.

It has been mentioned that more than one instance of the SlaveDaemon class can be uploaded inside the same node. For the platform effects, each of these instances will become a logical node added to the platform. When a daemon is uploaded, a whole instance of the Java virtual machine is created. When a second daemon is uploaded, the JVM is not completely created again, but a fraction of it. For each daemon subsequently uploaded, there will be a partial copy of the JVM. These actions consume mainly memory resources.

3 Architecture of the Platform

The planning and modeling phases for the platform has brought sustantiable benefits. The first one is the possibility of changes in the implementation of the services for the sublayers without affecting the superior layers or other services. The latter benefit is that new services can be added in any layer, enriching it and extending its capabilities.

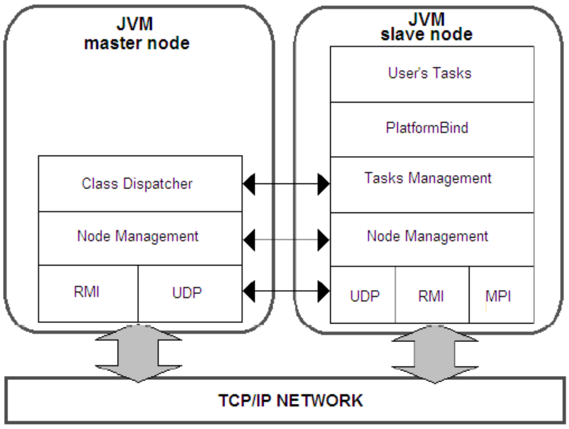


Fig. 1. Architecture of the daemons.

Master Daemon. The stack has 3 service layers:

1. Communication Channel Layer
2. Node Management Layer
3. Classes Dispatcher Layer

In the first layer the services that the master daemon uses to communicate with all the components of the platform are located. This layer is subdivided in two parts, UDP and RMI. The UDP part provides the service of message delivery and an easy structure to send them; both services are widely used in the entire platform. The RMI part offers the service of distributing the classes which are exposed thorough a remote interface.

The Node Management layer registers all the daemon processes that are uploaded in the platform. Also it makes polls on the nodes to know if they are available.

The Class Dispatcher layer makes the User Class Definition to balance the load. It uses the information of nodes to indicate to each node which classes to obtain and execute.

Slave Daemon. This stack has 5 service layers:

1. Communication Channel Layer
2. Node Management Layer
3. Tasks Management Layer
4. Platform Bind

5. Tasks of the user

The first layer is similar to the one in the Master Daemon; however this layer adds MPI services. The RMI part provides some facilities to obtain the master daemon classes. The MPI part has the MPI functions properly codified, solving the concurrency problems found in the normal library.

The Node Management layer creates the name of the node, also can request the name of each node.

The Tasks Management layer, as its name refers, treats the tasks received from the Master Daemon. This daemon indicates to each slave daemon which class obtains by assigning an ID for each task.

The PlatformBind acts as an interface for the Platform and the classes of the user. The layer provides communication services among tasks using messages with an UDP socket, passing messages with MPI, management of threads and the service of memory sharing.

The last layer performs the tasks of the user. Though the tasks are managed on the layer 3, in this layer they are executed. The distinction of layer 3 and 5 is about the services that each layer offers, such as putting the tasks in a pool. This brings up the extensibility concept.

4 Implementation

The implementation of the platform has been made in layers through the codification of the diverse classes. Next, the most important considerations for the development of each layer are presented.

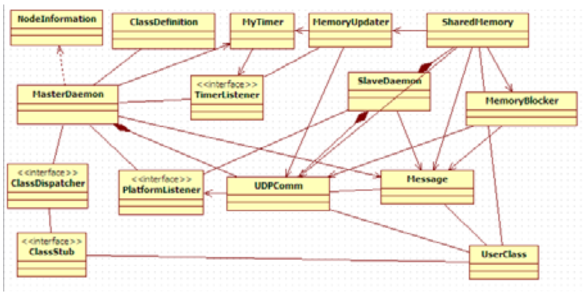


Figure 2. Class Diagram of the platform

Naming of the nodes. When a daemon is uploaded in a host, the daemon notifies to the platform that a new node has been created. A node is the instance of the SlaveDaemon class. For a user, the name of a node is not relevant, because the user does not need to know in which node its classes will be executed. However for the message passing, it is necessary to know the location of both sender and receiver. For this reason each node has a unique name in the platform. First the slave daemon takes its name directly from the node itself; if the name is localhost, the daemon renames the node on the application level. The name assigned will be NODE_XYZ, where

XYZ is a random number. If the selected name is in use, the Master Daemon notifies it and the slave tries to obtain a new name.

Message structure. A unique communication point is used by all of the functionalities of the platform, such as the memory, the platform management and the user messages. It is necessary that the message has a structure that allows recognizing the origin, destiny and purpose of the message. In the platform, the messages are represented by the class Message.

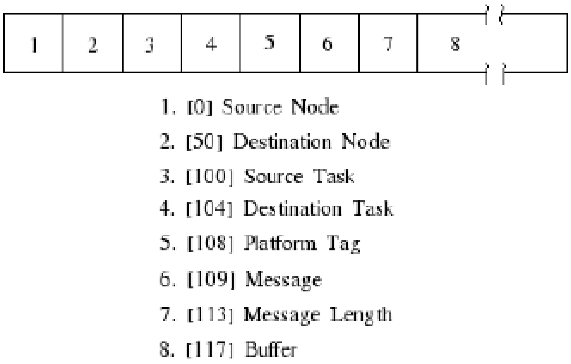


Figure 3. Message structure

If the field of the message Platform Tag indicates that the message comes from a user, the field Message may be used as desired. If the message comes from the platform, the field Message may have one of the following values:

1. DROP NET. This message indicates to all of the nodes of the platform to end the application.
2. ADD NODE. This message indicates that a node has registered in the platform network.
3. LIST NODES. This message is generated by the Master Daemon to advise to each node the list of available and registered nodes in the platform.
4. PING. This message is generated by the Master Daemon to poll to the nodes if they are still active.
5. PONG. This message is generated by a Slave Daemon in response to a PING message.
6. IP MASTER. This message is generated by the Master Daemon to inform the IP address of the master node to the slave nodes.
7. FETCH CLASS. It is a message to inform to the Slave Daemons the class that must be obtained to be next executed.
8. CLASS DEFINITION. This message is generated by the user to inform to the Master Daemon about the definition of the classes that will be used.
9. RENAME NODE. This message is generated once a node changes its name.
10. MEMORY MESSAGE. This message contains the changes made to the shared memory that should be informed to all of the nodes of the network.

11. BLOCKER MESSAGE. This message contains the addresses of the shared memory that have been blocked.
12. RELEASE MESSAGE. This message contains the address of the shared memory that have been liberated.
13. BARRIER MESSAGE. This message indicates that a node has been synchronized.
14. CLEAR BARRIER. This message indicates that the nodes have been liberated to continue their execution.

Communication sockets. To perform the communication through the platform, a UDP Multicast socket is employed. The use of these sockets brings simplicity and quickness to the protocol, which are not present in TCP, not even a Multicast implementation. Were TCP sockets employed instead of UDP in a network of N nodes, each node must create N-1 sockets to have communication with each of the remaining nodes of the platform. This justifies the use of UDP Multicast.

The Class-D IP addresses, which are between 224.0.0.0 and 239.255.255.255, are employed on multicast. Any application may use one of these addresses, and this assignment does not affect the address assigned by any network interface. With a correct address and a standard port UDP, the node can receive and transmit multicast messages.

Any delivery made in multicast will be received by any node which uses this socket.

Shared memory. The platform offers architecture with shared memory. Then, it is possible to offer to the developer a memory area common for all of the nodes of the platform, which is updated by replies. Each node possesses a full replica of the shared memory. This memory is designed to make pagination to the disc through small pages. The developer then disposes of a memory limited theoretically by the size of his hard drive. The data banks have an original size -configurable by the user- of 4096 bytes.

All of the instances of the user class that are found on the same node receive the same instance of shared memory. It is said that the shared memory offers concurrency in two levels, node and platform level.

The access to the shared memory is made through put*() and get*() methods, which allow to read and write primitive types of values to the memory -such as byte, int, float, double and byte arrays. All the addresses of the shared memory contain a byte, so the user shall be responsible of the movements needed to store primitive values which require more than one byte. The platform offers constants with the size of each of the primitive types. Similarly to the implementation of a hardware shared memory, the platform offers to block specific addresses. This functionality can be used as a basic synchronization mechanism. However several considerations about latency must be reviewed, because once a block is made, it will take some time before it is notified to all of the nodes of the platform.

Update and replicate. Once a process or task updates the shared memory of its node -by writing with the method put*()- this update must be notified to the other instances of the shared memory, so they know the new updated values.

The notification of these updates requires a great amount of messages that should be sent into the network. This provokes traffic in the network, resource consumption and extra time dedicated on attending each message.

To overcome this situation, the updates are made with delays. This is due to the writing process is regularly accompanied by another similar processes, rarely the writing is isolated. The late updates are controlled by a temporizer -with a time period configurable by the user. When a user updates the shared memory, the temporizer initiates a time counting. Once the count ends, all of the updates made on the memory are collected and then sent to all of the nodes of the platform. This mechanism reduces the network traffic and the consumption of resources. However, the execution time will be delayed and also the solution to the application of the user. This can be handled by modifying the temporizer time; when the time is reduced, the waiting time will be reduced, but the traffic will increase. It is decision of the user which factor to affect.

Address block. The shared memory allows the memory address block. Each time that an address is blocked, the block is registered with the ID of the task that made the block. From this moment, only this task may access to this address, to read or even write on it. Any other process which wants to read or write this address will be blocked until the address is liberated by the blocker task. The memory blocks are communicated through the whole platform with a message described previously. Unlike the messages that update content, the messages that indicate blocks are sent immediately when a task blocks an address, including also all of the messages prepared to be sent to the nodes -it means, this message acts as a trigger for the temporizer defined previously. With this, it is assured that the content is updated before making the block or unblock.

Regularly, the tasks will use the memory blocks as a manner to assure the concurrency, and as a manner to keep synchronization between them. This is why the block notifies or liberation of memory addresses is performed immediately, not delayed as the content update and replication.

Platform Synchronization. The synchronization points are common on the software tools that perform parallel or distributed tasks. A synchronization point consists into make all of the different processes to reach a point where they will expect the other processes to reach it. When all of the nodes reach the synchronization point, their execution will continue.

In the design of this platform, the synchronization points have been included to be programmed, and they are known as Barriers. These Barriers occur in two stages: the first stage synchronizes the node itself, because a node may be performing more than one task of the used; so the node will be synchronized once all of its performed tasks get the same defined point. The second stage synchronizes the entire platform, which will wait for all the nodes to reach the same defined point, once the first stage is reached for every node. The master node will be responsible of receiving all the notifications of each synchronized node. Once the notifications for each node are received, the master node will send the liberation message to all the nodes, and their execution will continue.

To achieve the synchronization, two classes of the concurrency tools offered by java have been employed: they are the *CyclicBarrier* and the *CountDownLatch*.

5 Conclusions

With the main objective of solving optimization problems on high scale by using parallel algorithms, a new platform has been developed. This platform is directed to take advantage of the cluster architecture.

One of the most popular tools in the scientific area is MPI. It is used to develop solutions by using a mechanism of message passing. However, a more natural model can be applied for some problems by using the concept of shared memory, which is not included in MPI. This problem has been completely solved with this platform.

This construction with shared memory has been developed by using concurrency mechanisms offered by the Java Language.

The facilities offered by the language for the creation of thread pools allows their management and reduces the traffic that implied the creation an destruction of threads. It has also support for the creation and application of policies about the execution of threads.

The utilities for concurrency of Java include an asynchronous queue. With this, it is not necessary to create critic zones or semaphores for the addition or substraction of elements, because a concurrency control is included in the same class. Asynchronous queues were employed in the platform for the message pass.

Some parts of the platform, especially in the shared memory area, include the semaphores concept of Java.

The disposal of this platform will allow to the students of the Faculty of Computer Sciences at the Benemérita Universidad Autónoma de Puebla, the development of concurrent, distributed and parallel applications through a unique programming language which requires only the classes and characteristics offered by the platform.

Acknowledgments. The heading should be treated as a 3rd level heading and should not be assigned a number.

References

1. Graham E. Faggs .Perfomance analysis of MPI collective operations. Cluster Computing. ISSN : 13867857. pag. 127-143. Kluwer Academic Publisher.
2. LAM/MPI Team at Open Systems Lab. LAM/MPI Users Guide. 2007.
3. David. D. *A Review: Java Packages and Classloaders*. 2001.
4. Goetz. B. *Concurrent Collection Classes in Java*. *IBM Home Page*. 2003
5. Ian Foster. *Designing and Building Parallel Programs.* Addison-Wesley. 1995.
6. Paul E. Black, *Algorithms and Theory of Computation*. Appearing in the Dictionary of Computer Science, Engineering and Technology. 2000. CRC. Press LLC.
7. Filman, Robert E, Daniel P. Friedman, *Coordinated Computing: Tools and Techniques for Distributed Software*. McGraw Hill Higher Education. (1984)

Architecture for the Interoperability among Ubiquitous Components

Anabel Pineda[1], Rolando Menchaca[2] and Giovanni Guzmán[3]

[1] – Computing and Systems Department
Matamoros Institute of Technology, Matamoros, Mexico
apineda@itmatamoros.edu.mx
[2,3] - Centre for Computer Research, National Polytechnical Institute, Mexico City, Mexico
{rmen, jguzmanl}@cic.ipn.mx

Abstract. Implementing spontaneous interoperability among nomadic components is currently one of the most relevant open problems in ubiquitous computing. There are three main issues related with achieving spontaneous interoperability, namely, initialization, discovery and interaction among components. In this work we present a novel architecture which is based on ontologies and peer-to-peer algorithms that solve these three aspects in an integrated way. The architecture is composed of a set of services implemented using JXTA and an ontology-based inference engine. Using our infrastructure, clients (mobile or static) are able to perform semantic and geographic-aware queries to the ubiquitous environment and dynamically find instances of a desired service. We also present a couple of applications that were developed using our proposed architecture. These applications allow us to show the potential of our infrastructure to implement two classic ubiquitous computing scenarios, both using mobile devices and heterogeneous wireless networks.

1 Introduction

Ubiquitous Computing, also known as Ubicom, is a relatively new paradigm first defined by Mark Weiser that describes a class of computing systems whose main goal is to improve the quality of life of human users by means of a seamless integration of hardware and software components into people's everyday activities. The services provided by these systems are ubiquitous in the sense that they can be provided by almost any everyday object, moreover, ubiquitous hardware and software components tend to disappear into the environment and the user only perceives their advantages while the complexity is hidden as much as possible [1]. The enormous potential of this new paradigm has attracted the attention of an increasing community of researchers who are working to materialize the seminal ideas of Mark Weiser [1]. There are many research areas directly related with the ubiquitous computing, for instance, software engineering, human-computer interaction, semantic computing, computer networks, natural language and others. In this work we focus on defining new software architecture that foster the constructions of ubiquitous systems by providing a set of fundamental services that have been identified as useful and desirable for almost any ubiquitous system.

Received 02/07/07
Accepted 21/10/07
Final version 25/10/07

To be more precise, a ubiquitous system is in essence a saturated environment with computing and communication components that are integrated with the everyday tasks of their human users [2]. The Ubicom community has identified two fundamental characteristics of a ubiquitous system: they are physically integrated into everyday objects and their components have to be capable of interacting with other ubiquitous components without need of manual re-configurations. Here, we go one step further of the second characteristic by designing and implementing a novel software architecture that solves the three main issues related with achieving spontaneous interoperability. To solve these three problems we developed a set of protocols that enable heterogeneous components to dynamically recognize other and interact among them. The computer networks that provide communication support to the ubiquitous systems have to be heterogeneous and allow the interoperability of wired, infrastructure-based wireless and ad hoc networks. Moreover, these networks have to be highly dynamic and support continuous arrival and departure of new components without need of explicit re-configurations [3]. To cope with these requirements we employ a peer-to-peer (P2P) software architecture. P2P systems are highly dynamic, composed of a set of independent software elements with equivalent functionality and they do not require centralized management. These characteristics endow to P2P systems with properties such as scalability, flexibility [4], fault tolerance and a fairly simple management [5] that can be quite useful when designing an infrastructure for the development of ubiquitous systems in general and in particular for the infrastructure proposed in this work.

Nowadays, there are several P2P systems available, for example Gnutella and Napster. However, most of these systems are designed for specific application, such as file sharing. Therefore, our software architecture uses the P2P infrastructure of JXTA in order to enable services among peers which may be hidden behind NAT, firewalls, to dynamically join and leave the P2P network, with the possibility of changing their locations. This is the general purpose for which the infrastructure JXTA was designed, to provide interoperability, platform independence and ubiquity. Inside the existent P2P infrastructures, JXTA is the one that fulfills most of the characteristics that our architecture needs.

Although JXTA provides the ideal characteristics to our architecture, it does not currently provide an adequate solution to the discovery service problem because is not enough to provide a basic advertisement-search mechanism. JXTA needs a flexible discovery service system to enable to locate all available services conforming to a particular functionality.

The rest of document is organized as follows: Section 2 presents a short introduction and the main characteristics of current discovery service protocols; in Section 3 we respond to the question why JXTA?; Section 4 describes the need to extend JXTA discovery service, Section 5 describes a novel architecture solution (ArCU) –integrating a set of services using JXTA with ontology-based inference engine– and the implementation of USE (Ubiquitous Services Environment). Finally Section 7 presents the conclusions of present work.

2 Discovery Service Protocols

There are recent solutions related with the integration of peer-to-peer (P2P) infrastructure and ontologies to discovery services [19][20]. These proposals are focused in offer precision in the discovery of services with wished functionalities. Currently, there are many protocols that provide a solution for discovery service, one of the most important requirements to reach spontaneous interoperability. Next, we present a short introduction and the main characteristics of them. Especially, we will focus in the discovery service because it is the most relevant for our proposal.

2.1 Universal Plug and Play (UPnP)

Universal Plug and Play is a technology developed in the UPnP Forum [6] for automatically configuring devices, discovering services and providing P2P data transfer over an IP network. UPnP technology is built upon IP, TCP, UDP, HTTP and XML, among others. The UPnP discovery protocol is based on the Simple Service Discovery Protocol (SSDP). The discovery process of UPnP based in SSDP is as follow: given an IP address, when a device is added to the network, the UPnP discovery protocol allows the device to advertise its services to control points on the network. Similarly, when a control point is added to the network, the UPnP discovery protocol allows that control point to search for devices of interest on the network. In both cases, the fundamental exchange is a discovery message containing a short essential description about the device or its services, i.e., its type, unique identifier, and an URL to obtain more detailed information. The principal limitation of SSDP is that it does not support the search for multiple types in the same request and attribute-based search [7].

2.2 Service Location Protocol (SLP)

Originally, the Service Location Protocol (SLP) [8] was proposed as an Internet Engineering Task Force (IETF) standard track protocol, to provides a framework to allow networking applications to discover the existence, location, and configuration of networked resources in networked resources, such as devices and services. SLP eliminates the need for a user to know the name of a network host that supports a service. Rather, the user supplies the service name and a set of attributes, which describes the resource.

The resources are modeled as clients that need to find servers attached to the enterprise network at a possibly distant location. For cases where there are many different clients and/or available resources, the protocol is adapted to make use of nearby Directory Agents that offer a centralized repository for advertised services.The basic operation in SLP is that a client attempts to discover the location for a resource. In small installations, each resource is configured to respond individually to each client. In larger installations, resource will register their services with one or more directory agents and clients contact the directory agent to fulfill request for resource location information. This is intended to be similar to URL specifications and make user of URL technology. The principal limitations are the ability to reflect SLP's orientation toward enterprise service discovery and heavyweight directories [8].

2.3 Salutation

The Salutation architecture was developed by the Salutarion Consortium, to solve the problems of discovery service and utilization among a broad set of appliances and equipment in a wide-area or mobile environment [9]. The Salutation architecture is composed of two elements: Salutation Lookup Manager (SLM) and Transport Manager. The SLM functions as a service broker for services in the network. The SLM can classify the services based on their meaningful functionality, called Functional Units (FU). The services are discovery by SLM by means of a comparison of the requerid service types with the types stored in the SLM directory. The discovery service process can be performed across multiple Salutation Lookup Managers, where one SLM represent its client while communicating with another SML to discover services [10].

2.4 Bluetooth

Bluetooth is a radio standard and communications protocol designed for low power consumption, with short-range radio frecuency [11]. Bluetooth defines its own protocol stack, including a service-discovery protocol, SDP. This protocol is based on unique identification numbers (UUIDs), with several predefined services such as phones, printers, modems, and headsets. The Bluetooth specifications are developed and licensed by the Bluetooth Special Interest Group.

Bluetooth communication is P2P, so it does not assume a fixed network infrastructure. Thus, discoverability is based on actual physical proximity rather than closeness in the IP routing infrastructure. In that sense, in comparison to the rest of the services discoveries, it simplifies the discovery and setup services. A Bluetooth device advertise all its services, making them more accessible, without the need to worry about network addresses, permissions and all other considerations related with typical networks.

2.5 Jini

Jini is a distributed service-oriented architecture developed by Sun Microsystems [12]. Jini is a simple insfrastructure for providing services in a network, and for creating spontaneous interactions between theses services. Services can join or leave the network in a robust fashion, and clients can rely upon the availability of visible services, or at least upon clear failure conditions [13].The service advertisement takes the form of interface descriptions. This simple form of the advertisement mechanism can be easily employed to provide high-level abstraction both for software and hardware entities in the network. Jini discovery insfrastructure provides a good base foundation for developing a system with components distributed in the network that need to discover each other. However, the Jini discovery process is tightly bound with the simple interface description advertisement. This leads in a loss of expressive power in the component description. For example, Jini discovery and lookup protocols are suficient for service clients to find a print service. However, they are not suficient for clients to find a print service through their geographical localization or particular functionality such as color laser printer service. This is a limitant because can create problems in a mobile environment. Furthermore, the simplicity of the Jini architecture also leads to the cross-domain service interoperablity problem.

2.6 JXTA

JXTA (short for "juxtapose") is a set of open protocolos that facilities Peer-to-Peer communication. This technology allows connecting a wide variety of devices that they can be anything with an electronic hearbeat [14]. JXTA is based upon a set of open XML protocols, this way allow exchange messages and collaborate independenlty of programming language, platform or network transport.

In JXTA the peers are organized in *peergroups* to represent types of services, location, etc. All network resources in JXTA such as peers, peergroups, pipes and services are represented by advertisements that are XML documents that announce the existence and some properties of these resources. Every Advertisement in JXTA has a string Name Field. For the search, JXTA Advertisement usually uses their name to indicate the type of service the peers provide. Advertisements provide a uniform way to publish and discover network resources and they have a lifetime to specify the lifetime of its associate resource.

The general purpose of JXTA is providing interoperability across verying P2P systems and communities, platform independence to support diverse languages, systems and networks, and ubiquity, in which every device has a digital heartbeat.

3 Why JXTA?

JXTA provides three aspects that the rest of the services previously mentioned individually they do not provide.

- The JXTA infrastructure adopt P2P systems characteristics: highly dynamics, set of independent software elements with equivalent functionality and decentralized management. These characteristics endow to JXTA with properties such as scalability, flexibility, fault tolerance and a fairly simple management.
- JXTA infrastructure provide interoperability, platform independence and ubiquity, and
- The opportunities for extending JXTA are manifold because its support arbitrary XML so it can integrate emerging standards (such as the Ontology Web Language [OWL] for description) as a relevant approach for refining searches.

These characteristics are necessary for the development of ubiquitous systems and for that reason we decide to use JXTA infrastructure.

4 Extending JXTA Discovery Service

In the introduction, we have mentioned that although JXTA provides the ideal characteristics to our architecture, it does not currently provide an adequate solution to the discovery service problem, because is not sufficient to provide a basic advertisement-search mechanism. JXTA needs a flexible discovery service system to enable to locate all available services conforming to a particular functionality. For that reason, we intend to integrate peer-to-peer algorithms and ontologies as alternative to approach to refined search.

Ontology in computer science is usually defined as an explicit specification of a conceptualization of a domain [15]. But, why is it important to use ontologies? We can use ontologies to describe a shared conceptualization of domain of services, devices and other concepts that could influence the discovery service process, such as different kinds of context [16], for example, in a particular case, geographical localization of a printer.

The following section will discuss discovery service enabling ontologies, as our architecture integrates such ontologies with JXTA, and which is the procedure to discovery service.

5 Architecture

In this section we present the design of ubiquitous computing architecture (ArCU) as well as a description of each of its components. We emphasize on the P2P nature of our architecture that is achieved by using JXTA [14] as our communication infrastructure. We also discuss how our ontology-based descriptions and inferences provide extra benefits to ArCU. As one of the most evident benefits of using JXTA is the fact that the JXTA's protocols implement a virtual overlay network that hides all the details about the particular instantiation of the physical communication network. In this way, our components or peers are able to interact with other components not regarding the type of communication network (wired, behind a firewall, infrastructure-based wireless) they are using [14]. As it is shown in Fig. 1, ArCU is composed by the following elements: One or more Clients (ArCU-MC) that can be mobile or static, one or more Ubiquitous Services (ArCU-US) and at least one Basic Inference Service (ArCU-BIS).

5.1.1. Mobile Clients (ArCU-MCs)

Using Clients or Mobile Clients, users are able to find and interact with the Ubiquitous Services provided by the environment. The ubiquitous services are discovered by means of queries that are issued to the Basic Inference Service (ArCU-

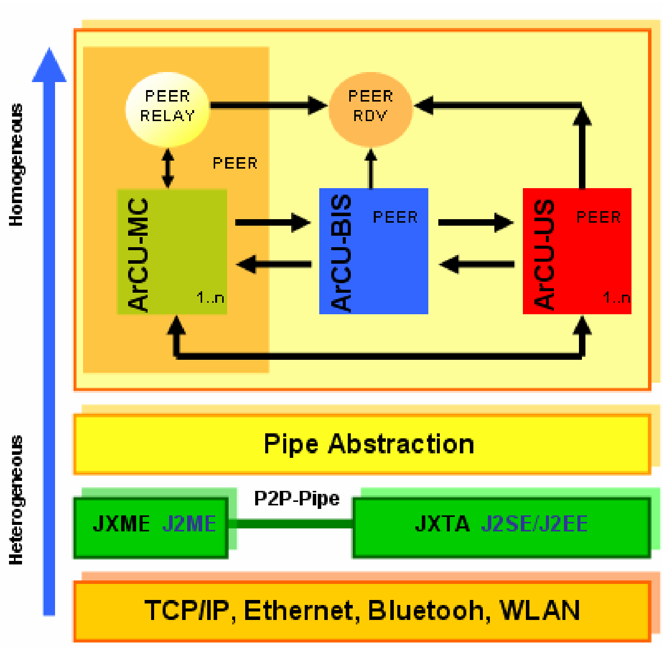


Fig. 1 ArCU Architecture.

BIS). Every Ubiquitous Service publishes a set of device-independent user interfaces (codified in XML) that can be analyzed and displayed by the ArCU-MCs. Every ArCU-MC displays the interfaces in the way that best fits its hardware capabilities. The flexibility provided by the device-independent definition is very important because a very wide range of devices may act as ArCU-MCs and some of them (i.e. cellular phones) may have restricted displaying capabilities.

On the other hand, small devices such as cellular phones or PDAs commonly have restrictions on their storage and computing capabilities and they may experience continuous disconnections due to mobility or lack of battery power. To cope with all these restrictions imposed by the nature of the hardware devices employed in ubiquitous systems we use the JXTA version for mobile devices, namely, JXME [14].

5.1.2. Ubiquitous Services (ArCU-US)

The Ubiquitous Services (ArCU-USs) are services viewed as functional components. That functionality is typically implemented as software components; some services specifically support, or are offered by devices (hardware components) where they characteristics and capabilities may play a role in the description and behavior of the services it offers. Examples of Ubiquitous Services: a projector, a printer, a public display, databases, software presentation and so on. The ArCU-USs employs JXTA announcements to publish their services in the environment. These announcements include semantic descriptions codified in OWL [17] about the services they offer.

5.1.3. Basic Inference Service (ArCU-BIS)

The Basic Inference Service (ArCU-BIS) is the most important component of ArCU. This component is subdivided in three elements: Communications Administration Service, Ontology Administrator and Search Engine.

The Communication Administration Service is in charge of the establishment and maintenance of the communication channels. The Communication Administration Service employs JXTA pipes as its communication abstraction. The JXTA pipes provide a virtual pipe that is able to communicate peers that do not have a direct physical link, or in other words, that reside in different types of networks or behind different firewalls. The endpoints of the pipes are dynamically linked (at run-time) to the endpoints of the peers and hence, peers are able to move from one domain to another in a transparent way. Moreover, using the pipe abstraction, our services can transparently recuperate from failures in any of the physical endpoints. This way, the Communication Administration Service is able to hide the fact that some devices may be mobile and change from one network to another.

The Ontology Administrator has the responsibility of creating and managing a dynamic ontology. That means? For example, a scenario is illustrated in the Fig. 2(a) with three available services. If 15 minutes later a service leave the environment such as show the Fig. 2(b).The Ontology Administrator has to refresh in no more of 5 minutes the ontology state as can be seen in the Fig. 2(c), all this in order to maintain the ontology as small as possible to help reduce the time and space complexity of the semantic searches. The Fig. 3 illustrates a section of a device ontology proposed by [18]. To this work, we adopt the use of this device ontology because helps to describe devices and their services in a rich and expressive way to facility a semantic discovery of services.

Finally, the Search Engine is in charge of performing the semantic and geographic-context-aware searches that are issued by the clients of the ubiquitous environment. As a result of these searches, Search Engine returns references to the services that meet the criteria specified by the clients. These references are further used by the clients to access the services.

5.1.4 ArCU Communication Infrastructure

ArCU builds on the basic advertisement-search JXTA mechanism. We purpose three important additons that peers providing services using ArCU must implement:

1. Pointer to XML document specify to graphic description of interfaces.
2. Pointer to OWL file that describe the service.
3. A XML Service Advertisement with need information to publish and invoke the documents previously mentioned. As it is shown in Fig. 4.

The OWL Web Ontology Language [17] is designed for use by applications that need to process the content of information instead of just presenting information to humans. OWL facilities the interpretability of content than that supported by XML, RDF, and RDF Schema (RDF-S) by providing additional vocabulary along with a formal semantics.

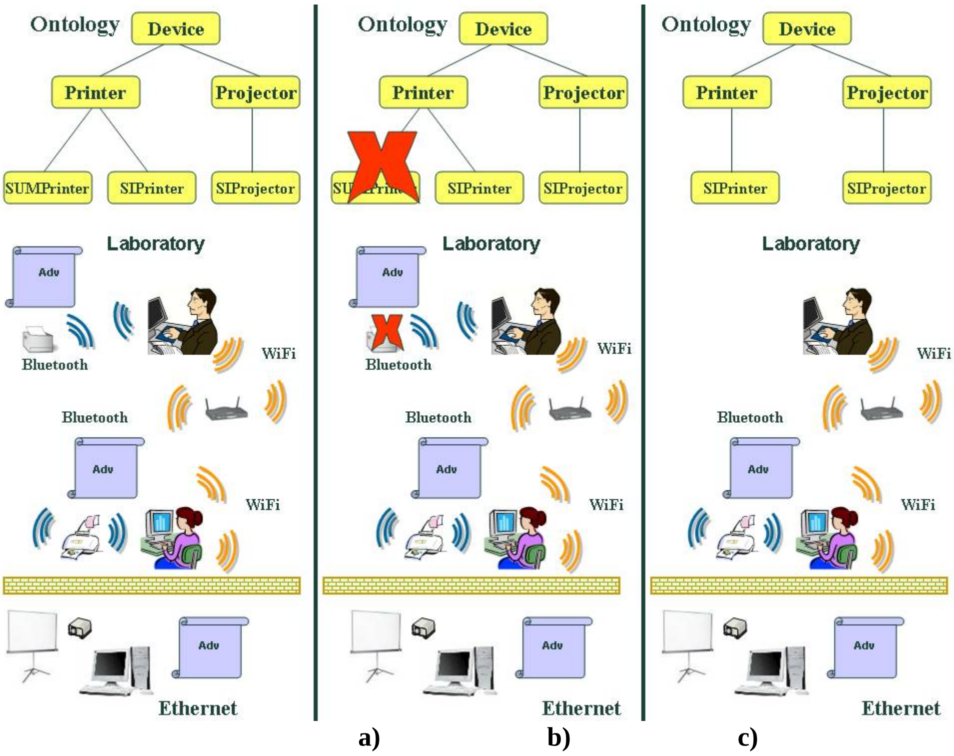


Fig. 2 Dynamic Ontology Scenario.

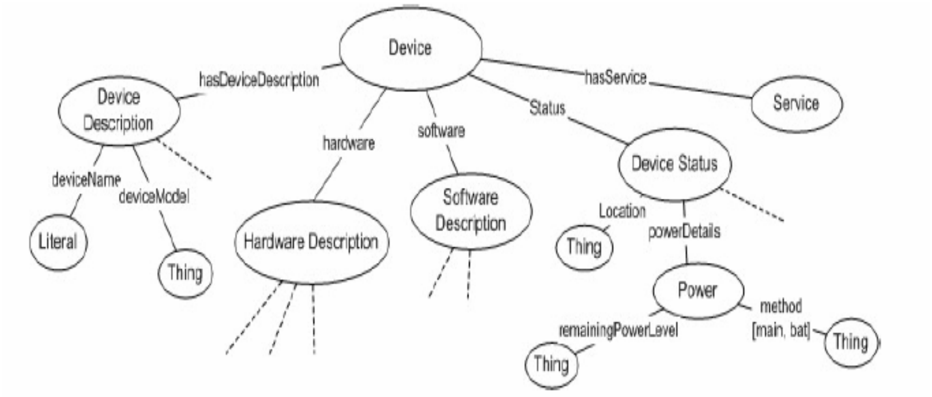
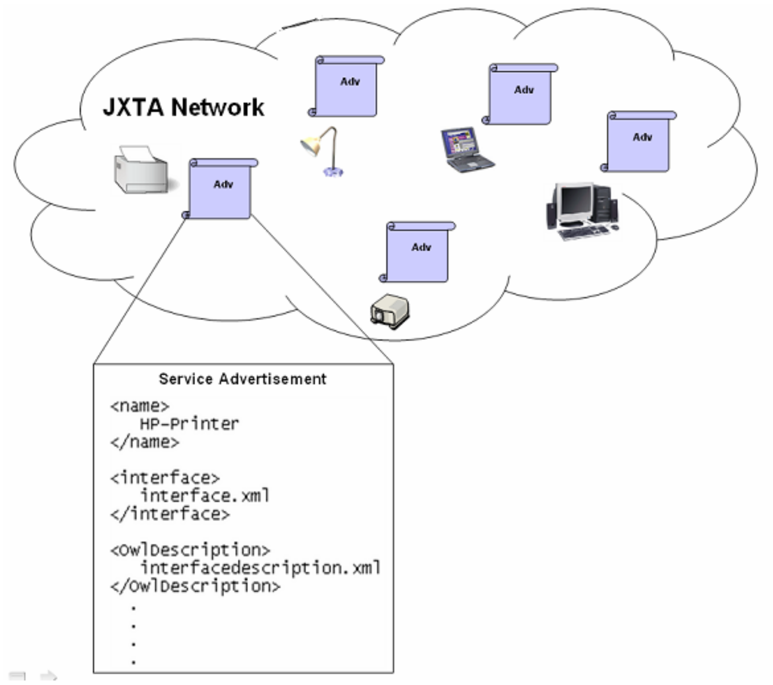


Fig. 3 Device Ontology.



```
<name>
   HP-Printer
</name>

<interface>
   interface.xml
</interface>

<OwlDescription>
   interfacedescription.xml
</OwlDescription>
  .
  .
  .
  .
```

Fig. 4 A service advertisement in ArCU.

In the case of our architecture that ArCU-BIS peer wish to matches a set of searching criteria a service need retrieve OWL file that describe the service. This remote communication is kept as previously mentioned by usign JXTA pipes as mechanism for exchanging messages between services. These messages in JXTA are also based in XML format, standarized by W3C same as OWL [17].

JXTA and OWL uses XML, the integration of both was straightforward since it doesn't commit to some standard of communication.

5.1.5 ArCU Discovery Service Mechanism

To explain the ArCU Discovery Service Mechanism is necessary describes the interactions among ArCU components for to publish and discover services. As we previously mentioned, the communication among the components of ArCU is carried out using standard JXTA protocols. A typical communication sequence is shown in Fig. 5.

1. The ArCU-BIS publishes its announcement in the distributed hash table implemented by a special-type peer known as rendezvous peer.
2. The ArCU-US peer also publishes its announcement in the distributed hash table implemented by a rendezvous peer (it may or may not be the same peer as in the step 1).

3. In order to make invocations over the services implemented by the ArCU-BIS, an ArCU-US has to find the announcement of an ArCU-BIS peer. In JXTA the management of the advertisements is carried out by the rendezvous peers who implement the distributed hash table.
4. Upon receiving a query, the rendezvous peer looks for an index of the announcement in the distributed hash table. If it finds the index, the rendezvous peer relies the query to the peer that published the announcement (in this case the ArCU-BIS). When the ArCU-BIS receives the query, it replies with its advertisement that is further received by the ArCU-US.
5. With the advertisement, the ArCU-US acquires the capability of invoking the services provided by the ArCU-BIS. In this way, the ArCU-US is able to record it services in the database of the ArCU-BIS.
6. The ArCU-MCs are implemented by a special type of peer that is called edge peer. These peers need to be coupled with another special type of peer that is called relay peer. So, in order to access the services of the ubiquitous environment, every ArCU-MC has to establish communication with a relay peer.
7. The functionality of a pair edge-relay peer is equivalent to the one of a regular JXTA peer. When the ArCU-MC is connected to a relay peer, it sends a query to a rendezvous peer looking for the announcement of an ArCU-BIS peer.
8. The rendezvous peer looks in the distributed hash table for the index of the announcement. If it finds the index, the rendezvous peer relies the query to the peer that published the announcement (in this case the ArCU-BIS).
9. When the ArCU-BIS receives the query, it replies with its advertisement that is further received by the ArCU-MC.
10. The ArCU-MC can now issue requests to the ArCU-BIS looking for a service that matches a set of searching criteria.
11. When the ArCU-BIS finds the service it replies with the service's advertisement.
12. Now, the ArCU-MC is able to send a request to the ArCU-US asking for its device-independent graphic interface.
13. Finally, the user can employ this interface to interact with the ubiquitous service.
14. It is important to insist that the graphic interface is described in XML document that is analyzed by the client device and then displayed according to its hardware capabilities.

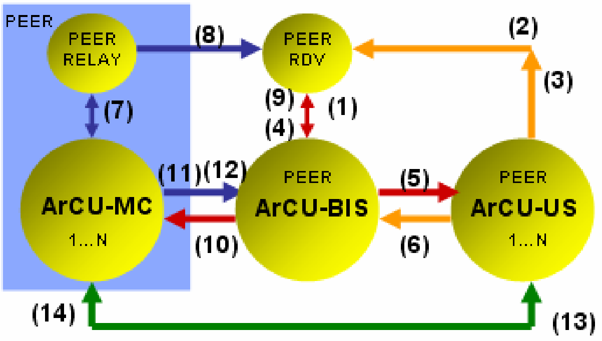
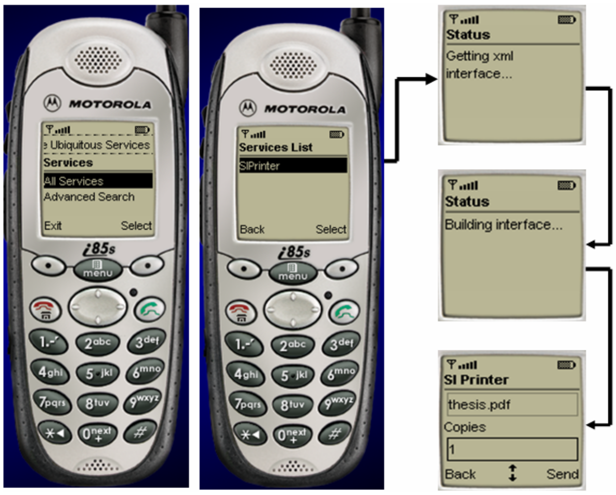


Fig. 5 Interaction among components in our ubiquitous computing platform.

5.1.6 Prototype

In order to materialize the architecture proposed we have been developed USE. USE is a Ubiquitous Services Environment. Into USE we implemented two classics ubiquitous computing scenarios, both using mobile devices and heterogeneous networks. In this section, only we illustrate one scenario. Fig. 6 show images of the graphics interfaces used in the print service scenario described below.



Getting xml interface...

Building interface...

thesis.pdf

1

Fig. 6 Print service graphics interfaces in USE.

Scenario: Rolando is a researcher in the area of ubiquitous computing and he is currently attending a meeting in the main conference room of an important research center. Later, he will be presenting some of his latest results in the same room and he needs to make some printouts to distribute them among his colleagues. To do so, Rolando uses one of the ubiquitous services provided by the conference room. Despite the fact that this is the first time that Rolando visits this particular research center, he is able to use his PDA to look for a public printer that is also located in the main conference room. As a result, the PDA gets a list of printing services that meet the criteria specified by Rolando. Once Roland selects the desired service, the PDA gets the graphic interface of the selected service. Finally, Rolando uses the interface to select the file to be printed and to send the job to the printer.

We have experimentally confirmed the functionality with real devices. Particularly we use a Sony Clié PEG-UX50 Handheld with PALM OS 5.2 with WiFi and Bluetooth technologies. With this experiment, we were able to test the potential of our infrastructure.

6 Conclusions

In this paper we presented a software architecture that allows clients (fixed or mobile) to perform semantic and geographic-context-aware searches over the services and resources contained in a ubiquitous environment. Our architecture employs a distributed hash table that works in conjunction with a set of inferences processes that are based on ontologies. In our proposal, the ubiquitous environment acts a dynamic repository that contains the ontologies that describe the services available in the environment in a given point in time. To make or architecture scalable, the algorithms employed to implement the repository were designed with the objective of reducing as much as possible the time and space complexity involved in the semantic searches. The communication among components is carried out using the standard JXTA protocols which is based on the interchange of XML documents. In the same way, the user interfaces are specified using XML documents, and each client is free to display those interfaces in the way that best matches its hardware and software capabilities. These last two design choices allow us to implement heterogeneous systems that can be composed of a wide variety of (off the shelf) hardware and software platforms.

Acknowledgments

The authors of this paper wish to thank the CIC, SIP, CONACYT, IPN and ITM for their support. Additionally, the authors wish to thank the reviewers for their pertinent comments.

References

1. Weiser, M., “The Computer for the 21st Century”, Scientific American, Vol. 265, No. 3, September 2001, pp. 94-104.
2. Satyanarayanan, M., “A Catalyst for Mobile and Ubiquitous Computing”, IEEE Pervasive Computing, Vol. 1, No. 1, January - March 2002.
3. Menchaca, R. and Favela, J., “Arquitecturas P2P para el Desarrollo de Sistemas Móviles y Ubicuos”, Septiembre 2003.
4. Balakrishnan, H., Kaashoek, F., Karger, D., Morris, R. and Stoica, I., “Looking Up Data in P2P Systems”, Comunications of the ACM, Vol. 46, No. 2, February 2003, pp. 43-48.
5. Kubiatowicz, J. M., “Extracting Guarantees from Chaos”, Comunications of ACM, Vol. 46, No. 2, February 2003, pp. 33-38.
6. Universal Plug and Play. www.upnp.org.
7. Guttman, E., “Service Location Protocol: Automatic Discovery of IP Network Services”, IEEE Internet Computing, Vol. 3, No. 4, August 1999, pp. 71-80.
8. Edwards, W. K., “Discovery Systems in Ubiquitous Computing”, IEEE Pervasive Computing, Vol. 5. No. 2, April 2006, pp. 70-77.
9. Salutarion Arquitecture Specification, v. 2.1, Salutation Consortium, 1999.
10. Chakraborty, D., Perich, F., Avancha, S. and Joshi, A., “DReggie: Semantic Service Discovery for M-Commerce Applications”, in Workshop on Reliable and Secure Applications in Mobile Environment, Symposium on Reliable Distributed Systems, October 2001.
11. Specification of the Bluetooth System, v.1.1 core, Bluetooth Consortium, 2001; http://bluetooth.com.

12. Waldo, J., The Jini Specifications, edited by Ken Arnold. Addison-Wesley Professional, Second edition. December 15, 2000.
13. Arnold, K., Wollrath, A., O'Sullivan, A., Scheifler, R. and Waldo, J., The Jini Specifications. Addison-Wesley, Reading, MA, USA, 1999.
14. Proyect JXTA 2.0 Super-Peer Virtual Network. http://www.jxta.org/project/www/docs/JXTA2.0protocols1.pdf .
15. Gruber T. "Toward Principles forthe Design of Ontologies Used for Knowledge Sharing". Technical Report KSL-93-04, Knowledge Systems Laboratory, Stanford University, CA, 1993.
16. Schmidt, A., "Ubiquitous Computing—Computing in Context", Submitted for the degree of Doctor of Philosophy, Lancaster University, England, U.K. November 2002.
17. http://www.comp.lancs.ac.uk/~albrecht/pubs/.
18. OWL Web Ontology Language: Overview. http://www.w3.org/TR/owl-features/
19. Bandara, A., Payne, T., De Roure, D. and Clemo, G., "An Ontological Framework for Semantic Description of Devices", the 3rd International Semantic Web Conference, Japan, November 2004.
20. Verma, K., Sivashanmugam, K., Sheth, A., Patil, A., Oundhakar, S. and Miller, J.,"METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services", Journal of Information Technology and Management, under review.
21. Paolucci, M., Sycara, K., Nishimura, T. and Srinivasan, N., "Using DAML-S for P2P Discovery", in Proceedings of the International Conference on Web Services, 2003.

Towards a Context-Aware and Pervasive Multimodality*

Manolo Dulva Hina [1,2], Chakib Tadj [1], Amar Ramdane-Cherif [2], and Nicole Levy [2]

[1] Université du Québec, École de technologie supérieure,
1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada
{manolo-dulva.hina.1@ens.etsmtl.ca, ctadj@ele.etsmtl.ca}
[2] PRISM Laboratory, Université de Versailles-Saint-Quentin-en-Yvelines,
45, avenue des États-Unis, 78035 Versailles Cedex, France
{rca, nicole.levy}@prism.uvsq.fr

Abstract. Pervasive multimodality aims to realize anytime, anywhere computing using various modes of human-machine interaction, supported by various media devices other than the traditional keyboard-mouse-screen used for data input and output. To achieve utmost efficiency, the modalities for human-machine interaction must be selected based on their suitability to a given interaction context (i.e. combined user, environment and system contexts) and the availability of supporting media devices. To be fault-tolerant, the system should be capable of finding replacement to a failed media device. This paper presents a paradigm of such computing system. Proposed solutions to some technical challenges including the establishment of relationships among interaction context, modalities and media devices, and of the mechanism for the system's incremental acquisition of knowledge on various interaction contexts and their corresponding selections of suitable modalities are presented in this work.

1 Introduction

In the very near future, we shall be living in a society in which pervasive computing (also known as ubiquitous computing) [1, 2] will no longer be a luxury but a way of life. In such a computing system, a user can continue working on a computing task, using various applications, whenever and wherever he wants. The infrastructure of pervasive computing [3] will be available and the applications seamlessly adapting accordingly to the user's context [4] and available resources. Multimodality [5], on its part, promotes the use of different modalities for human interaction (i.e. data entry and data output), the choice of modality being a function of the user's context. Media devices, subject to their availability and suitability to context, may be selected to support then chosen modality. Hence, pervasive multimodality shall be a computing trend of the future, one in which computing adapts to the needs of the users, including those with disabilities. To further enhance its functionalities, a pervasive multimodal system may be designed with machine learning [6, 7] capability, a mechanism in

* This research work has been made possible through the funding provided by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the scholarship grants from Décanat des études, of Université du Québec, École de technologie supérieure.

Received 21/08/07
Accepted 19/10/07
Final version 25/10/07

which the system "learns" from previous experiences to improve its performance, promotes autonomy and is itself fault-tolerant. The design of such system is filled with technical challenges that need optimal solutions. This paper presents our view and proposed solutions to the challenges to achieve pervasive multimodality. The rest of this paper is structured as follows. Works related to ours are listed in section 2. Section 3 lists down some of the technical challenges and our proposed solutions. The main contents, the matters related to interaction context and multimodality, and the context learning and adaptation are discussed in sections 4 and 5. The paper is concluded in section 6.

2 Related Work

Multimodality advocates the use of various modes for human interaction (data input/output) with a computer beyond the traditional keyboard-mouse-screen input and output devices. A few recent works in this domain include an interface for wireless user interface [5] and the static user interface [8]. Multimodality also refers to the fusion of two (or more) modalities. Sample works in this area include the combined speech and pen inputs [9] and the combined speech and lips movements [10]. Multimodality, as a domain in human-computer interface, aims at providing increased usability to the user, such that a modality that is weak for a given setting may be replaced by another but more appropriate modality. For example, using speech instead of mouse and keyboard as input device/s is more suitable for users doing computing in a moving car. Compared with others, our research goal on pervasive multimodality is to realize anytime, anywhere computing where users will use modalities that are suitable to their given context.

In [4], Coutaz et al explained the importance of context. Research has gone a long way since Dey provided the basic definition of context [11] as applied in context-aware computing [12]. Rey and Coutaz updated the definition in [13] and coined the term "*interaction context*" (IC) to mean the user, the environment and the system's contexts. Our work focuses on the IC in pervasive multimodal computing and considers both static and dynamic context data, including sensed, derived and profiled context information. There has been an active research going on in pervasive and mobile computing. The Prism model in Project Aura [14], for example, demonstrates a user's moving aura (i.e. user's profile and task). Our work has extended this concept by considering an incremental learning system in which acquired knowledge becomes part of pervasive information, along with user's profile, task and preferences. We inject the features of adaptability and autonomy into the system via machine learning.

3 Technical Challenges

Here, we list down some software engineering challenges of pervasive multimodality via technical requirements that need to be addressed and also describe our approach.

Our goal is to model a pervasive computing system that senses its current IC and chooses appropriate modalities and their supporting media devices. The design of such a system needs to address the key requirements cited below:

Requirement 1: *Provide a relationship between a modality and an IC (i.e. combined user, environment and system contexts) and a relationship between a modality and media devices.* Given that the application domain is multimodality, what parameters constitute the user, environment and system contexts? On what basis a specific modality is considered suitable to an IC? How media devices are selected to support a particular modality?

Requirement 2: *Provide a mechanism that allows the system to acquire incremental acquisition of knowledge related to IC-modalities-media devices scenario.* What machine learning methodology should be adopted if the system is to learn scenarios incrementally? How does it acquire knowledge on new IC scenarios?

Requirement 3: *Provide a mechanism allowing the system to be fault-tolerant on failed media devices.* If a chosen media device fails (i.e. absent or not functional), what media device replacement gets selected, and on what ground?

The technical challenges are addressed by the proposed solutions given below.

Proposed solution to requirement 1: The modalities for human-machine interaction are manual, visual and vocal both for data input and output (details in next section). An IC is composed of user, environment and system context parameters that are all related to modalities. The relationship to consider is how specific modality becomes suitable to an IC parameter and by how much (i.e. high, medium, low or inappropriate). To that effect, all media devices must be grouped in such a way that a relationship between modalities and media group may be established.

Proposed solution to requirement 2: We adopt machine learning; the system is trained with scenarios (i.e. interaction content – modalities) and each one learned is stored in a repository as an exemplar. Using case-based reasoning with supervised learning, a new IC (pre-condition scenario) is compared against stored exemplars; if a match is found, the corresponding post-condition scenario is implemented. Otherwise, a new case is considered for learning.

Proposed solution to requirement 3: We aim to design an autonomous, adaptive and fault-tolerant system. In case of faulty media device, a replacement is searched. Media devices are ranked by priority. The faulty top-ranked device is automatically replaced by second-ranked device (if available) then by the next-ranked device, and so on until a replacement is found. When replacement is not possible, the currently-chosen optimal modality is up for replacement.

4 Interaction Context and Multimodality

An *interaction context*, $\mathsf{IC} = \{\mathsf{IC}_1, \mathsf{IC}_2, \ldots, \mathsf{IC}_{\max}\}$, is a set of all possible interaction contexts. At any given time, a user has a specific interaction context *i* denoted as IC_i, $1 \leq i \leq \max$, which is composed of variables that are present during the conduct of the user's activity. Each variable is a function of the application domain which, in this work, is multimodality. Formally, an IC is a tuple composed of a specific user context (UC), environment context (EC) and system context (SC). An instance of IC is given as:

$$\mathrm{IC_i} = \mathrm{UC_k} \otimes \mathrm{EC_l} \otimes \mathrm{SC_m} \tag{1}$$

where $1 \leq k \leq max_k$, $1 \leq l \leq max_l$, and $1 \leq m \leq max_m$, and max_k, max_l and max_m = maximum number of possible user contexts, environment contexts and system contexts, respectively. The Cartesian product (symbol: ⊗) denotes that IC yields a specific combination of UC, EC and SC at any given time.

The user context UC is composed of application domain-related parameters that describe the state of the user during his activity. Any specific user context k is given by:

$$\mathrm{UC_k} = \bigotimes_{x=1}^{\max_k} \mathrm{ICParam_{kx}} \tag{2}$$

where $\mathrm{ICParam_{kv}}$ = parameter of $\mathrm{UC_k}$, k = the number of UC parameters. Similarly, any environment context $\mathrm{EC_l}$ and system context $\mathrm{SC_m}$ are specified as follows:

$$\mathrm{EC_l} = \bigotimes_{y=1}^{\max_l} \mathrm{ICParam_{ly}} \tag{3}$$

$$\mathrm{SC_m} = \bigotimes_{z=1}^{\max_m} \mathrm{ICParam_{mz}} \tag{4}$$

As stated, multimodality selects the modality based on its suitability to the given IC. Here, *modality* refers to the logical interaction structure (i.e. the mode for data input and output between a user and computer). A modality, however, may only be realized if there is/are media devices that would support it. Here, a *media* refers to a set of physical interaction devices (and some software supporting the physical devices). With natural language processing as basis, modalities are grouped as follows: (1) *Visual Input* ($\mathrm{VI_{in}}$), (2) *Vocal Input* ($\mathrm{VO_{in}}$), (3) *Manual/Tactile Input* ($\mathrm{M_{in}}$), (4) *Visual Output* ($\mathrm{VI_{out}}$), (5) *Vocal Output* ($\mathrm{VO_{out}}$), and (6) *Manual/Tactile Output* ($\mathrm{M_{out}}$). Multimodality, therefore, is possible if there is at least one modality for data input and at least one modality for data output:

$$\mathrm{Modality} = (\mathrm{VI_{in}} \vee \mathrm{VO_{in}} \vee \mathrm{M_{in}}) \wedge (\mathrm{VI_{out}} \vee \mathrm{VO_{out}} \vee \mathrm{M_{out}}) \tag{5}$$

Accordingly, media devices themselves are grouped as follows: (1) Visual Input Media (VIM), (2) Visual Output Media (VOM), (3) Oral Input Media (OIM), (4) Hearing Output Media (HOM), (5) Touch Input Media (TIM) (6) Manual Input Media (MIM), and (7) Touch Output Media (TIM). See Fig. 1.

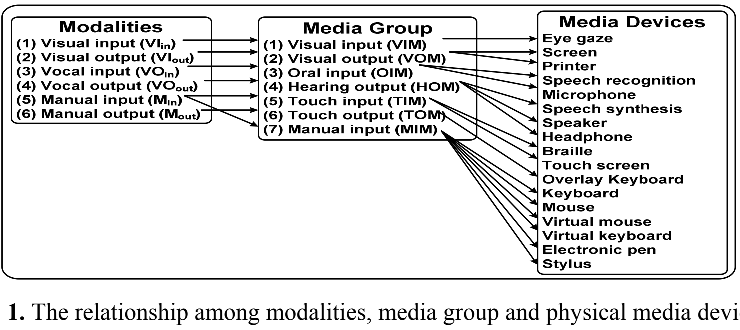


Fig. 1. The relationship among modalities, media group and physical media devices.

For the relationship between modalities and media devices, let there be a function $\mathbf{g_1}$ that maps a modality to a media group, given by $\mathbf{g_1}$*: Modality → Media group*. This is shown in Fig. 1. There is often a case when two or more media devices belong to one media group. In such a case, media device selection is determined through their priority rankings. Hence, let there be a function $\mathbf{g_2}$ that maps a media group to a media device and the media's priority ranking, denoted by $\mathbf{g_2}$**:** *Media group → (Media device, Priority)*. Sample elements of the two stated functions are:

$\mathbf{g_1}$ = {($\mathrm{VI_{in}}$,VIM), ($\mathrm{VI_{out}}$,VOM), ($\mathrm{VO_{in}}$,OIM), ($\mathrm{VO_{out}}$,HOM), ($\mathrm{M_{in}}$,TIM), ($\mathrm{M_{in}}$,MIM), ($\mathrm{M_{out}}$,TOM)}

$\mathbf{g_2}$ = {(VIM, (eye gaze, 1)), (VOM, (screen, 1)), (VOM, (printer, 1)), (OIM, (speech recognition, 1)), (OIM, (microphone, 1)), (HOM, (speech synthesis, 1)), (HOM, (speaker, 2)), (HOM, headphone, 1)), etc.}

A modality's suitability to IC is equal to its collective suitability to IC's individual parameters. Instead of binary (suitable or not), our measure of suitability is that of *high, medium, low* or *inappropriate*. *High suitability* means that the modality being considered is the preferred mode for computing; *medium suitability* means the modality is simply an alternative mode, hence, its absence is not considered as an error but its presence means added convenience to the user. *Low suitability* means the modality's effectiveness is negligible and is the last recourse when everything else fails. *Inappropriateness* recommends that the modality should not be used at all.

If the collective IC is composed of *n* parameters, then a modality being considered has *n* suitability scores, one score for each parameter. The following conventions are adopted:

1. A modality's suitability to an IC parameter is one of the following: H (high), M (medium), L (low), and I (inappropriate). Mathematically, H = 1.00, M = 0.75, L = 0.50, and I = 0.
2. The modality's suitability score to an IC is given by:

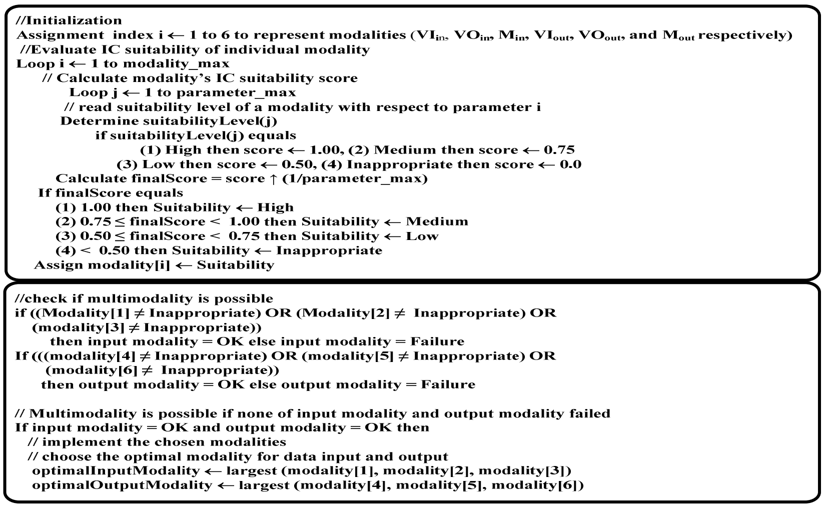
$$\mathrm{SuitabilityScore_{modality}} = \sqrt[n]{\prod_{i=1}^{n} \mathrm{context\_parameter}_{\,i}} \tag{6}$$

where i = parameter index and n = total number of parameters. Given the calculated value, a modality's IC suitability is given by:

$$\mathrm{Suitability_{modality}} = \begin{cases} \mathrm{H\ if\ SuitabilityScore_{modality}} = 1.00 \\ \mathrm{M\ if\ } 0.75 \le \mathrm{SuitabilityScore_{modality}} < 1.00 \\ \mathrm{L\ if\ } 0.50 \le \mathrm{SuitabilityScore_{modality}} < 0.75 \\ \mathrm{I\ if\ SuitabilityScore_{modality}} < 0.50 \end{cases} \tag{7}$$

Fig. 2 shows the algorithm for determining the suitability of modalities to a given IC and if multimodality is possible (i.e. equation 5). Checking the possibility of multimodality is done by checking that not all of input modalities (i.e. specified by indexes 1, 2 and 3) are scored "inappropriate". The same is true for output modalities (i.e. specified by indexes 4, 5 and 6). The *optimal input modality* is chosen from a group of input modalities, and is one with the highest IC suitability score. The same principle applies to the selection of *optimal output modality*. Subject to the availability of media devices, an optimal modality is ought to be implemented; all other modalities are considered optional. In the absence of supporting media devices, an alternative modality is chosen and is one with the next highest score. The process is repeated until the system finds a replacement modality that can be supported by currently available media devices.

If multimodality is possible and the optimal modalities are chosen, then supporting media devices are checked for availability. Using function $\mathbf{g_1}$, the media group that support the chosen modality may be identified. Given that **Modality** = {VI_{in}, VO_{in}, M_{in}, VI_{out}, VO_{out}, M_{out}} and **Media Group** = {VIM, OIM, MIM, TIM, VOM, HOM, TOM} and that $\mathbf{g_1}$**:** *Modality → Media group*, then formally, for all media group **p**, there exists a modality **q** such that the mapping between **p** and **q** is in set $\mathbf{g_1}$**,** that is $\forall$**p: Media group,** $\exists$**q: Modality** $|$ **p** $\rightarrow$ **q** $\in$ $\mathbf{g_1}$. Using function $\mathbf{g_2}$, the top-ranked media devices that belong to such media group are also identified. Given function $\mathbf{g_2}$, a media device **d**, the priorities **p1** and **p2** where Priority: $\mathbf{N_1}$ (positive numbers excluding zero), then the specification for finding the top-ranked device for a media group **m** is $\exists$**m:** Media group, $\forall$**d:** Media device, $\exists$**p1:** Priority, $\forall$**p2:** Priority $|$ **d • m** $\rightarrow$ **(d, p1)** $\in$ $\mathbf{g_2} \wedge$ **(p1 < p2).**



```
//Initialization
Assignment  index i ← 1 to 6 to represent modalities ($VI_{in}$, $VO_{in}$, $M_{in}$, $VI_{out}$, $VO_{out}$, and $M_{out}$ respectively)
 //Evaluate IC suitability of individual modality
Loop i ← 1 to modality_max
    // Calculate modality's IC suitability score
        Loop j ← 1 to parameter_max
       // read suitability level of a modality with respect to parameter i
      Determine suitabilityLevel(j)
            if suitabilityLevel(j) equals
                    (1) High then score ← 1.00, (2) Medium then score ← 0.75
                (3) Low then score ← 0.50, (4) Inappropriate then score ← 0.0
      Calculate finalScore = score ↑ (1/parameter_max)
   If finalScore equals
      (1) 1.00 then Suitability ← High
      (2) 0.75 ≤ finalScore <  1.00 then Suitability ← Medium
      (3) 0.50 ≤ finalScore <  0.75 then Suitability ← Low
      (4) <  0.50 then Suitability ← Inappropriate
  Assign modality[i] ← Suitability

//check if multimodality is possible
if ((Modality[1] ≠ Inappropriate) OR (Modality[2] ≠  Inappropriate) OR
   (modality[3] ≠ Inappropriate))
      then input modality = OK else input modality = Failure
If (((modality[4] ≠ Inappropriate) OR (modality[5] ≠ Inappropriate) OR
     (modality[6] ≠  Inappropriate))
     then output modality = OK else output modality = Failure

// Multimodality is possible if none of input modality and output modality failed
If input modality = OK and output modality = OK then
  // implement the chosen modalities
  // choose the optimal modality for data input and output
   optimalInputModality ← largest (modality[1], modality[2], modality[3])
   optimalOutputModality ← largest (modality[4], modality[5], modality[6])
```

Fig. 2. Algorithm to determine modality's suitability to IC and if multimodality is possible.

Let there be a *media devices priority table* (MDPT) (see Table 1) which tabulates all media groups, and each media group's set of supporting media devices, arranged by priority ranking. Let **T** = {T_1, T_2… T_{max_table}} be the set of MDPT's. The elements of table $T_n \in$ **T**, where n = 1 to *max_table*, are similar to elements of function $\mathbf{g_2}$. Every T_n is unique; no two MDPT's are identical. To create a new table, at least one of its elements is different from all other tables that have already been defined. The priority ranking of a specific media device may be different in each MDPT. In general, any given IC scenario and its suitable modalities is mapped/assigned to a specific MDPT.

5 Context Learning and Adaptation

In concept, *machine learning* (ML) is about programming that optimizes a system's performance through the use of sample data or past experiences. ML is important when human expertise does not exist. Hence, learning rule is formulated from acquired data [7]. A machine is said to have "learned" if its performance in doing a task improves with its experience [6]. In this work, the objective of adopting ML is for the system to learn various IC's and each one's selection of appropriate modalities and supporting media devices. Each learned IC is a knowledge learned and is stored in a repository (i.e. it becomes an "exemplar"). When the same IC case reoccurs, the system automatically adopts the IC case's corresponding multimodality selections with little or no human intervention.

Table 1. A sample media devices priority table (MDPT).

Media Group	Media Devices				
	Priority = 1	Priority = 2	Priority = 3	::	Priority = *n*
Visual Input	Eye Gaze				
Oral Input	Microphone, Speech Recognition				
Touch Input	Touch Screen	Braille Terminal			
Manual Input	Mouse, Keyboard	Virtual Mouse, Virtual keyboard	Electronic Pen	Stylus	Braille
Visual Output	Screen	Printer	Electronic Projector		
Hearing Output	Speaker	Headphone, Speech Synthesis			
Touch Output	Braille	Overlay Keyboard			

System knowledge acquisition begins with the establishment of a priori knowledge, those related to IC parameters. An example of an IC parameter is shown in Table 2. As shown, the IC parameter is the "user location". The value of this parameter is deduced from the data taken by a sensor (i.e. a GPS). To formulate this specific a priori knowledge, some specified values of latitude and longitude are assigned with specific meanings (we call them "conventions"). When sample sensor readings are taken, the system compares them with the conventions and concludes whether the user is "at home", "at work" or "on the go". Then, the expert (i.e. end user) is supplies his perceived suitability score of each modality for each user location convention (see Table 2(b)). Hence, based on the given value of an IC parameter (e.g. user location), the system easily retrieves the suitability score of each modality.

Table 2. Sample User context parameter – convention and modalities selections

(a): User location convention table using GPS values

Convention No.	Latitude	Longitude	Meaning
1	<*value*$_{11}$>	<*value*$_{12}$>	*At home*
2	<*value*$_{21}$>	<*value*$_{22}$>	*At work*
3	not <*value*$_{11}$> *AND* not <*value*$_{21}$>	not <*value*$_{21}$> *AND* not <*value*$_{22}$>	*On the go*

(b): Modality selection based on user location

Type of Modality	User location = At home	User location = At work	User location = On the go
Visual Input	H	H	L
Visual Output	H	H	H
Vocal Input	H	H	H
Vocal Output	H	H	H
Manual Input	H	H	H
Manual Output	H	H	H

In general, if a system is to become reliable in its detection of the suitability of all modalities to a given IC, it needs the most a priori knowledge on context parameters as possible. In our work, an end user can add, modify, and delete one context parameter at a time using our layered virtual machine for incremental definition of IC (implemented but not shown here due to space constraints). When all the a priori knowledge are collected and grouped together, it forms a tree-like IC structure, as shown in Fig. 3. Every new IC parameter is first classified as either UC or EC or SC parameter and is appended as a branch of UC or EC or SC. Then, the conventions of the parameter are identified as well as the modalities' suitability scores in each convention.

There are cases, however, when a certain IC parameter's value could nullify the importance of another IC parameter. For example, the declaration

user_handicap (blind) *nullifies* light_intensity()

states that UC parameter "user handicap" nullifies the EC parameter "light intensity". As such, whatever light intensity value or convention is identified by a sensor is simply ignored in the calculation of the overall modality's suitability to the given IC.

Distinct scenarios that the system had encountered are stored in the knowledge database as an exemplar while a current one becomes a "case". A *case* is composed of three elements: (1) *the problem* – the IC in consideration, composed of UC, EC and SC parameters and their values or conventions, (2) *the solution* – the final IC suitability of each modality, and (3) *the evaluation* – the relevance score of the solution.

When the ML component receives a new scenario (i.e. new IC), it converts it into a case, specifying the problem. Using the similarity algorithm, it compares the problem in the new case against all the available problems/exemplars in the knowledge database. The scenario of the closest match is selected and its solution is returned. The evaluation is the score of how similar it is to the closest match. If no match is found (relevance score is low), the ML component takes the closest various scenarios and regroup and organized them to find the solution of the new case. The user may or may not accept the proposed solution. In case of expert refusal, a new case with supervised learning is produced, the problem to the case resolved. Afterwards, ML component adds the new case in its knowledge database. This whole learning mechanism is called *case-based reasoning with supervised learning*.

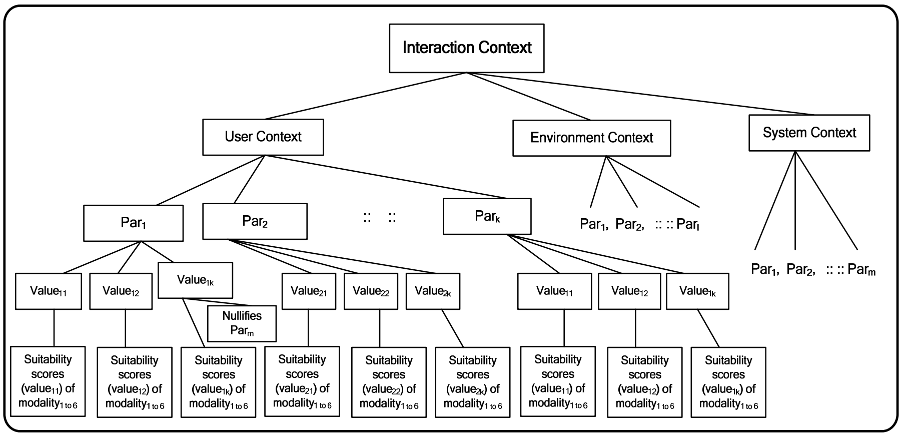

Interaction Context
User Context
Environment Context
System Context
Par_1
Par_2
:: ::
Par_k
Par_1, Par_2, :: :: Par_l
Par_1, Par_2, :: :: Par_m
$Value_{11}$
$Value_{12}$
$Value_{1k}$
Nullifies Par_m
$Value_{21}$
$Value_{22}$
$Value_{2k}$
$Value_{11}$
$Value_{12}$
$Value_{1k}$
Suitability scores ($value_{11}$) of $modality_{1 to 6}$
Suitability scores ($value_{12}$) of $modality_{1 to 6}$
Suitability scores ($value_{1k}$) of $modality_{1 to 6}$
Suitability scores ($value_{21}$) of $modality_{1 to 6}$
Suitability scores ($value_{22}$) of $modality_{1 to 6}$
Suitability scores ($value_{2k}$) of $modality_{1 to 6}$
Suitability scores ($value_{11}$) of $modality_{1 to 6}$
Suitability scores ($value_{12}$) of $modality_{1 to 6}$
Suitability scores ($value_{1k}$) of $modality_{1 to 6}$

Fig. 3. The structure of stored IC parameters.

Inspired by [15], we modify their similarity scoring scheme to reflect the needs of our system. Hence, given a new case (NC) and an individual case stored in the knowledge database (MC), the similarity of the problem between the two cases, that is NC against MC as denoted by the subscripts, is equal to their similarity in the case's UC, EC and SC and is given by:

$$\mathrm{Sim}(\mathrm{NC},\mathrm{MC}) = \tfrac{1}{3}\mathrm{Sim}(\mathrm{UC}_{\mathrm{NC}},\mathrm{UC}_{\mathrm{MC}}) + \tfrac{1}{3}\mathrm{Sim}(\mathrm{EC}_{\mathrm{NC}},\mathrm{EC}_{\mathrm{MC}}) + \tfrac{1}{3}\mathrm{Sim}(\mathrm{SC}_{\mathrm{NC}},\mathrm{SC}_{\mathrm{MC}}) \quad (8)$$

The similarity between the UC of NC against the UC of MC is given by:

$$\mathrm{Sim}(\mathrm{UC}_{\mathrm{NC}},\mathrm{UC}_{\mathrm{MC}}) = \frac{\sum_{i=1}^{\max_{\mathrm{NC}}} \mathrm{Sim}(\mathrm{UC\_Par}_{i_{\mathrm{NC}}},\mathrm{UC}_{\mathrm{MC}})}{\max(\mathrm{UC\_Par}_{\mathrm{NC}},\mathrm{UC\_Par}_{\mathrm{MC}})} \quad (9)$$

where *UC_Par_i, i* = 1 to max, is the individual UC parameter, *$max(UC\_Par_{NC}, UC\_Par_{MC})$* is the greater between the number of UC parameters between NC and MC, and $\mathrm{Sim}(\mathrm{UC\_Par}_{i_{\mathrm{NC}}},\mathrm{UC}_{\mathrm{MC}}) = \max_{j=1 \text{ to } \max_{\mathrm{MC}}} \mathrm{Sim}(\mathrm{UC\_Par}_{i_{\mathrm{NC}}},\mathrm{UC\_Par}_{j_{\mathrm{MC}}})$ where $\mathrm{UC\_Par}_{j_{\mathrm{MC}}} \in \mathrm{UC}_{\mathrm{MC}}$ and $\mathrm{Sim}(\mathrm{UC\_Par}_{i_{\mathrm{NC}}},\mathrm{UC\_Par}_{j_{\mathrm{MC}}}) \in [0, 1]$ is the similarity between a specific UC parameter *i* of NC and parameter *j* of MC.

For the similarity measures of EC of NC against EC of MC, and the SC of NC against SC of MC, the same principle as Equation 9 must be applied, with the formula adjusted accordingly to denote EC and SC, respectively, yielding:

$$\mathrm{Sim}(\mathrm{EC}_{\mathrm{NC}},\mathrm{EC}_{\mathrm{MC}}) = \frac{\sum_{i=1}^{\max_{\mathrm{NC}}} \mathrm{Sim}(\mathrm{EC\_Par}_{i_{\mathrm{NC}}},\mathrm{EC}_{\mathrm{MC}})}{\max(\mathrm{EC\_Par}_{\mathrm{NC}},\mathrm{EC\_Par}_{\mathrm{MC}})} \quad (10)$$

$$\mathrm{Sim}(\mathrm{SC}_{\mathrm{NC}},\mathrm{SC}_{\mathrm{MC}}) = \frac{\sum_{i=1}^{\max_{\mathrm{NC}}} \mathrm{Sim}(\mathrm{SC\_Par}_{i_{\mathrm{NC}}},\mathrm{SC}_{\mathrm{MC}})}{\max(\mathrm{SC\_Par}_{\mathrm{NC}},\mathrm{SC\_Par}_{\mathrm{MC}})} \quad (11)$$

Equation 8 assumes that the weights of UC, EC and SC are equal (i.e. each is worth 33.3%). This figure is not fixed and can be adjusted to suit the need of the expert. An ideal case match is a perfect match. However, a score of 90% means that a great deal of IC parameters is correctly considered and is therefore 90% accurate. The expert, however, decides the threshold score of what is considered as an acceptable match.

When the IC-appropriate modalities are satisfactorily identified, the media devices supporting the modalities are checked for availability. If available, the devices are simply activated. Otherwise, a replacement is searched. Via MDPT, the media device that is next in priority is searched. The process is repeated until a replacement is found (see Fig. 4). Formally, given a failed device **d** of priority **p1,** the specification for finding the replacement media device $\mathbf{d_{rep}}$ is ∃m: Media Group, ∀**drep:** Media Device, ∃**p1:** Priority, ∀**p2:** Priority | **(p1 = p1 + 1) ∧ (p1 < p2) ∧ m → (drep, p1) ∈ g₂ • $\mathbf{d_{rep}}$.**

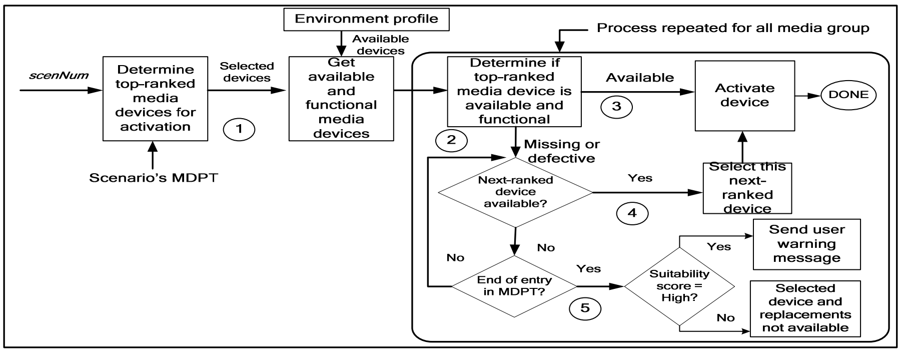

Environment profile
Available devices
Process repeated for all media group
scenNum
Determine top-ranked media devices for activation
Selected devices
1
Get available and functional media devices
Scenario's MDPT
Determine if top-ranked media device is available and functional
Available
3
Activate device
DONE
2
Missing or defective
Next-ranked device available?
Yes
4
Select this next-ranked device
No
No
End of entry in MDPT?
Yes
5
Suitability score = High?
Yes
Send user warning message
No
Selected device and replacements not available

Fig. 4. Algorithm for finding replacement to a failed media device.

6 Conclusion

In this paper, we presented the major challenges in designing the infrastructure of pervasive multimodality. We address those challenges by presenting the elements that comprise a given interaction context, the grouping of modalities, media groups and media devices. We establish the relationship among them and provide their formal specifications. Machine learning is used to build an autonomous and interaction context-adaptive system. Such learning system needs a priori knowledge on context parameters and the methodology to augment it incrementally. Also, the acquisition of various scenarios is presented in this paper. Finally, we demonstrate one fault-tolerant characteristic of the system by providing the mechanism that finds a replacement to a failed media device.

References

1. Weiser, M., The computer for the twenty-first century. Scientific American, Vol. **265**(3), (1991) 94 - 104.
2. Vasilakos, A.V. and Pedrycz, W. Ambient Intelligence, Wireless Networking, Ubiquitous Computing, ArtecHouse, USA (2006).
3. Satyanarayanan, M., Pervasive Computing: Vision and Challenges. IEEE Personal Communications, Vol. **8**(4), (2001) 10-17.
4. Coutaz, J., et al., Context is key. Communications of the ACM, Vol. **48**(3) (2005), 49-53.
5. Ringland, S.P.A. and F.J. Scahill, Multimodality - The future of the wireless user interface. BT Technology Journal, Vol. **21**(3), (2003) 181-191.
6. Mitchell, T., Machine Learning, McGraw-Hill (1997).
7. Alpaydin, E., Introduction to Machine Learning, Cambridge, MA, USA, MIT Press (2004).
8. Oviatt, S.L. and Cohen, P.R., Multimodal Interfaces that Process What Comes Naturally. Communications of the ACM, Vol. **43**(3), (2000) 45 - 53.
9. Oviatt, S.L., Designing the User Interface for Multimodal Speech and Gesture Applications: State-of-the-art Systems and Research Directions. Human Computer Interaction, Vol. **15**(4), (2000), 263-322.

10. Rubin, P., Vatikiotis-Bateson, E. and Benoit, C., Audio-Visual Speech Processing. Speech Communications, Vol. **26**(1-2) (1998).
11. Dey, A.K. and Abowd, G.D., Towards a Better Understanding of Context and Context-Awareness. in 1st Intl. Conference on Handheld and Ubiquitous Computing, Karlsruhe, Germany: Springer-Verlag, LNCS 1707 (1999).
12. Dey, A.K., Understanding and Using Context Springer Personal and Ubiquitous Computing, Vol. **5**(1), (2001) 4 - 7.
13. Rey, G. and Coutaz. J., Le contexteur: une abstraction logicielle pour la réalisation de systèmes interactifs sensibles au contexte in Proceedings of the 14th French-speaking Conference on Human-Computer Interaction (Conférence Francophone sur l'Interaction Homme-Machine) IHM '02, Poitiers, France, ACM Press (2002).
14. Garlan, D., et al., Project Aura: Towards Distraction-Free Pervasive Computing. IEEE Pervasive Computing, Special Issue on Integrated Pervasive Computing Environments, Vol. **21**(2), (2002) 22 - 31.
15. Lajmi, S., Ghedira, C. and Ghedira, K., Une méthode d’apprentissage pour la composition de services web. L’Objet, Vol. **8**(2), (2007), 1 - 4.

A Service Oriented Architecture for Context-Aware Systems

Moeiz Miraoui, Chakib Tadj

LATIS Laboratory, Université du Québec, École de technologie supérieure
1100, rue Notre-Dame Ouest, Montréal, Québec H3C 1K3 Canada
{Moeiz.Miraoui.1@ens,ctadj@ele}.etsmtl.ca

Abstract. Devices in a pervasive system must be context-aware in order to provide services adapted to the global context. Architectures of context-aware systems should take into account many settings of a pervasive environment to be useful. Several context-aware architectures were proposed however most of them are specific to a particular application and have some limitation which reduces their usability by developers. In this paper and based on our previous work on defining context and context-awareness, we will propose a multi-agent service oriented architecture for context-aware systems. The architecture is based on the concept of service which plays an important part in the operation of a pervasive system. Our architecture is a hybrid form between client/server and peer-to-peer models. In addition of the advantages of those models, the proposed architecture enhances the extensibility, reusability, security of context-aware systems and takes into account the dynamic aspect of a pervasive environment. We will discuss in deep the contribution of the proposed architecture and its relation with other ones.

1 Introduction

In a pervasive system, the user's environment is populated with communicating smart devices. These devices provide adapted services to both the user and application. This adaptation is made according to the global context. Devices sense the global context and react proactively (without an explicit intervention of the user) to the changes of it. The goal is to help user in his everyday life's tasks. To-do so, devices in pervasive system must be context-aware thus context is a key concept in such systems. Context must be well understood to provide a better adaptation. In previous work [1] we have proposed a definition of both context and context-awareness based on the concept of service because it seems for us that this latter plays a crucial role in the adaptation task. Those definitions are more generic than others and make a good compromise between abstraction of the term and limitation of the set of required information for adaptation.

In the last few years many architectures of context-aware systems were proposed to support the development of these systems. Most of them are specific to particular applications and have limitations in many levels (management of contextual information, communication between devices, flexibility and reusability). In this paper we will present a multi-agent architecture for context-aware systems based on

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Copmuting Science 29, 2007, pp. 236-244

Received 31/08/07
Accepted 19/10/07
Final version 24/10/07

the concept of service. Our architecture takes into account the dynamic aspect of pervasive systems, more generic (applicable to a large variety of applications) and modular which enhance its reusability.

This paper is organized as follows: section II presents a review of previous architectures of context-aware systems, discuss them and then present our multi-agent architecture. We will argument both the use of such architecture (multi-agent) and the importance of the concept of service. In section III a discussion is presented to show the originality of our approach and our contribution. Finally, we will conclude this paper and present our further work.

2 Context-Aware Architectures

2.1 Previous Architectures

Most of proposed architecture of context-aware systems makes a separation between the context sensing and the using process. This allows an abstraction of low level details of sensing and enhances the extensibility and reusability of systems. Among proposed architectures, there are basically two categories depending on whether the contextual information are centralized or distributed. The first strategy consists of using a context server where will be grouped the collected information from different sensors and provided to applications on demand. Examples of this category are, CoBrA [3], SOCAM [4], etc. In the second category, contextual information is distributed between entities of the system and uses the peer-to-peer communication model for information exchange. Examples of this category are the HYDROGEN architecture [2], the context toolkit [5] and the contexteur [6]. A survey on different architectures is done by Baldauf et al. [7]. It shows that most of them are layered architectures [8, 9, 10, 11] with basically the following layers:

- Sensing layer: it enables the physical sensing of contextual information using different kind of sensing.
- Interpretation layer: it makes transformation of gross information sensed by the previous layer to significant and useful information.
- Reasoning layer: it is not included in all architectures and permits the deduction of new information from existing ones and makes a prediction of context from previous cases which add the aspect of intelligence to the system.
- Management and storing layer: makes the management of contextual information (add, retrieve, up-date, searching, etc.) and the storing in an appropriate format.
- Adaptation layer: makes service adaptation according to context.

In the following we take a look at some architecture and analyze the approaches used in each. The context toolkit [5] was proposed to support development of context-aware systems. It is a layered architecture to separate the acquisition and the representation of context from the delivery to applications. It is based on context widgets acting similarly as GUI widgets to hide the complexity of physical sensors used by applications which gives context more abstraction and provides reusable and customizable building blocks of context sensing. The layers of the architecture are: sensors, widgets, interpreters, services and discoverers. The architecture is simple to implement, offer distributed communication between devices and reusable widgets however the discovery mechanism is centralized which makes it not a perfect peer-to-peer model, it has a limited scalability when the number of components increases, the event handling system consists of creating a thread for each event which implies an important overload of the system and does not include a context reasoning engine.

HYDROGEN [2] is architecture and a software framework to support context awareness. It is a three-layered architecture to suit special needs of mobile devices: adaptor layer, management layer (context server) and application layer. The context server contains all information sensed by adaptor layer and provides application layer with needed context or other devices by using a peer-to-peer communication. The hydrogen approach considers context as all relevant information about an application's environment and describes it using an object oriented model. This approach is very simple to implement, the three-layered architecture is located on one device thus making it robust against network disconnections, takes into account limited resources of a mobile devices (memory, CPU, etc.) and overcome the problem of centralized architecture by using a peer-to-peer communication model between devices. However the adaptor layer makes both the sensing and the interpretation of context which does not offer the abstraction level needed for context in context-awareness and makes context very dependant on sensors which affect the reusability of the architecture's components also the architecture does not include a reasoning layer about context which makes it a simple database of contextual information rather than a smart device that adapt dynamically according to current context.

SOCAM [4] is a service oriented context-aware middleware architecture for the building and rapid prototyping of context-aware mobile services in an intelligent vehicle environment. It is a layered architecture with the following layers: context providers, context interpreter (context reasoning and context knowledge), service locating service, context-aware mobile services and a context database. It is based on the client/server model where the context interpreter collects contextual information from context providers (internal/external) and context database and delivers them to both context-aware mobile services and service location service. The main strength of the SOCAM architecture is its reasoning system which uses ontologies to describe context enabling then formal analysis of domain knowledge (context reasoning). It uses two classes of ontologies: domain specific and generalized and multiple reasoners can be incorporated into context interpreter to support various kinds of reasoning tasks. However the architecture is used for the development of a small specific application (intelligent vehicle) which limits its usability for a wide range of pervasive systems. Both reusability and extensibility of the architecture are not argumented by the authors in particular, the component of the architecture are specific to the application developed and can not easily used for other systems (more open than a vehicle environment and with different characteristics) also to add/remove a device an explicit update of components is needed this is added to the major problem of a centralized model (when the server falls down).

CoBrA [3] is broker-centric agent architecture to support context-aware computing in smart spaces. The broker is an autonomous agent that manages and controls the context model of a specific domain and runs on a resource-rich stationary computer. The broker has a layered architecture composed of context knowledge base, context reasoning engine, context acquisition module and privacy management module. The broker acquires context information from devices, agents (applications) and sensors located in its environment and fuse them into a coherent model which is then shared with devices and their agents. CoBrA uses ontologies to describe context thus enabling a good reasoning mechanism and a better share of contextual information, it uses a centralized design for storage and processing because mobile devices in a pervasive system have limited computing resources and introduces the user's privacy policy. However the centralized design is also its weakness, when the broker falls down all the system is affected and become useless. The fact of using a server for context in a pervasive system contradict the nature of context in such a system which is generally distributed in addition to overloading the system with the using of a server running on a stationary computer.

Several other architectures exist but they do not differ a lot from the ones listed in this paper which seem to us the most relevant.

The modeling of contextual information is a fundamental step for every adaptation system. The modeling task consists of providing an abstract representation of contextual information in both data structure level and semantic level which ease their manipulation. Many methods of modeling were proposed having particularities related to used techniques. A detailed discussion of these methods is out of the scope of this paper but Strang and al. [12] did a survey on them and distinguished basically the following models for context representation: a) Key-value models, b) Mark-up scheme models, c) Graphical models, d) Object oriented models, e) Logic based models, and f) Ontology based models.

According to the authors, the context representation based on ontology model is a promising method for the design of context management system adapted to a pervasive computing environment.

The proposed architecture is specific to a particular application domain (human-computer interaction, mobile computing, etc.). Context server based systems have the principal problem of centralized systems: if the server breaks down, the other devices of the system will be automatically affected. It needs more effort for the administration task. Also it contradicts the nature of contextual information in pervasive system which is generally distributed. The peer-to-peer architecture in particular the one proposed by Dey and al. [5] doesn't allow an efficient reasoning on context because the method used to represent context is limited (key-value model) and does not support a robust logic reasoning. The architecture proposed by Rey and al. [6] requires a strong communication between entities and does not take into account the limited resources (energy, processing, etc.) of devices in a pervasive system and the possibility of loosing connection even though it uses a distributed architecture which offers many advantages compared to the context server architecture. To summarize, there are two major aspects that need more attention and work in most proposed architectures:

- Flexibility: architectures must be more flexible and take into account the dynamic changes in a pervasive environment (add, suppression of devices).

- Reusability: architectures must offer reusable modules to ease their integration in other pervasive systems in order to decrease the effort of development.

2.2 Service based Architecture

The main objective of a pervasive system is to provide proactively adapted services to both the user and the applications. Thus the concept of service plays a crucial part for the operation of such a system. As we defined in our previous work [1] context and context-awareness based on the concept of service, we will use those definitions in order to design architecture for context-aware systems based on the same concept. The architecture is peer-to-peer and multi-agent (the use of multi-agent approach will be augmented progressively).

A pervasive system is composed of a set of communicating smart devices and provides adapted services according to the global context in various forms (various qualities of services) by using different media and modalities (figure 1). In our approach, for every device, a service will be modeled with a deterministic finite state automata whose states are the forms of services (among several forms) offered by a device. Transitions between states (from a service form to another) are due to changes in values of contextual information (since we defined context [1] as any information that trigger a service or change the quality (form) of a provided service). So we can easily limit the set of contextual information for every service and enable it to adapt accordingly.

For example calls indication for a cellular phone has several forms: ring tone, vibrator, ring tone with vibrator and silent. For simplicity, we will look to the two forms of this service: ring tone and ring tone with vibrator. Initially the cellular phone indicates calls with ring tone. It senses the level of noise of its environment; if it is high (over a fixed value) it changes automatically the form of service to ring tone with vibrator to draw the attention of the user. Figure 2 shows the finite state automata for calls indication.

In the following we will give the architecture for a service inside a device then the architecture of a device in a pervasive system and finally the global architecture. Figure 3 show that a service of a device may be provided in several forms. These forms are related with deterministic finite state automata. For each form there are a set of media and modalities depending on it.

In a pervasive system, a service has the following characteristics:

- Reactive entity: a service perceives the changes of the global context using its device’s sensors and reacts by changing its form or its release.
- Autonomous: It can be provided in various forms independently of other services and devices.

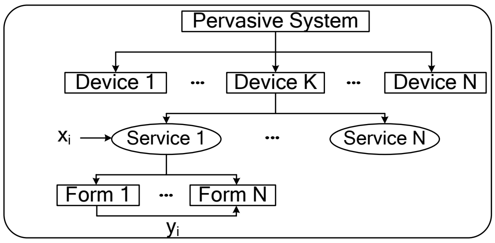


Fig. 1. A device provide several services in different forms: The change in value of x_i will trigger service1. The change in value of y_j will change the form of service.

- Proactive: It can be equipped with mechanisms enabling it to take initiatives to react (to predict the future form of a service).
- Sociability: it can exchange contextual information with other services to make aggregation (fusion) of context or to acquire useful information which is not provided by its main device.

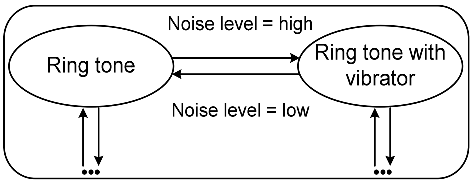


Fig. 2. A part of the finite state automata for calls indication.

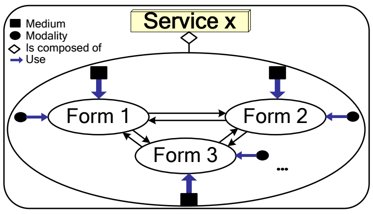


Fig. 3. Example of a service having three forms.

These are the principal characteristics of a software agent. For this reason, each service of a device will be modeled by an agent with internal states. These agents will be controlled and managed by a central agent (device agent) who plays the part of a context server for other agents (service agents). Each agent must be registered for a set of contextual information that concerns it. The service agent will be notified by the central agent each time the value of one of these information changes (to limit information flow). Also the communication with other service agents of the same device will be made only via the central agent (figure 4). In the case of a simple sensor, it will be modeled by a central agent without service agents since it provides one type of services (collect information).

The central agent of a device perceives the changes of the values of contextual information via the device's sensors or by communication with the other central agents of the other devices composing the pervasive system. If a change of contextual information concerns one of its service agents, then it communicates to it the change of value. With this intention, each central agent holds a list of contextual information which concerns each one of its service agents. With the reception of this information, the concerned service agent reacts either by a transition in its finite state automata to change the form of service or by triggering the service it self (initial form).

The central agent has layered architecture containing in addition of basic layers (sensors, interpretation, reasoning, management/storing and adaptation) two modules of communication: internal module for exchange of information with the device service agents and an external module for the exchange of information with the other central agents of the pervasive system. If a central agent perceives or senses the change of value of a contextual information (through the sensors of the device) and which concerns or not one of its service agents it broadcasts the new value to all central agents to enable other agents to react and will be notified by concerned central agents. This will allow the broadcaster central agent to: a) remember next time the central agents concerned with that information and b) up-date the list of active central agents in the pervasive system. For that reason we envisage to add for each central agent a cache memory for this type of transactions with an emptying system FIFO (first in first out) when the memory is full. Also to use an internal cache memory for internal transactions with the service agents (figure 5). The change of value of the same contextual information can be broadcasted by several central agents. This enables it to apply a fusion process for better control of errors.

Inside a device the central agent is a server to the service agents of the device (client/server model) and in the whole pervasive system, it can be either a server or a client for the other central agents (peer-to-peer model). The agent service has just two tasks: a) trigger a service and b) change the form of a service (make a transition in his automata) all the others tasks are made by the central agent (server).

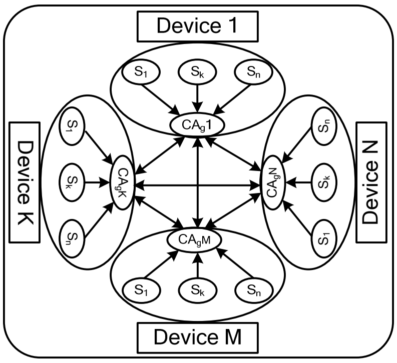


Fig. 4. The global multi-agents architecture. **S_i**: Service agent i, **CA_g i**: Central Agent i, ↔: Communication.

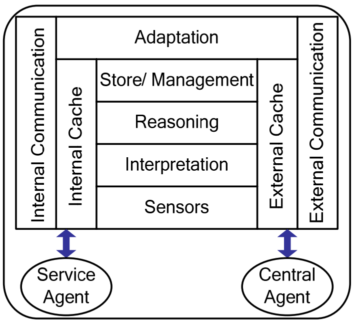


Fig. 5. Central agent architecture.

3 Discussion

The architecture that we proposed is hybrid architecture: client/server in the level of the multi-agent system (central agent is a server and service agents are clients) of each device and a peer-to-peer in the level of multi-agent system of the set of devices that compose the pervasive system (central agents). This makes possible to benefit from the advantages of the two architectures. Peer-to-peer architecture allows distributing contextual information and the processing relative to this information between the entities (devices) of the system. This reduces the load of processing and decrease the information flow between them. Client/server architecture allows an effective control of concurrent and multiple accesses to contextual information (central agent) and discharges the clients (service agents) from the operations of storage and information management.

In the multi-agent peer-to-peer architecture for the central agents, if a device of the pervasive system breaks down, its central agent will be inaccessible (inactive), but it does not affect the operation of the system. Only the services provided by that device become unavailable. A replacement by services provided by others devices is possible.

In the same context, it is possible to add new devices to the system without disturbing it since it will detect the new device automatically if it exchanges information with one of the components of the system (extensible dynamic architecture).

This architecture is reusable for the following reasons: the multi-agent system of a device (central agent and service agents) can be used in another pervasive system with minor modifications because the services offered by a device are the same ones (example: a GPS device always provides the geographical coordinates whatever its use).

The encapsulation of service details of a device as well as the contextual information management model within the device reinforce the protection of the device against undesirable accesses which can damage its quality of services (the access is done via an interface: central agent).

In this architecture, we did not consider the user's context because it's implicitly included in the device's context (example: the localization of the user is detected by a GPS that is with the procession of the user).

4 Conclusion and Further Work

The main characteristic of devices in a pervasive system is their context-awareness in order to provide adapted services. The design of architecture for such systems is a basic task for their development and implementation. Most of proposed architectures do not provide a lot of help to developers and have some limitations. In the light of existing architectures of context-aware systems, and based on the concept of service, we have proposed a multi-agent architecture. We discussed the characteristics and strong points of such architecture that can be summarized in the following points: hybrid architecture between client/server and peer-to-peer which takes advantages from both architectures. Extensible with few modifications thus dynamic (add and removal of devices). Reusable since the services of each device remain almost unchanged whatever the enclosed system thus it is possible to use it with its multi-agent system in another application. Easy to implement by using ready components of multi-agent domain in particular the mechanism of communication.

Our outlined architecture requires completion on the level of communications inter-devices and intra-devices by using the existing means of communication in multi-agent systems as well as the context representation model. We plan to use ontology which seems to us suitable for such architecture. Finally we will work on the reasoning layer of the central agent to make it more intelligent.

References

1. M. Miraoui, C. Tadj, "A service oriented definition of context for pervasive computing," in Proceedings of the 16th International Conference on Computing, IEEE computer society press, Mexico city, Mexico, November 2007.
2. T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger & J. Altmann, "Context- awareness on mobile devices – the hydrogen approach". In Proceedings of the 36th Annual Hawaii International Conference on System Sciences, pages 292–302. 2002
3. H. Chen, T. Finin, A. Joshi, "An ontology for context-aware pervasive computing environments, " Knowledge Engineering Review, vol. 18, pp. 197-207, 2003
4. T. Gu, H. K. Pung, D. Q. Zhang, " A middleware for building context-aware mobile services", in the proceedings of IEEE Vehicular Technology Conference (VTC 2004). Milan, Italy
5. A. K. Dey, G. D. Abowd, " A Conceptual framework and a toolkit for supporting rapid prototyping of context-aware applications", anchor article of a special issue on Context-Aware ComputingHuman-Computer Interaction (HCI) Journal, Vol. 16 (2-4), 2001, pp. 97-166.
6. G. Rey, J. Coutaz, " Le contexteur : capture et distribution dynamique d'information contextuelle ", ACM transaction pp. 131-138 Mobilité & ubiquité, 2004
7. M. baldauf, S. Dustdar, F. Rosenberg, "A survey on context-aware systems" International Journal of Ad Hoc and Ubiquitous Computing forthcoming, 2004
8. P. Korpipää, J. Mäntyjärvi, J. Kela, H. Keränen, E. Malm. "Managing Context Information in Mobile Devices". IEEE Pervasive Computing. 2003
9. P. Fahy, S. Clarke. "CASS – Middleware for Mobile Context-Aware Applications". MobiSys 2004
10. G. Biegel, V. Cahill. "A Framework for Developing Mobile, Context-aware Applications". In Proceedings of 2nd IEEE conference on Pervasive computing and Communications, Percom 2004
11. M. Román, C. Hess, R. Cerqueira, A. Ranganat, R. H. Campbell, K. Nahrstedt. "Gaia: A Middleware Infrastructure to Enable Active Spaces". In IEEE Pervasive Computing, Oct-Dec 2002.
12. T. Strang, C. Linnhoff-Popien, " A context modeling survey" In the first International Workshop on Advanced context modeling Reasoning and management. UbiComp 2004

Multi-Agent System for the Distribution and Location of Learning Objects

Lluvia E. Ponce[1], Karla Torres[1], Juan C. Ramírez[1], Fabiola López[2] ,
Darnes Vilariño[1]

[1] Facultad de Ciencias de la Computación, BUAP
Puebla, México
[2] Dirección de Modalidades Alternativas de Educación, BUAP
Puebla, México

pomell81@gmail.com, karlatorresmx@hotmail.com, jcramirezmx@gmail.com,
darnes@cs.buap.mx, fabiola.lopez@siu.buap.mx

Abstract. The Learning Objects Paradigm (LO) is currently being used for the design of online courses with certain success. Nevertheless, the technologies that currently allow the sharing are not completely developed. In this paper, a system for the management of a distributed repository of Learning Objects is presented. This system considers the market as a Multi-Agent System –and it is named "Learning Objects Market (LOM)". In this market, Provider Agents enter to publish their LO; Consumer Agents request the LO that they need; and an Administrative Agent controls and monitors all of the transactions that occur inside the market. These actions guarantee the accomplishment of policies for login and permanence of the agents on the market. The interactions between all of the agents inside or outside of the LOM are presented, because depending on the type of service requested, the administrative agent throws a group of agents with particular tasks. An strategy and a negotiation protocol has been developed for both provider and consumer agents, on the process of buying and selling a LO. The results obtained with the first implementation of the LOM are also presented.

Keywords: Agents, Market, Learning Objects, Negotiation process.

1 Introduction

The quick advances on technology, the competitivity in the labour market and the need to be more prepared in a globalized world where the knowledge must be acquired with the least possible delay, are some causes for using new learning methodologies, in order to satisfy the demands for increasing the performance of a person in the process of teaching and learning.

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Copmuting Science 29, 2007, pp. 247-256

Received 29/06/07
Accepted 19/10/07
Final version 24/10/07

The education leaves behind the traditional methods for teaching and starts with new learning systems. This creates new concepts, such as distance-learning, e-Learning, Virtual Learning Environments, etcetera. These new concepts generate new techniques for knowledge sharing, which shall be applied by professors and researchers around the world. They generate thousands of courses that are combined to produce content structures. These courses are created on web pages or e-Learning platforms, and are produced as monolithic. In other words, their content is not reusable once a course is divided; they lack of any interoperability because they are not multiplatform, and the production of these courses is slow and inefficient.

Most of the solutions to these problems is found on the Learning Objects Paradigm. A Learning Object (LO) is defined as "a digital or non-digital entity supported by technology which can be used, reused or referenced during the learning process" [1]. In this order, the courses are formed by LO and given their characteristics, the LO can be reused to generate new courses.

The use of the LO to generate courses introduce some problems such as: How to create them? How to store them? How to find them? How to use them? How to upgrade them?

The current paper is centered on the solution of two of these problems: the storage and the location. The development of an informatic system is proposed, which can publish, locate and control these objects. The system has been organized in a model of a Learning Objects Market (LOM).

The LOM has been implemented by following the Agent-Oriented Paradigm, and it is supported by the Services-Oriented Architecture (SOA) proposed by De Roure [2]. The basic components of a SOA are the following:

Provider Agents, which represent the interests of the LO developer users. They interact with the market to put on disposal several LO for Consumer Agents. The providers will keep certain rights and policies for access to the LO.

Consumer Agents, which represent the interest of course developer users. They dispose of tools to easily locate, access and negotiate use conditions for LO.

An Administrative Agent which regulates all the transactions made inside the market.

Before a LO is given from a Provider Agent to a Consumer Agent, they both must negotiate the conditions of use for the LO. Some of these conditions are price and permissions such as use, distribution and lending. This is done on a so-denominated Negotiation Stage.

The final agreements must be legalized in a contract that must be signed by both provider and consumer.

The paper is organized as follows: Section 2 describes the components of the LOM as a Multi-Agent System and the interactions of each agent inside the system. Section 3 contains the protocols and negotiation strategies that are used by the agents. Section 4 presents the communication language that allows to the agents to perform their tasks. Section 5 describes the JADE platform, which is used as the implementation tool for the agent behaviors. Finally, Section 6 gives the obtained results through the development of the system.

2 LOM Components as a Multi-Agent System

A general model of the market has been developed. This model shows the components of the SOA that have been implemented by following the Agent-Oriented Paradigm. To solve the previously presented problems, a Multi-Agent System has been proposed. It is composed by the following entities: Provider Agents (PA), Consumer Agents (CA) and an Administrative Agent (AA).

With the purpose of attending simultaneously the queries made by the different PAs and CAs, the AA tasks have been divided in a group of subtasks. Each of the subtasks is performed by independent agents -such as register agents, login handler agents, LO Localizer Agent and Contract Register Agent. These agents are conveniently created by the AA depending on the service requested by any entity of the market. It is important to remark that, based on the proposed model, there are several agents that 'reside' inside the market, and others that are external to the market. Both kinds of agents are listed next.

Resident Agents of the LOM:

Administrative Agent (AA). This agent creates the LOM, creates the agents involved on the MAS, manages the LOM, delivers LO to the CA and waits for queries sent by the Mini-Reporter Agent to create CA inside the LOM.

Interface Agent for the AA (IAAA). This agent waits for actions made by users, and access remotely to the Market Data Base to send response to the user queries.

Consumer Agent. This agent registers on the LOM, locates LO, negotiates about the use of determined LO and logs out of the LOM.

Register Agent (RA). This agent registers to both PAs and CAs to enter and exit the LOM.

Login Handler Agent (LHA). This agent publishes LO, retires LO, and allows to PAs and CAs to drop out of the market.

LO Localizer Agent (LOLA). Locates to the LO Provider Agents.

Contract Register Agent (CRA). Delivers a contract for the use of a LO.

External Agents to the LOM:

Interface Agent for the CA (IACA). This agent waits for actions of the consumers, shows agreed contracts and policies for login and permanence inside the market.

Mini-Reporter Agent (MRA). This agent communicates with the market, by representing a consumer.

Provider Agent. This agent registers in the LOM, publishes LO, negotiates contracts for use of a LO and delivers LO.

Interface Agent for the PA (IAPA). This agent waits for actions of the providers, shows agreed contracts and policies for login and permanence inside the LOM, and keeps the key to access to the LO.
The proponed Multi-Agent System is shown on the Fig. 1.

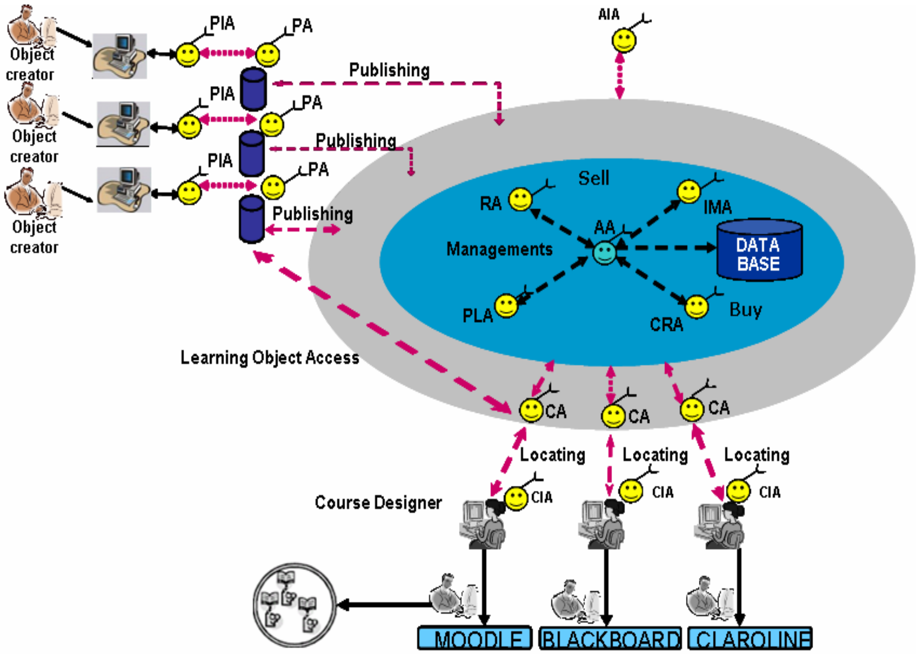


Fig. 1. General Model of the Learning Objects Market

2.1 Interactions of the Multi-Agent Systems

In order to obtain a good performance of the LOM an interaction process between the agents should exist. Due to the tasks performed by each agent are not isolated, in other words, the execution of a task implies the finishing of a previous task. The interaction process is described next.

The user that creates a LO is represented by a PA that performs the tasks described on the previous section. The developer puts on disposal of the customers their objects through these agents. Both the PA and the user that creates LO interact with the IAPA to maintain communication. The PA and the IAPA do not form the market; generally they are located on a remote node. Any action or request made by the user that creates LO must be delivered to the PA through the IAPA, and the PA will then communicate with the market and eventually, the PA will communicate with the IAPA to deliver to the user the obtained results of these actions and queries.

The consumer user of LO is represented by a CA that is in charge of obtaining the LO requested by the user to create his own courses. Both the CA and the consumer user interact between them via a IACA. The CA resides inside the LOM while the IACA is located on the computer of the consumer user, so the communication between them is made remotely. Any action or query performed by the consumer user is delivered to the CA via the IACA, and the CA will communicate directly to the market. Eventually, the CA will communicate with the IACA to deliver to the user the results obtained of these actions or queries.

The objects that are given or published by the PAs are stored on a global repository. The control and management of the LO is in charge of the AA.

The AA is always in execution state. Due this agent is in charge of receiving requests from the PAs and CAs, it verifies the kind of request and creates the corresponding agent to the performed task, it may be a RA, RHA, LOLA or CRA, to attend the query.

Regarding to the social structure of the market, it may be possible that the PAs and CAs have different interests about how to obtain or offer a LO. So before a LO may be delivered, a process of negotiation about the use of the LO must be initialized. The process will conclude with the sign of a contract that includes the terms and conditions of use of the LO. The duties, benefits and penalties in case of breach of contract are also established in this contract. The PAs and CAs will be provided with capacities of dynamic negotiation. The core of these agents is described next, in other words, the communication language, strategies and protocols that are used on the Negotiation Stage.

3 Negotiation

On Garima's research [3], a model of the negotiation agent is presented. This agent is able to operate in different environments and to adapt dynamically to them, due to the architectural division of the agent in modules. In the current paper, the model proposed by Garinma is used as a base for the design of an agent able to accomplish dynamic negotiation.

Conceptual model of the agent. An agent is composed by an static core and interchangeable components. Each agent is composed of three basic modules (see Fig. 2).

Communication Module. This is responsible of the communication between agents. This is a static module that supports the ACL FIPA language.

Protocol Module. It contains general rules for the negotiation. When an agent starts the negotiation, based on the negotiation table, it decides which negotiation protocol can be used, and dynamically it loads the correct module.

Strategy Module. It contains policies, a set of goals, actions and action rules. The chosen strategy depends on the success rate and the selected communication protocol. If there is not available information for these conditions, a default strategy model is uploaded to the agent.

As it is presented on the Fig. 2, an agent is capable to dynamically choose any appropriated combination for Protocol and Negotiation Strategy for a particular negotiation case. The result of applying these combinations may be stored for making future decisions. Then, in the moment of choosing a combination Protocol-Strategy that is considered to bring a successful negotiation, the agent may use the acquired experience, or apply a different combination that has never been applied or with a low success ratio. This is done to avoid a predictable performance that may be applied against it by other agents.

The following Protocols and Strategies are proposed to be included in the agent programming:

Protocol Module: Monotonic Awarding Protocol (MAP) and Rubinstein's Alternate Offers Protocol (RAOP).

MAP. The MAP works following these steps:

1. The negotiation goes in rounds.
2. On the first round, the agents simultaneously propose a deal of the group of negotiations.
3. The agreement is reached once the deal of an agent is as good as its proposal.
4. If the agreement is not reached, then another round of simultaneous proposals proceeds.
5. Any agent must propose a deal that is worst than a previously proposed deal.
6. If any agent makes an award in any round, then the negotiation ends with a conflictive agreement.

Using the MAP, the negotiation is guaranteed to end -with or without agreements- after a finite number of rounds.

RAOP. There are N agents, $Agents = \{A_1,...A_n\}$. The agents need to reach an agreement about a given situation. It is assumed that the agent may take actions on the negotiation only in certain moments of possible periods of agreement $T = \{0,1,2,...\}$. These periods are determined and known in advance by the agents. On every period $t \in T$ of the negotiation, if it has not previously finished, the agent makes its offer in the time t. In this moment, there may occur an agreement –respect to the situation specified on the negotiation. Each of the agents may accept the offer (Yes), reject it (No) or leave the negotiation (Out). If the offer is accepted by all of the agents, the negotiation ends and the offers are implemented. If at least one of the agents decides to get out of the negotiation, it ends in conflict. If all of the agents stay in the negotiation, but at least one rejects the offer, the negotiation continues on the period t+1. The rejecting agent makes a counter-offer, the rejected agent responses, and the negotiation continues.

Strategy Module. In the negotiation context, the LOM has two agents: the PA that wants to sell its LO to the best possible price, and the CA that wants to acquire a LO to the best possible price.

The intervals that the price of the LO may take have been defined. The negotiable parameters considered by the PA are the following:

1. Desired price. It is the amount that the LO creator wants to obtain.
2. Minimal acceptable price. It is the lowest price that the LO creator will sell its LO.

Because of these parameters, the PA has to respect the following parameters for the price variation, defined on the range:

$$[p\_minp, p\_desp] \qquad (1)$$

The negotiable parameters considered by the CA are the following:

1. Desired price. It is the price that the consumer is disposed to pay for the LO.
2. Maximum acceptable price. It is the highest price that the agent is disposed to pay for the LO.

Due to these parameters, the CA has to respect the following parameters for the price variations, defined on the range:

$$[p\_desc, p\_maxc] \qquad (\mathbf{2})$$

Three strategies for negotiation have been designed and implemented. Each proposal made by the PA or the CA -by following the previously described protocols- is found on the defined ranges, respectively for the kind of agent.
The proposed strategies to be included in the concerning module are the following:

Strategy of Constant Increases-Decreases.
Strategy of Increases-Decreases by a Parabolic-Function.
Strategy of Priority Negotiations by Increases-Decreases.

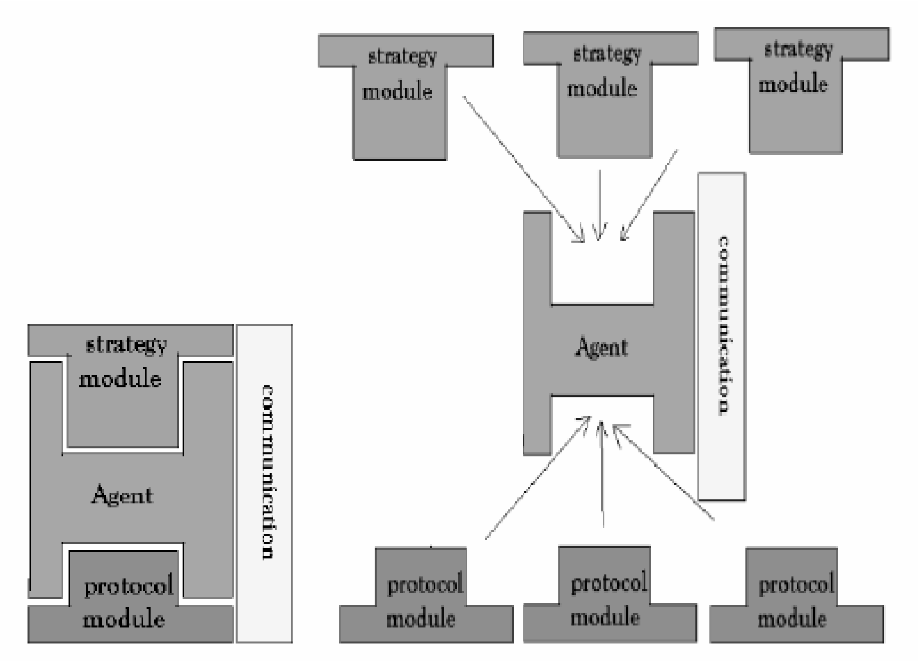


Fig. 2. General Model of the Learning Objects Market

4 Communication between Agents

In order to perform the negotiation and the tasks of the agents involved in the LOM, a comprehensive communication language between them is needed. An agent may reject a query of another agent, so they must be capable to speak between them, do decide which actions to perform or which data to obtain.

ACL has been chosen as the communication language between the agents.

An ACL message is a communicative act that an agent delivers so the receiver performs certain action.

The classical syntax is a LISP type, with several parameters such as the following.

1. About the communication. They contain the sender and receiver and intermediate entities.
2. About the content. Such as the language, ontology, etcetera.
3. About the intention, such as what message is replying, which is the following protocol, etcetera.

This can be expressed in XML or any other descriptive language.

ACL has a group of communicative acts. On the following Table, the strategies performed for negotiation and communication between the agents and the LOM are presented.

Table 1. Communicative Actions with ACL.

Communicative Act	Meaning
Cfp	Request a proposal.
Refuse	Reject a query.
Not-understood	The message is not understand.
Propose	Make a query.
Reject-proposal	Reject a proposal.
Accept-proposal	Accept a proposal.
Inform	Give information to the receiver.
Failure	Explains why the action failed.
Request	Asks to perform an action.
Agree	Accepts query.
Cancel	Cancels query.
Query-if	Asks something to the receiver.
Subscribe	Subscribe to a service of a receiver.

5 Implementation of the Multi-Agent System

The MAS has been developed in the JADE platform ((Java Agent DEvelopment Framework). This platform is implemented in Java. Jade simplifies the implementation of Multi-Agent Systems through a middleware that follows the FIPA specifications. It also contains a set of graphic tools which help on the debugging and deployment phases. The platform of the agent can be distributed into different hosts, they do not share the same operative system and the configuration can be controlled by a remote graphic interface.

5.1 Agents Behavior in JADE

A behavior represents a task that an agent can achieve. The behaviors are classified in three groups:

1. One-shot behavior: Behaviors that complete immediately and whose action() method is executed only once.
2. Cyclic behavior: Behaviors that never complete and whose action() method executes the same operations each time it is called.
3. Generic: Behaviors that embed a status and execute different operations depending on that status. They are completed when a given condition is met.

It is necessary that the implementation of one of these behaviors inherits from the jade.core.behaviours.Behaviour class, to get an agent to achieve an implemented task for a behavior object. It is enough to add the behavior to the agent through the addBehaviour() method of the Agent class. Behaviors can be added at any time, when an agent starts –setup() method, or in another behaviors [4]. Any class that inherits of Behaviour must implement the action() method where are described all the types of behaviors that an agent use to execute each one of its tasks.

Each agent of the LOM has its own tasks. They are implemented based on some of the previous behaviors that were mentioned above. As an example, the implemented behaviors of the administrative agent and the types of behaviors that the agent uses to achieve each one of its tasks, are described next.

1. AnalizaMensajesBehaviour: These behavior responses queries of PA or CA. It is a cyclic behavior because a query can arrive there at any time.
2. AnalizaRequestBehaviour: These behavior responses queries of the internal agents of the market. The behavior is similar to the latter.
3. CrearConsumidoresBehaviour: This behavior creates CAs in the market. They received the name of MRA. The behavior is one-shot type because for each consumer user there is a CA.

In the similar way the other behaviors of agents have been implemented.

6 Conclusions and Recommendations

The assigned tasks of each agent that is in the MOA have been fully implemented. To reach our objective, the communication protocols and all of the messages that are used in the interaction process have been defined, with the main purpose to achieve the goals of the agents.

According to the Negotiation phase, the MAP and the RAOP have been implemented. The latter one, that considers the strategies of constant increments and decrements makes these movements with a quadratic function.

Currently the first version of the LOM has been implemented in JADE. This middleware has shown an excellent performance. The Multi-Agent System can be consulted in the following web page: www.dgmae.buap.mx/invdes/mercadoOA.

The developed agents senses the environment, they take the information and then send and receive messages according to the external data. This task implies the implementation of new behaviors. The authentication process achieved in the LOM is the key access validation; an agent receives a key once it is registered in the market, and however it is not enough to maintain the security in the process of MOA. Now we are working in authentication of PA and CA, another task to implement is the process of sending and receiving messages in a secure way to avoid the losses of information.

Acknowledgements. Research financed by the Trust SEP-UNAM, Project 159/2006.

References

1. IEEE 1484.12.1, Learning Technology Standards Committee, Learning Object Metadata Standard.
2. D. De Roure, N. Jennings, N.R. Shadbolt., The Semantic Grid: A future e-Science Infraestructure.hnical report UKeS-2002-02.
3. P. Marcin., A. Abraham1, Agents Capable of Dynamic Negotiations. Computer Science Department Oklahoma State University Tulsa, OK, 74106, USA.
4. G. Caire, JADEProgramming-Tutorial-for-beginners, JADE Documentation, 2003, p. 8.

Policies for a Regulated and Distributed Repository of Learning Objects

María Erika Olivos Contreras[1], Fabiola López y López[2], Darnes Vilariño Ayala[1]

[1] Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Blvd. 14 sur y Av. San Claudio Edificio135, Puebla, México.
[2] Dirección de Modalidades Alternativas de Educación, Benemérita Universidad Autónoma Puebla

merikaoc@hotmail.com, darnes@cs.buap.mx, fabiola.lopez@siu.buap.mx

Abstract. Designing on-line courses is an expensive process, because to do that, the collaborative work of several experts is needed. To reuse on-line educational material, the learning object paradigm has been used in the last years. Learning Objects (LO) either can be stored in a central data base or they can be distributed along a network; the problem with public distribution and access is that the quality, the standardization and the updating of LO cannot be guaranteed. For this reason, a regulated and distributed repository of LO has been developed. Since such a repository works in a similar way than an electronic market, we have called it Learning Object Market. The market is conceived as a multi-agent system and to obtain good results in any transactions on it, several policies have been defined. The aims of this paper are first, to describe the different types of policies needed in a LOM and, second, to present an ontology to specify them.

Keywords: Agents, Market, Learning Objects, Polices, Contracts, Ontology.

1 Introduction

Designing on-line courses is an expensive process, due to the need of a collaborative work between several experts. To reuse on-line educative material, the learning object paradigm has been created and promoted in the last years. Learning Objects (LO) are electronic educational material designed in such a way that the student can easily obtain the contents of a particular topic. Generally, LO are made of multimedia material that can be logged in from a web page or from a learning management system.

A LO can be stored in a central data base or they can be distributed along a network; moreover, their use can be private or public. The problem with public distribution and access of LO is that their quality, standardization and updating cannot be guaranteed. For this reason, a regulated and distributed repository of LO has been developed. Since such a repository works in a similar way than an electronic market, we have called it Learning Object Market (LOM).

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Copmuting Science 29, 2007, pp. 257-267

Received 29/06/07
Accepted 19/10/07
Final version 24/10/07

The market was conceived as a multi-agent system where agents can play one of three different roles: producers, consumers or managers.

In order to obtain good results in any transactions in a market, several kinds of policies are needed. Furthermore, since the market is modeled as a multi-agent system, a way to represent policies that can be understood by agents is required. The work presented in this paper represents a first approach to do that. On the other hand it is necessary to follow the signed contracts inside the LOM, once the parts have reached an agreement about a determined LO. So, the main objective of the paper is to specify through an ontology, a set of policies that can be applied in a Learning Object Market and the formalized transaction contracts. The paper is organized as follows. First, a general description of the LOM is provided. Next section shows a typology of the policies that are needed by a market to work, and the contract that is handled in the LOM is defined. After that, the ontology is shown. Finally, our conclusions and future work are discussed.

2 The Model of Learning Object Market

An electronic market can be considered as a virtual space where some encounters take place. There are providers of goods, and they use the market to shown their products and to find buyers of them. There are also buyers that, based on a need, go to the market to find and select the goods that satisfy their requirements. Once an encounter is made, a contract over the agreements can be signed by both providers and consumers. In an electronic market, there are also managers whose function is to monitor any transaction done in the market and, in the case of conflict, to apply punishment to any transgressor of an agreement.

In our case, we have used such metaphor to develop a virtual place to publish, to locate and to make deals in order to reuse learning objects. The reasons to prefer this metaphor over another (i.e. a distributed database) can be explained as follows. First, learning objects could be developed by experts that do not necessarily know in advance, the users of such objects. Second, the learning objects offered in the market must comply with certain requirements that can guarantee their quality and standardization. Third, the users of learning objects could have wider options from which an appropriated object can be selected. Fourth, once a consumer finds a learning object that satisfies the needs, it can start a negotiation process to make agreements over the rights to use the object, the time to use it or the frequency to update the content of the object. To formalize such agreements a contract must be signed. Fifth, since agents in this context may not know each other, it is necessary a third part that regulate both the members that can participate and the deals that are made in the market.

The proposed Multi-Agent System is formed by the following instances: Administrative Agent (AA), Interface Agent for the Administrative Agent (IAAA), Consumer Agent (CA), Register Agent (RA), Input Handler Agent (IHA), LO Localizer Agent (LOLA),Contract Register Agent (CRA).

These agents inhabit the market. But there are other agents that reside in the computers of the providers and consumers, such as: Interface Agent for the Consumer

Agent (IACA); Mini-Agent Reporter (MAR) invisible to the user; Provider Agent (PA); Provider Interface Agent (PIA).

This market is managed by an administrative user, which is responsible of verifying the compliance of the applied policies for login and permanence in the market. The market is represented by agents. Then the administrative user, the service providers and the service consumers are represented by an Administrative Agent (AA), provider agents (PA) and consumer agents (CA), respectively. The design of the LOM considers several internal agents that are launched by the AA to the market depending on the service requested there. To guarantee a good behavior of the agents in the LOM, it is necessary to specify policies for login, permanence and dropout of every agent in the market. Also a contract must be established between the negotiating parts, which includes terms and conditions of use of the requested LO. This latter part is the essence of this research.

The objectives and tasks that each one of these agents must perform have been previously defined. For the best performance of the market, all of the access, policies for register and permanence of the agents have been defined, so as the contract that must be signed by both the consumers and provider agents about the use and handling of a determined Learning Object.

3 Types of Policies for a LOM

A policy is applied in practice to rule and solve conflicts. It can create coherence and obligatory rules to everyone "living in the same area". Its purpose is to unify the interpretations about repetitive and concrete aspects [15].

It is important to mention that to define the LOM policies, three different electronic markets already functioning have been analyzed, and these are eBay [9], MercadoLibre [9] and Amazon [9].

According to the interactions that must be done in a market we identify 6 different types of policies:

1. To register new members.
2. To retire members that does no comply with the regulations of the market.
3. To publish a LO.
4. To evaluate a member of the LOM.
5. Contract following.
6. Enter an agent to the LOM.

For the purpose of this work we define policies as a set of norms that must be applied when certain kinds of interactions between agents in a LOM must be done. To do so, we based our work in the normative agent-based framework developed by López, Luck and d'Inverno [3]. In such framework, a normative agent is defined as an autonomous agent which behavior is determined by the duties that must be satisfied, the prohibitions that limit the pursued goals and the social commitments that are created during social interactions.

Moreover, norms are defined as the mechanisms which are used by society to adecquate the behavior of the agents that inhabit inside it. Its structure includes the following components.

Normative Goal. What shall be done.
Receiver. The entity which executes the goal.
Beneficiary. The entity for whom the goal is executed.
Context. The moment in which a norm shall be executed.
Exception state. The entities or situations when a goal shall not be obliged.
Rewards. If the norm is accomplished, what is obtained by the entity.
Punishments. If the norm is not accomplished, what is obtained by the entity.

Nowadays, many organizations offer their services on Internet. We can treat the involved entities as service producers and service consumers. It is necessary that these organizations develop their own frameworks where there can be negotiations between these producers and consumers, because certain restrictions must be applied – established by the organization itself- which are known as service policies. These policies must include at least the following:

- A form to verify the identity of the user (Authentication).
- Rules to determine what, when and where can be done inside the framework (Authorization).
- Rules that determine the penalties applied to any failing entity, for breach of contract or any abnormality registered during the compliance of a contract.

Based on de Roure [1], the lent of a service has a life cycle that consists on three phases:

- The creation of the service. In this step, the owner of the service puts it in disposal of any user. This system is considered as 'dynamic' because the providers are always creating and destroying services.
- Proxy of a service. This step includes a service provider and a service consumer. Here a contract is established in order to legalize the agreed services and the terms
- The use and disposal of the service.

Policies to register new members Provider and Consumer Agents (PRPCA)

- To become a member of the LOM, a provider or consumer agent must represent a professor of any of the following institutions: BUAP, CENIDET, UAEH and UNAM.
- The Provider/Consumer agent is responsible of its own access password for the LOM. It is much obliged to give the password to access to the market.
- The Provider/Consumer agent must be located on a server with a valid URL.

-The Provider/Consumer agent will be notified of any policy change on the LOM.

Now, according to our definition, policies 1, 2 and 3, which determine the behavior of an agent by considering the previous considered policy, can be expressed by using norms as follows.

Norm 1. N1P1RPCA.

Normative Goal. General data sent to the LOM.
Receiver. Provider or Customer Agent.
Beneficiary. Manager Agent.
Context. On the time when the agent sends its general information to register.

Exception state. Empty.
Rewards. The agent is registered to the LOM.
Punishments. The agent is not registered in the LOM.

Norm 2. N2P1RPCA.
Normative Goal. Knowledge areas sent to the LOM.
Receiver. Provider or Customer Agent.
Beneficiary. Manager Agent.
Context. On the time when the agent registers.
Exception state. Empty.
Rewards. The agent is registered to the LOM.
Punishments. The agent is not registered in the LOM.

Norm 3. N3P1RPCA.
Normative Goal. Access code of the user sent to the LOM.
Receiver. Provider or Customer Agent.
Beneficiary. Manager Agent.
Context. On the time when the agent fills correctly its register.
Exception state. Invalid access code.
Rewards. The agent enters to the LOM.
Punishments. The agent does not enter in the LOM.
From these, the following politics were defined.

Policies to publish a new LO (applied only to providers) (PPLOS). The purpose of these policies is to guarantee the requirements so the Producer Agent may publish a LO in the Market.

- Any LO must satisfy the quality and standardization requirements established by SCORM or LOM.
- The LO must include an academic guarantee.
- If both Policies 1 and 2 are satisfied, the LO will be published.

Policies to Evaluate both Providers and Consumers (PEPC). The intentions of these policies are to evaluate the behavior of the agents inside the LOM.

- For each agent, a reputation file will be created, referring its behavior in the LOM.
- The institutions that are part of the market will monthly receive a report of the behavior of their agents.

Policies of Contract Following for Providers and Consumers (PCFPC)

- Any contract will be registered in the LOM.
- The Manager Agent will monitor the compliance of the norms that will form part of the contract.
- The Provider/Consumer agent must agree with the conditions of use and terms of the LOM.

Policies for Incoming Providers (PIP)

- To enter to the LOM the PA must be registered.
- The PA can enter to the LOM only when publishing a LO, agreeing with the publishing policies.
- The PA can enter to the LOM only to delete a LO.
- The PA can enter to the LOM only to delete and to submit a new LO.

- The PA can enter to the LOM to unsubscribe itself from the LOM.

Policies for Incoming Customers (PIC)

- The CA must be registered on the LOM to get access to it, by agreeing with the register policies.
- The CA can enter to the LOM only to request the list of LO providers.
- The PA can enter to the LOM only to unsubscribe itself from the LOM.

To establish the process of negotiation for the LO, the following scheme for a contract is proposed. This contract will eventually be signed by both the PA and the CA involved on it.

4 Contracts in a LOM

Another important component to make any deal in our LOM is a contract. A contract can be defined as a private agreement that can be oral or written, between parts that obligate and require the accomplishment of a good or service [14].

The general structure of a contract includes general and specific clauses. The general clauses are those written previously by a single person or entity in a general and abstract way, in order to fix the normative content of future contracts with their own elements. These clauses include Definitions, Object, Contract length, Price and Payment method, Communication, Confidentiality, and Contract resolution. The specific clauses are those which intend to satisfy certain conditions of high interests for most of the members which can operate inside the market. It includes: Description of the presented contents, Exclusivity, Outsourcing and Intellectual propierty.

Some general clauses include the agent ID, sheet, keyword, contract date, LO delivery date and payment for the LO. For the specific case of this work, we will use the following template for all contracts.

FIRST CLAUSE. Object of the contract. The CONSUMER AGENT (CA) is responsible of making the agreed payment on the negotiation for the LO. On the accorded date, the SUPPLIER AGENT (PA) will deliver the LO on a period of ___working days, as agreed on the same negotiation. The LO should not be modified on this period from the contract date until the delivery date.

SECOND CLAUSE. Value of the contract. The CA will pay to the PA the amount of ___for the total value of the LO, as agreed on the first clause of this contract. This amount does not include the Added Value Tax (IVA in Mexico), which represents the 21% of the contract value.

THIRD CLAUSE. Payment mode. The CA must pay to the PA the agreed price on the following way. A first payment of ___ on the moment of signing the current contract, and the rest of the payment ___ once the PA accomplishes this contract. The PA can cancel this contract in case of breach of the payment. If the CA is interested again on purchasing the LO, a new amount of ___should be paid by the CA. The PA can review the cancelled value and other offers by other agents in order to choose the best market option to renew the LO sale. The renewal of this contract should be made a year later from the sign of this contract.

FOURTH CLAUSE. Penalty for breach of payment of the contracted LO. The CA must pay to the PA an amount calculated as the 0.5% of the total contract value, for each delayed day of payment, and until a total amount of 10% of the total value of the contract. This amount will be discounted by the PA to the unpaid debts of the current contract from the account of the CA.

FIFTH CLAUSE. Duties acquired by the CA. The CA is obliged, not only to the duties acquired by purchasing a LO, but also to the followings. (a) To absolutely reserve the information on the LO and to maintain the exclusivity of the LO. (b) To make the payment accorded in this contract on time and form. (c) If the CA does fail on any of these duties, the PA has the right to penalize to the CA on the fail.

SIXTH CLAUSE. Duties of the PA. The PA is obliged, not only to the duties acquired by purchasing a LO, but also to the followings. (a) To deliver the LO named ___ on both date and form established on this contract. (b) If the PA does fail on any of these duties, the CA has the right to penalize to the PA on the fail.

SEVENTH CLAUSE. Unilateral ending or suspension of this contract by the CA. The Manager Agent (MA) may suspend or end this contract only on the following cases. (a) Breach of Contract by the CA on the established terms of the payment of the LO, or in the conditions and presentation of the object services of this contract. (b) If the PA decides to end or suspend this contract, the CA will pay, as maximum, the 20% of the first payment value. This amount does not imply any future duty for the CA with the PA.

EIGHTH CLAUSE. Unilateral ending or suspension of this contract by the PA. The Manager Agent (MA) may suspend or end this contract only on the following cases. (a) Breach of Contract by the PA on the established terms of the delivery of the LO, or in the conditions and presentation of the object services of this contract. (b) If the PA has not delivered the agreed LO on the following thirty (30) days of the contract signing, the CA is exonerated from any responsibility of any duty acquired by this contract, or for the delayment in the satisfaction of any assistance on its charge, only when the delayment is proved to be Force Majure or unforeseen circumstances. The CA is enabled to temporarily suspend the following contract under these situations, and only the PA is capable to continue with the goal task of this contract.

NINTH CLAUSE. Author Copyrights. The PA unconditionally guarantees that the elements used on text, figures, pictures, design and trademarks that are delivered to the CA as included on the LO, are properties of the PA or it has written permission to use them. The CA is free of any responsibility or complaint by the right owners of the material. Just in case that the CA requires certain material included on the LO, the CA may take its own files, or use free license files under public licenses, or reproduce the required material, in the latter they will be deducted on the current contract. All of the LO will have the written credits that correspond to the PA. The CA will keep the right of storing one copy of this LO. The PA will unconditionally retain its copyrights, including the invention rights and the author copyrights on the contracted design.

TENTH CLAUSE. Improvements. For security reasons and to prevent falsification by the steal of identity, this contract can only be considered as valid from the moment when both PA and CA obtain their keys.

ELEVENTH CLAUSE. Contract extension. In case that the contract can not be accomplished on the accorded terms, the CA is much obliged to make a written request for more time to satisfy the contract. This extension should be documented and delivered at least two days before the due date of the contract.

TWELFTH CLAUSE. Contract cancellation. This contract may be cancelled by an agreement between the PA and CA, in a period established in the current contract, or at least before two months since the sign of the contract, or due to the expedition of an administrative document that orders the termination, or at the accorded date by both parts of the contract. In the termination document the adjustments, revisions and acknowledges shall be accorded. The document shall also contain the agreements, conciliations and transactions that shall be reached to put an end on the presented situations, so the cancellations can be declared on peace and safe. As a proof, the current contract is subscribed in ___ at the ___days in the month of ___ of the year 20_.

In order for the agents to interpret the policies previously described about access and control of the available transactions on the LOM, an ontology is currently under development. The main objective of this ontology is that the agents shall be enabled to respond with certain events to the analyzed information.

5 An Ontology for the Policies of a LOM

An ontology is an explicit specification of a concept or some part of the whole concept. The ontology includes a list of terms and the specification of the meaning of each term. In this specification, objects, process, resources, capabilities and many other elements can be described [6].

The most important information that shall be considered on the LOM policies is described next. The information that shall be stored in the contract between a Consumer and a Provider Agent after a negotiation process shall be also considered, in order for the good use of a Learning Object.

There are many methodologies to develop ontologies, so in order to develop an ontology for the LOM, Methodology [2] is being used. Nowadays, this is one of the most used methodologies around the world. As a tool for development, Protégé_3.1 [18] is also being used. Protégé is based on the ontological language OWL (Ontology Web Language).

The first step on Methodology is to create the glossary of terms that belong to the developed domain –also known as Taxonomy. For the LOM, the domain considers the policies and the contract. The taxonomy is composed by the following concepts –in Spanish-:

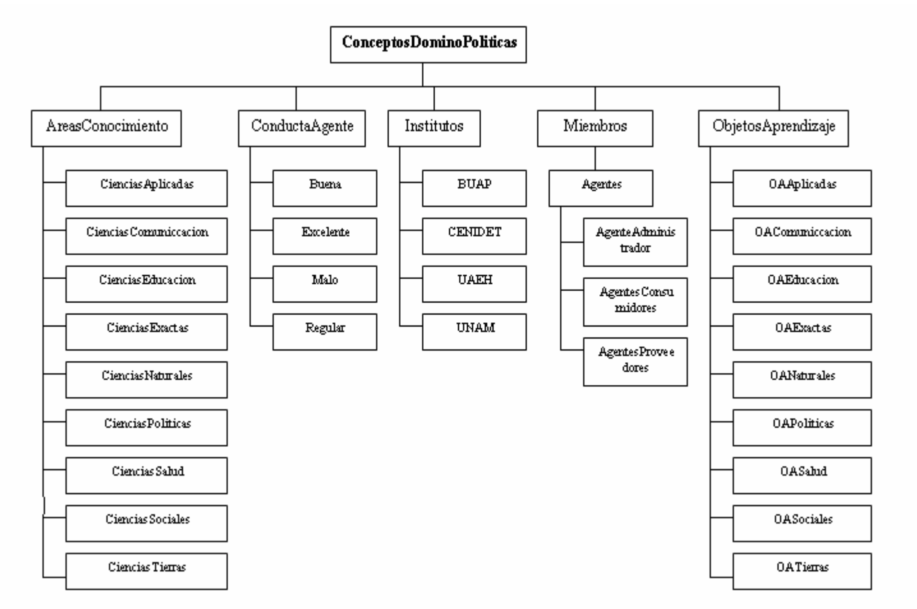


Fig. 1. Concepts of Command of Policies

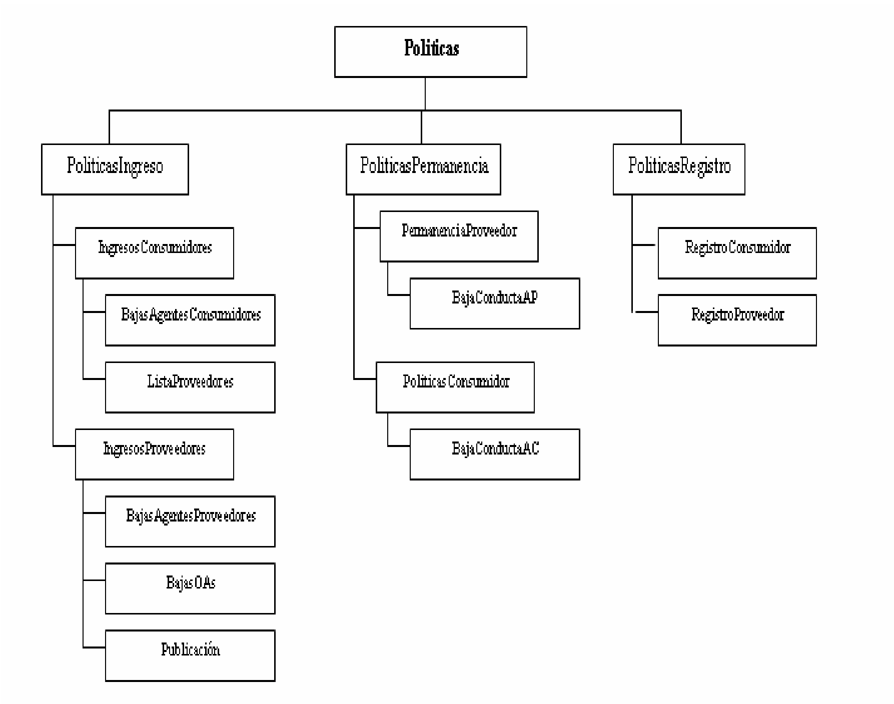


Fig. 2. Policies.

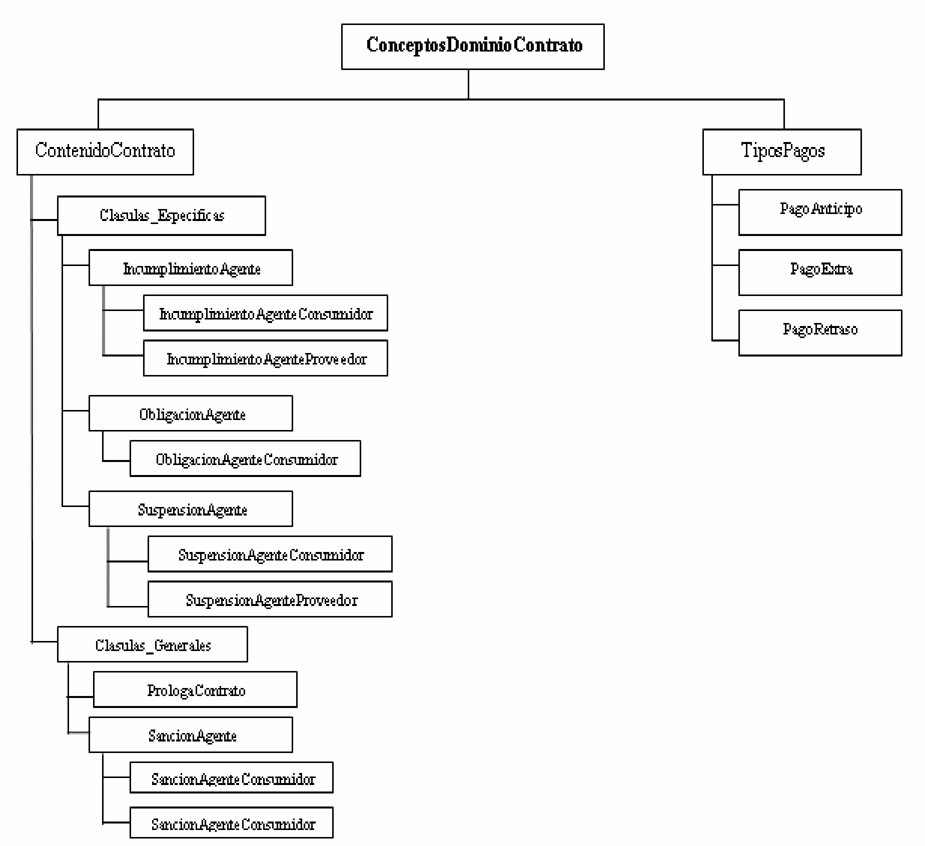


Fig. 3. Concepts of Command of Contract

The relationship between classes has been established, for each relation a group of rules and axioms have also been defined. With these rules the inferences are made and with the axioms the conditions that shall be accomplished by the different concepts are declared. In order to find inconsistencies, the current ontology has been validated with the RACER software tool -Renamed ABox and Concept Expression Reasoner [13].

6 Conclusions

In order to develop the Multi-Agent System, the platform JADE_has been chosen because it satisfies the FIPA standards. The agents have been implemented and based on behaviors, and depending on the sent and received messages. The communication language that is used for the agents is ACL. Currently the LOM is in testing phase. It can be tested on the website: www/dgmae/mercadoOA

In the current work the policies for login and permanence of any agent in the LOM have been defined. It also explains the contract that any provider and consumer will sign about the use of a LO. This has been developed by using domain ontology.

Nevertheless, it is necessary that the agents shall be enabled to interpret the ontology to create new behaviors, in order to create new actions to be performed. The research is currently centered in this topic.

References

1. De Roure, N. Jennings, and N. Shadbolt, The Semantic Grid: A future e-science infraestructure. Grid Computing, 2003, pp. 437-470.
2. Fernández M., Gómez-Pérez A. y Juristo N. “METHONTOLOGY: From Ontological Art Toward Ontological Engineering”. Spring Symposium Series on Ontological Engineering. (AAAI’97). Págs. 33-40. Stanford. California. USA. Marzo, 1997.
3. F. López and M. Luck. “A Model of Normative Multi-Agent Systems and Dynamic Relationships”. In Lindemann et al. [4], Pág. 259–280.
4. F. L. y López and A. A. Márquez. An Architecture for Autonomous Normative agents. In ENC ’04: Proceedings of the Fifth Mexican International Conference in Computer Science (ENC’04), pages 96–103, Washington, DC, USA, 2004. IEEE Computer Society.
5. G. Sánchez, “Sistema de Mercado para acceder Objetos de Aprendizaje”. Tesis Profesional, de la Facultad de Ciencias de la Computación, BUAP. Junio 2006.
6. Gruber T. “A Translation Approach to Portable Ontology Specifications”. Knowledge Acquisition, vol.5 no.2, Págs. 199-220. 1993.
7. M. Wooldridge. “An Introduction to Multiagent Systems, Chapter Intelligent Agents”, pages 15–46. John Wiley & Sons, 2002.
8. Van Heist G. Schreiber A. Y wielinga B. „Using Explicit ontologies in KBS Development“. International Journal of Human-computer studies, vol 45 pag. 183-292. 1997.
9. http://www.amazon.com/,http://www.ebay.com/,http://www.mercadolibre.com/noviembre 2006
10. http://www.gestiopolis.com/canales7/ger/importancia-de-las-politicas-para-la-comunicación-organizacional.htm 2007
11. http://www.hipertexto.info/documentos/ontologias.htm
12. http://protege.stanford.edu, 2006
13. http://www.racer-systems.com/ septiembre de 2005
14. http://es.wikipedia.org/wiki/Contrato#Elementos_del_contrato 2006
15. http://es.wikipedia.org/wiki/Pol%C3%ADtica, 2004

Comparative Evaluation of Free/Open Source Software to Develop Virtual Reality Systems

Eduardo Islas Pérez[1], Ivonne Ávila Gutierrez[1], Ilse Leal Aulenbacher[1] and
Benjamin Zayas Pérez[1]

[1] Instituto de Investigaciones Eléctricas, Av. Reforma 113 Col Reforma,
Cuernavaca, Morelos, México, 62490
{eislas, ivon_ag, ileal, zayas}@iie.org.mx

Abstract. In this paper we describe an evaluation methodology for virtual reality (VR) free/open source software tools. A Multi Criteria Decision Making (MCDM) methodology based on a criteria set and weight assignment was applied. The analysis and evaluation aimed to help decision makers to select the most appropriate software tools to develop VR applications. The selected tools were used to develop a virtual reality system to teach concepts related to generation, transmission and distribution of electricity from a power plant to consumption centres.

1 Introduction

The success of a virtual reality project or system relies on many factors. One of the most important aspects is good planning, in which software resources must be considered and managed. An adequate selection of VR software tools may determine the success or failure of a project.

In order to develop a VR system with a high degree of interaction, immersion and realism, we need different types of free/open source and commercial software tools. Therefore, software analysis and evaluation must be carried out to identify the most appropriate tools that can be integrated to support the development process. The results of these activities should reflect in the selection of software with the right features according to the system requirements. These results can improve the planning and developing of a project.

The system described in this paper explores the use of VR as a training tool to help learners (utility workers and students) to become familiar with the equipment and facilities of a power system: from electricity generation in a power plant, to the distribution lines for domestic supply, going through the transmission towers and the substations. This application provides the users with different levels of immersive, interactive and three-dimensional experience allowing them to explore the power system freely in order to know the elements and equipment involved with power generation, transmission and distribution.

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Copmuting Science 29, 2007, pp. 268-281

Received 18/09/07
Accepted 19/10/07
Final version 24/10/07

2 Free/Open Source Software

Besides the obvious low cost of Free/Open Source Software (FOSS), there are many other reasons why public/private organizations are adopting this kind of technology [1]. The most important are: security, reliability/stability, open standards / vendor independence, reduced reliance on imports, developing local software and capacity.

– **Security**. Development method, program architecture and target market can greatly affect the security of a system and consequently make it easier or more difficult to violate. There are some examples where FOSS systems are superior to proprietary systems [2].

  Three reasons are often cited for FOSS's better security record:

  - **Availability of source code**: Availability has made it easier for developers and users to discover and fix vulnerabilities as soon as they are found.
  - **Security focus, instead of user-friendliness**: It is more focused on robustness and functionality, rather than ease of use.
  - **Roots**: These systems are mostly based on the multi-user, network-ready Unix model. Because of this, they come with a strong security and permission structure.

– **Reliability/Stability**. FOSS is well known for their stability and reliability. For example, Vaughan and Steven conducted a reliability test between Red Hat Linux, Caldera Systems OpenLinux and Microsoft's Windows NT Server 4.0. The result was that NT crashed once every six weeks but none of the FOSS systems crashed at all during a period of 10 months [3].

  In other example Prof. Miller from Wisconsin University has been measuring reliability by feeding programs random characters and determining which ones resisted crashing and freeze-ups (Fuzz testing). This approach is unlikely to find subtle failures, the study found that their approach still manages to find many errors in production software and is a useful tool for finding software flaws. What is more, this approach is extremely fair and can be broadly applied to any program, making it possible to compare different programs fairly [4].

– **Open standards and vendor independence**. Open standards give users flexibility and the freedom to change between different software packages, platforms and vendors. Proprietary, secret standards lock users into using software only from one vendor and leave them at the mercy of the vendor at a later stage, when all their data is in the vendor's proprietary format and the costs of converting them to an open standard is prohibitively high.

– **Reduced reliance on imports**. A major incentive for developing countries to adopt FOSS systems is the enormous cost of proprietary software licenses. Because virtually all proprietary software in developing countries is imported, their purchase consumes precious hard currency and foreign reserves. These reserves could be better spent on other development goals in developing countries.

– **Developing local software capacity**. It has been noted that there is a positive correlation between the growth of a FOSS developer base and the innovative capacities (software) of an economy. There are three reasons for this:

- **Low barriers to entry:** FOSS, which encourages free modification and redistribution, is easy to obtain, use and learn from.
- **FOSS as an excellent training system**: The open and collaborative nature of FOSS allows a student to examine and experiment with software concepts at virtually no direct cost to society. Likewise, a student can tap into the global collaborative FOSS development network that includes massive archives of technical information and interactive discussion tools.
- **FOSS as a source of standards**: FOSS often becomes a de facto standard by virtue of its dominance in a particular sector of an industry. By being involved in setting the standards in a particular FOSS application, a region can ensure that the standard produced takes into account regional needs and cultural considerations.

FOSS has significant market share in many markets, is often the most reliable software, and in many cases has the best performance. FOSS scales, both in problem size and project size and often it has far better security, perhaps due to the possibility of worldwide review. Total cost of ownership for FOSS is often far less than proprietary software, especially as the number of platforms increases. These statements are not merely opinions; these effects can be shown quantitatively, using a wide variety of measures. This does not even consider other issues that are hard to measure, such as freedom from control by a single source, freedom from licensing management (with its accompanying risk of audit and litigation) [5].

In the case of our application, we need to accomplish some of these features. For instance, security is needed for systems that run over the Internet or a public network; in fact, we aim to develop this kind of applications in the near future. In the reliability/stability context, not only the mentioned aspects are relevant but also the stability of developers or vendors. For example, the well-known company Sense 8 is no longer available in the VR field since a couple of years ago, leaving some customers without any kind of support for their VR systems or projects.

3 VR Software Description

We have identified four types of software tools commonly used to develop VR applications, which are: toolkits and graphic environments for programming and developing VR applications, tools for 3D modeling, tools for developing mathematical models and tools for 3D visualization. Figure 1 depicts this type of software tools. The analysis and evaluation described in this paper are mainly based on this classification.

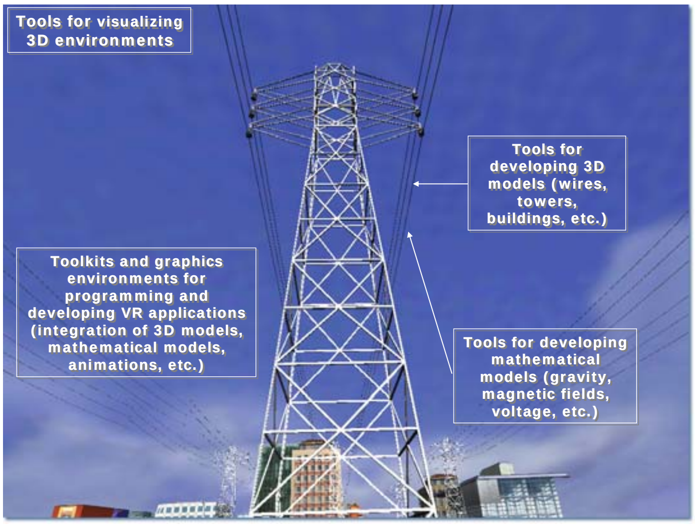

Tools for visualizing 3D environments
Tools for developing 3D models (wires, towers, buildings, etc.)
Toolkits and graphics environments for programming and developing VR applications (integration of 3D models, mathematical models, animations, etc.)
Tools for developing mathematical models (gravity, magnetic fields, voltage, etc.)

Fig. 1. Types of software tools involved in a VR system development

- **Toolkits and graphic environments for programming and developing VR applications.** Although there are many kinds of toolkits to develop three-dimensional environments or virtual worlds, we have only considered two types of tools:
  - **Graphic environments to develop VR applications**. These tools provide a graphical environment to develop applications. Basic nodes or primitives are used to build more complex virtual worlds. In addition to the graphical interface, this type of tools also offer some built-in basic animations and object behaviors, which make the development of applications easier than with development toolkits. In particular, a good background in programming is not needed to develop applications. This is one of the biggest advantages of these environments. However, the limited functionality they provide precludes the development of complex applications.
  - **Toolkits for programming VR applications.** A toolkit is an extensible library of object-oriented functions designed specifically for developing VR applications. Toolkits are in a middle stage between low-level graphic language, such as Open GL [6] and graphic environments such as Open Inventor [7]. These tools afford functionality through a rich set of function libraries such as: connectivity with input/output hardware, behaviors, animations, lighting techniques, etc. Complementary functions can also be programmed by developers using high-level programming languages such as C++ or Java.
- **Tools for 3D modeling**. Three-dimensional models can be created and edited with this type of tool. The usage of a diversity of techniques, objects, scenes and environments can be replicated. Some of these tools allow developing simple object animations, object behaviours and special effects through scripting. These tools are important in the sense that they create the visual part of a VR application.

- **Tools for developing mathematical models**. With this kind of tool, object behaviors can be modeled in a virtual environment. Simulations are based on mathematical models of behaviors such as gravity, inertia, weight, acceleration, etc. Object behaviour makes the representation of physical settings more realistic.
- **Tools for 3D visualization**. These types of tools are used for model visualization, interacting with virtual objects in a scene and exploring a virtual environment. These tools are divided into two categories: general-purpose or developed for a specific application. Usually, toolkits or other virtual reality software offer viewers for their particular applications. However, those general-purpose viewers are limited in functionality. Viewers with extended functionality can be built with the tools aforementioned in this section. These viewers can be distributed to final users without buying additional development licenses for toolkits and graphic environments.

4 VR Software Evaluation

The MCDM methodology used in this project is based on some concepts from the methodology described in [8] which was applied in the evaluation of VR hardware and software tools [9] and appraisal of Learning Management Systems (LMSs) [10]. It is worth pointing out that at some degree it is a general-purpose methodology, in the sense that depending on the kind of items to be evaluated, a set of matching criteria (or parameters) must be defined. We applied this methodology to evaluate different VR software tools. However, due to lack of space, only the evaluation details of toolkits and graphic environments for developing VR applications are presented in this paper. Results for other software tools are only shown.

a) **Identification and selection of evaluation parameters**. The list of parameters considered in the toolkits evaluation and graphic environments is shown in Table 1:

Table 1. Evaluation parameters for toolkits and graphic environments

	Parameter		Parameter
1.	*Drivers to ease hardware integration*	2.	Use of communication networks
3.	Classes or functions library	4.	Multiplatform and portability
5.	Import and export 3D models and scenes	6.	Import and export animations
7.	Geometries library	8.	Optimization
9.	Audio	10.	Realism level
11.	Animation	12.	Rendering and visualization
13.	Use of databases	14.	Open source and versions
15.	Availability of demos	16.	Management aspects
17.	Company Profiles		

b) **Value assignment for each parameter**. Table 2 shows in detail the features that are taken into consideration when grading and scaling the *Drivers to ease hardware integration* parameter. Details on the other parameters have been intentionally omitted due the lack of space.

Table 2. Grading for drivers to ease hardware integration

Features
Drivers to ease the use of:
1. Video equipment (*HMDs*, *eyeglasses*, *CAVEs*, etc.)
2. Audio equipment (*headphones*, *speakers*).
3. Haptic devices (*gloves*, *cybergrasp*, etc.)
4. Equipment for movement (input devices)(*mice*, *josticks*, etc.)
5. Positional gadgets (*trackers*, *nestbirds*, etc.)

c) **Weight assignment for all of the parameters**. Weights assigned to each parameter were: 1, 1.5 and 2 where 1 means an optional parameter, 1.5 a parameter to improve immersion, interaction or development and 2 means a very important parameter.

d) **Identification and selection of tools**. Because of the great number of software tools available nowadays and based on the most important attributes, we made a pre-selection of software tools. The final list of the evaluated software tools is shown in Table 3.

Table 3. Identification and selection of software tools

Software Tools	**Company**	**Price (USD)**
MetaVR [11]	MetaVR, Inc.	10,500.00
IRRLicht [12]	IRRLicht	FOSS
Cult3D [13]	Cycore	7,700.00
Torque [14]	GarageGames, Inc.	395.00
Open Inventor [7]	Mercury Inc.	5,000.00
Horizon Scene Graph [15]	DigiUtopikA Lda.	Not available on line
OpenGL Performer [16]	Silicon Graphics	Not available on line
Panda 3D [17]	Disney and Carnegie Mellon University	FOSS
Java 3D [18]	Sun Developer Network	FOSS
OpenSceneGraph [19]	*OpenSceneGraph*	*FOSS*
X3D [20]	Web 3D Consortium	FOSS
VR Juggler [21]	Iowa State University's Virtual Reality Center	FOSS

e) **Analysis and evaluation of each tool**. Table 4 shows the results for the OpenSceneGraph evaluation using an MCDM method. This method is the additive value function and non-hierarchical weight assessment which is briefly described below [10].

$$MAX\ V\left(x_j\right)=\sum_{i=1}^{n} w_i v_i\left(x_{ij}\right) \tag{1}$$

where:

x_{ij} = The value of criterion *i* for alternative *j*

$v_i(x_{ij}) =$ A single criterion value function that converts the criterion into a measure of value or worth. These are often scaled from 0 to 1, with more being better. In this method these values were not scaled

$w_i =$ Weight for criterion *i*, representing its relative importance.

$n =$ Number of criterions

Table 4. Evaluation results of OpenSceneGraph

Parameter	x_{ij}	$v_i(x_{ij})$	w_i	$w_i v_i(x_{ij})$
Drivers to ease hardware integration	*1*	*1*	*2.0*	*2.0*
Use of communication networks	2	2	1.5	3.0
Classes or functions library	4	4	2.0	8.0
Multiplatform and portability	2.5	2.5	2.0	5.0
Import and export 3D models and scenes	3	3	2.0	6.0
Import and export animations	3	3	2.0	6.0
Geometries library	2	2	1.0	2.0
Optimization	5	5	2.0	10.0
Audio	1	1	1.5	1.5
Realism level	3	3	2.0	6.0
Animation	3	3	2.0	6.0
Renderization and visualization	4	4	2.0	8.0
Use of databases	2	2	1.5	3.0
Open Source and versions	4	4	1.0	4.0
Demos availability	3	3	1.0	3.0
Management aspects	3	3	2.0	6.0
Company profiles	1	1	1.5	1.5
			$\sum_{i=1}^{n} w_i v_i(x_{ij})$	**81.0**

f) **Obtaining a graph to compare tools**. The results obtained for this group of tools are shown in Figure 2. The results show the best commercial and FOSS toolkits and graphic environments for programming and developing VR applications. Figure 3, Figure 4 and Figure 5 show the evaluation for the other types of software considered in this assessment, which were obtained by applying this methodology.

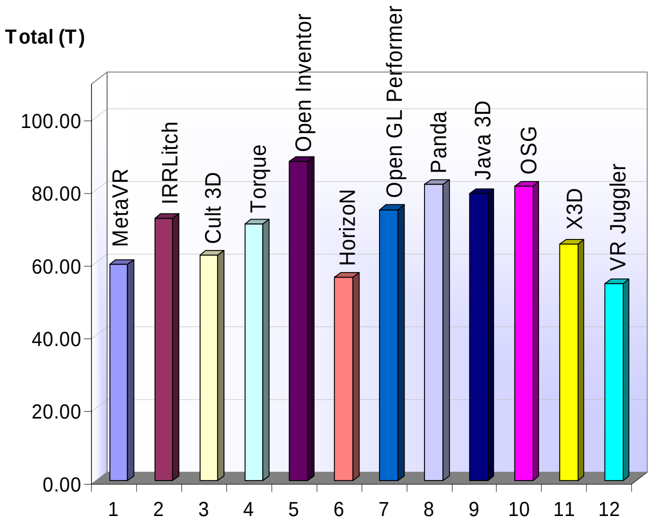


Fig. 2. Toolkits and graphic environments for programming and developing VR applications

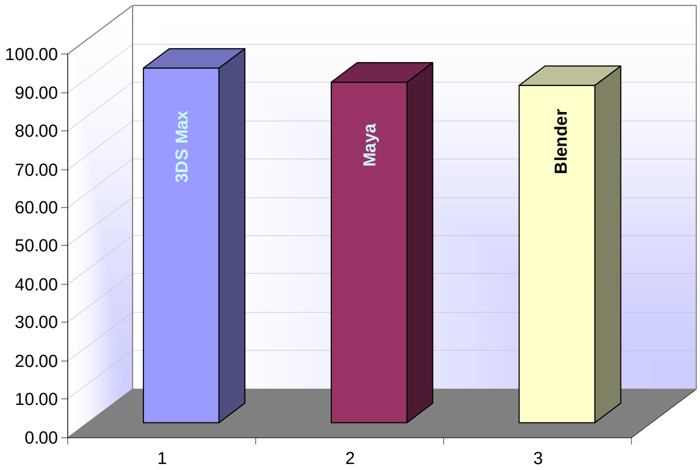


Fig. 3. Tools to develop 3D models (3DS Max, Maya, Blender)

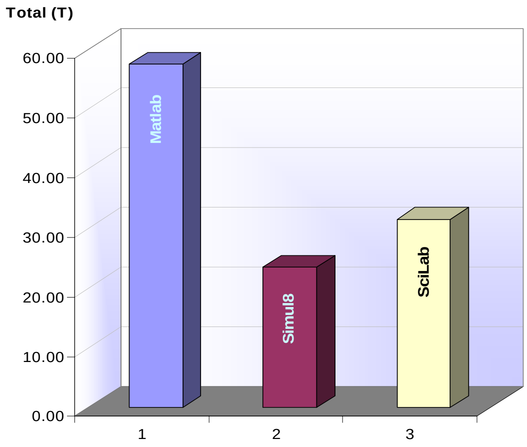


Fig. 4. Tools to develop mathematical models (Matlab, Simulink, SciLab)

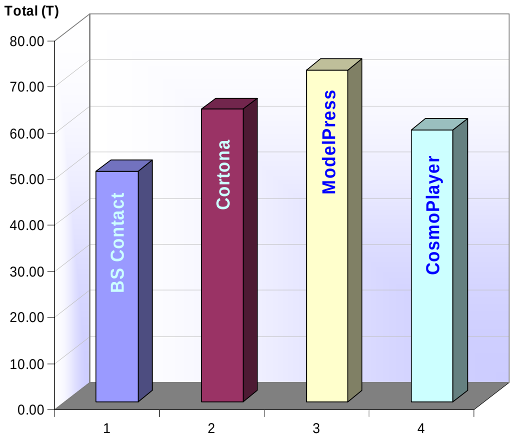


Fig. 5. Tools to visualize 3D environments (BS contact, Cortona, ModelPress, CosmoPlayer)

g) **Documentation of results and conclusions**. According to the results analysis and evaluation, the best FOSS tools to develop VR applications can be identified. These results can help make decisions about the best configuration to build a platform considering the software tools evaluated in previous stages.

Based on the evaluation results, the best combination of FOSS to develop an interactive virtual environment is shown in Figure 6.

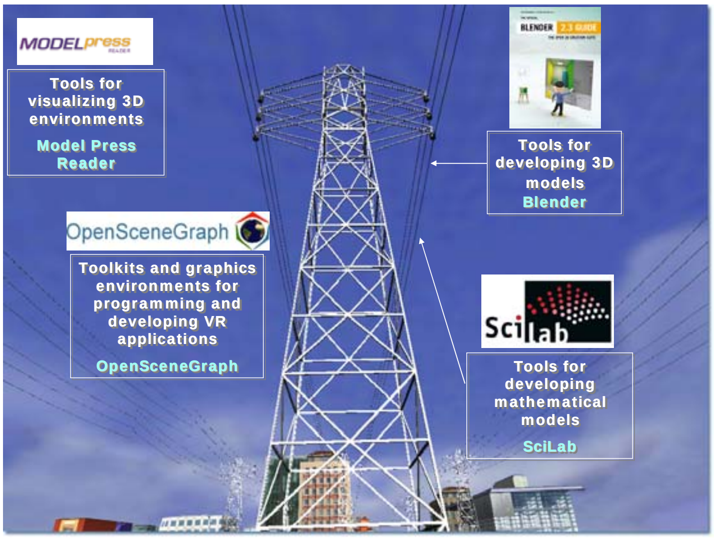

MODELpress READER
Tools for visualizing 3D environments
Model Press Reader
BLENDER 2.3 GUIDE
Tools for developing 3D models
Blender
OpenSceneGraph
Toolkits and graphics environments for programming and developing VR applications
OpenSceneGraph
Scilab
Tools for developing mathematical models
SciLab

Fig. 6. FOSS tools recommended for developing an interactive virtual environment

5. Development of a VR Application using FOSS

With the results we were able to know the best possible combination of software tools to develop a virtual power system tutorial. In this section we provide some snapshots to illustrate the way Blender was used to create 3D models and OpenSceneGraph to program and develop the VR application [22].

5.1 Tool for 3D Modeling

We propose the use of Blender (v 2.44) to develop the 3D models for the tutorial development. This FOSS tool runs on several platforms (Windows, MacOS, Linux, FreeBSD, Irix and Solaris). Furthermore, it has very important animation features such as: physics features and particles functions [23].

5.2 Tools for programming and developing VR applications

The use of OpenSceneGraph (v 1.9.8) is justified because it was the best VR FOSS tool obtained from the evaluation. One of its most important features is that many of its libraries were developed in C++ .Net. Additionally, it includes optimization methods such as: culling, different levels of detail, etc. Finally it presents many features for rendering and visualization [19].

5.3 Developing a Virtual Power System Tutorial using FOSS

The virtual power system tutorial allows a better learning experience than just reading or viewing photographs, because users can acquire knowledge using some of the activities proposed by Moshel and Hughes to improve users learning: constructivist, constructionist and situated [24]. Constructivist learning involves the exploration of prebuilt worlds and discovery, which is obtained with the exploration of the virtual world. Furthermore the users learn by means of situated learning because the students can have interaction with the virtual world using most of their senses to explore a power system in an immersive environment that gives him/her the sense of actually being there. This kind of VR applications with additional features can be further applied to personnel training in power plants, equipment maintenance, etc. without the risk of accidents or equipment damage.

In the developed tutorial the user/student can interact with all the objects situated in the virtual power system. The following figures show some power system tutorial screenshots [22]. For example, figure 7 shows an exterior view of a power plant; this particular application is focused on a fossil power plant where electricity is generated using petroleum. Other kind of power plants could be modeled and integrated into the system to learn about them. For example, the user can learn about differences between geothermal, hydropower and nuclear power plants.

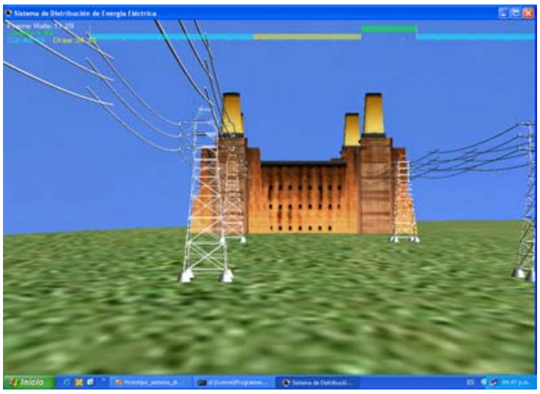


Fig. 7. A screenshot of the exterior view of a fossil power plant

In figure 8 a view of the transmission lines and towers is depicted, where the user can learn about different voltage levels for transmission (400, 230 and 161 KVs) and how the lines arrive at a power substation in order to reduce the level of voltage (34.5, 23, 13.8, 6.6, 4.16 y 2.4 kVs) for the distribution system.

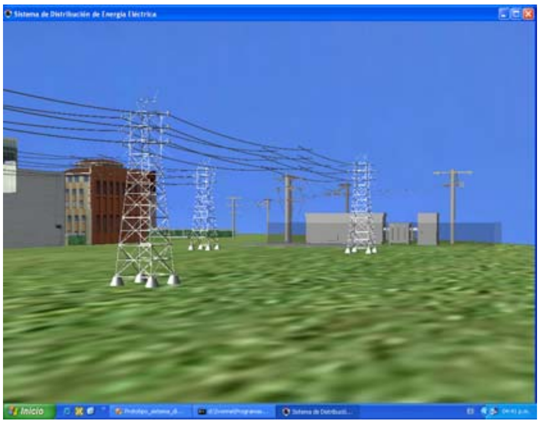

Sistema de Distribución de Energía Eléctrica
Inicio
Sistema de Distribució...

Fig. 8. Transmission towers and a power substation screenshot

Figure 9 shows a narrow view of the electrical substation. In the tutorial, the user can navigate and interact with the substation equipment to learn how voltage is reduced for its use in the distribution system and gain knowledge about different components of a substation: switchgears, power transformers, surge protection, controls, metering, etc.

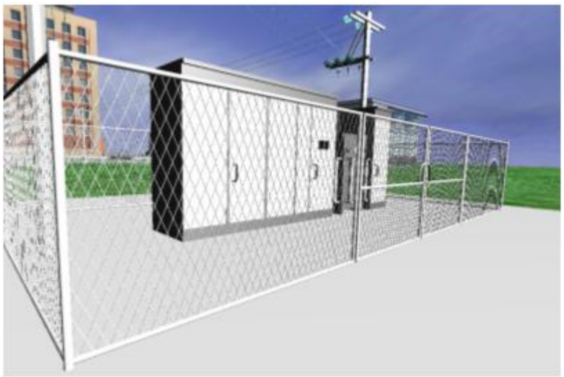


Fig. 9. A narrow view of the electrical substation

Figure 10 shows a screenshot of a city. In this part, the user can learn about the distribution of electricity at low voltages (440, 220 and 127v) and how electricity is supplied to final consumption centers such as: buildings, streets, houses, factories, etc.

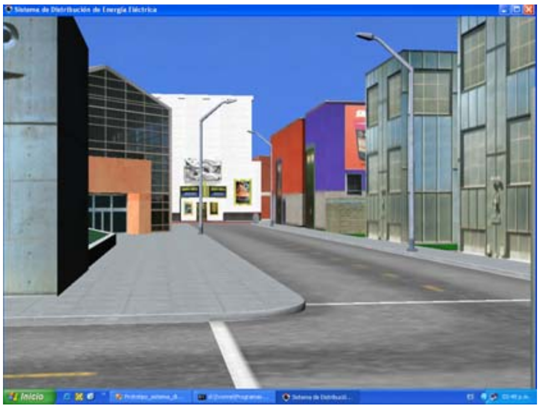


Fig. 10. A screenshot of the use of electricity in consumption centers

6 Conclusions and Future Work

The methodology suggested in this paper helped to evaluate objectively, the characteristics and functionality of commercial and open source software tools. A Multi Criteria Decision Making methodology based on a criteria set and weight assignment was useful to facilitate the selection of VR software tools. The Virtual Reality Group reduced time and effort in the development process of virtual reality systems. In so doing, we have obtained further information to improve and refine the methodology.

Future work includes, review of the evaluation methodology according to the fast changes in technology and keep track of software updates in order to obtain a reliable and updated evaluation. New criteria should be introduced to take into account factors derived from our implementation experience. For instance, geometric format compatibility, reliable documentation, functionality, and development support.

Additionally, in the near future real physical behavior will be added to the objects for simulation. We will develop mathematical models for behaviors using SciLab which is the top rated FOSS tool according to the evaluation. After that, we will add those mathematical models (for example gravity, inertia, weight, magnetism, etc.) to the power plant tutorial in order to have a more realistic environment.

References

1. Wikibooks. FOSS A General Introduction/Why FOSS?, http://en.wikibooks.org/wiki/FOSS_A_General_Introduction/Why_FOSS%3F#_ref-19, last modified 13 May 2007.
2. Pescatore, J., Commentary: Another worm, more patches, CNet News.com; available from http://news.com.com/2009-1001-273288.html?legacy=cnet&tag=nbs ; 20 September 2001.

3. Vaughan-Nichols, S. J., Can You Trust This Penguin?, ZDNet SmartPartner. http://web.archive.org/web/20010606035231/http://www.zdnet.com/sp/stories/issue/0,4537,2387282,00.html ; 1 November, 1999.
4. Miller B., Fuzz Testing of Application Reliability, http://pages.cs.wisc.edu/~bart/fuzz/fuzz.html, Last modified: Jun 2 2006.
5. Wheeler D., Why Open Source Software / Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers, http://www.dwheeler.com/oss_fs_why.html, Last modified: April 16, 2007.
6. Open GL. The Industry's Foundation for High Performance Graphics, http://www.opengl.org/, visited on Jun 2007.
7. Open Inventor from Mercury, Overview, Open Inventor main features, 2002, http://www.tgs.com/pro_div/oiv_overview.htm, visited on Jun 2007.
8. Pérez M., Zabre E. and Islas E., Prospectiva y ruta tecnológica para el uso de la tecnología de realidad virtual en los procesos de la CFE, Instituto de Investigaciones Eléctricas, Cuernavaca México, Technical Report IIE/GSI/022/2003, 2003.
9. Islas E., Zabre E. and Pérez M., Evaluación de herramientas de hardware y software para el desarrollo de aplicaciones de realidad virtual; Boletín IIE, vol. 28, pp. 61-67, Apr-Jun 2004.
10. Islas E., Pérez M., Rodriguez G., Paredes I., Ávila I. and Mendoza M., E-learning Tools Evaluation and Roadmap Development for an Electrical Utility, Journal of Theoretical and Applied Electronic Commerce Research, Vol 2, pp. 63-75, Apr 2007.
11. Virtual Reality Scene Generator (VRSG) With Tracker Support, MetaVR, http://www.metavr.com/products/vrsg/immersim.html, visited on Jun 2007.
12. Irrlicht, Features, http://irrlicht.sourceforge.net/features.html, visited on Jun 2007.
13. Cult3D, Welcome, http://www.cult3d.com/, visited on June 2007.
14. Torque, Torque Game Engine SDK, http://www.garagegames.com/products/1, visited on June 2007.
15. DigiUtopikA, HorizoN Scene Graph, Portugal, http://www.utopika.net/horizon/EN/horizonsg.html, visited on Jun 2007.
16. OpenGL Performer, Overview, http://www.sgi.com/products/software/performer/overview.html, visited on Jun 2006
17. Panda3D, About Panda3D, What is Panda3D, http://www.panda3d.org/what.php, visited on Jun 2007.
18. Sun Developer Network, Java 3D API, http://java.sun.com/products/java-media/3D/, visited on Jun 2007.
19. OpenSceneGraph, Homepage, http://www.openscenegraph.com/index.php, visited on Jun 2007.
20. Web 3D Consortium, X3D Developers, http://www.web3d.org/x3d/, visited on Jun 2007.
21. VR Juggler, Juggler Suite of Tools Project Overview, Iowa State University, http://developer.vrjuggler.org/, visited on Jun 2007.
22. Avila I., Desarrollo de un prototipo de realidad virtual como tecnología de apoyo para capacitación de personal en CFE, Tesis de Licenciatura, Agosto 2005
23. Blender, Features and Gallery, http://www.blender.org/features-gallery/, visited on Jun 2007.
24. Moshel M and Hughes C., Virtual Environments as a Tool for Academic Learning, in K. Stanney (Ed.), The Handbook of Virtual Environments Technology, Erlbaum, Mahwah, NJ, 2002.

Energy Saving Analysis for the Wavelet Transform Processing

Héctor Silva-López[1], Sergio Suárez Guerra

Center of Computational Research, National Technical Institute, México, A.P. 75-476, C.P. 07738, Zacatenco, DF, México.
hsl@fis.cinvestav.mx[1], ssuarez@cic.ipn.mx

Abstract. The applications of the wavelet transform which are executed in mobile devices usually operate with batteries of limited capacity. The problem to be solved is how to increase the efficiency of the batteries used so that their time life can be measured in days instead of hours. On the other hand, the ability to adapt the changes of the work charge conditions for a real-time system with energy saving is handled in this article using an algorithm for graphic applications (by means of discrete wavelet transform) which presents a considerable energy saving in its processing. These applications will be executed in a variable voltage processor. The algorithm selects the way of execution of each level of discrete wavelet transform so that the energy saving is maximized and that the deadline is not lost. The problem is presented as a problem of dynamic optimization with discrete constraints. The experimental results were focused on images that showed that the filters Haar, Daub4 and Daub6 had a smaller execution time for the algorithm presented. These filters had a smaller execution time for smaller images size. The energy saving is greater for smaller images and smaller for bigger ones. An extra saving is obtained if the algorithm is executed in a shared memory environment.

1 Introduction

Although wavelet have their roots in approximation theory [7] and signal processing [14], they have recently been applied to many problems in computer graphics. These graphics applications include image editing [4], image compression [12], and image querying [3]. These applications impose strict quality of service requirements in the form of timing constraints. Ignoring energy consumption, operating the CPU at its highest speed operation quickly drains the batteries. Thus there is a tradeoff between reduced energy consumption and quality of service.

Voltage scaling technology has the potential to exploit such variability in the case of meeting timing constraints. By adjusting the operating voltage of the processor, the energy consumption and speed can be controlled [1]. Power regulator and variable voltage

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Copmuting Science 29, 2007, pp. 285-294

Received 02/07/07
Acepted 19/10/07
Final Version 20/10/07

processors with response times in the microseconds range are available [10]. Fast response time makes it practical to dynamically adjust the voltage at run time.

Recently, researches have attempted to apply Dynamic Voltage Scaling (DVS) to video decoding to reduce power [5, 8, 9, 11]. These studies present approaches that predict the decoding time of incoming frames or Group of Pictures (GOPs), and reduce or increase the processor setting based on this prediction. As a result, idle processing time, which occurs when a specific frame decoding completes earlier than its playout time, is minimized. In [6] an alternative method called Dynamic is proposed as an improvement to these techniques. The Dynamic approach is designed to perform well even with high motion videos by dynamically adapting its prediction model based on the decoding experience of the particular video clip being played. The same authors present another alternative method called frame data computation aware (FDCA) in [2]. FDCA dynamically extracts useful frame characteristics while a frame is being decoded and uses this information to estimate the decoding time.

The objective of this work is to develop a new DVS technique for the energy saving. An algorithm that can be used for any graphic application that uses the wavelet transform for its implementation is presented. Some energy saving will be obtained for each level of the transformed one, and the sum of energy savings of all the levels is considered the total energy saving.

This article is structured as follows: In section 2, a general description of wavelet transform is given. In section 3, the problem is formulated and the algorithm is described. Section 4 shows a practical example to demonstrate the function of the algorithm. The simulation of experiments is described in section 5. And finally, the conclusions are described in section 6.

2 Wavelet Transform

Of the many processes available for image compression, two of the most popular transformations are the Discrete Cosine Transform (DCT) used in the common JPEG format, and the Discrete Wavelet Transform (DWT) used in the newer JPEG 2000 format. The DWT differs from the traditional DCT in several fundamental ways. The DCT operates by splitting the image into 8x8 blocks that are transformed independently [13]. Through this transformation process, the energy compaction property of the DCT ensures that the energy of the original data is concentrated in only a few of the transformed coefficients, which are used for further quantization and encoding [15]. It is the discarding of the lower-energy coefficients that result in image compression and the subsequent loss of image quality. Unfortunately, the rigid 8x8 block nature of the DCT makes it particularly susceptible to introducing compression artifacts (extraneous noise) around sharp edges in an image. This is the “halo effect” seen in over compressed web images. Because the artifacts become more pronounced at higher compression ratios, JPEG's suitability for line drawings and cartoon-like images is significantly impaired.

In contrast to the DCT, the DWT operates over the entire image, eliminating artifacts like those caused by the 8x8 DCT blocking. Like the DCT, the fundamental wavelet transform is completely reversible, meaning that if the forward and reverse transforms are applied in sequence, the resulting data will be identical to the original. In addition, the DWT is based on sub-band coding where the image is analyzed and filtered to produce image components at different frequency sub-bands [18]. This produces significant energy compaction that is later exploited in the compression process. The wavelet's two-dimensional nature results in the image visually being divided into quarters with each pass through the wavelet transformation. A key effect of this transformation is that all of the high pass quadrants in the image contain essentially equivalent data [16]. In the field of wavelets, the Haar wavelet is traditionally used for rudimentary image compression because of its algorithmic simplicity and low computational complexity due to an integer based design [18].

When the wavelet is applied and through its filtering process produced the scaling function coefficients (low frequency) and wavelet coefficients (high frequency). From the application of the Haar wavelet, it is evident that the scaling function coefficients is simply the average of two consecutive pixel values, while the corresponding wavelet coefficient is the difference between the same two pixel values. The scaling function coefficients appear to contain all the image data, while the wavelet coefficients appear to be zero (black).

Figure 1 presents the process of the wavelet transform. It begins with an original image, when the transformed by row occurs, the image is divided in two halves, the first half corresponds to the scaling function and the second half corresponds to the wavelet transform; each is called half sub-band.

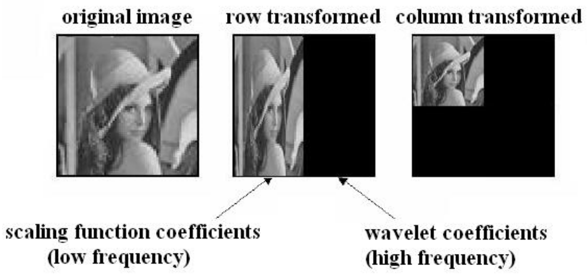


Figure 1.First Level Transform.

When the following transformed by column is performed, this image is divided in four equal parts or sub-bands. The first quadrant or sub-band corresponds to the scaling function and the other three sub-bands, to the wavelet transform; in this first step, four sub-bands are obtained. This process is repeated for the scaling function for the second level and so on successively. In the end, the image is seen as it is shown in figure 2.

3 Formulation of the Problem

The problem can be formulated as follows. Each time a new image arrives at or leaves the system, the problem is to determine the mode (speed) of execution of the sub-bands such that no sub-band misses its deadline and the energy savings of the system is maximized. Each image in the system execute in a discrete voltage/frequency processor. Note that a solution to this problem must be computed each time a new image arrives or leaves the system, therefore a solution with probably cause deadlines to be missed.

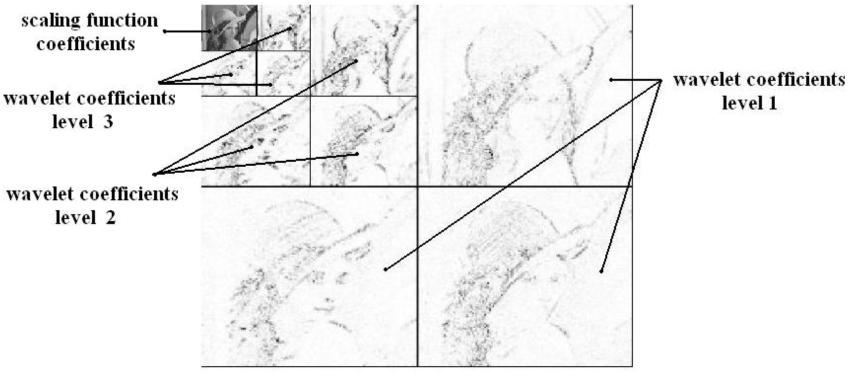

scaling function coefficients
wavelet coefficients level 3
wavelet coefficients level 2
wavelet coefficients level 1

Figura 2. Wavelet Transform for three levels.

The problem is formulated as:

$$\max_{\{u(k)\}_{k=0}^{2}} J = \sum_{k=0}^{2} S_k[u(k)]$$

$$\begin{aligned} \text{subject to} \quad & x(k+1) = x(k) - u(k) \\ & x(0) = 1.0, \\ \text{with :} \quad & x(3) = 0, \\ & u(k) \le x(k), \\ & u(k) \in \{1.0, 0.8, 0.6, 0.4, 0.15\} \ \textit{for} \ k = 0,1,2 \end{aligned}$$

where

$k = correspond to\ sub-bands$

$S_k[u(k)] = energy\ saving$

$x(k) = variable\ of\ state$

$u(k) = control\ variable,(speed\ to\ select).$

Bellman's optimality principle is used by compute the state variable in each stage, as follows:

Step 1: x(3) is calculated as follow:

$$J_3^*\{x(3)\} = 0$$

Step 2: x(2) $\in \{1.0,0.8,0.6,0.4,0.15\}$ is:

$$J_2^*\{x(2)\} = \max_{u(2)}\{S_2[u(2) + J_3^*\{x(2) - u(2)\}\},$$

where:

$$x(3) = 0 = x(2) - u(2), u(2) \leq x(2), u(2) \in \{0.15,0.4,0.6,0.8,1.0\}$$

Step 3: x(1) $\in \{1.0,0.8,0.6,0.4,0.15\}$, the equation that corresponds is:

$$J_1^*\{x(1)\} = \max_{u(1)}\{S_1[u(1) + J_2^*\{x(1) - u(1)\}\},$$

where:

$$u(1) \leq x(1), u(1) \in \{0.15,0.4,0.6,0.8,1.0\}$$

Step 4: x(0)=1.0 is:

$$J_0^*\{1.0\} = \max_{u(0)}\{S_0[u(0) + J_1^*\{1.0 - u(0)\}\},$$

where:

$$u(0) \leq 5, u(0) \in \{0.15,0.4,0.6,0.8,1.0\}$$

4 Example of the Algorithm

To illustrate the execution of the proposed algorithm, we consider the image Lena and we use five discrete speed levels {1.0, 0.8, 0.6, 0.4, 0.15}, in a discrete voltage/frequency processor. The CPU utilization can be measured while the system is under various states of loading. Obviously there is no way (yet) to measure CPU utilization directly. CPU utilization must be derived from measured changes in the period of the background loop. The average background loop period should be measured under various system loads and then the CPU utilization can be obtained. The CPU utilization is defined as the time 'not' spent executing the idle task (is a task with the absolute lowest priority in a multi-tasking system). The amount of time spent executing the idle task can be represented as a ratio of the period of the idle task in an unloaded CPU to the period of the idle task under some known load.

% time in the idle task = (average period of the background task with no load) * 100%
/ (average period of the background task with some load) [1]

Based on DVS technique, we adjust processor speed for each sub-band with slack reclamation.

Table 1 shows the time in microseconds for each sub-band of the image. Five levels are considered. The average execution time of the idle task is 195 microseconds, this time was calculated in the same was as the one presented in [17] for which an Analyzer of Logical State was used to measure the data that flow through the data and direction bus when a task is being executed in background. The time obtained is the average period of the task in background without load. Immediately, the time of the idle task with load is obtained, when each sub-band of the second column of table 1 is executed, and finally, the third column is obtained from equation 1. The fourth column is the result of subtracting the third column from 100 in order to obtain the use of the CPU.

After the executing the first sub-band of the level 1, to the maximum speed, we can obtain the percentage of utilization of this sub-band, the percentage idle is assigned to sub-bands 2 and 3, the executing speed the these sub-bands depending the result of first sub-band. Later, for the others levels, calculating the percentage of utilization for the first sub-band of each level and assigning the percentage idle to the two following sub-bands. In the table 2 is presented the percentage of energy saving level by level. The total energy saving of the all image is 44.19 %.

Sub-bands	T (microseconds)	% Idle	% CPU
1	974	20.02	79.97
2	535	36.49	63.55
3	388	50.26	49.74
4	859	22.70	77.30
5	417	46.76	53.24
6	353	55.24	44.76
7	595	32.77	67.23
8	368	52.99	47.01
9	287	67.94	32.06
10	445	43.82	56.18
11	313	62.30	37.70
12	267	73.03	26.96
13	342	57.02	42.98
14	281	69.39	30.61
15	247	78.95	21.05

Table 1. Sub-bands vs. Average background loop Period.

The energy saving per level of the third column of table 2 was the one that was obtained per sub-band. For the fourth column, the percentage of the total energy saving per level was calculated on the basis of the maximum saving that could be obtained per level with respect to the obtained from the third column. For example, for the first level, the maximum saving that could be obtained is 20,02 for the first sub-band, 36,49 for second sub-band and 50,26 for the third sub-band, which gives a maximum total of 106.77. The total saving obtained per sub-band was of 40,04; then the percentage of saving energy is obtained from a rule of three which is 37.50% for the first level, and so on for the other levels.

Level	Sub-band	% Energy Saving	Total
1	1	0	
	2	20.02	
	3	20.02	= 37.50 %
2	4	0	
	5	22.70	
	6	22.70	= 36.40 %
3	7	0	
	8	32.77	
	9	32.77	= 42.64 %
4	10	0	
	11	43.82	
	12	43.82	= 48-91%
5	13	0	
	14	57.02	
	15	57.02	= 55.53 %

Table 2. Energy Saving for level.

5 Simulation Experiments

The algorithm proposed on the basis of our criterion of optimization presented in this paper is verified in this experiment simulation, using different images and different wavelet filters. The objective of this experiment simulation is to measure the energy saving for different image sizes. Each image size will be observed for different wavelet filters. It can be observed that different execution times are obtained if different wavelet filters are used. The execution time for each level is physically measured in microseconds, having used a Laptop Sony, model VGN-T350P, with a processor Intel Centrino at 3,2 GHZ, with 512MB of RAM and running in the Operating System Fedora Linux version 5.0. The used function to measure the time is psched_get_time.

Figure 3 presents the execution times obtained when performing the wavelet transform for each level, using different filters. It can be seen that Villa2 filter obtained the worst time and that the Haar filter obtained the best one, within the first 3 levels. For the rest of the levels, all the filters tended to have an equal time, tending to zero, mainly because the transformed in level 5 was made only for a block of 32x32 pixels. This Figure 3 shows that the three best filters are the Haar, Daub4 and Daub6, and it is from these filters that the execution time for each level, using different image sizes as shown in figure 4, will be obtained.

Figure 4 shows that the smaller the image, the smaller execution time per level. This same behavior can be observed for the filters Daub4 (figure 5) and Daub6 (figure 6). The difference between the execution times for the three types of filters is conserved in these three figures. It is important to mention that the times (for each filter) were almost equal when they were proven with other images, which lead to conclude that these times are constant for any type of image of the same size.

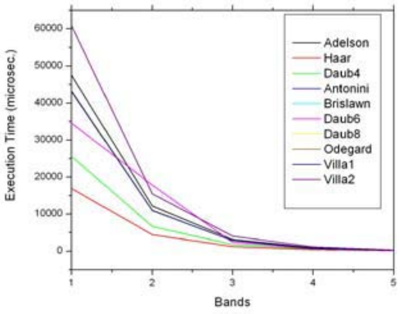


Figure 3. Execution time for different filters

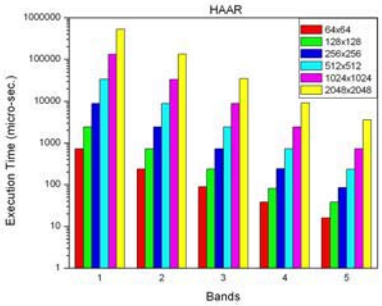


Figure 4. Execution time for filter Haar.

The energy saving for these three filters, using different image sizes, is presented in figure 7. For small image sizes, a greater energy saving is observed and for bigger image sizes a smaller energy saving is obtained.

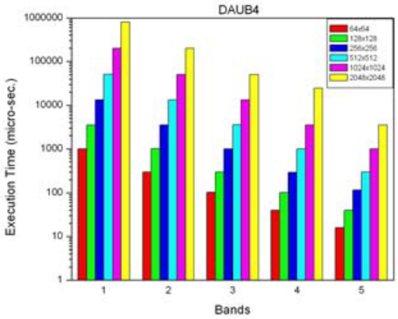


Figure 5. Execution time for filter Daub4.

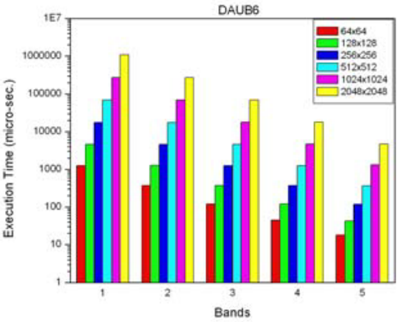


Figure 6. Execution time for filter Daub6

The last step consisted of running the algorithm in parallel form in order to observe how much the energy saving would be reduced in this environment. The algorithm was programmed to be executed by shared memory. The Pthread library, which fulfills with the standards POSIX, and it allowed working with two threads of execution at the same time. Figure 8 shows that there was an extra energy saving of the order of 28.77% in average.

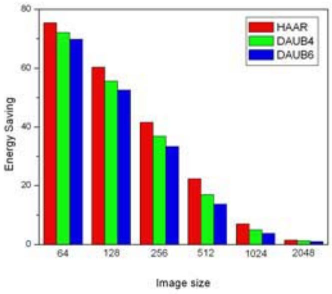


Figure 7. Energy saving for different size images.

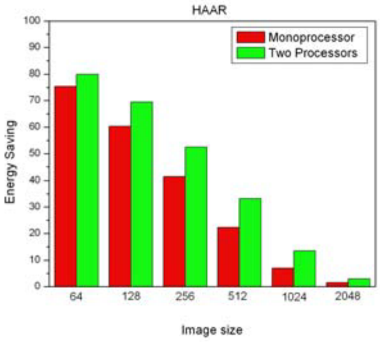


Figure 8. Energy saving for two threads.

6 Conclusions

A method of optimization for the discrete wavelet transform applied to images running in a processor of variable speed is being proposed in this article. The problem is presented as a linear problem with discrete constraints. The proposed approximate solution is based on the Bellman equation. An energy saving of the 44.19% is obtained for an image. Comparing the execution times among the different filters it is observed that the Villa1 filter is the worst and that the Haar filter is the best, but approaching the fifth level, the execution time tends to zero for all the filters. Taking into account only the three best filters, the execution time for different image sizes was obtained; thus observing that for small image sizes, the smaller execution time was obtained. The execution time for other images was also obtained and the results lead to conclude that these times are equal for any type of image of the same size. The energy saving for different image sizes was obtained, thus concluding that a greater saving energy occurs for small image sizes, whereas for a bigger size, a smaller energy is saved. Finally, when executing the algorithm in a shared memory environment, an extra saving of the order of 28.77% in average was obtained.

References

[1] A. Chandrakasan, S. Sheng and R. W. Brodersen, "Low-power CMOS digital design", IEEE Journal of Solid State Circuirs, 27, pp. 473-484, April 1992.

[2] B. Lee, E. Nurvitadhi, R. Dixit, C. Yu, and M. Kim, "Dynamic Voltage Scaling Techniques for power efficient video decoding", Journal of Systems Arquitecture, pp. 633-652, Available online 18 April 2005.

[3] Charles E. Jacobs, Adam Finkelstein, and David h. Salesin, "Fast multiresolution image querying", In Proceedings of SIGGRAPH 95, pages 277-286, ACM, New York 1995.

[4] Debora Berman, Jason Bartell, and David Salesin, "Multiresolution painting and compositing", In Proceedings of SIGGRAPH 94, pages 85-90, ACM, New York, 1994.

[5] D. Son, C. Yu, and H. Kim, "Dynamic Voltage Scaling on MPEG Decoding", International Conference of Parallel and Distributed Systems (ICPADS), June 2001.

[6] E. Nurvitadhi, B. Lee, C. Yu, and M. Kim, "A Comparative Study of Dynamic Voltage Scaling Techniques for Low-Power Video Decoding", International Conference on Embedded Systems and Applications (ESA), Jun 23-26, 2003.

[7] Ingrid Daubechies, "Orthonormal bases of compactly supported wavelets", Communications on Pure and Applied Mathematics, 41(7): 909-996, October 1988.

[8] J. Pouwelse, K. Langendoen, R. Lagendijk, and H. Sips, "Power-Aware Video Decoding", Picture Coding Symposium (PCS'01), Seoul, Korea, April 2001.

[9] J. Pouwelse, K. Langedoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor", 7th ACM Int. Confe, on Mobile Computing and Networking (Mobicom), pp. 251-259, Rome Italy, July 2001.

[10] M. Fleischmann, "Crusoe power management reduces the operating power with LongRun", in Hot Chips 12, Aug. 2000.

[11] M. Mesarina and Y. Turner, "Reduced Energy Decoding of MPEG Stream", ACM/SPIE Multimedia Computing and Networking 2002 (MMCN'02), San Jose CA, 18-25 January 2002.

[12] R. Devore, B. Jawerth, and B. Lucier, "Image compression Through wavelet transform coding", IEEE Transactions on Information Theory, 38(2):719-746, March 1992.

[13] Santa-Cruz, D, T. Ebrahimi, J. Askelöf, M. Larsson and C. A. Christopoulos, "JPEG 2000 Still Image Coding Versus Other Standards," *Proceedings of SPIE* 89 *45th Annual Meeting, Applications of Digital Image Processing XXIII*, Vol. 4115, San Diego, CA, July 30-August 4, 2000, pp. 446-454.

[14] Stephane Mallat, "A theory for multiresolution signal decomposition: The wavelet representation", IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7):674-693, July 1989.

[15] Subramanya, S.R., "Image Compression Techniques," *IEEE Potentials*, Vol. 20, No. 1, February/March 2001.

[16] Topiwala, P.N., Wavelet Image and Video Compression, Kluwer Academic Publishers, 1998.

[17] Trader Michael, "Embedded Real Time Techniques for Calculating CPU Utilization", Spring Embedded Systems Conference, Course ESC-449, San Francisco, USA, March 2004.

[18] Villasenor, J., Belzer, B. and J. Liao, "Wavelet Filter Evaluation for Image Compression", *IEEE Transactions on Image Processing*, Vol. 4, No. 8, pp. 1053-1060, August 1995.

[19] Welstead, S, Fractal and Wavelet Image Compression Techniques, SPIE Optical Engineering Press, 1999.

Novel Hardware Architecture for a Digit Serial Multiplier GF(2^m) for Cryptographic Applications

Mario A. García-Martínez[1] , Carlos Tejeda-Calderón[1] and Rubén Posada-Gómez[1]

[1] División de Estudios de Posgrado e Investigación, Instituto Tecnológico de Orizaba, Orizaba
Veracruz, México.
{marioag1955, vctejeda}@yahoo.com.mx, pgruben@yahoo.com

Abstract. This paper presents the implementation in FPGA (*Field Programmable Gate Array*) of a digit-serial multiplier that operates efficiently over finite fields *GF(2^m)*. Arithmetic operations on *GF(2^m)*, specially multiplication, are fundamental for cryptography, error control coding and digital signal processing. Design and construction of efficient architectures to perform this arithmetic operation is of great practical concern. Bit-serial architectures for *GF(2^m)* multiplication are very efficient for space complexity, but they are not suitable for time complexity; in the other way, bit-parallel architectures compute this operation in a single clock cycle, but they require a great amount of physical chip-area. In digit-serial multiplication, groups of bits known like *digits,* are processed in a single clock cycle. This allows us to design the circuit in a rank of time and space efficiencies which can be defined by the selected digit size. We have constructed a digit serial multiplier that operates in the field *GF(2^{239}).* It is a field recommended by the NIST (*National Institute of Standard Technology*) for Elliptic Curve Cryptosystems (ECC). We have used the tools of computational package ISE Version 8.1i of Xilinx for our design: VHDL (*Hardware Description Language*) to describe the circuit, and ModelSim for the simulations of the multiplier, which has been implemented in a FPGA Spartan 3 in a card prototype of Digilent.

Keywords: *FPGA, digit size multiplication, finite fields, cryptography.*

1 Introduction

Arithmetic operations over finite fields *GF*(*2^m*) are widely used in cryptography, error control coding and signal processing. In particular, multiplication is specially relevant since other arithmetic operators, such as division or exponentiation, which usually utilize multipliers as building blocks. Hardware implementation of field multiplication may provide a great speedup in procedure's performance, which easily exceeds the one observed in software platforms.

We represent a finite field with *q* elements as *GF(q)*. For computational applications the fields of extension *GF(2)*, represented by *GF(2^m)* are very important due to his possible representation by digital logic. Representation of field elements has a fundamental importance to determine the efficiency of arithmetical architectures

A. Gelbukh, S. Suárez, H. Calvo (Eds.)
Advances in Computer Science and Engineering
Research in Copmuting Science 29, 2007, pp. 295-303

Received 16/06/07
Accepted 03/07/07
Final version 24/10/07

to compute the basic operations on the field. There are different basis to represent the elements of the field *GF(2^m)*, for example: polynomial or standard basis [1], normal basis [2] and dual basis [3]. Use of certain basis determines a particular type of algorithms and architectures for *GF(2^m)* multiplication, associated with time and space complexity of the circuit. Considering *GF(2^m)* like a vector space on *GF(2),* the elements of the field can be seen like vectors of *m*-bits. In this situation, the arithmetic operation of sum is relatively little expensive, whereas the multiplication is the most important and one of most complex. One of the main application area of the digit-serial multipliers on finite fields *GF(2^m)* is cryptography. Nowadays, these systems require a great efficient performance in speed, area requirements, power consumption and security.

Generally, software implementations of arithmetic operations on finite fields require great resources of calculation and great amounts of memory, which affect the performance of a calculation system. Due to this recently we found in the state-of-the-art several proposals of hardware implementations of such operators [4], [5], [6], [7], [10]. We presented here the FPGA implementation of a digit serial multiplier that operates over the field *GF(2^{239})*, that is a field recommended by the NIST (*National Institute of Standard Technology*) for Elliptic Curve Cryptosystems (ECC). We used reconfigurable devices like FPGA´s, due to its characteristic of reprogramming, which allows to a greater facility in verification and redesign process.

2 Algorithm Description

Finite field multiplication of two elements *A* and *B* in *GF(2^m)* to obtain a result *C = A*B mod p(x)* (where *p(x)* it is the irreducible polynomial) can be made with different logical architectures: serial, parallel or digit serial. Digit serial multiplication algorithm introduced in [4] for binary fields *GF(2^m)*, is very efficient in area consumption, time and power, and we have use it in this work. Several coefficients of the multiplying *B* are processed at the same time. The number of coefficients that are parallel processing is named the *digit size*, and it is defined as *D*. Let *d= [m/D]* be the total digit number.

Let *A,B* be:

$$A=\sum_{j=0}^{m-1} a_j\alpha^j\ ,\ \ B=\sum_{i=0}^{d-1} B_i\alpha^{D_i}$$

Where
$$B_i=\sum_{j=0}^{D-1} b_{D_{i+j}}\alpha^j\ ,\quad 0\le i\le d-1 \tag{1}$$

$$C\equiv AB \bmod p(x)=A\sum_{i=0}^{d-1} B_i\alpha^{D_i} \bmod p(x) \tag{2}$$

$$\begin{aligned} C &\equiv [B_0 A + B_1(A\alpha^D \bmod p(x)) \\ &+ B_2(A\alpha^D\alpha^D \bmod p(x)) + ... \\ &+ B_{d-1}(A\alpha^{D(d-2)}\alpha^D \bmod p(x))] \bmod p(x) \end{aligned} \quad (3)$$

Then, we present the next algorithm for multiplication.

Digit Size Multiplier Algorithm:

```
==============================================
Input: $A = \sum_{j=0}^{m-1} a_j \alpha^j$ , where $a_i \in GF(2)$ and $B = \sum_{i=0}^{[\frac{m}{D}]-1} B_i \alpha^{D_i}$

     where $B_i$ is defined in equation 1.

Output: $C \equiv A * B = \sum_{i=0}^{m-1} c_i \alpha^i$ where $c_i \in GF(2)$
1. $C \leftarrow 0$
2. for $i = 0$ to $[\frac{m}{D}] - 1$ do
3.     $C \leftarrow B_i A + C$
4.     $A \leftarrow A\alpha^D \bmod p(x)$
5. end for
6. Return $(C \bmod p(x))$
==============================================
```

3 Digit Serial Multiplier Architecture

The digit size multiplication of *A(α)* and *B(α)* over finite fields is an operation more complex than addition and requires 3 steps for its calculation: [4] [9] [10].

- A polynomial multiplication
- A main reduction operation module the irreducible polynomial.
- A final reduction operation module the irreducible polynomial.

Figure 1 shows the architecture of digit serial/parallel multiplier traced from *LSD-First* algorithm in [4]. This architecture is also called single accumulator multiplier (*SAM*) since it uses a polynomial multiplication that is the multiplier core.

These architectures are widely used in hardware implementations for cryptographic applications. As you can see, the entrance polynomials *A* and *B* are 163 bits polynomials. They are introduced to multiplier core where the partial products and sums are computed. This operation is defined as $C= B_i*A+C$. Later, the main reduction $A = A*\alpha^D \mathit{mod}\ p(x)$ occurs, and finally the reduction operation *C mod p (x)* is made. Polynomial multiplication circuit (*multiplier core*) computes the intermediate results (partial additions and products) and stores them in the accumulator *C*.

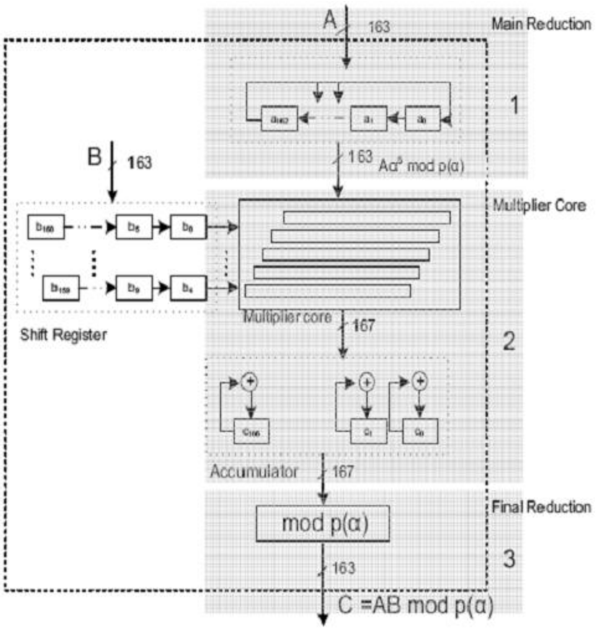


Figure 1. Digit multiplier architecture using a digit size (*D*=5) for *GF* (2^{163}).

In this operation are obtained *m* columns and *D* rows in each clock cycle. Figure 2 shows the structure of the multiplier core (step 3 of the algorithm).

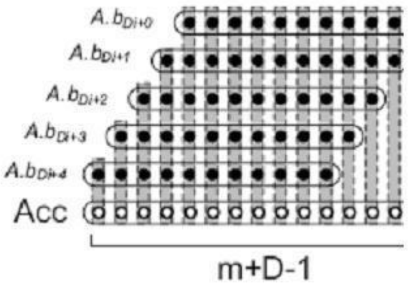


Figure 2. Multiplier core using a digit size (*D*=5) for *GF* (2^{163}).

Function of *main reduction circuit* is to shift *A* left by *D* positions and to reduce the result mod *p*(*x)* (step 4 of the algorithm). The figure 3 shows the structure of main reduction circuit.

The *final reduction circuit* reduce the contents in the accumulator to get the final result *C* (step 6 of the algorithm). Figure 4 shows the structure of final reduction. The figures 2, 3 and 4 denotes an AND gate with a black dot and a XOR gate as a vertical line between two black dots.

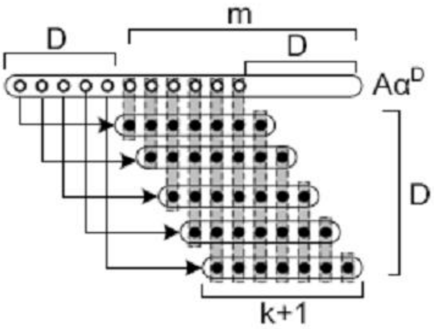


Figure 3. Main reduction circuit using a digit size (*D*=5) for *GF* (2^{163}).

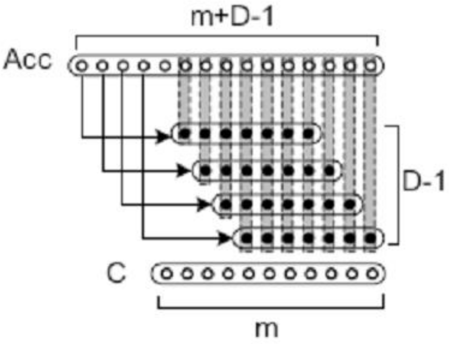


Figure 4. Final reduction circuit using a digit size (*D*=5) for *GF* (2^{163}).

4 Implementation of Digit Serial Multiplier in FPGA

All tests and measurements were made in a prototype card of Digilent that contains a FPGA SPARTAN3: XC3S200-FT256. The FPGA contains 200,000 gates, 960 cell logic blocks (CLBs) and 1,920 slices.

4.1 Previous calculations of space and time complexities

The space complexity based on the number of logic gates is shown in Table 1.

Table 1. Space complexity of the digit multiplier.

Irreducible Polynomial	Area Complexity (Gates)
General	$(m+k)D$ XORs + $(m+k+1)D$ AND*s*

In table 2 is shown the space requirements for a finite field $GF(2^{239})$ that use the irreducible polynomial $P(x)= x^{239} + x^5 +1$.

Table 2. Space complexity of the digit multiplier that operates in the finite field $GF(2^{239})$ for different values of D.

m	*D*	***Space complexity***	
		Gates	**Slices**
239	5	2445	1222
239	10	4890	2445
239	30	14670	7335
239	60	29340	14670

We have determined the time complexity previously making the following considerations:

Frequency of FPGA that is handled by prototype card SPARTAN 3 is 50 *Mhz*.

Then:

f = 50 *Mhz* ; T= 1 / 50 MHz = 0.02 *µseg in one clock cycle*
T_{LSDE} = 0.02 *µseg * clock cycles (d= Total digits or clock cycles)*
Where: *d = m/D clock cycles*

Example of time using the digit multiplier over a finite field $GF(2^{239})$.

Digit Multiplier with m=239 bits and D=20
$d=m/D$ = 12 *digits or clock cycles*
T= 0.02 µseg * 12 = **0.24 *µseg***

Figure 5 shows the time complexity in microseconds using the finite field $GF(2^{239})$ for different values of D. The frequency used is 50 *MHz*, with a T = 0.02 *µseg*.

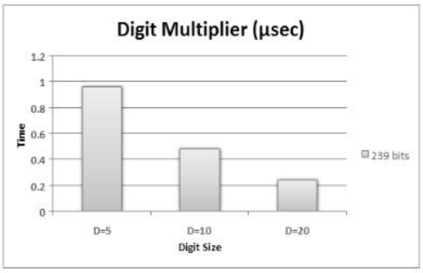


Figure 5. Time complexity in *µseg* for digit serial multiplier for different values of *D*.

4.1 Implementation Results

Figure 6 shows the area complexity (*slices*) reported by the synthesis tool for different values of *D*. We can observed that for *m*=239 bits the space of FPGA is used almost in its totality with a value of *D*=5, that it requires 1,918 slices of a total of 1,920 slices from the FPGA.

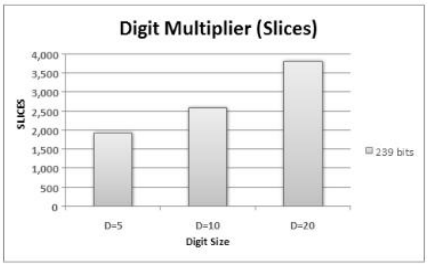


Figure 6. Area complexities (*slices*) for digit serial multiplier in FPGA.

In the figure 7, we presented the time complexity according to the clock cycles.

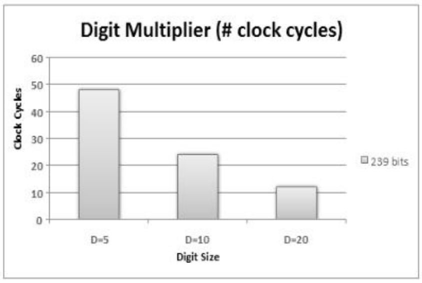


Figure 7. Time complexity of the digit serial multiplier according to the clock cycles for different values in *D*.

4 Comparison with Others Architectures

We compared our multiplier with some multipliers proposed in [7], [9], [10].

Table 3. Time complexity for Software/Hardware implementations reported in [7].

Implementation	sw/hw	*m*	Mult.	Platforms
López (1999)	SW	162	10.5mS	UltraSparc 300 Mhz
Savas (2000)	SW	160	18.3µS	Micro ARM 80 Mhz
Rodríguez (2000)	SW	163	5.4µS	Pentium II 450 Mhz
Rosner (1998)	HW	168	4.5mS	FPGA-XC4052 16 Mhz
Orlando (1999)	HW	167	0.21mS	FPGA-XCV400 76 Mhz
Lee (2000)	HW	192	2.88µS	*Not implemented*
García (2004)	HW	239	3.1 µS	Virtex-300 (75 Mhz)

Table 4. Time complexity for digit serial architectures reported in [9].

m=167	Digit Size	Clock (Mhz)	Montgomery (msec)
	4	85.7	0.55
	8	75.5	0.35
	16	76.7	0.21

Table 5 Time complexities of digit multipliers reported in [4].

Digit Size D=16	Field *m*	Platform	Time (msec)
	155	VLSI 40 Mhz	3.9
	155	Xilinx FPGA XC4020XL, 15 Mhz	18.4
	113	Xilinx FPGA XCV300, 45 Mhz	3.7
	155	VLSI, 66 Mhz	5.7
	167	Xilinx FPGA XCV400E, 76.7 Mhz	0.21

In the presented tables, we can observe a greater efficiency in operation time of our multiplier compared with the results reported in the state of the art. The digit multiplier that we have presented computes a multiplication in a time of 0.24 *μseg* using a digit size D=20 for a finite field $GF(2^{239})$.

4 Conclusions

We have presented the implementation in a FPGA Spartan3 of Xilinx, of a digit serial multiplier that operates in the field $GF(2^{239})$ and that uses an irreducible polynomial $P(x) = x^{239} + x^5 +1$, which are values suggested by the NIST for cryptographic applications of elliptical curves ECC. Has been shown that with the selection of the digit D, can be obtained an efficient implementation in the FPGA considering the time and space complexities that are required for specific applications. A direct application of our multiplier will be the construction of a cryptoprocessor for Elliptic Curve Cryptography, considering that is an important component for several systems that they require of a great performance in speed, area, power consumption and security.

Acknowledgments. We thank the suggestions of anonymous referees that helped us to improve the final presentation of this paper.

4 References

1. Mastrovito, E.D.: VLSI Architectures for Multiplication Over Finite Fields GF(2^m). Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes. Proc. Sixth Int Conf., AAECC-6, New York: Springer-Verlag, Roma, pp. 297-309, July 1988.
2. Omura,J and Massey, J.: Computational Method and Apparatus for Finite Field Arithmetic. U.S. Patent Number 4,587,627, May 1986.
3. Fenn, S.T.J., Benaissa, M., Taylor, D.: GF(2^m) Multiplication and Division Over the Dual Basis. IEEE Trans. Computers, vol.45, no.3, pp. 319-327, March 1996.
4. Song, L., Parhi, K.: Low Energy Digit Serial Parallel Finite Field Multipliers. Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis. 1997.
5. Kumar, K., Wollinger, T.: Optimized Digit Multipliers for Elliptic Curve Cryptography. Communication Security Group (COSY). Ruhr-Universitaet Bochum, Germany, 2005.
6. Paar, C.: A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields. IEEE Trans. Computers, vol. 45, No. 7, pp. 856-861, 1996.
7. García-Martínez, M.A.: Construcción de operadores básicos sobre Campos Finitos $GF(2^m)$. PhD Tesis . Cinvestav, IPN. México D.F. December, 2004.
8. Orlando, G.: Efficient Elliptic Curve Processor Architectures for Field Programmable Logic. ECE Dept. Worcester Polytechnic Institute, Germany. 2002
9. Paar, C.: Reconfigurable Hardware in Modern Cryptography. ECE Dept. Worcester Polytechnic Institute, Germany. 2006.
10. Baz, E.: Implementación en Hardware Reconfigurable de Multiplicadores sobre Campos Finitos GF(2^m). Master Tesis. División de Estudios de Postgrado. Instituto Tecnológico de Orizaba. Orizaba Ver. December, 2006.

Using a Temporization Technique for Page Encryption on Secure Processors

Osvaldo Espinosa Sosa, Luis Villa Vargas and Oscar Camacho Nieto

Center for Computing Research CIC-IPN, Av. Juan de Dios Batíz, esq. Miguel Othón de Mendizábal, México, D.F., 07738. México
espinosa@cic.ipn.mx , lvilla@cic.ipn.mx, oscarc@cic.ipn.mx
Phone: +(55)57296000 ext. 56519

Abstract. At the present days, there are many people making great efforts to improve security in computer systems with the aim to reduce problems caused by hardware attacks, viruses and intruders as well as piracy of software. Several techniques have been proposed in order to encrypt information allocated in main memory in a way that only the microprocessor can access such information through encryption and decryption processes. This work proposes utilization of a system with temporization to encrypt and decrypt pages of main memory at regular intervals of time, in this way every page encryption permits key change with the main objective of improve security in the system. The main idea is that security system causes the minor possible degradation in microprocessor performance. In this paper is analyzed the effect on performance of an encryption system in a superescalar microprocessor. This work shows that making an adequate selection of period of time and memory page size can be incremented the security level in a high grade without affecting performance beyond 10% compared with the reference architecture. Two types of reference systems are studied: the first one using Direct Encryption Mode and a second one using Counter Mode.

1 Introduction

At the present time is evident that software industry is having billionaire loses due to illegal duplication of application programs, that is commonly called piracy [1]. At the same time, new attacks constantly appears and are produced by malicious software that take advantage of operating systems vulnerabilities and hardware weaknesses in computing systems, especially memory system. In order to reduce these problems, several techniques have appeared at microprocessor level [3]. In these techniques the microprocessor is the only entity authorized to access to information, any other hardware component is considered vulnerable to the attacks due to the fact that anybody can be monitoring the information flowing through the buses [4]. Programs are then stored in memory in an encrypted way and only can be decrypted on chip, taking into account that actual processors contain the first two levels of cache memory in the same silicon die that the microprocessor, this means in the same integrated circuit.

Received 10/07/07
Accepted 19/10/07
Final Version 24/10/07

The intention of this paper is to show the effect in the performance of a superescalar microprocessor due to the inclusion of an encryption and decryption system including the capacity to encrypt pages of main memory at regular intervals of time, changing on every page encryption the key to be utilized with the purpose of increase the system security. It is clear that this encryption system will add more latency to the main memory access and this will affect the overall processor performance. It is important to find a trade off between the period of time and the size of memory pages to be encrypted in such a way that performance will not be affected severely and it can offer an adequate security level. The paper contains the methodology used to evaluate the proposal included in this work which uses an execution driven simulator to obtain detailed statistics of realized experiments and finally we have conclusions and bibliographic references.

2 The Memory Encryption Process

On previous work it has been proposed to encrypt data contained in main memory to offer a good security level against attacks [2]. The encryption and decryption system is usually inserted between level 2 of memory cache and main memory due to it is the place where processor performance is degraded in a minor quantity and because of the fact that levels one and two of cache memory normally reside on chip, it serve as a bus interface with an insecure external world. When the processor performs a read operation to main memory, the data obtained must be decrypted to be used by the processor; likewise, when a write operation to main memory is performed the data must be encrypted before.

There are two approaches to do this: the first one is called *Direct Encryption Mode* where the encryption/decryption engine is placed serially between main memory and the second level cache. It encrypts and decrypts data moving between level 2 of cache and main memory. This encryption mode has the characteristic of exhibit the whole latency of encryption system and then an access to main memory increments its normal latency adding the encryption engine latency resulting in a higher total latency. The second approach is called *Counter Mode.* Unlike Direct Encryption system it does not need to wait until data arrives from memory (it does not work serially), instead it encrypts information known yet at the memory access moment such as the address and/or a counter value and encryption can be done in parallel with memory access. Once data and encrypted information are obtained then these are XOR'ed to produce something called data pad and the encryption engine latency is hidden with the memory access latency. Both cases are shown in fig. 1. We can notice the minor total latency of Counter Encryption Mode.

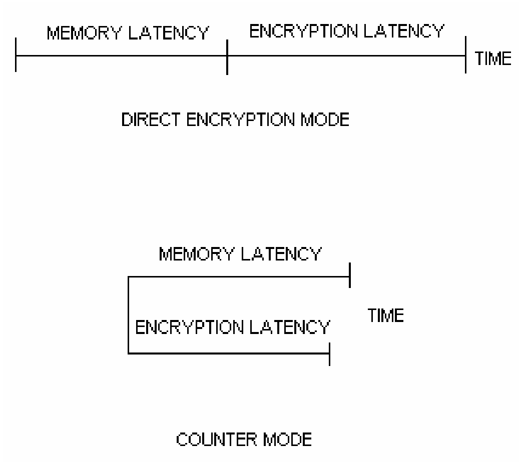


Fig 1. Two approaches for encryption/decryption

For encryption/decryption is usually utilized the AES algorithm (Advanced Encryption Standard) which is an algorithm of fixed parameters. The AES requires as input a data block of 16 bytes in order to encrypt a cache memory line of 64 Bytes, four AES blocks are required as shown in figure 2. The encryption or decryption process can use always the same key, which represent a vulnerability due to the possibility that algorithm can be broken by an expert intruder. Other option is to change the key after certain number of encryptions and decryptions, however when the key is replaced the main memory must be re-encrypted causing that system could be stopped a large interval of time (in the order of seconds) in a system working at frequencies in the order of Ghz. To improve security in a computer system without important performance degradation we propose a system where periodically keys are replaced using a pull of keys, even more we can use different keys for each memory page, as we are going to explain in the next section.

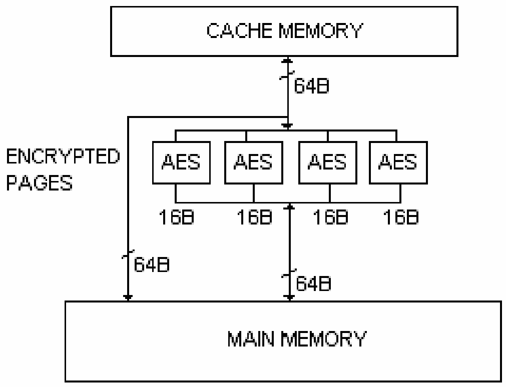


Fig. 2. Encryption and decryption circuit detail.

3 Proposed Architecture

The encryption system is shown in figure 3. The main memory is divided in pages of fixed size. There is a *timer* that activates the key replacing mechanism periodically at regular intervals of time. Keys are generated in a random form. Every time this mechanism is activated a memory page is decrypted using the old key and re-encrypted using the new key. Afterwards, the encrypted information is sending back data to main memory.

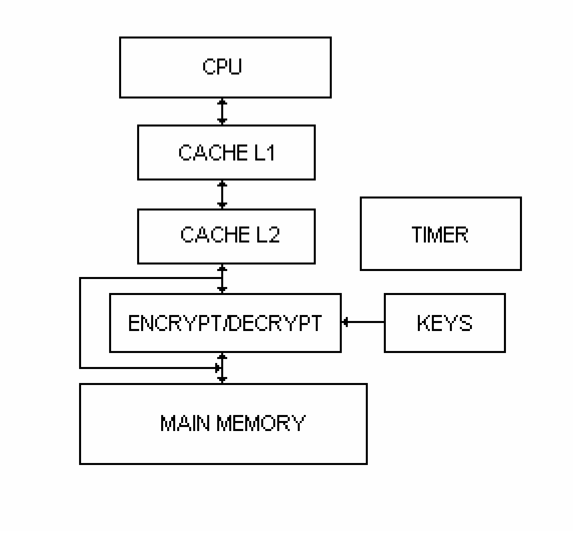


Fig. 3. Proposed architecture.

The next time (after a period of time) a new page will be encrypted using the same process and following a Round Robin scheme to select the page in main memory using a different key and in this way offer a higher level of security in comparison with proposals that use only one key. Using this new page encryption model, the security level increases without an important performance degradation. In order to know what memory pages are active on memory, the processor has a special group of page registers working together with operating system which is the responsible of resource management (memory in this case). The processor includes a set of special registers to store old keys because of the fact that these are used to decrypt memory pages before it use the new one to encrypt data and send it back to memory.

It is important to mention that when a memory page is re-encrypted, information pass through the encryption circuit and then data is given back to main memory without affecting cache memory. This process is mainly affected by memory page reads because data must be read before it can be used unlike memory page write accesses which are more flexible in the time of execution of an operation.

4 Methodology

We have evaluated our model using the simplescalar 3.0 simulator tool set which performs a very detailed simulation of a superescalar processor with out of order execution [5]. The simulator was configured as an Alpha 21264 because this architecture has been considered the best superscalar processor at its time of appearance. This processor contains two level 1 cache memories (Instructions and data) of 64 KB 2-way associative with 64 byte blocks. The second level of cache is unified with size of 1 MB being 8-way associative with 64 byte block. The simulator has as input a set of application programs called benchmarks; we used especially the SPEC CPU 2000 suite which is composed of twelve applications of fixed point and fourteen floating point programs. In this work the performance is monitored whenever we change the page size or the period of time for page re-encryptions. Simulations was made executing 5 X 10^8 instructions skipping the first 1 X 10^8 instructions with the aim of eliminate initialization effects on statistics. Results are shown in terms of IPC average of the 26 SPEC CPU programs.

5 Evaluation

We can take a look first at the case with Direct Encryption Mode. As we can see on fig. 4 the average performance for all applications decrease when we add the page encryption scheme in relation to the system that only uses encryption for normal read and write accesses (this latter case corresponds to 100% on performance, this is our baseline model). The performance reduction is higher if we reduce the interval of time for page re-encryption (number of cycles between page re-encryptions).

If the page size is constant (4 KB) and considering that microprocessor frequency is 1 Ghz, the figure 4 shows that best results are obtained for configurations of 1 X 10^6 cycles (1 ms) and 1 X 10^5 cycles (0.1 ms) between re-encryptions, in both cases performance is degraded only 0.7% and 6.95% respectively. In the previous experiments we consider acceptable a result if performance is not degraded beyond 10%. Using these two best configurations, we vary the page size to see the effect on the average performance.

Figure 5 shows the average results for configurations of pages of 4 KB, 8 KB, 16 KB y 32 KB with an interval of 100,000 cycles between page re-encryptions. We can notice that pages of 4 KB are suitable because performance is diminished only 6.95% and increasing page size the performance is degraded in a major form, in this case re-encryption of pages is performing 10,000 times in a second with the implication of an important improve on security and an acceptable reduction in performance.

It is shown on fig. 6 the result of use a period of time equivalent to 1,000,000 cycles where we can see an improvement on performance respect to the previous case in which it is possible to increase the page size with a minor reduction on performance.

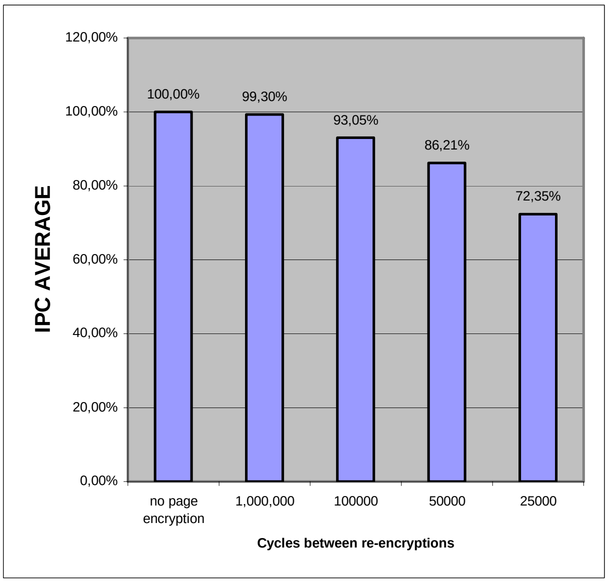


Fig. 4. Page encryption at different periods

It is shown that pages could be up to 32 KB without degrading performance beyond 6% of the average performance.

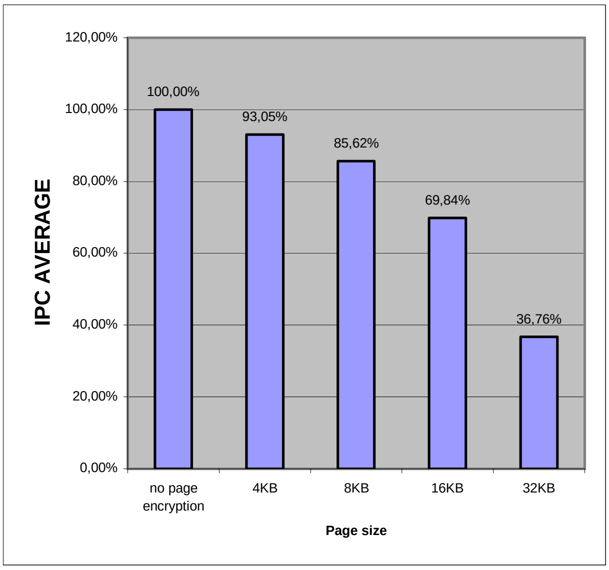


Fig. 5. Effect of page size (100,000 cycles of period)

It must be considered that in the previous case only 1000 pages will be re-encrypted but in contrast this scheme permits to reduce the total quantity of keys to be used by the system 8 times (if we use pages of 32KB). Now, we can analyze the case for Counter Mode Encryption. Results obtained for a constant 4 KB pages and different periods of time are shown on fig. 7.

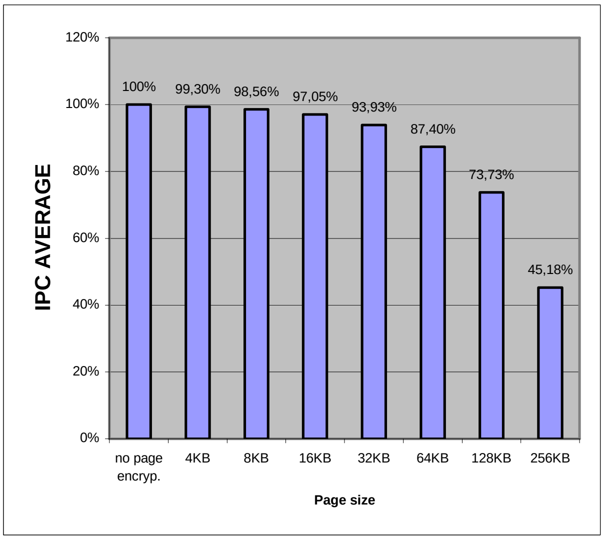


Fig. 6. Effect of page size (1000000 cycles of period)

Results can show that with 4 KB pages in Counter Mode Encryption inclusive with periods of 50,000 cycles, results are tolerable, lower than 10% on average in performance degradation.

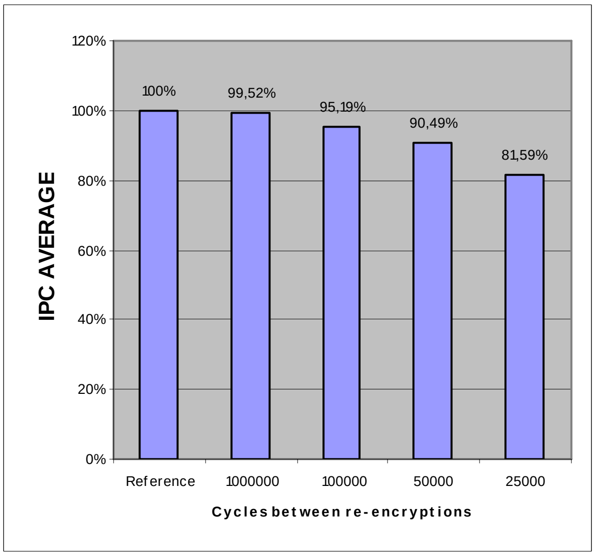


Fig. 7. Page encryption at different periods

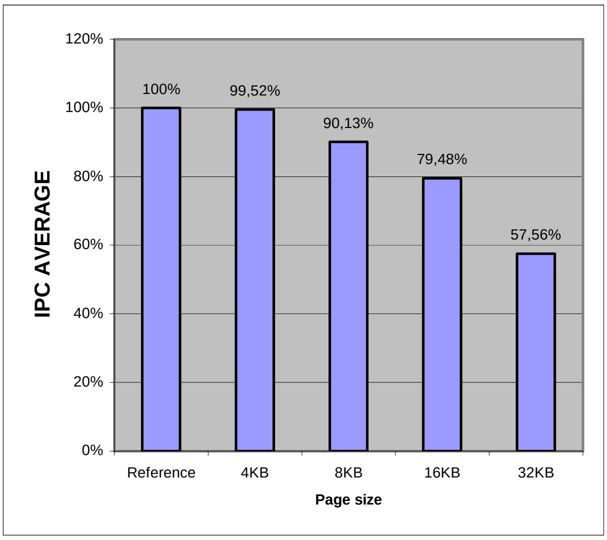


Fig. 8. Effect of page size (100,000 cycles of period)

The fact is due to a minor memory latency that experiments the system with this encryption mode and then it supports more page re-encryptions. Although the average in performance losses is acceptable for three cases: 1000000, 100000 and 50000 cycles between re-encryptions, we can take into account that for 50000 cycles there are some particular programs which experiments losses of 15% on performance. Compared with Direct Encryption Mode this system could tolerate memory latency better.

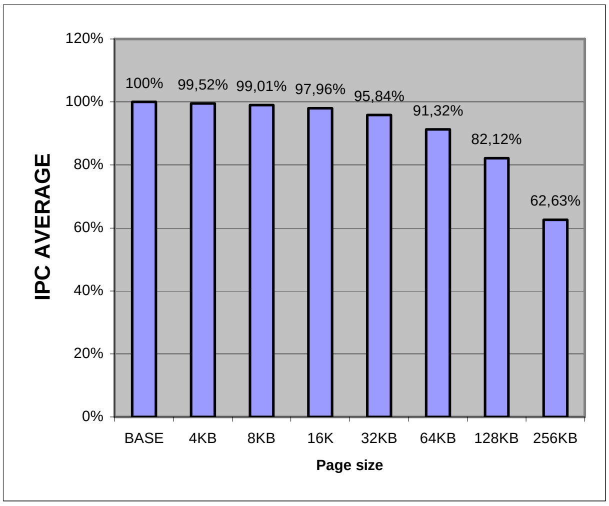


Fig. 9. Effect of page size (1,000,000 cycles of period)

If we take the period of 100,000 cycles as a constant and we vary the page size, it is clear that we can grow up the page size up to 8 KB with acceptable results and it is important to notice the fact that in that case are necessary a minor number of keys (two times) with page re-encryption every 0.1 ms. Fig. 8 contains the exposed information in a graphic form. Again, data obtained with periods of 1000000 cycles and pages of 4 KB indicate a better system performance as it is shown in fig. 9. We can notice that page size could be increased up to 64 KB having a great performance and reducing the number of keys sixteen times (pages of 64 KB). In the sane manner, it is important to see that the greater the number of keys the greater is the security but more resources are needed and performance is affected. In contrast, the greater the page size the lower is the security.

6 Conclusions

In this work it was studied the effect of the inclusion of page encryption using a temporized scheme. It was shown that it is possible to increase the security level of processors when they include capacities of change encryption keys with periodicity and even better using one key for every page instead of one unique key for the whole memory. All without affect in an important manner the processor performance. The small loss of performance is well compensated with the level of security gained. As we can see with the results of experiments, it is important to consider the period of time used by page re-encryption as well as the page size in order to obtain an adequate trade-off between security and performance. Results obtained show us that Counter Mode Encryption has a lower effect on the system but maybe it could be more easily to be attacked that Direct Encryption Mode. The reason of this is that one part of encrypted data (such as memory address can be observed on buses) could be known by an intruder. As a counterpart Direct Encryption Mode offers better levels of security with a more performance impact due to higher memory latency.

References

1. Yang,Zhang,Gao. Fast secure processorfor inhibiting software piracy and tampering. Proceedings of the 36th International Symposium on Microarchitecture MICRO 36-2003.
2. Ruby B. Lee, Peter C. S. Kwan. Architecture for protecting critical secrets in microprocessors. Proceedings of the 32nd Annual International Symposium on Computer Architecture 2005.
3. T.Kgil, L.Falk and T. Mudge. ChipLock: support for secure microarchitecures. Workshop on architectural support for security and anti-virus, 2004.
4. Chenyu Yan, Brian Rogers et. Al. Improving cost, performance and security of memory encryption and authentication. International Symposium on computer architecture ISCA 2006.
5. Burger,Dough. The simplescalar toolset, version 3.0 *Computer Architecture News*, **25** (3), pp. 13-25, June, 1997.