Research on Computing Science

Avances en la ciencia de la computación en México

Prólogo de Juan Luis Díaz de León Santiago

Alexander Gelbukh
Macario Hernández Cruz
(Eds.)

Alexander Gelbukh
Macario Hernández Cruz
(Eds.)

Avances en la ciencia de la computación en México

Con prólogo de Juan Luis Díaz de León Santiago

Research on Computing Science, vol. 2

Centro de Investigación en Computación
Instituto Politécnico Nacional
México

Editores

Alexander Gelbukh, Macario Hernández Cruz
Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Col. Zacatenco, CP. 07738, México, D.F., México.
E-mail: gelbukh@cic.ipn.mx, macario@correo.cic.ipn.mx;
http://www.gelbukh.com

Published by the Instituto Politécnico Nacional
Centro de Investigación en Computación
Av. Juan Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México DF, Mexico

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

Coordination:
Luis Hernández Lara
Salcedo Camarena, Teresa
Sandoval Reyes, Alejandro
Torres Frausto, Raquel
Vargas García, Soila
Zamora Camarena, Ma. Trinidad

The editors and publisher of this book have made their best effort in preparing book, but make no warranty of any kind, expressed or implied, with regard to information contained in this book.

All rights reserved. No part of this publication may be reproduced, stored n a retri system, or transmitted, in any form or by any means, including electronic, mechan photocopying, recording, or otherwise, without prior permission of Instituto Politécnico Nacional, except for personal or classroom use provided that copies the full citation notice provided on the first page of each paper.

ISBN 970-36-0026-3

© Instituto Politécnico Nacional, México, 2003
Printed in Mexico
Printing: 500

Prólogo

libro es el segundo volumen de la serie *Research on Computing Science*. Las ntribuciones aquí presentados fueron seleccionadas por el Comité Editorial en base arbitraje internacional estricto de entre los trabajos presentados en el 4° Congreso ıcional de Computación CORE-2003, el cual se realizó en las instalaciones del :ntro de Investigación en Computación (CIC) del Instituto Politécnico Nacional 'N), del 6 al 7 de mayo de 2003.

propósito fundamental del CORE, desde su creación en 1998, ha sido brindar a comunidad estudiantil nacional un espacio de integración académica, donde los ›irantes a maestros o doctores en ciencias de la computación cuenten con un foro ‹eñado *ex profeso*, donde puedan exponer sus desarrollos, avances en temas de tesis ›piniones sobre temas de su interés. Con este tipo de actividades, los estudiantes nen la oportunidad de desenvolverse como investigadores que presentan productos sus investigaciones y con ello, se logra un paso importante en su formación ntífica.

Quienes asisten al CORE, tienen la oportunidad de escuchar a investigadores de nivel, en conferencias donde exponen los recientes avances en las áreas de interés congreso; estas intervenciones de conocidos científicos forman un caldo de cultivo para ampliar los horizontes académico-científicos de los estudiantes ticipantes en cada una de las ediciones del CORE.

En sus orígenes el CORE estaba planeado como la celebración anual de la semana udiantil del CIC, la cual debía incluir eventos culturales, científicos y deportivos; este evento, surge el capítulo estudiantil de ACM (*Association for Computing :chinery*) en el CIC. Posteriormente, CORE evolucionó hasta convertirse en un ıgreso formal en el cual participan estudiantes e investigadores de todo el país e lusive del extranjero incluyendo, en alguna ocasión, concursos de prototipos de tware y hardware.

libro contiene más de treinta trabajos distribuidos en once secciones, a saber: :ligencia Artificial, Lógica y Razonamiento, Agentes, Interfaces en Voz y ıguaje Natural, Reconocimiento de Patrones, Procesamiento de Imágenes, Internet eguridad, Sistemas de Información, Ingeniería de Software, Ingeniería de Cómputo resentaciones cortas de sistemas.

nuestro deseo que el material presentado en este libro cumpla las expectativas os estudiantes de maestría y doctorado, cuyo trabajo involucre alguna o algunas de áreas afines a las ciencias de la computación. Pretendemos que los lectores tengan mano un diagnóstico preciso y bien documentado del sentir estudiantil respecto los avances recientes en Ciencias de la Computación y de las perspectivas de arrollo a corto y mediano plazo.

El Comité Organizador del CORE 2003 agradece la entusiasta participación autoridades del CIC-IPN, asistentes, autores, revisores y encargados de la logístic congreso. Asimismo, deseamos manifestar nuestro agradecimiento por el brindado, para el desarrollo de este trabajo, a las siguientes instituciones: Insti Politécnico Nacional, CONACyT México, ACM y RITOS-2.

Juan Luis Díaz de León Santi

Comité Editorial

lvarado Mentado, Matías (Instituto Mexicano del Petróleo)
alladares Ocaña, Leandro (CIC-IPN)
antú Paz, Erick (Lawrence Livermore National Laboratory, Estados Unidos)
oello Coello, Carlos A. (CINVESTAV)
Albornoz Bueno, Álvaro (ITESM-Ciudad de México)
la Fuente, Pablo (U. de Valladolid, España)
ecouchant, Dominique (CNRS, Francia)
de León Santiago, Juan Luis (CIC-IPN)
elipe Riverón, Edgardo Manuel (CIC-IPN)
alicia Haro, Sofía (CIC-IPN)
elbukh, Alexander (CIC-IPN)
ómez Sánchez, Eduardo (U. de Valladolid, España)
onzalo, Julio (UNED, España)
ernández Cruz, Macario (CIC-IPN)
akowiak, Sacha (U. Joseph Fourier, Francia)
mus Olalde, Cuauhtémoc (CIMAT)
pez López, Aurelio (INAOE)
artín Vide, Carlos (U. de Vigo, España)
artínez Enriquez, Ana María (CINVESTAV)
artínez Muñoz, Jorge (CIC-IPN)
ejía Lavalle, Manuel (Instituto de Investigaciones Eléctricas)
onroy, Raúl (ITESM-Estado de México)
ontes y Gómez, Manuel (INAOE)
una Gómez, César A. (Instituto Mexicano del Petróleo)
lomar, Manuel (U. de Alicante, España)
nario, Daniel (Carleton University, Canadá)
lechano Ferragud, Vicente (U. Politécnica de Valencia, España)
ñas, Anselmo (UNED, España)
rez Cortés, Elizabeth (UAM-Iztapalapa)
ieda, Luis (UNAM)
da Silva, Antonio (U. Técnica de Lisboa, Portugal)
dríguez Brisaboa, Nieves (U. De A Coruña, España)
dríguez Liñares, Leandro (U. de Vigo, España)
dríguez, Horacio (U. Politécnica de Cataluña, España)
Shulcloper, José (ICIMAF, Cuba)
Francisco (U. de Castilla-La Mancha, España)
itaolaya Salgado, René (CENIDET)
eremetov, Leonid (Instituto Mexicano del Petróleo)
orov, Grigori (CIC-IPN)
Azuela, Humberto (CIC-IPN)
ube de Lima, Vera Lúcia (U. Católica do Rio Grande do Sul, Brasil)
Succar, Enrique (ITESM-Cuernavaca)
iemykh, Andrei (CICESE)
Horebeek, Johan (CIMAT)
laseñor Pineda, Luis (INAOE)

Árbitros invitados

Bertossi, Leopoldo (Carleton University, Canadá)
Castro García, Miguel Alfonso (UAM-Iztapalapa)
González Ares, Luis A. (U. de A Coruña, España)
Insaurriaga González, Marco Antonio (U. Católica do Rio Grande do Sul, Brasil)
Loukanova, Roussanka (Dept. of Linguistics, Uppsala University, Suecia)
Martinez Barco, Patricio (U. de Alicante, España)
Pérez-Piñar López, David (U. de Vigo, España)
Prieto Guerrero, Alfonso (UAM-Iztapalapa)
Ríos Viqueira, José (U. de A Coruña, España)
Rodríguez Morales, Roberto (ICIMAF, Cuba)
Saiz Noeda, Maximiliano (U. de Alicante, España)
Serolli Pinho, Márcio (U. Católica do Rio Grande do Sul, Brasil)
Suárez Guerra, Sergio (CIC-IPN)

Colaboradores

Jiménez Hernández, Hugo (CIC-IPN)
Lugo García, Tania (CIC-IPN)
Martínez Muñoz, Jorge (CIC-IPN)
Quiané Ruiz, Jorge A. (CIC-IPN)
Rodríguez Elizalde, J. Marcelo (CIC-IPN)

Con colaboración de:

CONACyT, México
ACM
RITOS-2

Contenido

ıteligencia artificial

imulation of mobile robots in virtual environments 1
Jesús Savage, Emmanuel Hernández, Gabriel Vázquez, Humberto Espinosa, Edna Márquez

stimación adaptativa de parámetros en un sistema dinámico no lineal 14
Yuriria Cortés Poza

ONTROLINT: Sistema de inteligencia artificial aplicado a la enseñanza de la ıditoría de estados financieros 21
Guillermo Adolfo Cuéllar Mejía

:conocimiento automático de actividades humanas 32
José A. Montero Valverde, Víctor Hugo Zárate Silva

ógica y razonamiento

ıálisis de estabilidad para problemas dinámicos de satisfacción de ;tricciones 44
Manuel Iván Angles Domínguez, Hugo Terashima Marín

ıálisis de terminación de reglas, un enfoque con red de Petri 56
Joselito Medina Marín, Xiaoou Li

rative n-Literal Rule for Solving Satisfiability Problems 68
Rafael Rivera-Lopez, Olga Leticia Prado-de-la-Paz, Juan Frausto-Solis

nbiente web para la construcción, simulación y visualización 3D de tómatas finitos y lenguajes regulares 78
Claudia Noguerón, Manuel Chim, Juan Carlos Tovilla, Leandro Balladares Ocaña

ıálisis, diseño e implementación de un simulador de combate aéreo basado lógica difusa 89
Francisco Cano, Fernando Juárez, Servando Ortega, Edmundo Camarillo, Leandro Balladares Ocaña

;entes

plementación de protocolos de interacción FIPA compatibles para agentes la herramienta CAP-AgentTool 107
Ernesto Germán Soto, Rigoberto Velásquez Elenes, Omar Olivas Zazueta

tema multi-agente para control de manufactura flexible basado en la teoría restricciones 118
Jorge Martínez Muñoz, J. Juan Guerra Vásquez

erfaces en voz y lenguaje natural

tema multiagentes para el comando oral de un robot móvil 126
Oscar Pérez Márquez, Luis Villaseñor Pineda

Desarrollo de interfaces de voz para aplicaciones web.....
J. Quiané-Ruiz, J. Manjarrez-Sánchez

Estudio sobre la navegación en la Web usando voz.....
Areli Torres González, Luis Villaseñor Pineda

Procesamiento de textos electrónicos para la construcción de un corpus.....
Sofía N. Galicia-Haro

Reconocimiento de patrones

Caracterización de series de tiempo con escalamiento multidimensional.....
V. Rivera-Mancera, E. Bautista-Thompson, J. Figueroa-Nazuno

Evaluación de red neuronal artificial polinomial (RNAP) para el modelado de series de tiempo.....
E. Guzmán Ramírez, E. Bautista Thompson & J. Figueroa Nazuno

Simulación computacional de la actividad eléctrica cerebral.....
Lizette P. Castro, L. A. Torres

Procesamiento de imágenes

Arquitectura hardware para el seguimiento de objetos utilizando procesamiento piramidal.....
Marco Aurelio Nuño Maganda, Miguel Arias Estrada

La fusión de imágenes a nivel características para la detección y seguimiento de objetos.....
Janeth Cruz Enriquez, Leopoldo Altamirano Robles

Recuperación de formas 3D usando luz estructurada.....
Venustiano Soancatl Aguilar, Leopoldo Altamirano Robles

Internet y seguridad

SymWeb: Modelado de una aplicación de simulación web mediante OOHDM.....
Omar Sosa-Tzec, Víctor Menéndez-Domínguez, Luis Vargas-Villamil

Aplicando análisis de proximidad en Web-Mapping.....
Miguel F. Mata, Miguel Torres

Desarrollo de una autoridad de certificados digitales para una intranet universitaria.....
Dorian Omar González Cortés, Jesús Eduardo Torres Castañeda, Petar Zaprianov Doychev, Arturo Pruneda Martínez

Sistemas de información

Análisis y empleo de herramientas OLAP en el sistema de información corporativa de la CFE.....
Liliana Argotte, Norma Jácome, Manuel Mejía

ngeniería de software

echnical Recommendation Sheets (HRT) for Support of Software
evelopment Process 276
Gustavo Arellano, Hanna Oktaba

)iseño de un portal cultural basado en patrones de diseño para el Web 283
Plutarco Galavíz Carmona, Jaime Muñoz Arteaga,
Carmen Santiago Díaz

ıgeniería de cómputo

ıteracción háptica de alto desempeño con objetos virtuales dinámicos
eformables 294
Omar A. Domínguez Ramírez, Vicente Parra Vega

ontrol remoto universal a distancia DTMF por línea telefónica 311
Andrés Gerardo Fuentes Covarrubias,
Fermín Pascual Estrada González, Ricardo Fuentes Covarrubias

iseño y construcción de una caja de actividad motora con tecnología de
icrocontroladores para la captura y análisis de datos de ratas en
:perimentación 325
Francisco J. Heredia López, Karen Z. Navarrete Kao

lgoritmo de cálculo de la carga atendida para redes digitales multiservicio de
ında angosta 333
Julio Carlos Delgado Hernández

·esentaciones cortas de sistemas

)bot Command Center (Roc2) 338
Jesús Savage, Emmanuel Hernández, Gabriel Vázquez

·ototipo de un sistema multiagentes para el comando oral de un robot móvil 341
Oscar Pérez Márquez, Luis Villaseñor Pineda

:sualización y control de una computadora desde varios equipos: Enseñanza
·tual 344
Juan Carlos May Pat, Alejandro de Jesús Ortiz Cervera

ograma de control en lenguaje ensamblador para el módulo de control de
caja de actividad motora con tecnología de microcontroladores para
captura y análisis de datos de ratas en experimentación 346
Francisco J. Heredia López, Karen Z. Navarrete Kao

dice de autores 349

Simulation of Mobile Robots in Virtual Environments

Jesús Savage [1], Emmanuel Hernández [2], Gabriel Vázquez [3],
Humberto Espinosa [4], Edna Márquez [5]

Laboratory of Intelligent Interfaces , University of Mexico, UNAM.
[1] savage@servidor.unam.mx, [2] deyael@mezcal.fi-b.unam.mx,
[3] bowser@mezcal.fi-b.unam.mx, [4] jhespinosa@ieee.org,
[5] edna@mezcal.fi-b.unam.mx

Abstract. In this paper we describe a system to simulate and operate Virtual and Real robots (ViRBot). The system consists of several layers that control the operation of real and virtual robots. A 3D graphic engine interface allows to test multiple virtual robots, that are a close simulation of the real ones, the virtual robots are able to execute the same commands, using a robot's programming language, that real robots can, including behaviors, movement equations and sensors readings.

Introduction

ıe development of algorithms to control mobile robots it is necessary to simulate using virtual robots, in order to test their performance before implementing in the real ones. This is done because sometimes it is too expensive or unfeasi- to execute hundreds of testing operations with a real robot. Therefore, we have :loped a system, named the ViRBot, where operational algorithms for mobile ıts can be tested using virtual robots with the same conditions as if they were the ones. The main idea is to be able to control the real and virtual robots indis- :ly and to test the mobile robots algorithms using first the virtual robot and the one later. The ViRBot system allows the testing of multiple virtual robots, at same time, of different forms and sizes, multiple viewpoints, wired frame es, shaded or textured polygons, interaction with the system using different faces such as keyboard, mouse, joy pad and internet. Every part of the ViRBot developed by the authors of this document at the laboratory of Human Inter-

he virtual environment was developed using a virtual reality toolkit called WTK :e8 that gave us a powerful tool to program 3D world applications using mod- 1]. Using this tool is also possible to create boxes, tables, buildings, factories, ;eons, labyrinths and everything needed to construct different environments the virtual robots can be tested.

:lbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México,

2 Mobile Robot Operation

There are several architectures to control mobile robots, our architecture is ba the Aura architecture developed by Ronald Arkin [2] [3]. In our approach the robot's operation is formed by several layers, see Figure 1, each one having cific function that in whole control the behavior of the robot.

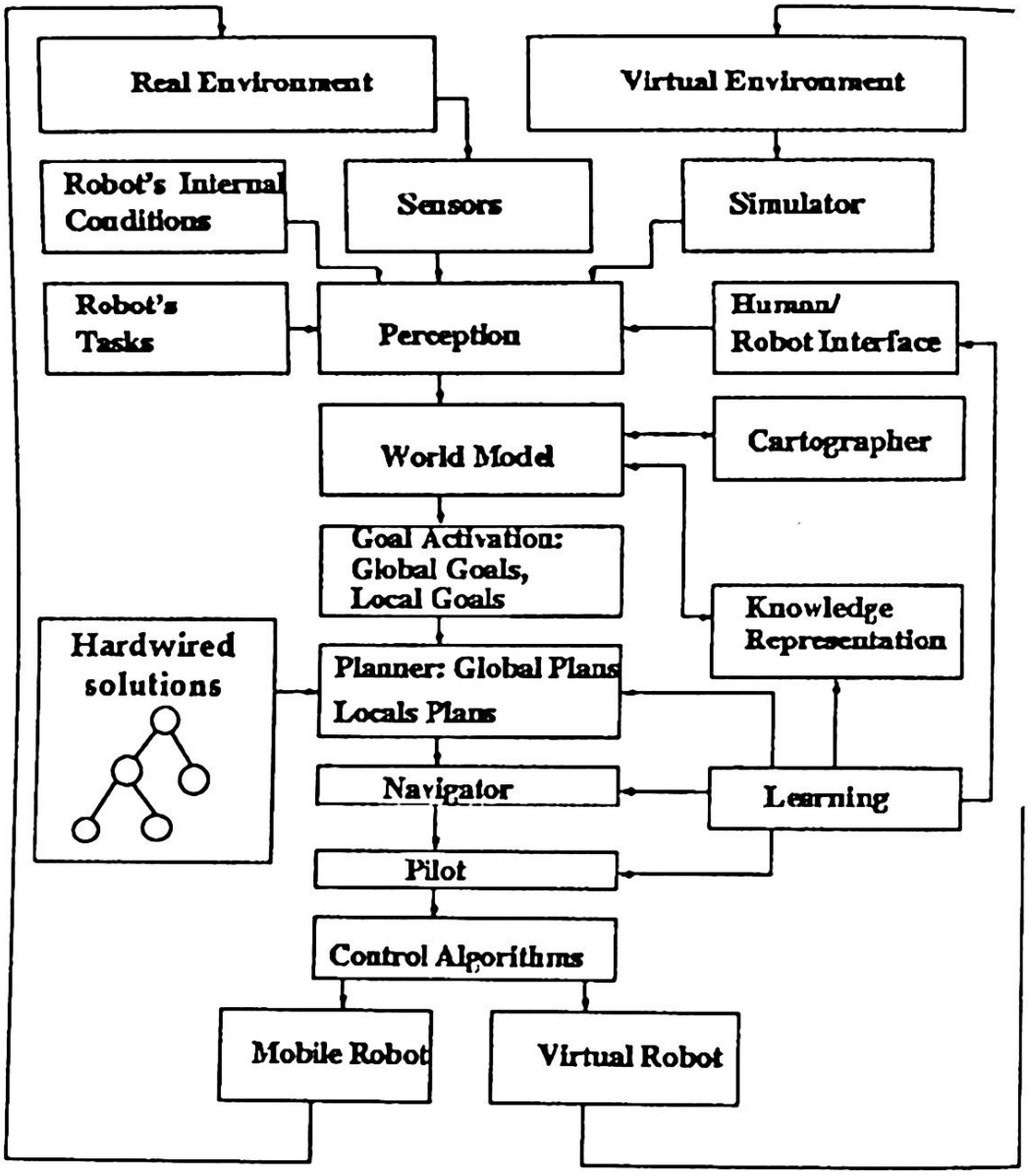


Fig. 1. Virtual and Real Robot (ViRBot) System.

Each of the layers are described briefly in the following sections with mo tails on the virtual environment and the simulator.

Perception

of the main objectives of the perception module is to obtain a symbolic presentation of the data coming from the robot's internal and external sensors, om the Human/Robot interface, from the robot's tasks and if chosen from the nulator. The symbolic representation is generated after applying digital signal ocessing algorithms on the data generated by the sensors. With this symbolic presentation a belief is generated, later with the world representation situation cognition is created.

Internal Sensors

ch of the sensors is represented by a data structure that has the following ele-nts: sensor's name, sensor's type, sensor's position in the robot and a set of the sor's values. We use a B14 robot, developed by Real World Interfaces, that it has wheel encoders and a battery level detector.

External Sensors

robot has an array of 8 infrared sensors, 8 sonar sensors, 20 tactile sensors and a leo camera [8].

Simulator

the ViRBot system, it is possible to simulate each of sensors' signals according to hysical model of the sensors. Thus, when the system is tested using the virtual ot, the simulator provides each of the sensors' values. A user may include his simulation algorithms [3]. The simulation of the sensors is explained in detail ection 3.

Human Robot Interfaces

communication between the robot and a user is done by several means: speech ognition, text provided by keyboard and control pads.

Robot's Tasks

robot is programmed to perform certain tasks, such as picking objects and de-ring them to another place, during the day at a certain time.

2.7 World Model

The world model module uses the belief generated by the perception modu together with the information provided by the cartographer and the knowledg resentation it generates a situation that needs to be solved.

2.8 Cartographer

For every room of the working environment there is a representation of how the rooms are interconnected between them, as well as each of the known ob included in them. The links between rooms are used later to form a tree that the root the origin room and one or several leafs have the destination room.

The known obstacles are defined as polygons which consist of a clockw dered list of its vertex. Representing the obstacles as polygons makes easi search for a solution to navigate from one point to another. They are represer a fact in an expert system as follows:

(polygon type obstacle' room obstacle' name x0 y0 ... xn yn)

where type represents the type of obstacle, such as a wall, a desk, etc. The xi represents the coordinates of the obstacle. From this obstacle representation den areas are created, which are areas not allowed for the robot to enter, built by growing the polygons by a distance greater than the radius of the ro consider it as a point and not as a dimensioned object [7]. Figure 2 shows bidden areas of the experiment's room. It is possible to create the config space in this way because our robot has cylindrical shape.

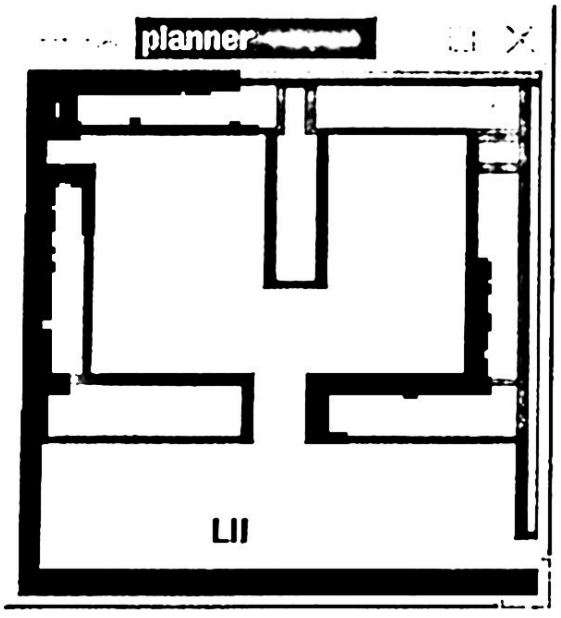


Fig. 2. Symbolic representation of the experiment's room.

Knowledge Representation

expert system is used to represent the robot's knowledge. In an expert system, knowledge is represented by rules, each one contains the encoded knowledge of expert, that is, the actions that the robot would do if certain conditions were met. The environment was defined as facts in an expert system, we used the expert stem shell called CLIPS, developed by NASA [4], in which we add a TK graphics vironment and sockets communication that allows to send data to it and to the bot through Internet.

0 Goal Activations

ven a recognized situation, a set of goals is activated in order to solve them.

Options

of hardwired procedures that solve, partially, specific problems.

Planner

basic problem of planning the robot's movements is reduced to: Given the ini- position and heading of the robot **A** in space **W**, a path τ specifying a continuous quence of positions and headings for the robot **A** must be found in order to avoid lisions with the obstacles $\mathbf{B}_i$'s, beginning in the initial position and heading and ishing in the goal position and heading. If there is not such path, the impossibility solve the problem is reported.

Planning is defined as a procedure or guide for accomplishing an objective or This requires searching a space of configurations to find a path that corre- nds to a set of the operations that will solve the problem. For example, in our tem we may want the robot to move from one place to another and to take a ture. This problem will be solved by finding a sequence of operations that leads m an initial state to a goal state. Then the objective of planning is to find a se- nce of physical operations in order to achieve the desired goal.

The robot's planning organization consists of several hierarchical layers. The top er is the planner, which takes as input an initial and a final state and an environ- nt description, and produces an overall plan that will lead the robot from the ial to the final state. If the command was to move the robot from one room to ther, the planner would find the best sequence of movements between rooms the robot reaches its destination. Inside each room it also finds the best path sidering the known obstacles, that represent some of the objects in the room. h of the objects in the environment is represented by a set of polygons whose are by the planner, in order to find the best path avoiding those that interfere with goal.

2.13 Navigator

The navigator takes the overall plan and finds a set of trajectories that cons distances and angles that the robot needs to execute in order to reach its destin Given a set *(x1, y1, x2, y2...xn, yn)* of points obtained by the planner and a tim visit them, the navigator finds the angle of rotation θi, a distance *di* and a speed reach all of them.

2.14 Pilot

The pilot takes the trajectories generated by the navigator and executes them. cally, it has to move the robot a distance D_i and turn θ_i radians in each step i search for unknown obstacles to the planner and tries to avoid them using beha control. look for

2.15 Controller

It controls the robot's motors and reads incoming data from the motors and ser The mobile robot that was used to test the ViRBot system is a cylindrical type bile robot equipped with a wheeled base that allows two types of motion: transl (movement parallel to the robot head alignment) and rotation (movement pe dicular to the robot head alignment), see figure 3. The robot has motion contrc that control its movements, it also has odometers to measure distances.

Fig. 3. The Robot B14.

Learning

system can learn how to solve new problems using genetic algorithms, babilistic methods, Hidden Markov models, etc.

Sensors' Simulation

simulation module allows the creation of virtual robots with multiple sensors of ferent types such as contact, reflective, sonar and infrared sensors.

Contact sensors. In order to simulate contact sensors we developed a simple polygon cross algorithm between a contact prism (sensor) and all the world (obstacles), so if the prism cross an obstacle the sensor's status change to on, indicating a collision with an obstacle.
Reflective sensors. These types of sensors are useful to create line followers robots. The virtual robot gets the track/floor values returned by the sensors in the robot's sensor position. To simulate them an average of the texels are calculated in the position where the sensors are. Texel is a texture element, the base unit of a texture over a 3D polygon [1].
Sonar sensors. Sonar sensors are simulated by getting the position where the sonar beam strikes an object.
Infrared sensors. As in the sonar sensors to simulate the infrared ones it is necessary to get the position where the infrared beam strikes an object.

Collisions with obstacles are checked all the time, the purpose of this is to simu- the robot's physical interaction with its environment. Thus the virtual robot can pass through any simulated object, if it strikes an obstacle then it needs to exe- additional behaviors related to the strike. The collision algorithm is based in nding box and polygon crossing.

user of this system can incorporate its own simulation of a particular sensor, a sonar specialist may test the operation of a particular sonar sensor whose sical behavior has been modeled and include it in the system using standard ++ routines.

andom noise can be added to the readings of the sensors, the user can select ei- if noise is added or not, as well as the mean and the variance of the Gaussian lom variables that are added.

Robot's Movements

simplest way to define the movement of a robot is through a sequence of ght lines [5], see figure 4. Basically to follow a straight line the robot requires a sition from a straight line to another, and this is obtain by coordinate transform a-

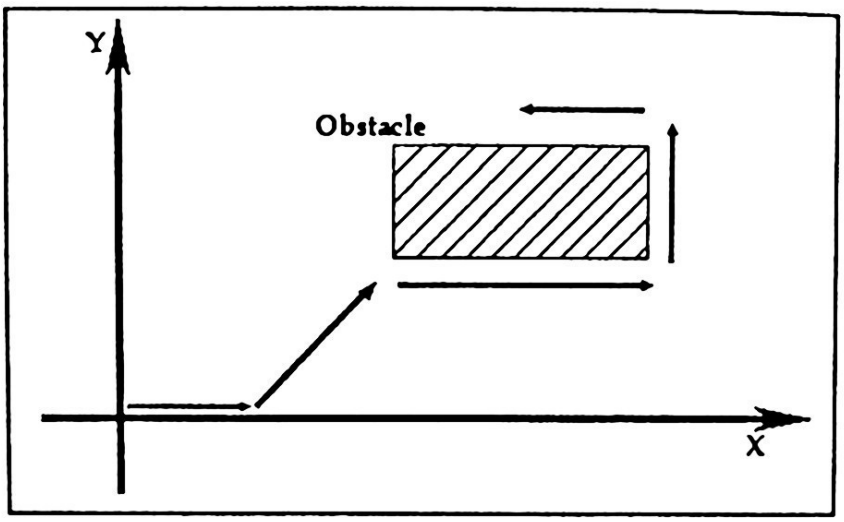


Fig. 4. Movement of the robot through straight lines.

The robot's position is described, in time i, by (x_i, y_i, θ_i) where θ_i represents the of the robot with respect of x axis. The system of coordinates x', y' represent new axis of coordinates after a rotation and displacement of the robot. Fig shows these systems of coordinates from which following equations are obtaine

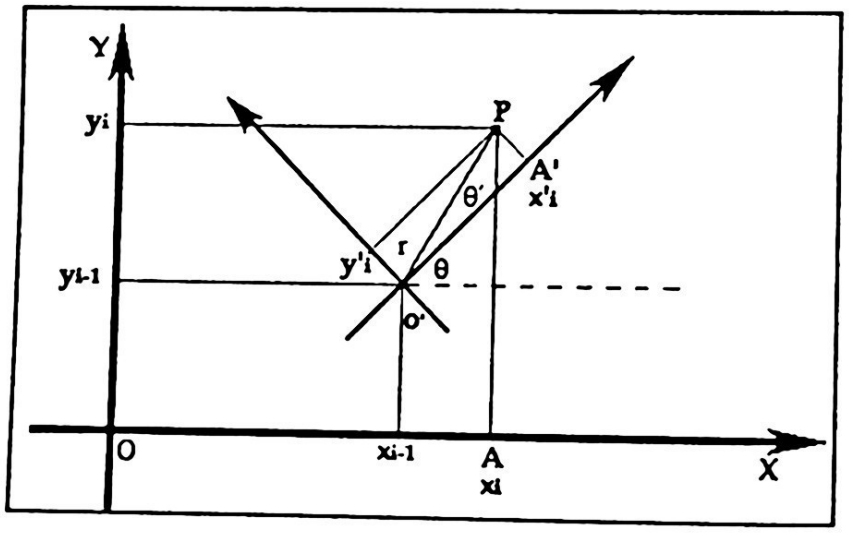


Fig. 5. Coordinate system.

$$x_i = \overline{OA} = x_{i-1} + r*\cos(\theta+\theta') = x_{i-1} + r*\cos(\theta)*\cos(\theta) - r*\sin(\theta)*\sin(\theta) .$$

$$y_i = \overline{AP} = y_{i-1} + r*\sin(\theta+\theta') = y_{i-1} + r*\sin(\theta)*\cos(\theta') + r*\cos(\theta)*\sin(\theta') .$$

$$x_i' = \overline{O'A'} = r*\cos(\theta') .$$

$$y_i' = \overline{A'P} = r*\sin(\theta') .$$

$$x_i = x_i'\cos(\theta) - y_i'\sin(\theta) + x_{i-1} .$$

$$y_i = x_i'\sin(\theta) + y_i'\cos(\theta) + y_{i-1} .$$

$$\theta_i = \theta_i' + \theta_{i-1} . \quad (7)$$

and y_i' represent the displacement of the robot and θ' the angle of rotation in the axis. If $y_i'=0$, then if the robot rotates θ_i radians and advance a distance $x_i=d_i$ the equations for the new position of the robot are:

$$x_i = d_i*\cos(\theta_i) + x_{i-1} . \quad (8)$$
$$y_i = d_i*\sin(\theta_i) + y_{i-1} . \quad (9)$$

An important part of the simulation are the movement equations, that have to be ved in any step of the simulation engine. These equations take in consideration speed and acceleration that the robot needs to follow, the next equations are ed:

Position: $$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 . \quad (10)$$

Speed: $$\frac{dx(t)}{dt} = a_1 + 2a_2 t + 3a_3 t^2 . \quad (11)$$

Acceleration: $$\frac{dx(t)^2}{dt^2} = 2a_2 + 6a_3 t . \quad (12)$$

Given the initial and final position, and the initial and final speed the a_i s con- ınts are found.

user of this system can incorporate its own simulation of the movement of the ɔot, thus a researcher wishing to test a control algorithm may include it in the stem, using standard C/C++ routines.

o make more realistic the movements of the simulated robots, random noise can added to the final movements, the user can select either if noise is added or not, well as the mean and the variance of the Gaussian random variables added.

Simulation Process

simulation process [6] is described as follows, see figure 6:

Create and initialize the 3D world. The world is created adding all the objects that will exist in it, including rooms, models, lights and textures.
Create the robot. Here, using simple objects or 3D models, all the robots that will be used are composed, loaded and initialized to work in the system.
Assign default behaviors to the robot. Default values and tasks needed to make the robots work are attached.
Initialize the communication. Here all the communication devices are initialized. If it is required, the communication with the real robot via Internet sockets starts.
Wait for an order. Constantly the data coming from an internet socket, joy pad and keyboard buffers are sensed, and the information received is analyzed.

- Execute an order. When a valid command is received it is necessary activ certain tasks needed to execute the command.
- Finish 3D world. Kill all processes attached to world; send finish commun tion signals and close all windows related.

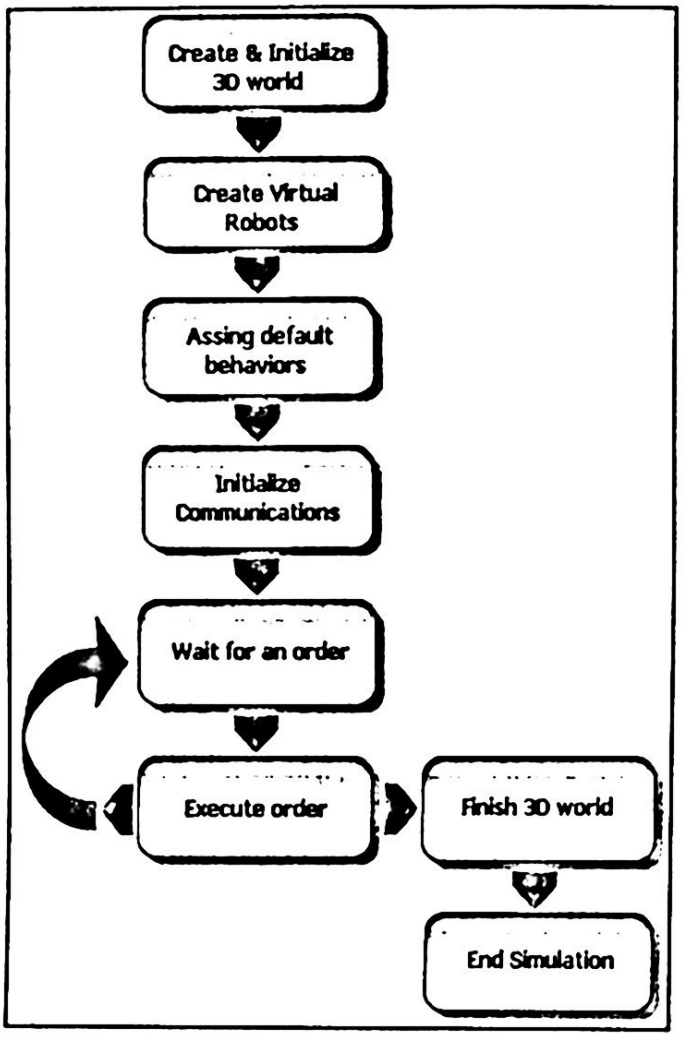


Fig. 6. Blocks diagram of the simulation process.

6 Robel

To improve the way to program and test robots, we have developed a robot' gramming language, robot behavior language or Robel, as an easy way to progr and assign basic tasks (commands) to the robots. In order to use Robel, it is nec sary that in the robot has been installed a Robel Virtual Machine (RVM), that cute commands written in that language. Basically any virtual robot in the simu tion is loaded with the RVM. We developed the RVM for a Real World Interfa robot, for a custom robot based on the 6811 micro controller and for the virt robots. The Robel's instruction set is composed by 16 basic instructions which

classified in movements, status, sensors' readings, internet interfacing and condi-
als:

Movements. These type of instructions are used to move the robot in its environ-
Instructions as moving, rotating and moving after rotating are examples of
For example the command mv d_i, θ_i, t_i rotates the robot θ_i radians, then the
advances a d_i distance, in decimeters, and in t_i seconds the action is executed.

tatus. As in real world, it is necessary that the robot knows where in the world it
ocated. Instruction as getting and setting position are examples of it.

ensors. Maybe the most important commands are the sensor related commands,
it is possible to get the values of all the defined sensors that the robot has.

onditionals. Instructions that control the flow of the program.

ser's functions. An user can include his own subroutines written in C/C++ al-
ng him to include complex behaviors to the virtual robots. Figure 7 shows the
lation of the robots on the virtual environment. The upper left window shows a
view of the 3D environment; the lower left shows a faraway view of one of the
ts; the upper right window shows the point of view of one the robots; and the
right window shows a console where the commands are typed.

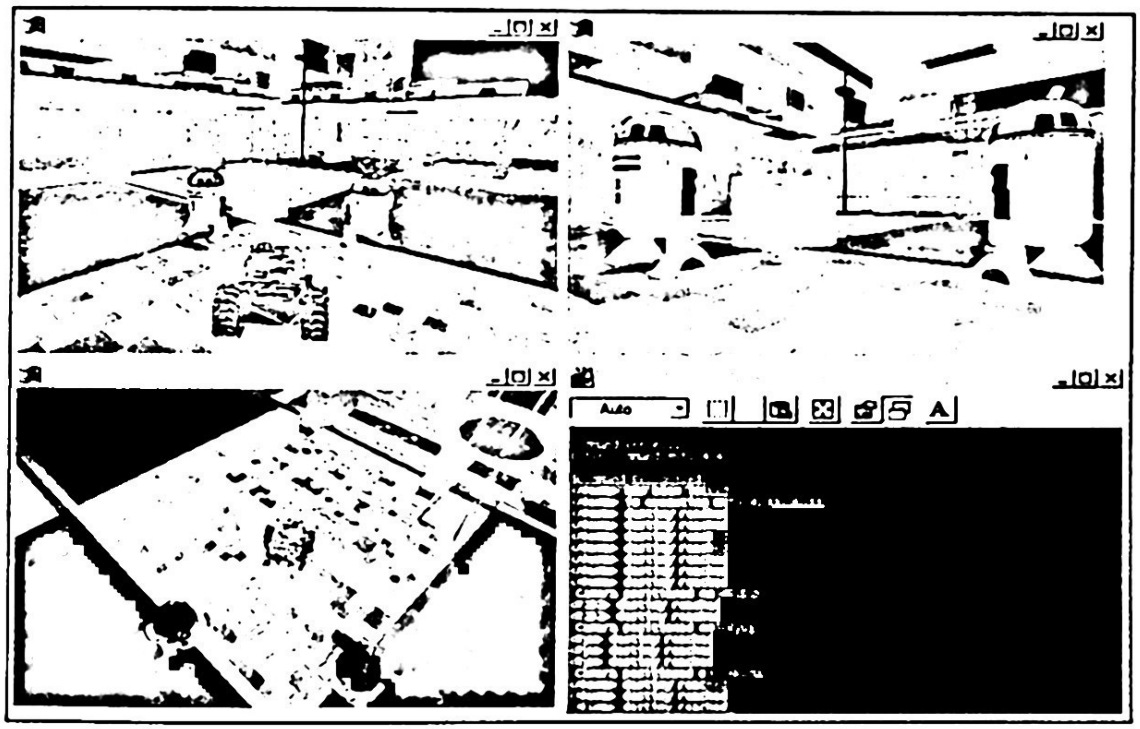


Figure 7. Simulation of mobile robots.

Robel Program

following code is an example of Robel programming language. This algorithm
generated using genetic algorithms in order to evade dynamic obstacles [9].

```
#DEFINE MAX_VALUE 5.0
#DEFINE MIN_VALUE 4.5
int state
float s1, s2
state <- 0          @1
while 1 < 2         @2
READ:                                   @3
       shs sonar 2                      @4
       s1 <- RegF1                      @5
       shs sonar 4                      @6
       s2 <- RegF1                      @7
       if state = 0                     @8
            if s1 < MIN_VALUE           @9
                 state <- 1             @10
                 goto ETI1              @11
            endif                       @12
            if s2 < MIN_VALUE           @13
                 state <- 2             @14
                 goto ETI1              @15
            endif                       @16
       endif     @17
       if state = 1                     @18
            if s2 < MIN_VALUE           @19
                 state <- 2             @20
                 goto ETI1              @21
            endif                       @22
            if s1 > MAX_VALUE           @23
                 state <- 0             @24
                 goto ETI1              @25
            endif                       @26
       endif     @27
       if state = 2                     @28
            if s1 < MIN_VALUE           @29
                 state <- 1             @30
                 goto ETI1              @31
            endif                       @32
            if s2 > MAX_VALUE           @33
                 state <- 0             @34
                 goto ETI1              @35
            endif                       @36
       endif     @37
ETI1:                                   @38
       if state = 0                     @39
            mv 3 0                      @40
            goto READ                   @41
       endif     @42
       if state = 1                     @43
            mv 0 0.5236                 @44
            goto READ                   @45
       endif     @46
       if state = 2                     @47
            mv 0 -0.5236                @48
            goto READ                   @49
       endif     @50
BYE:                                    @51
wend                                    @52
```

Conclusions

ng the ViRBot system to control virtual and real mobile robots is an easy and way to develop complex behaviors in the robots. The same algorithms tested the virtual robots have been tested, with slight variations, in two different platns: a custom robot with a 6811 micro controller, and a real world interface morobot B14, with Linux operating system. The advantage of using simulated ots instead of the real ones is that we can test extensively robots' algorithms in simulated ones before the final versions are executed in the real robots.

ferences

oley J., Van Dam A., Feiner S. Hughes J. "Computer Graphics: Principles and Practice C" Addison Wesley, 1996.

.C. Arkin and R. Murphy. "Autonomous Navigation in a Manufacturing Environment." EE Transaction on Robotics and Automation 6(4): 445-452 1990.

.C. Arkin. Behavior-Based Robotics. Cambridge, MA: The MIT Press, 1998.

iarratano, J., and Riley, G. "Expert Systems: Principles and Programming", 3rd Edition, oston, PWS Publishing Company, 1998.

C. Latombe, "Robot Motion Planning", Massachusetts, USA: Kluwer Academic Pubshers, 1991.

orldToolKit Reference Manual Release 9. Engineering Animation Inc. 1999.

omás Lozano-Pérez, "An algorithm for planning collision-free path among polyhedral ostacles," Communications of the ACM, vol.22, pp.560-570, October 1979.

alavanis K, Hebert T., Kolluru R., and Tsourveloudis N., "Mobile Robot Navigation in Dynamic Environments Using an Electrostatic Potential Field," IEEE Transactions on stems, man, and cybernetics, Vol. 30, no. 2 pp 187-196. March 2000.

ade Mattias. "ETS Retro service Chalmers University of Technology". Sweden, 2002.

Estimación adaptativa de parámetros en un sistema dinámico no lineal

Yuriria Cortés Poza

Universidad Nacional Autónoma de México
yuriria@ciencias.unam.mx

Resumen. En este trabajo se construye una red neuronal, entrenada mediante un algoritmo genético, capaz de estimar los parámetros de un sistema dinámico no lineal con comportamiento caótico, a partir de una serie de tiempo de las variables del sistema. Se tomó como caso de estudio un *ratchet* determinista, un sistema de en una partícula en un potencial periódico asimétrico, sobre la cual actúa una fuerza externa armónica. Este sistema complejo, altamente no lineal y no integrable, es de sumo interés tanto por sus aplicaciones biológicas como por la dinámica que presenta. La red estima de manera eficiente y confiable los parámetros en este sistema, tanto en los casos en los que la partícula en el potencial exhibe un comportamiento periódico, como en los casos en donde el comportamiento es caótico.

1 Introducción

En un sistema dinámico existen parámetros que caracterizan su comportami Frecuentemente estos parámetros son desconocidos y no pueden ser medidos tamente, por lo cual, el monitoreo debe de llevarse a cabo mediante observaci del sistema. Estas observaciones son cantidades medibles que dependen de rámetros. El objetivo entonces es determinar el valor de estos parámetros, a parti una serie de tiempo de alguna variable del sistema. El problema de la *estimación parámetros*, requiere, en la mayoría de los casos la solución de un problema optimización [Bard 74, Beck 77, Gallant 87]. Cuando el sistema dinámico que siendo modelado es no lineal, el problema de estimación de parámetros se complicado, y el desempeño de métodos tradicionales que existen en la literat puede ser bastante pobre. La estimación de parámetros en sistemas no lineales problema computacional importante con aplicaciones en diversas áreas, tanto ficas como tecnológicas [Jorgensen 01, Park, Froment, 98, Whigham, Rechnagel

La técnica que aplicamos nosotros en este trabajo, combina las redes neuronal con la teoría de algoritmos genéticos, utilizando un criterio de mínimos cuadr para determinar la convergencia. El método encuentra el valor de los parámetro un sistema dinámico no lineal de forma eficiente y confiable. La información requiere para la estimación son únicamente las mediciones tomadas del compc miento del sistema, es decir, una serie de tiempo de las variables importantes.

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México

me conocimiento alguno del operador de estimación como lo hacen métodos licionales, y siempre se converge a un óptimo global.

método que proponemos se examina bajo diferentes condiciones aplicándolo a sistema dinámico no lineal, con comportamiento caótico, que en los últimos años despertado un gran interés [Cortés 2000, Magnasco 93, Sarmiento, Larralde 99]. bajamos con una partícula en un potencial periódico asimétrico en el que actúa fuerza externa armónica dependiente del tiempo. Este sistema, llamado *Ratchet* gran interés por sus propiedades dinámicas, ya que la fuerza externa aplicada, combinación con la asimetría del potencial, producen direccionalidad en el flujo partículas, comportamiento que depende sensiblemente de los parámetros del ema (la masa de la partícula, la fricción, y la amplitud y la frecuencia de la fuer- externa aplicada). Este problema es altamente no lineal y para ciertos valores de parámetros presenta un comportamiento caótico. Entre las aplicaciones que tie- los ratchets esta el entender el transporte unidireccional de motores moleculares, buscar nuevos métodos para la separación de partículas, entre otros [Reimann 2].

Ratchet determinista

nque en algunos trabajos se considera la presencia de ruido, muy recientemente, habido motivaciones para entender con detalle las propiedades de transporte de hets determinísticos [Jung 1996, Mateos 00, 02]. Estos ratchets tienen en general dinámica caótica clásica que determina las propiedades de transporte. En este ajo no se toma en cuenta ningún tipo de ruido, por lo que la dinámica es deter- ista. La ecuación de movimiento está dada por:

$$m(d^2x/dt^2) + \bullet(dx/dt) + dV(x)/dx = F_0 \cos(\omega_D t) \quad . \qquad (1)$$

m es la masa de la partícula, • el coeficiente de fricción, V(x) el potencial ódico asimétrico externo, F_0 la amplitud de la fuerza externa aplicada, y ω_D la uencia de la fuerza. El potencial del ratchet está dado por (Fig.1):

$$V(x) = V_1 - V_0 \text{sen}[2\bullet(x-x_0)/L] - (V_0/4) \text{ sen } [4\bullet(x-x_0)/L] \qquad (2)$$

L es la periodicidad del potencial, V_0 la amplitud y V_1 una constante arbitra- El potencial se desplaza una cantidad x_0 de tal forma que el mínimo se encuen- el origen. Ya que la ecuación de movimiento es no lineal, obtendremos órbi- eriódicas y caóticas dependiendo del valor de los parámetros: m, F_0 y ω_D.

Fig. 1. Potencial asimétrico dado por la ecuación (2).

3 Sistema computacional

El problema que resolveremos consiste en encontrar los parámetros m y F_0 *ratchet* determinista, a partir de una serie de tiempo de la velocidad y la posici la partícula. Resolveremos esto utilizando una red neuronal entrenada median algoritmo genético. A continuación (Fig 2) se muestra el funcionamiento del ma.

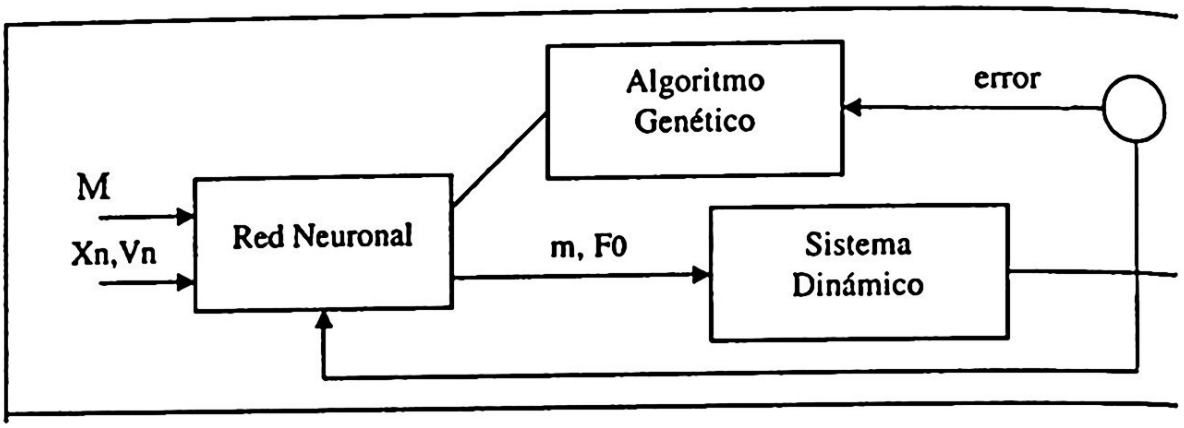


Fig. 2. Funcionamiento del Sistema. Donde *Xn=(x1,x2,...,xn)* es la posición de la partíc *Vn=(v1,v2,...,vn)* es la velocidad. *M* es la matriz de pesos de la red, la salida de la red son los parámetros estimados.

3.1 Red neuronal

3.1.1 Estructura de la red

La red neuronal que utilizamos está compuesta por tres capas: 1) la capa inicia tendrá en este caso 10 neuronas (nodos). Cada nodo de esta capa recibirá entrada un valor de la posición de la partícula en el tiempo t. Elegimos 10 neu para esta capa, por ser un número mediante el cual obtenemos suficiente inf ción del sistema para trabajar sin que disminuya la eficiencia. 2) la capa medi 4 neuronas. El encontrar el número óptimo de neuronas para la capa media, en ral se hace de forma muy empírica. Si son demasiadas neuronas, la velocid proceso disminuye, ya que el número de operaciones que se deben de llevar aumenta. Si el número de neuronas es muy poco el sistema tardará más en ger. 3) la capa final, compuesta por 2 neuronas. Cada neurona de esta capa, pro rá una salida, que eventualmente debe de ser la solución buscada, i.e., los dos metros estimados. Así, tendremos una matriz de pesos de U de 10x4 entre inicial y la capa media y una de V de 4x2 entre la capa media y la capa fina cialmente los pesos se generan aleatoriamente y son un número real $p \in [0,1$ dos matrices, tendrán que ajustarse en el entrenamiento para producir la salic seada. La función de activación • utilizada es la función sigmoide: $\bullet(x)=1/($ [Hertz, Krogh, Palmer 91].

Propagación de la señal

partir de una matriz de pesos y un vector de entrada, la red neuronal debe de :ular una vector de salida. Esto se hace de la siguiente forma:

Sea X=[x1,x2,...,x10] el vector de entrada y U10x4 y V4x2 las matrices de peso la capa inicial y la media y la media y la final respectivamente. La salida de neuronas de la capa inicial es la entrada de la capa media. Como primer paso emos calcular la salida de las neuronas de la capa media, Y=[y1,...,y4] que está por:

$$Yj = \bullet \left(\sum_i (xi\ uij)\right)$$

de i=1,2,...,10, y j=1,2,...,4, uij es el elemento i,j de la matriz **U** y $\bullet(x)=1/(1+e^{-x})$. salida de la capa final Z=[z1,z2] sé calcula de la manera similar, sin utilizar la :ión de activación:

$$Zj = \sum_i (yi\ vij)$$

de i=1,...,4, y j=1,2 y vij es el elemento i,j de la matriz **V**.

salida **Z** son los parámetros estimados:

$$a = z_1 \quad w = z_2$$

i=1,...,4, y j=1,2 y vij es el elemento i,j de la matriz **V**.

Entrenamiento

forma en que entrenaremos la red neuronal será mediante algoritmos genéticos ldberg 89]. A este tipo de red se le suele llamar Red Neuro-Genética. Los indi- os, o posibles soluciones, con los que va a trabajar el algoritmo genético, son ices de pesos de la red neuronal, cada una de las cuales, junto con la entrada de :namiento, producirá una salida diferente de la red. Los pasos que se siguen son siguientes:

Inicialización

enera una población inicial aleatoriamente, es decir, se crea un cierto número eterminado de individuos los cuales son posibles matrices de pesos para la red. ste caso, cada dato en vez de ser un bit (como es usual en los algoritmos genéti- será un elemento de matriz.

Evaluación

vez inicializada la población, se calcula la salida que va a producir la red neu- con cada matriz de pesos, y la entrada de entrenamiento. La salida de la red los dos parámetros estimados. Para poder evaluar al individuo en cuestión, se :ne una serie de tiempo de la posición y la velocidad de la partícula utilizando uevos parámetros estimados en la ecuación de movimiento. Calculamos el error las dos series utilizando mínimos cuadrados. En base a este error le asignare- al individuo o matriz de pesos correspondiente una medida de desempeño (fit- Este procedimiento se repite para cada individuo de la población.

3.2.3 Selección con elitismo

Cuando ya ha sido evaluada la población y cada individuo tiene asignado un se lleva a cabo una selección. En este trabajo aplicamos el método de la ruleta método de selección estándar proporcional. Mediante este procedimiento se una nueva población con el mismo número de individuos que la población anter Cada individuo tendrá una probabilidad de ser elegido proporcional a su medid desempeño, de esta manera, los mejores individuos tendrán más probabilidad elegidos, sin embargo, los peores individuos también pueden llegar a ser elegid evitando de esta manera caer en un óptimo local y permitiendo una mayor dive dad. El mejor individuo de la población actual siempre pasará a la siguiente.

3.2.4 Cruza y mutación

A la nueva población seleccionada se le van a aplicar los operadores cruza y ción. Para la cruza, se seleccionan dos individuos de la nueva población. Se elige punto de cruza pc, que será un renglón y una columna de la matriz. Se dividen ambos individuos, y se concatena el principio del primero con el final del segund viceversa. Para la mutación, se elige uno o dos puntos de mutación, que serán bién un renglón i y una columna j de la matriz. El dato a_{ij} será sustituido por constante proporcional a su valor original, ka_{ij}, donde k es un valor elegido riamente dentro de un rango generalmente (0,1). De la población total, un porcen PC de individuos será cruzado y un porcentaje PM de individuos será mutado. porcentaje de mutación PM, es muy grande, puede provocar que se pierda infor ción histórica sobre los individuos, si es muy pequeño, puede caerse en un local, ya que los individuos producidos por cruzamiento, no producen informa nueva. En esta implementación elegimos PC=80% y PM=20%, ya que fueron valores que produjeron los mejores resultados.

Una vez producida una nueva generación, se evalúa, y se vuelven a aplicar operadores cruza y mutación. Esto se repite hasta que se encuentre un individuo produzca una salida satisfactoria.

4 Resultados

Hicimos un número grande de pruebas de nuestro sistema para diferentes valores los parámetros. Se observó que el sistema es confiable y eficiente tanto para periódicas como para órbitas caóticas.

Graficando el comportamiento de la evolución del desempeño de la red genética a través de las generaciones (Fig. 3), se observa, como es característico los algoritmos genéticos, una mejora dramática en las primeras generaciones, posteriormente una aproximación mas lenta al 1, siguiendo un comportamiento saturación. En general, en todas las ejecuciones realizadas se alcanza una medida desempeño mayor a 0.999 en las primeras 100 generaciones.

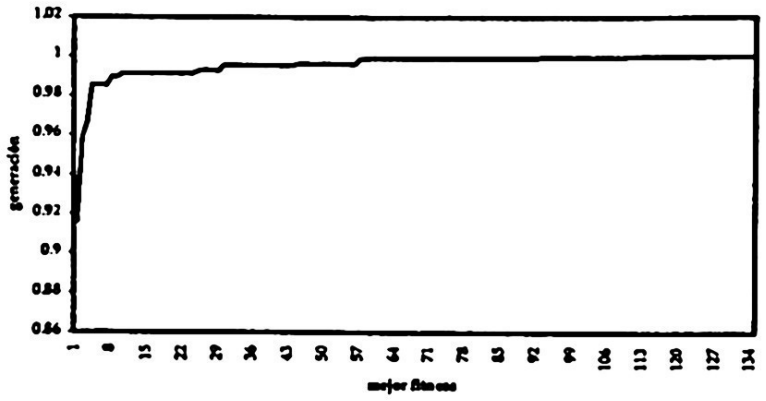


Fig. 3. Medida de desempeño del sistema (fitness) a través de las generaciones.

.onclusiones

era de esperarse, las trayectorias caóticas reales se separan de las trayectorias .ıidas utilizando los parámetros estimados, que generalmente tienen un error del de 10^{-6}. Esto es debido a la alta sensibilidad que tienen los sistemas caóticos a perturbaciones en las condiciones iniciales o los parámetros en este caso aney 92, Ott 92, Peitgen 92]. El tiempo que las trayectorias permanecen juntas aumentar si disminuimos el error en la estimación, lo que se logrará con un :ro mayor de iteraciones del programa. Dependiendo de la aplicación que se le dar al problema, se decide la precisión que se necesita, tomando en cuenta mientras mas preciso sea el resultado el tiempo de ejecución será mas largo.

.onsideramos que la contribución de este trabajo, es: por un lado, la creación de :stema computacional útil para resolver el problema de estimación de paráme- Se muestra de esta manera la gran utilidad de las redes neuronales y los algo- genéticos, como una herramienta computacional, capaz de superar otras téc- computacionales con mucha ventaja. Por otro lado el trabajo contibuye a abrir ımino en la difícil tarea de caracterizar la dinámica de los ratchets, donde, a del gran auge que ha tenido la investigación en el tema en los últimos años, queda mucho por hacer, y dadas las múltiples aplicaciones que tienen los *rat*- consideramos que las contribuciones que se puedan hacer en este campo, por de nuestro enfoque pueden resultar de suma importancia.

rencias

ioldberg, D. E. Genetic algorithms in search, optimization and machine learning. *Addi- ɔn-Wesley Publishing Company.* NY 1989

ſichell, M. An introduction to genetic algoritms: complex adaptive systems. *MIT ˜ress.* EEUU 1998

3. Poli, R., Langdon, W. Foundations of genetic programming, *Springer Verlag*, NY
4. Bard, Y. Nonlinear Parameter Estimation. *Academic Press*, NY, 1974
5. Beck, J.V. , Arnold, K.J. Parameter estimation in engineering and science. *series in probability and mathematical statistics*. J. Wiley, NY, 1977
6. Gallant, A.R., Nonliner statistical models. *Wiley*, NY, 1987
7. Jorgesen, S.E., Fundamentals of Ecological Modelling 2da ed. *Elsevier*, Amsterd 1994
8. Park,T., Froment, G. "A hybrid genetic algorithm for the estimation of parameter detailed kinetic models", *Computers chem. Engng.* 22, 1998
9. Whigham, P.A., Rechnagel, F. "Predicting chlorophyll-a in fresh water lakes by ing process/based models and genetic algorithms. *Ecological Modeling*, 146, 200
10. Astumian, R.D., Hänggi, P., "Brownian Motors", *Physics Today*, Nov. 2002.
12. Cortés, E., "Ratchet motion induced by a correlated stochastic force". *Physica A* 2000
13. Jung, P., Kissner, J.G., Hänggi, P., "Regular and chaotic transport in asymetric peri potentiales: inertia ratchets". *Phys. Rev. Lett*, 76, 1996
14. Hänggi, P., Bartussek, P., Nonlinear Physics of Complex Systems, *Lecture Note Physics*, 476, Berlin 1996.
15. Magnasco M.M., Forced Termal Ratchets, *Physical Review Letters*, 71-10, 1993.
16. Mateos, J.L., "Current reversals in deterministic ratchets: points and dimers", *D*, 168, 2002
17. Mateos, J.L., "Chaotic Transport and Current Reversal in Deterministic Ratch *Physical Review Letters*, 84-2, 2000
18. Reimann, P., "Brownian motors: noisy transport far from equilibrium", *Phys. Rep* 361, 2002.
19. Sarmiento A., Hernán Larralde, "Deterministic transport in ratchets", *Physical E*, 59-5, 1999
20. Fausett L., Fundamentals of Neural Networks: Architectures, Algorithms and Appl tions, *Prentice Hall*, NJ 1994
21. Hetz, J., Krogh, A, Palmer, R.,Introduction to the theory on neural computation. *son Wesley*, NY, 1991
22. Kohonen, T., Self-Organization and Associative Memory, 2da. ed., *Springer-*Berlin, 1984
23. Rojas, R. Neural Networks. A Systematic Introduction. *Springer Verlag*. Berlin 1
24. Rumelhart, D.E., McClelland, J.L., Parallel Distributed Processing, *MIT Press*, bridge, 1986
25. Tenorio, M.F., Lee, W.T. "Self organizing network for optimum supervised learni *IEEE Transactions on Neural Networks*, 1, 1990
26. Devaney, R.L. Chaotic dynamical systems, *Addison Wesley*, NY, 1992
27. Greborgi, C., Ott, E., Yorke, J.A., "Chaos, strange attractors, and fractal basin bou ries in nonlinear dynamics", *Science*, 238, 1987
28. Ott, E. Chaos in Dinámical Sistems, *Cambridge University Press*, NY, 1993
29. Peitgen, H.O., Jürgens, H., Saupe, D. Chaos and Fractals: New Frontiers of Scie *Springer-Verlag*, NY, 1992
30. Rasband, S.N., Chaotic dynamics of nonlinear systems, *John Whikey & Sons*, NY,

CONTROLINT: Sistema de inteligencia artificial aplicado a la enseñanza de la auditoría de estados financieros

Guillermo Adolfo Cuéllar Mejía

Departamento de Ciencias Contables, Facultad de Ciencias Contables Económicas y Administrativas, Universidad del Cauca,
Popayán, Calle 5ª, Carrera 4ª Esquina, Departamento del Cauca, Colombia

E-mail: gcuellar@ucauca.edu.co, gcuellar15@hotmail.com
Teléfono: 2593-0767

Resumen. Se presenta el desarrollo de una herramienta de apoyo a la enseñanza de la auditoría de estados financieros, utilizando técnicas de Inteligencia Artificial y Sistemas Expertos para las actividades de auditoría no estructuradas (dictamen, evaluación del control interno, planeación, evaluación de riesgos, etc.). El motivo de este trabajo de debe a que hasta ahora se ha aplicado la informática a las actividades de auditoría estructuradas (balances, razones financieras, muestreo estadístico, pruebas, circularización etc.) y a las semiestructuradas (consultas en base de datos, análisis de información etc). Para el desarrollo del sistema inteligente utilizamos la metodología CommonKADS para la adquisición y modelación del conocimiento, e implementamos un prototipo usando el Shell para sistemas expertos Crystal.

ıtroducción

ıseñanza de las ciencias contables a nivel universitario adolece de un problema ha sido muy difícil de solucionar: su contenido sumamente teórico y su nıento de la práctica real de las empresas. Este problema es reiterativo en cada las áreas del conocimiento que cubren esta campo de la ciencia: contabilidad, ;is de estados financieros, auditoría de estados financieros, régimen impositivo

de estas áreas, la auditoría de estados financieros tiene mucha analogía con la :ina, pues básicamente esta es *el examen independiente de la información de uier entidad* [1], pero a diferencia de los aspirantes a médicos, los futuros ɔres no tienen a su disposición empresas que les permitan realizar sus exámenes las mismas para adquirir la pericia necesaria para desarrollar una buena práctica ;ional. En las universidades se realizan prácticas sobre empresas simuladas en ıales el docente trata de trasmitir sus conocimientos prácticos de auditoría, sin plenamente sus objetivos, pues la enseñanza continúa siendo teórica y abstracta

---

'rnational *Federation of Accountants*, I.F.A.C. Guías Internacionales de oría.

bukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, -2003, pp. 21–31, 2003. © Centro de Investigación en Computación, IPN, México.

sin aterrizarse a realidad práctica por muchos esfuerzos que se haga al respecto, es muy difícil trasmitir la experticia que da la práctica cotidiana en un aula.

Los sistemas de inteligencia artificial o sistemas expertos se han aplicando éxito en múltiples campos: medicina, geología, química, ingeniería, etc. para real tareas muy diversas como (interpretación, predicción, diagnóstico, planificación, instrucción, control, etc.) Las actividades administrativas, financier contables también son campos en los que se pueden aplicar este tipo de sistema inteligencia artificial, pues se realizan muchas de las tareas antes descritas y, ade éstas cumplen la mayoría de los requisitos que son necesarios para poder desarr un sistema experto (las tareas requieren conocimiento especializado, auténticos expertos en la materia, los expertos son escasos, la pericia necesit localizada en distintos lugares, la mayoría de las tareas requieren soluci *heurísticas*). Ahora bien, no en todas las tareas que se realizan en el campo contabilidad y las finanzas es necesario utilizar los sistemas de inteligencia artifi Así, en las tareas de auditoría que están perfectamente estructuradas, son mecánicas y pueden expresarse en forma *algorítmica* (preparación de bala cálculo de razones, muestreo, circularización) se puede, y es conveniente, utiliz *informática convencional* (programas informáticos normales, tratamientos de bases de datos); en las tareas que estén semi-estructuradas se pueden utiliza *sistemas de ayuda a la decisión* (hojas de cálculo, sistemas de consulta de arch sistemas de representación y análisis de datos); reservándose los *sistema inteligencia artificial* para las tareas que estén muy poco o nada estructuradas, este tipo de tareas se requiere mucho del juicio de un experto y se utilizan *heurísticas* para llegar rápidamente a una solución, dado que el campo de soluc puede ser muy amplio.

Sin embargo, es muy poco lo que se ha investigado para solucionar las tare estructuradas de las ciencias contables, en especial la auditoría, median utilización de sistemas de inteligencia artificial, y la aplicación de estos sistemas la enseñanza universitaria es inexistente. No existe en la revisión bibliogr efectuada, un sistema de inteligencia artificial que apoye la enseñanza universitar las ciencias contables, y específicamente un sistema de inteligencia artificial apoyo a la enseñanza de la auditoría a nivel universitario y tampoco se encon metodologías que permitan el desarrollo de sistemas de inteligencia artificial aplicaciones didácticas contables, lo que se considera como una gran falencia, son abundantes las aplicaciones de la Inteligencia Artificial a diversas disciplina conocimiento, muchas de las cuales tienen gran similitud con las ciencias cont por lo que los resultados obtenidos podrían ser también similares.

Lo poco que existe se ha orientado en su mayoría a sistemas de intelig artificial para solucionar problemas de auditoría de las firmas transnacional servicios contables y por tanto la escasa literatura al respecto se encuentra en inglés, solo en dos universidades españolas, investigadores han trabajado proble ajenos a dichas trasnacionales pero también es poco lo relacionado con la intelig artificial como apoyo a la enseñanza universitaria.

El principal objetivo de nuestro trabajo fue el diseño de un sistema de intelige artificial como apoyo a la enseñanza de la auditoría a nivel universitario y establ divulgar una metodología que permita el desarrollo de sistemas de intelig artificial con aplicaciones didácticas contables. Como objetivos específic

.empló desarrollar un sistema inteligente para la enseñanza de la auditoría en el _po concreto de la evaluación del control interno de una empresa, para que sirviera)oyo al estudiante para determinar el grado de confianza que este le merece y así 'minar el tipo de pruebas a aplicar, su alcance y la oportunidad de las mismas.

Estos objetivos del proyecto se justifican en razón a que la ausencia de una :ica real de las ciencias contables en las aulas universitarias ha llegado a constituir verdadero escollo para la calidad del servicio profesional a prestar, pues en has ocasiones se tienen dudas acerca del desempeño de un contador recién sado de las aulas universitarias y por esto se exige un tiempo mínimo de práctica :sional simultáneo con los estudios o posterior al grado universitario para otorgar itorización del ejercicio de la profesión contable. Es así que para el caso del área uditoría las Normas Internacionales de Auditoría en su Norma General No. 1)lece: "*La auditoría debe desempeñarse por una persona o personas que posean adiestramiento técnico adecuado y pericia profesional*" [2], pero esta pericia :sional exigida nunca será lograda a cabalidad en el aula de clases, por muchas laciones que se realicen por la dificultad antes anotada de trasmitirla por amente por la mediación de un profesor.

vestigaciones relacionadas

lugar a dudas, la rama de la contabilidad en la que se han desarrollado más mas de inteligencia artificial es la auditoría. Las razones de este desarrollo se en parte, al interés y al apoyo prestado por las grandes multinacionales de la _oría, que han desarrollado o financiado muchas de las investigaciones llevadas a en esta materia. En la tabla 1, se presenta una relación de algunos sistemas de igencia artificial famosos, unos comercialmente disponibles y otros sólo :ntados a nivel teórico en la literatura sobre investigación contable.

Visión general

sistemas expertos son programas informáticos que se derivan de una rama de la stigación informática llamada Inteligencia Artificial (IA) y que utilizan el cimiento y procedimientos para resolver problemas difíciles para personas :sionalmente entrenadas [Klein M, Methlie L, 90]. En los Sistemas Expertos, se na una forma de conocimiento, consistente en incluir, como parte integral y amental del sistema, aspectos como la experiencia, el conocimiento, la intuición, de un especialista en cierto campo del saber, a fin de que el Sistema Experto ese emular, lo mejor posible, su trabajo de toma de decisiones [Feigenbaum, 83], .es, 83].

programas de IA que logran la capacidad experta de solucionar problemas ando las tareas específicas del conocimiento se llaman *Sistemas Basados en)cimiento* o *Sistemas Expertos*. A menudo, el término *sistemas expertos* se va para los programas que contienen el conocimiento usado por los humanos rtos, en contraste al conocimiento recolectado por los libros de textos. Los

---

[2] WILEY, Larry P. Miller, Guía de Auditoría.. pág. 4.05

términos, *sistemas expertos* (ES) y *sistemas basado en conocimiento* (KBS), utilizan como sinónimos. Tomados juntos representan el tipo más extenso aplicación de IA[Pignani, J, 98].

Tabla 1: Investigaciones recientes en Sistemas Expertos Aplicados a la Auditoría

NOMBRE DEL SISTEMA	DESARROLLADO POR	HERRAMIENTA DE DESARROLLO
AUDITPLANNER	Deloitte, Haskins & Sells Foundation.	Lenguaje de ingeniería del conocimiento EMYCIN
RISK ADVISOR	Coopers & Lybrand	FFAST, desarrollada por Coopers & Lybrand, y Shel ART (Automate Reasoning Tool).
RICE (Risk Identification and Control Evaluation)	Arthur Andersen & Co	Este sistema forma parte d Proyecto FEAP (Front End Analysis and Planning).
ARISC (Auditor Response to Identified Systems Controls).	Peat, Marwick & Mitchell Foundation, y Artificial Intelligence Research Center de la Universidad de Minnesota	Shell Galen
INTERNAL-CONTROL-ANALYZER		Lenguaje de ingeniería del conocimiento EMYCIN.
TICOM (The Internal Control Model)	Peat, Marwick, Mitchell & Co.	Lenguaje PASCAL
DECISION SUPPORT (DS) o EY/ DECISION SUPPORT	Arthur Young; Ernst & Young.	Lenguaje PASCAL
EXPERTEST	Coopers & Lybrand	Shell de Coopers & Lybrand, en lenguaje LISP
COMPAS (Computerized Planning Advisory System)	Price Waterhouse	
AUDITOR	Peat, Marwick Mitchell Co	Shell AL/X (Advice Language/X).
CFILE (credit file analysis) LOAN-PROBE (versión posterior ampliada de CFILE)	Peat, Marwick, Mitchell Co	Shell INSIGHT2. Shell NEXPERT. Contiene 16 programas escritos en lenguaje PASCAL
EXPERTAX	Coopers & Lybrand.	Shell en lenguaje LISP
GC-X		Lenguaje LISP
AOD (Audit Opinion Decision)	Peat, Marwick, Mitchell Co	Sistema XINFO
CHECKGAAP		Shell BROWSE. Lenguajes C y Assembler
AUDI EXPERT	Universidad de Sevilla.	Shell CRYSTAL
AUDIT MASTERPLAN (AMP)	Institute of Internal Auditors	
EDP-XPERT	Peat, Marwick, Mitchell & Co.	Shell AL/X (Advice Language/X)

Cada sistema experto consiste de dos partes principales: *la base del conocimiento*; *razonamiento*, o motor de inferencia. *La base del conocimiento* de los sistemas rtos contiene el conocimiento efectivo y heurístico. *El conocimiento efectivo* es el cimiento del dominio de la tarea que se comparte ampliamente, encontrado amente en libros de textos. *El conocimiento heurístico* es el conocimiento riguroso, más experimental, más crítico del funcionamiento. En contraste con nocimiento efectivo, el conocimiento heurístico raramente se discute y es en gran individualista. Es el conocimiento de la buena práctica, del buen juicio y del namiento admisible en el campo. Es el conocimiento que es la base del " arte de inferir" [Pignani, J, 98].

representación del conocimiento formaliza y ordena el conocimiento. Una sentación ampliamente usada es *la regla de producción*, o simplemente *regla.* regla consiste en: una parte *SI*, y otra parte *ENTONCES* (también llamada como *condición* y *una acción*). Las listas de partes *SI* son un conjunto de condiciones na cierta combinación lógica. La porción del conocimiento representado por la es relevante a la línea del razonamiento que es convertido si la parte *SI* de la está satisfecha; por lo tanto, la parte *ENTONCES* puede ser concluida, o su de solucionar el problema ser tomada. En los sistemas expertos en que el miento se representa en forma de regla se llaman: *sistemas basados en reglas.*

primera época de la Inteligencia Artificial (IA) aplicada, tras la inicial búsqueda stica y micromundos formales, se centró en el desarrollo de lenguajes de sentación del conocimiento que hicieran más fácil el uso en la inferencia. En l, cada forma de representación lleva asociada de forma natural un tipo de encia. Con este énfasis en el conocimiento se inicia el contacto de la IA con los lemas reales, en los que son necesarias soluciones eficientes y procesos de rollo sistemáticos. Sin embargo, esta primera época se centró en el prototipado lo, en el uso de entornos de desarrollo muy orientados hacia una forma específica presentar e inferir, y con el convencimiento más o menos explícito de que la IA ada era un 'arte'. Algo que para cada caso, en cada tipo de conocimiento del nio, debería resolver por procedimientos artesanales a partir de un cierto diálogo cturado con el experto humano para 'capturar' su conocimiento [Musen, 93].

Metodología

vestigación se guiará con la metodología proporcionada por los siguientes los: C.M.M. Modelo de Capacidad de Madurez (*Capability Maturity Model*), dología de dirección de proyectos del PMI (*Project Management Institute*), dología Clásica de proyectos en sistemas, Marco de Referencia de Soluciones osoft MSF (*Microsoft Solutions Framework*) y CommonKADS (Common vledge-Based Analysis and Design Systems) . La propuesta solución se rolló utilizando la metodología Common-KADS y para el diseño físico del tipo se usó el Shell Cristal (Intelligent Enviroments Cristal versión 4.50 ıced to Ra Ma)

5 Diseño de Controlint

Para el desarrollo del módulo de evaluación del control interno del efe requerimos de aplicar técnicas basadas en el conocimiento. La metodología aplicamos es CommonKADS, con la que estamos cubriendo todo el proce desarrollo del software de este sistema basado en el conocimiento, desde los asp puramente de gestión hasta el diseño. Esta metodología pone en práctic directrices de modelado, gestión de riesgos y reusabilidad propias de la ingenier software y de la ingeniería del conocimiento, a través del conjunto de modelc estados de esos modelos y las bibliotecas de modelado.

5.1 Evaluación del Control Interno de Efectivo

Uno de los métodos utilizados por los auditores para la evaluación del control es el método de cuestionarios. Básicamente consiste en un listado de pregu través de las cuales se pretende evaluar las debilidades y fortalezas del siste control interno. Estos cuestionarios se aplican a cada una de las áreas en las cua auditor dividió los rubros a examinar. Para elaborar las preguntas, el audi tener el conocimiento pleno de los puntos donde pueden existir deficiencias formular la pregunta clave que permita la evaluación del sistema en vigencia empresa. Generalmente el cuestionario se diseña para que las respuestas neg indiquen una deficiencia de control interno. Algunas de las preguntas pueden tipo general y aplicables a cualquier empresa, pero la mayoría deben ser espec para cada organización en particular y se deben relacionar con su objeto social.

Las reglas que aplica el auditor para evaluar si el área tiene un control fuerte o débil, se establecen como proposiciones *SI ..... ENTONCES*. Si la resp a la pregunta es negativa, existe una debilidad de control interno, si es afirmati una fortaleza. Al finalizar el cuestionario el auditor debe en forma heu determinar el peso que tienen las fortalezas frente a las debilidades para así conc el área examinada es débil o fuerte en referencia al Control Interno y de esta establecer las pruebas que debe aplicar, así como la oportunidad y la extensión mismas. En la Tabla 2 se puede ver un ejemplo típico de un cuestionario ponderación que el auditor da a cada respuesta de acuerdo a su importancia rela

[3] Control Interno es el proceso conformado por las diversas disposiciones y creados por la alta dirección, desarrollado por el recurso humano de la organi con el fin de dar seguridad y confiabilidad a la información que se generada transacciones económicas, promover la eficiencia y la eficacia de las operacion ente económico y asegurar el conocimiento y cumplimiento de la normat interna y externa de la Entidad.

'abla 2: Cuestionario evaluación del Control Interno de Efectivo y su ponderación
= Fortaleza de C. I., D = Debilidad de C. I.)

	PREGUNTA	RESPUESTA	SIGNIFICA	PUNTOS
	Las entradas de efectivo se controlan mediante prenumeración o relación de cobros etc.?	SI	F	9
		NO	D	
	Los ingresos por aprovechamientos se reportan en forma oportuna a contabilidad?	SI	F	3
		NO	D	
	Los ingresos diarios de efectivo se depositan a primera hora del siguiente día hábil en la misma especie recibida?	SI	F	9
		NO	D	
	Todos los egresos de efectivo se realizan por medio de cheque?	SI	F	7
		NO	D	
	Los giros de efectivo se hacen siempre a nombre del primer beneficiario y se prohíbe el giro al portador?	SI	F	5
		NO	D	
	El giro de fondos por cheque se hace con restricciones como de páguese al primer beneficiario, para depositar en cuenta solamente?	SI	F	7
		NO	D	
	Las cuentas bancarias son conciliadas en forma mensual y se realizan los ajustes necesarios al mes siguiente?	SI	F	5
		NO	D	
	Los cheques se giran usando como mínimo dos firmas?	SI	F	8
		NO	D	
	Los dineros en poder de la empresa se mantienen en caja fuerte?	SI	F	9
		NO	D	
0	*Los talonarios de cheques y su secuencia numérica son controlados por la persona responsable?*	SI	F	5
		NO	D	
	Los soportes que respaldan el giro de cheques se anulan con un sello de Pagado donde se describa el número de cheque y cuenta?	SI	F	8
		NO	D	
2	*Las trasferencias de fondos entre un banco y otro quedan debidamente documentadas con las respectivas autorizaciones?*	SI	F	8
		NO	D	
3	*Existen restricciones para transferencias electrónicas de fondos?*	SI	F	8
		NO	D	
	La creación del fondo fijo se realiza mediante resolución de gerencia en la cual:			
4	*Se determina el monto máximo a girar por cada pago?*	SI	F	3
		NO	D	
5	*Se prohíbe fraccionar facturas?*	SI	F	5
		NO	D	
6	*Se determina claramente los pagos que se pueden realizar y los que están prohibidos?*	SI	F	2
		NO	D	
	Se establece el momento de realizar el reembolso del fondo fijo?	SI	F	1
		NO	D	
	Se fija fianza para el responsable del fondo fijo?	SI	F	2
		NO	D	
	Se obliga al reembolso al final del ejercicio?	SI	F	1
		NO	D	
	Se prohíbe hacer préstamos con los dineros del fondo fijo?	SI	F	5
		NO	D	

5.2 Reglas utilizadas en la evaluación del control interno de efectivo

El anterior cuestionario puede ser presentado como reglas así:

R1:
SI Las entradas de efectivo se controlan mediante prenumeración o relaci
pagos
ENTONCES
Existe una fortaleza de Control Interno de Efectivo y
aumenta en 9 el acumulado del puntaje en Fortalezas

R2:
SI Las entradas de efectivo *NO* se controlan mediante prenumeración o rel
de pagos
ENTONCES
Existe una debilidad de Control Interno de efectivo

R3:
SI Los ingresos por aprovechamientos se reportan en forma oportun
contabilidad
ENTONCES
Existe una fortaleza de Control Interno de Efectivo y
aumenta en 3 el acumulado del puntaje en Fortalezas

R4:
SI Los ingresos por aprovechamientos *NO* se reportan en forma oportun
contabilidad
ENTONCES
Existe una debilidad de Control Interno de efectivo

R5:
SI Los ingresos diarios de efectivo se depositan a primera hora del siguiente
hábil en la misma especie recibida
ENTONCES
Existe una fortaleza de Control Interno de Efectivo y
aumenta en 9 el acumulado del puntaje en Fortalezas

R6:
SI Los ingresos diarios de efectivo *NO* se depositan a primera hora del siguien
día hábil en la misma especie recibida
ENTONCES
Existe una debilidad de Control Interno de efectivo

..

R40:
SI Se prohíbe hacer préstamos con los dineros del fondo fijo
ENTONCES
Existe una fortaleza de Control Interno de Efectivo y
aumenta en 5 el acumulado del puntaje en Fortalezas

R41:
SI *NO* Se prohíbe hacer préstamos con los dineros del fondo
ENTONCES
Existe una debilidad de Control Interno de efectivo

Resuelto el cuestionario, el auditor debe proceder a evaluar el peso de las respuestas para así determinar el grado de fortaleza o debilidad del control interno de efectivo para establecer las pruebas que debe aplicar, así como la oportunidad y la extensión de las mismas. Este proceso se lleva a cabo con unas reglas de este tipo:

R42:
SI El acumulado de Fortalezas es mayor que 95
ENTONCES
El control interno del efectivo es muy fuerte, se debe realizar la Prueba P en la oportunidad O con el alcance A

R43:
SI El acumulado de Fortalezas es mayor que 75 y menor que 95
ENTONCES
El control interno del efectivo es fuerte, se deben realizar las pruebas P1, P2 y P3 en las oportunidades O1, O2 y O3 con los alcances A1, A2 y A3

R44:
SI El acumulado de Fortalezas es mayor que 55 y menor que 75
ENTONCES
El control interno del efectivo es débil, se deben realizar las pruebas P1, P2, P3, P4, P5, P6 y P7 en las oportunidades O1, O2, O3, O4, O5, O6, y O7 con los alcances A1, A2, A3, A4, A5, A6 y A7. La confianza en el control interno es baja. Se debe considerar la posibilidad de una salvedad sobre el efectivo.

R45:
SI El acumulado de Fortalezas es mayor que 35 y menor que 55
ENTONCES
El control interno del efectivo es muy débil, se debe realizar las pruebas P1, P2, P3, P4, P5, P6, P7, P8, P9 y P10 en las oportunidades O1, O2, O3, O4, O5, O6, O7, O8, O9 y O10 con los alcances A1, A2, A3, A4, A5, A6, A7, A8, A9 y A10. La confianza en el control interno es baja. Se debe emitir una salvedad sobre el efectivo si las pruebas realizadas no satisfacen en un 90% .

R44:
SI El acumulado de Fortalezas es menor que 35
ENTONCES
No se puede confiar en el control interno y debe considerarse una posible abstención de opinión sobre el efectivo.

Es así como el auditor toma decisiones acerca del resultado de la evaluaci control interno de efectivo. Estas decisiones las logra como resultado experiencia acumulada durante muchos años de evaluar dicho control interno proceso de ensayo y error que logra garantizar el éxito.

En la enseñanza de la auditoría no es posible trasmitir en forma idónea experiencia, pues para lograrlo se requiere de un trabajo individualizado con estudiante, lo cual no es posible en la mayoría de los casos dada la cantid estudiantes por cada aula.

Controlint permite al estudiante entender a su propio ritmo, las razones cuales se realiza el cuestionario de control interno, los resultados que origina respuesta al mismo y las alternativas se que tendrán al finalizar el cuestionari esta manera apoya la decisión que se tomará. El sistema tiene un módu explicación que da al estudiante las razones por la cuales se opta por determ conclusión, además puede explicar la razón subyacente en cada pregunta realiza

El sistema logra resolver el problema del alejamiento del estudiante la pr real de las empresas, pues simula la situación a la que se enfrentaría en el ejercic su profesión cuando de evaluar el Control Interno de un área específica se Controlint supliría al profesor en esta parte, pues se asemejaría a un experto que se comunica con cada estudiante en forma incansable y personalizada, gener la práctica de la que se adolece en las aulas universitarias hasta el momento.

6 Conclusiones

De la experiencia adquirida en éste dominio, pueden extraerse las sigui conclusiones:

1. La formación y mantenimiento de equipos multidisciplinares resulta determi en la obtención de soluciones informáticas aplicables a problemas reales.
2. La inclusión de un módulo de adquisición y mantenimiento del conocim orientado al auditor da utilidad real al sistema.
3. Las técnicas informáticas disponibles son suficientes para construir sistemas ofrezcan soluciones útiles a problemas reales.
4. Es necesario profundizar en técnicas de aprendizaje a partir de datos para faci el mantenimiento.

Referencias

1. Bailey, Larry P. Miller, Guía de Auditoría.. pág. 4.05 (1995).
2. Feigenbaum, E. La Quinta Generación, Ed. Planeta. . (1983).
3. Hayes-Roth F. Building Experts Systems, Addison-Wesley. (1983).
4. International Federation of Accountants, I.F.A.C. Guías Internacionales Auditoría.(1995).
5. Klein, M. Methlie, L. Expert Systems a Decision Support Approach Applications in Management and Finance. Addisson Wesley, Wokinghan (1990).

6. Montesinos Julve, Vicente: "En torno al problema de la división en Contabilidad". Técnica Contable, tomo XXVIII, noviembre, (1976)
7. Musen,. An Overview of Knowledge Acquisition Tools. En J. M. David, J.P. Krivine & R Simmons, eds., Second Generation Expert System. Springer-Verlag. (1993)
8. Pignani Juan Manuel. Sistemas Expertos. Universidad Tecnológica Nacional, Facultad Regional Rosario. Argentina. (1998)
9. Sánchez Tomás, Antonio. Sistemas expertos en auditoría. Tesis doctoral. Universitat de València, Valencia, (1994)
10. Sierra Molina, Guillermo J., Bonsón Ponte, Enrique, Núñez García, Carmen y Orta Pérez, Manuel. Sistemas expertos en contabilidad y administración de empresas. Desarrollo de aplicaciones usando Crystal. Rama. Madrid. (1995).
11. Sierra Molina, Guillermo J. Utilización de la Inteligencia Artificial en la Docencia de la Contabilidad. Ed. Kronos. Sevilla. (1996).
12. Sierra Molina, Guillermo J. y Orta Pérez, Manuel. La teoría de la evidencia en auditoria. Aplicaciones de los sistemas expertos. Actualidad Financiera, 1: 5 1-15 ABR (1996).
13. Villalobos Marco A.: "Shell en Internet, para la generación de prototipos de sistemas expertos basados en reglas, una aplicación en olivocultura". Revista Facultad de Ingeniería U.T.A. Chile Vol 4 (1997).

Reconocimiento automático de actividades humanas

José A. Montero Valverde[1], Víctor Hugo Zárate Silva [2]

[1] Instituto Tecnológico de Acapulco, tel. (744) 468-18-89, fax. (744) 468-18-87,
00374868@academ01.mor.itesm.mx

[2] Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Morelos,
tel. (777) 329-71-63, fax. 329-71-96
vzarate@campus.mor.itesm.mx

Resumen. El reconocimiento automático de la actividad humana, es un área de investigación reciente que ha incorporado varios enfoques de las ciencias computacionales (Inteligencia Artificial, Reconocimiento de Patrones, Visión por Computadora). El gran interés mostrado por los investigadores, se debe a la amplia gama de aplicaciones a que dan lugar estos trabajos, como: Interfaces Humano – Computadora, Desarrollo de Ambientes Inteligentes, Supervisión y Monitoreo, etc., este interés se pone de manifiesto con la existencia de dos grandes programas de investigación al respecto. El primero, conocido como DARPA VSAM (*Visual Surveillance and Monitoring*) patrocinado por el Departamento de Defensa de los Estados Unidos, con una duración de tres años a partir de 1997, donde participaron la Universidad de Carnegie Mellon (CMU), el Instituto Tecnológico de Massachusetts (MIT) y los Laboratorios Sarnoff. El segundo de estos programas (IMPROOFS), se desarrolla en Europa. En este trabajo, se pretende mostrar la mayoría de las técnicas que han sido utilizadas hasta el momento, en el reconocimiento automático de la actividad humana, asimismo, mostrar sus ventajas y desventajas. Además, se mencionan las principales áreas de desarrollo que han surgido como consecuencia de estos avances, así como los trabajos realizados en las mismas.

1 Introducción

La habilidad que tiene el sistema visual humano para reconocer el movimiento envidiable. El humano tiene capacidad de reconocer, tanto, movimiento estructurad *caminar, correr, saltar* y *nadar*; como movimiento descrito por patrones estadísticos movimiento del agua en un río y, las hojas meciéndose por el viento [33]. embargo, a partir de la década de 1980 y debido al continuo avance de dispositiv electrónicos, surgió un gran interés por parte de los investigadores de diferentes de las ciencias computacionales, en automatizar el reconocimiento de la activid humana. Esto se demuestra, con la existencia de dos grandes programas investigación al respecto, el primero conocido como DARPA VSAM (*Visu Surveillance and Monitoring*) patrocinado por el Departamento de Defensa de Estados Unidos, con una duración de tres años a partir de 1997, donde participaron Universidad de Carnegie Mellon (CMU), el Instituto Tecnológico de Massachuse (MIT) y los Laboratorios Sarnoff [21].

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 32–43, 2003. © Centro de Investigación IPN México.

El segundo programa (IMPROOFS) se desarrolla en Europa [30]. El área que ha mostrado un mayor interés (por el número de trabajos realizados) en el reconocimiento automático de las actividades humanas, es la representada por la comunidad de visión computacional. Los investigadores de esta área, se interesan en desarrollar modelos para interpretar lo *que está pasando en una escena* a partir de una secuencia de imágenes de video, en oposición a determinar lo *que hay en la escena* al analizar una imagen [7]. Este interés, se comprende debido a la existencia de aplicaciones sobre una amplia gama de tópicos: a) desarrollo de ambientes inteligentes [38], [7-8], [32], [61], [46], b) interfaz humano-computadora [49], [45], 61], [62], [8], [66], [9], c) vigilancia y supervisión [21], [63], [46], [31], [73], [15].

En este trabajo, se pretende mostrar la mayoría de las técnicas que han sido utilizadas hasta el momento, en el reconocimiento automático de la actividad humana, asimismo, mostrar sus ventajas y desventajas. Se muestran, asimismo algunos proyectos desarrollados considerando aspectos como, el número de sensores de visión utilizados, detección de personas aisladas o en grupo y, técnicas de visión de bajo nivel aplicadas. Además, se muestran las principales áreas de desarrollo que han surgido como consecuencia de estos avances, así como los trabajos realizados en las mismas.

La organización del presente trabajo es como sigue: la sección I muestra la introducción, donde se plantea de manera general el reconocimiento automático de la actividad humana, y áreas involucradas en el desarrollo de estos trabajos. La sección 2, trata lo referente a las técnicas que se han aplicado para reconocer de manera automática la actividad humana, se muestran en varias sub-secciones, varios niveles de enfoque, desde una perspectiva basada en visión. En la sección 3, se abordan las principales áreas de desarrollo, que han surgido como consecuencia de los trabajos realizados en este tópico.

2 Técnicas aplicadas

En el proceso de reconocer de manera automática la actividad humana, desarrollada por una persona [7], o por un grupo de personas [50], [33], se ha realizado un número considerable de trabajos, los cuales han aportado nuevos enfoques a este problema (Ver Tabla 1). A continuación se muestran las técnicas y métodos que han aplicado.

2.1 Métodos de análisis del movimiento

Generalmente, hay dos tendencias para el reconocimiento de acciones: a) reconocer el movimiento a partir de una estructura y b) reconocimiento del movimiento sin estructura. El segundo enfoque, emplea características de bajo nivel, tales como las propiedades de regiones de color que representan piel o ropa, que se utilizan para el seguimiento de objetos [18], [65]. Si se utiliza suficiente información de contexto[1], esta estrategia resulta eficiente y confiable. Sin embargo, sin el uso de estructura para

[1] Información de Contexto. Se considera como tal, toda descripción del medio ambiente estático observado por una o más cámaras, asimismo, a las tareas relacionadas con el medio observado [61], [80].

resolver problemas como auto-oclusión, el uso de movimiento sin estructura se convertir en una limitante significativa. A su vez, el primer enfoque, se basa construcción de modelos 2D ó 3D de objetos, así como en descripciones física- cuerpo humano [27].

Algunos trabajos utilizan modelos ocultos de Markov[2] (HMMs) para realiz correspondencia entre los modelos 3D de acciones y las características de bajo (ángulos de juntura de brazos, manos, torso, piernas, etc.,) [77], [8]. Aunque, enfoque de alto nivel, ofrece información más significativa de la acción computacionalmente costoso y altamente dependiente de las primitivas de bajo La extracción de características influencía directamente los métodos utilizados analizar el movimiento, los cuales pueden ser: representaciones probabilistas bas en estado, redes difusas o neuronales, o técnicas parecidas. El uso de caracterís redundantes puede ayudar cuando la incertidumbre es alta, pero si no se mar correctamente, pueden generar niveles de ruido mayores y degradar el rendim [34], [51].

2.2 Representación basada en vistas

Aún no hay un consenso general acerca de cuales movimientos y/o orientaciones cuerpo humano, son los más importantes para el análisis del movimiento. En sentido, la mayor parte de los trabajos realizados generan un enfoque *ad* motivado por la aplicación. Los trabajos que emplean más de una cámara pue proporcionar sin ambigüedad la posición de un objeto, pero enfrentan problemas correspondencia de imágenes y calibración de la cámara. Para desarrollar algorit dependientes de la vista, se necesita un modelo del mundo [6]. Sin embargo ía de los trabajos realizados en visión confía en la información proporcion a cámara monocular [35], cuya interpretación se relaciona directamente bservado por la cámara.

2.3 Clasificadores utilizados para reconocer movimientos

El objetivo de los clasificadores es encontrar la mejor correspondencia entre representación de características de una imagen y una metodología reconocimiento. Una técnica que represente de manera apropiada a una acción, ser, invariante en el tiempo y preservar el orden [8]. Más importante es, representación deba ser capaz de cuantificar el progreso en espacio y tiempo movimiento, y que pueda tolerar variaciones naturales, pero todavía pod distinguir de otros movimientos. Existen varios métodos que modelan las accio humanas de manera confiable [1], [17], [29]. Hay un gran número de clasificado que son populares por distinguir la característica *espacio-tiempo* generado movimiento humano, algunos se muestran en la Tabla 1.

---

[2] Modelos Ocultos de Markov. Enfoque probabilista basado en estados [67].

El orden temporal que caracteriza al movimiento humano, motiva de manera natural el uso de representaciones basadas en estado[3], tales como los modelos ocultos de Markov (HMMs) [67]. Uno de los primeros métodos basados en estado, usaron la deformación dinámica temporal (DTW[4]) para el manejo de gestos y acciones en el tiempo. Este proceso, es similar a la programación dinámica continua, sin embargo, trabaja mejor con datos espaciales, tales como reconocer movimiento a partir de parámetros de estructura. Gavrilla y Davis, aplicaron este método para obtener los ángulos en modelos 3D con el fin de reconocer secuencias [30] y, para comparar trayectorias de movimientos contra plantillas de trayectorias [54]. Bobick y Wilson [8] usaron formulaciones de programación dinámica para calcular las trayectorias de costo mínimo entre dos nodos en un grafo producido por trayectorias de movimiento [14].

Un enfoque mostrado en trabajos recientes en el reconocimiento de acciones [32], es el empleo de los modelos ocultos de Markov como un medio para modelar acciones complejas. Este método, explota el orden temporal que, de manera natural manifiestan los gestos y acciones humanos, usando una representación probabilista basada en estado. Algunos trabajos, usan una combinación de técnicas con el fin de reconocer de manera confiable la actividad humana, tal como el trabajo desarrollado por Ismail Haritaoglu y otros [33]. Donde, desarrollaron un sistema conocido como *W*^{4}, para detectar grupos de personas, partes del cuerpo y realizar un seguimiento de las mismas.

Enfoques recientes han utilizado a las redes neuronales con el fin de clasificar gestos [45]. El uso de modelos articulados (3-D) del cuerpo humano se ha utilizado para analizar una secuencia de imágenes, tomando en consideración que el cuerpo humano posee una estructura y movimiento complicado. En este enfoque, se fija el modelo 3-D en el cuerpo de la imagen en cada cuadro, una vez realizado esto, es posible medir las poses de este modelo en movimiento [42], [30]. Uno de los problemas que enfrenta este método, es la correspondencia del modelo con los cuadros. El procedimiento que realiza la correspondencia requiere una gran cantidad de cálculos para fijar el modelo en cada cuadro y cada vista de la imagen.

2.4 Sistemas de reconocimiento de acciones humanas

A continuación se muestran algunos sistemas para el reconocimiento de actividades humanas que se han desarrollado e implementados por distintos grupos de investigadores y en diferentes instituciones. *Pfinder,* es un sistema usado para obtener una descripción 3D de una persona en espacios grandes, enfoca el problema de seguimiento de una persona en escenarios complejos, no maneja oclusiones y, utiliza una cámara fija [33]. Tanto *Pfinder*, como *W*^{4} aplican un modelo estadístico con el fin de localizar personas, sin embargo *Pfinder* utiliza una distribución de una sola Gaussiana para cada color de píxel, mientras que *W*^{4} aplica una distribución bimodal para modelar la intensidad. Ambos sistemas utilizan siluetas para detectar las partes del cuerpo. *KidRooms* es un sistema de seguimiento basado en "*regiones cerradas del*

---

[3] Las representaciones basadas en estado son usualmente máquinas de estado finito que intentan segmentar y agrupar el movimiento en un orden natural de estados.

[4] DTW. *Dynamic Time Warping*

mundo". Estas, son regiones de espacio y tiempo en las cuales, el contexto específico de lo que existe en la región es conocido [6-8].

Smart Kiosk, es una aplicación para detectar y seguir personas frente un quiosco. Utiliza para ello, información de color, características del rostro, e información estéreo. Cuando las personas se encuentran cerca del quiosco, solamente puede seguir a una de ellas. El sistema *TI*, es un sistema de propósito general para detección de objetos en movimiento y reconocimiento de eventos. Los objetos movimiento, se detectan a través de cambios en la sustracción del fondo, usan distribución Gaussiana para modelar la intensidad de cada píxel. El seguimiento realiza utilizando técnicas de correspondencia de vecinos cercanos y de predicción primer orden. Se aplica para fines de vigilancia en interiores, no maneja pequeños cambios de objetos en el fondo

En CMU (Carnegie Mellon University), se desarrolló un sistema que permite a un operador humano monitorear actividades en un área extensa, usando una distribuida de sensores de vídeo [31], [15]. El sistema permite detectar y seguir a varias personas y vehículos y monitorear sus actividades en períodos extendidos tiempo. Desarrollaron, rutinas confiables para la detección de objetos en movimiento emplean una combinación de diferencia temporal y seguimiento de plantillas. objetos detectados se clasifican en humano, grupo de humanos, automóvil y camión se apoyan además en el análisis de formas y color. Los trabajos desarrollados en MIT (Massachusetts Institute of Technology) utilizan un conjunto de sensores distribuidos, con el fin de clasificar objetos y aprender patrones de actividad humana comunes e inusuales [61], [7-8].

3 Áreas de desarrollo

A continuación se mencionan las principales áreas donde incide el mayor número de trabajos relacionados con el reconocimiento automático de la actividad humana.

- *Desarrollo de ambientes inteligentes.* En esta aplicación se pretende establecer una conexión entre espacios cerrados (sala de juegos, salón de clases, sala reuniones) y computación ubicua [38]. Se intenta que las computadoras detecten al humano en la escena, lo reconozca, entienda su intención y comportamiento, con el fin de adaptarse a sus hábitos. A esta aplicación se conoce como "Observando a las Personas" (*Looking at People*), término utilizado por Pentland y otros. Algunos trabajos realizados son: *Intelligent Room*, *KidsRoom y, Smart Classroom* [61], [7-8], [32]. Los ambientes inteligentes, computadoras portátiles, interfaces de usuario perceptuales y la computación ubicua, son aspectos manejados por algunos investigadores [61], [46] como cuarta generación en la tecnología de la computación e información. Debido, que estos dispositivos de computación pasarán a formar parte de la vida cotidiana y se encontrarán en todas partes (ropa, casa, automóvil, oficina). Esta área investigación, se encuentra entre las de mayor importancia económica dentro la tecnología de la información y ciencias de la computación [74].
- *Interfaz humano-computadora.* En esta aplicación, se buscan formas naturales (expresión facial, lenguaje corporal, palabras habladas)

comunicación con la computadora con el fin de obtener interfaces cualitativamente mejores que las actuales. Es claro, que las máquinas deban ser capaces de entender y generar por lo menos, alguno de estos elementos de comunicación [66], [9]. Recientemente, un gran número de líderes industriales, incluyendo Bill Gates de Microsoft, han declarado de alta prioridad el desarrollo de interfaces visuales [36]. Se han realizado trabajos para auxiliar a personas con discapacidad que tienen la necesidad de comunicarse a través de una computadora [45], [61], [62], [8]. Pentland y otros, realizan trabajos con la finalidad de que las computadoras entiendan a las personas, y puedan trabajar de forma parecida a un asistente humano. Para ello, crean ambientes de trabajo donde utilizan datos provenientes de cámaras y micrófonos, con el fin de analizar gestos, seguimiento de una persona (cabeza, pies y manos) y voz. La mano humana, es un dispositivo de entrada capaz de coordinar varios grados de libertad al mismo tiempo. Por lo que se convierte en un medio adecuado para controlar aplicaciones complejas [25]. Este tipo de interfaces es particularmente apropiada en aplicaciones como: realidad virtual, multimedia y, operación remota [80], [28], [3].

Vigilancia y supervisión. El aspecto de seguridad, da lugar a una de las principales áreas de investigación relacionada con la comprensión automática de actividades humanas: La vigilancia y supervisión. Los trabajos realizados en esta área emplean cámaras para detectar personas, seguirlas y realizar un análisis limitado de su comportamiento. Uno de los primeros trabajos realizados en esta área fue el de Akita, quién demostró que el cuerpo humano podría ser seguido visualmente, en circunstancias muy limitadas y empleando conocimiento previo [2]. El proyecto DARPA VSAM, ha financiado un gran número de trabajos en ésta área [21]. El principal problema que enfrentan los investigadores, es la confiabilidad en la detección y seguimiento, en presencia de sombras y oclusiones, y construir una representación de patrones de movimientos válidos, de tal forma que los movimientos no-válidos, puedan ser identificados. Mientras, la norma en los sistemas de vigilancia es la obtención de respuestas cortas, en algunas aplicaciones los datos de sensores simplemente son almacenados; en este caso, la información relevante debe obtenerse a través de preguntas formuladas por el operador.

Aunque solamente se mostraron tres áreas de desarrollo, existen otras de gran interés, así como varias aplicaciones que da lugar cada una de las mismas.

Conclusión

reconocimiento automático de actividades humanas es tema de investigación que integra áreas como: Reconocimiento de Patrones, Inteligencia Artificial y, Visión Computacional entre otras. Se considera un área de interés reciente, la mayor parte de los trabajos se han realizado a partir de la segunda mitad de 1990, con el surgimiento de dos grandes proyectos: DARPA VSAM en E.U.A y el IMPROOFS en Europa. El interés primordial de estos trabajos, consiste en la integración de nuevos enfoques que conduzcan a la determinación automática de reconocer acciones

humanas en distintos escenarios, asimismo su interacción con personas u objetos circundantes.

La mayoría de los trabajos utilizan como fuente de información la proporcionada por cámaras de vídeo, aunque algunos han empleado la integración de sensores de distinto tipo. Las áreas de desarrollo donde han incidido, la mayor parte de trabajos realizados son: Desarrollo de Ambientes Inteligentes, Interfaces Humano Computadora y, Vigilancia y Monitoreo. Aunque se han logrado avances sustanciales, aún existen desafíos que enfrentar, algunos de los cuales son: *La segmentación continua de actividades*, la mayor parte de los trabajos, trata con el reconocimiento de acciones primitivas discretas, la segmentación y clasificación continua de actividades humanas es un área virtualmente inexplorada. Otra de las áreas que requiere atención es la integración de la información proveniente de distintos sensores (principalmente micrófonos y cámaras) con el fin de dar mayor confiabilidad al proceso, es decir, *técnicas de fusión de datos*.

Tabla 1. Técnicas utilizadas en trabajos de reconocimiento de gestos y acciones.

Método de Reconocimiento	Ventajas	Desventajas	Autores
Restricciones espaciales (trayectorias de movimiento)	Invariable a cambios de velocidad en las extensiones corporales.	Requiere aprendizaje supervisado	Campbell, Bobick
Condensación	Permite la extracción automática de la trayectorias del movimiento	Necesita inicialización supervisada	Black et al, Isard
Redes Neuronales	La decisión está disponible después de cualquier cuadro (tiempo-real).	No puede procesar una secuencia de cuadros.	Schlenzig otros, Linaje
Imágenes con historia y energía de movimiento	Emplea plantillas compactas para el reconocimiento.	Sensible a vistas: no maneja variación espacial.	Bobick
HMMs continuos y discretos	Maneja variación espacio-tiempo, es altamente escalable.	No hay decisión antes de completar la secuencia de observación.	Starner, Bobick
Movimiento basado en color y reglas.	Exactitud de reconocimiento de 95%	La persona debe usar ropa especial .	Hienz y otros
Aprendizaje recursivo usando inducción basada en reglas.	Reconocimiento rápido de poses, exactitud incrementada al aprender	Dificultad para generalizar debido al uso de reglas específicas.	Schlenzing otros
Modelos ocultos de Markov acoplados basado en primitivas (PCHMM).	Reconoce acciones complejas dependientes de contexto.	Reconoce la acción de una persona, sujeto a datos de entrenamiento.	Haibing y

Método de Reconocimiento	Ventajas	Desventajas	Autores
Detección y seguimiento basado en un modelo articulado del cuerpo humano.	Permite el manejo de oclusión y movimientos complicados.	La correspondencia del modelo en la secuencia de imágenes es costosa.	Yamamoto, Kakadiaris, Gavrilla.
Modelo espacio-temporal (combinación de modelos basados en apariencia y HHMs)	Reconoce varios patrones de actividades, y realiza un análisis del comportamiento	La precisión en el reconocimiento depende de los datos de entrenamiento.	Krahnstover, Yeasin, Sharma
Enfoque probabilista-posibilista	Obtiene un promedio de reconocimiento superior al 94 %.	Costo computacional, sensible a condiciones ambientales.	Kim, Bang, Bien
Representaciones invariantes a la vista (instantes dinámicos e intervalos)	Segmentación automática de video en acciones individuales. Aprende acciones humanas sin entrenamiento previo	Puede confundir acciones con el mismo número de instantes.	Rao, Mubarak
Enfoque lingüístico controlado por análisis de movimiento y técnicas de agrupamiento	Detección automática sin inicialización manual, reconocimiento confiable.	Computacionalmente costoso.	Duric, Li, Wechsler
Modelo jerárquico (tomando lo mejor de varias técnicas)	Además de interpretar la acción, analiza la probable causa que la origina.		Bourbakis, Bebis, Gattiker

Referencias

1 AGGARWAL, J.K. and Q. Cai, "Human motion analysis: a review", Computer Vision and Image Understanding, Vol. 73, No.3, pp. 428-40, March, 1999.

2 AKITA, K., "Image Secuence Analysis of Real World Human Motion", Pattern Recognition, vol. 17, no. 4, pp. 73-83, 1984.

3 BAUDEL, T. and M. beaudouin_Lafan, "Charade: remote control of objects with freehand gestures", Communications of the ACM, 36(7):28-35, 1993.

4 BERARD, F. , M. Black, A. Jepson, W. Newman, and M. Taylor, "The digital Office Overview", Proc. AAAI Spring Symp, Series, PP. 1-6, Stanford Univ., Mar. 1998.

5 BLACK, M. and A.D. Jepson, "Recognizing temporal trajectories using the condensation algorithm", Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition", pp. 16-21, Nara, Japan, April, 1998.

6 BOBICK, A. , "Movement, Activity, and Action: The Role of Knowledge in the Perception of Model", Royal Society Workshop on Knowledge-based Vision in Man and machine, 1997.

7 BOBICK, A. and J. Davis, "Real-Time Recognition of Activity using Temporal Templates", 3rd IEEE Workshop on Applications of Computer Vision, Sarasota, Florida, pp. 39-42, 1996.

8 BOBICK, A. and A. Wilson, "A State-Based Technique for the Summarization Recognition of Gesture", Proc., International Conference on Computer Visio Cambridge, Massachusetts, 1995.

9 BOLT, R. "Put-That-There: Voice and Gesture at the graphics Interface", Compu Graphics, vol. 14, no. 3, pp. 262-270, 1980.

10 BOULT, T. "Frame-rate Multi-Body Tracking for Surveillance", Proc. DARPA Understanding Workshop Monterey, Calif., San Francisco: Morgan Kauffman, 1998.

11 BOURBAKIS, N. , G. Bebis, and J. Gattiker, "A Sinergistic Model for Interpreti Human Activities and Events from Video: A Case Study", International Journal Artificial Intelligence Tools, vol. 9, no. 2, pp. 132-139, 2000.

12 BRAND, M. and I. Essa, "Casual Analysis for Visual Gesture Understanding", Symposium on Computational Models for Integrating Language and Vision, 1995.

13 BRAND, M. and V. Kettnaker, "Discovery and Segmentation of Activities in Video" IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No. 8, pp. 851, August 2000.

14 BREMOND Francois, Somboon Hongeng and Ramakant Nevatia, "Representation Optimal Recognition of Human Activities", DARPA Image Understanding Worksho pages 211-216, 1999.

15 BREMOND, F. and C. Medioni, "Scenario Recognition in Airborne Video Imagery" Proc. DARPA Image Understanding Workshop Monterey, Calif., San Francisco: Morga Kauffman, Nov., 1998.

16 CAMPBELL L. and A.F. Bobick, "Using Phase Space Constraints to Represent Huma Body Motion", Proc. Of the International Conference on Computer Visio Massachusetts, 1995.

17 CÉDRAS, C. and M. Shah, "Motion Based Recognition: A Survey", Image and Visio Computing, Vol. 13, No. 2, pp. 129-55, 1995.

18 CEN Rao and M. Shah, "View-Invariant Representation and Learning of Human Actio" PAMI, 22(8), pp. 747-757, August, 2001.

19 CIPOLLA, R. and N.J. Hollinghurst, "Human-robot interface by pointing uncalibrated stereo vision", Image and Vision Computing, 14(3):171-178, 1996.

20 CIPOLLA, R. , Y. Okamoto, and Y. Kuno, "Robust Structure from motion using motio parallax", In Proc. IEEE Fourth International Conference on Computer Vision, Berli May, pp. 374-382, 1993.

21 COLLINS Robert T., Alan J. Lipton, and Takeo Kanade, "Introduction to the Speci Section on Video Surveillance", IEEE, Transactions on Pattern Analysis and Machi Intelligence, Vol. 22, No. 8, August 2000.

22 DARRELL, T. and A. Pentland, "Active Gesture Recognition using Partially Observabl Markov Decision Process", 13th IEEE International Conference on Pattern Recognitio Vienna, Austria, vol. 3, pp. 984-988, August, 1996.

23 DURIC, Z. , F. Li and H. Wechsler, "Recognition of Arm Movements", Proc. of the IEEE International Conference on Automatic Face and Gesture Recognition, pp. 221 2002.

24 ETEMAD, K. and R. Chellappa, "Discriminant Analysis for Recognition of Human Images", J. Optical Soc. Am. A. vol. 14, pp. 1724-1733, 1997.

25 FOLEY, J.D., "Interfaces for Advanced Computing", Scientific American, 257:7, 1

26 FOLEY, J. , A. Van Dam, S. Feiner And J. Hughes, "Computer Graphics: Principles Practice", Addison-Wesley, 1990.

27 FUJIYOSHI, H. and A.J. Lipton, "Real-time human motion analisys by image skeletonization", Proc. of IEEE CVPR'97, pp. 928-934, 1999.

28 FUKUMOTO, M. , K. Mase and Y. Suenaga, "Finger-pointer: pOinting interface by image processing", Computation and Graphics., may, 18(5):633-642, 1994.

29 GAVILA, D.M., "The visual analysis of human movement: A survey", Computer Vision and Image Understanding, Vol. 73, No. 1, pp. 82-98, January, 1999.

30 GAVRILLA, D. and L.S. Davis, "3-D model-based tracking of humans in action: a multi-view approach", Proc. of IEEE CVPR'96, pp. 73-80, 1996.

1 GRIMSON, E., C. Stauffer, R. Romano, and L. Lee, "Using Adaptive Tracking to Classify and Monitor Activities in a Site", Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 22-29, 1998.

32 HAIBING Ren and Guangyou Xu, "Human Action Recognition in Smart Classroom", Proc. of the 5th IEEE International Conference of Automatic Face and Gesture Recognition, pp. 98-103, May, 2002.

33 HARITOUGLU, Ismail, David Harwood and Larry S. Davis, "W4: Real-Time Surveillance of People and Their Activities", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, August 2000, pp. 809-830.

34 HALL, David L. and , James Llinas, " An Introduction to Multisensor Data Fusion", Proceedings SPIE Conference sensor Fusion and aerospace Applications, Orlando Florida, April 1997.

35 HIENZ, K. Grobel, and G. Offner, "Real-Time Hand-Arm Motion Analysis using a Single Video Camera", 2nd International Conference on Automate Face and Gesture recognition, Killington, VT, pp. 164-169, 1996.

36 GATES, B., publicado en noviembre de 2000 (http://www.microsoft.com/billgates/pdc.htm).

37 ISARD, M.A., "Contour tracking by stochastic propagation of conditional density", Proc. European Conference on Computer Vision, pp. 343-356, Cambridge, UK, April, 1996.

38 ISHIGURO, H., C. Kim, M. Trivedi, and T. Sogo, "Monitoring Dynamically Changing Ubiquitous Vision System", Int. Joint Conf. Artificial Intelligence, 1999.

39 IVANOV, Y. and A. Bobick, "Recognition of Visual Activities and Interactions by Stochastic Parsing", IEEE Transactions on Pattern Analysis and Machine Intelligence,, Vol. 22, No. 8, pp. 852-872, August 2000.

40 IVANOV, Y., C. Stauffer, B. Bobick and W. Grimson, "Video Surveillance of Interactions", Proc. IEEE Workshop Video Surveillance, Fort Collins, Colorado, June, 1999.

41 JO, K., Y. Kuno, and Y. Shirai, "Manipulative Hand Gestures Recognition Using Task Knowledge for HCI", Proc. IEEE Conf. Automatic Face and Gesture Recognition, pp. 468-473, Nara, Japan, April, 1998.

42 KAKADIARIS, I.A. and D. Metaxas, "Model-Based estimation of 3D human motion with occlusion based on active multi-viewpoint selection", Proc. of IEEE, CVPR'96, pp. 81-87, 1996.

43 KIM, J., K. Park, W. Bang, and Z. Bien, "Continuos Gesture Recognition System for Korean Sign Language based on Fuzzy Logic and Hidden markov Model", IEEE Trans. on PAMI, vol. 21, no. 10, 2002.

44 KRAHSTOVER, N., M. Yeasin, and R. Sharma, "Towards a Unified Framework for Tracking and Analysis of Human Motion", In 4th IEEE Proc. International Conference on Automatic Face and Gesture recognition, pp. 47 -54, 2001.

45 LINAJE M., Pérez R. M., Martínez P., and Aguilar P.L., "Classifying gestures by self-organizing neural network", Proc. IEEE, 4th International Conference on Knowled Based Intelligent Engeneering Systems, Aug., Brighton, U.K., pp. 796-799, 2000.

46 LIPTON, A., H. Fujiyoshi and R. Patil, "Moving Target Detection and Classifica from Real-Time Video", Proc. IEEE Workshop Application on Computer Vision, I

47 MAES, P., "Designing Autonomous Agents: Theory and Practice from Biology Engineering and Back", Bradford Books/MIT Press, Cambridge (MA), 1991.

48 MASE, K., "Human Reader: A Vision-Based Man-Machine Interface", Computer Human-Machine Interaction", R. Cipolla and A. Pentland (eds), Cambridge Univ. 1998.

49 McNEIL, D., "Hand and Mind: What Gestures Reveal About Thought", Univ. Chicago, Press, 1992.

50 MIKIC, I., K. Huang and M. Trivedi, "Activity Monitoring and Summarization Intelligent Meeting Room", IEEE Conf. SMC, 2001.

51 MONTERO, V., José A. ., "Técnicas de Fusión de datos en Sistemas Distribuid Tiempo Real", Estado del Arte, ITESM Campus Morelos, 2000.

52 MUBARACK, Sha and D. Ayers, "Monitoring Human Behavior in an Environment", PAMI, Vol. 19, No. 7, pp. 780-794, July 1997.

53 MURBRIDGE, E., "Animal Locomotion", 1887.

54 NAGAYA, S., S. Seki and R. Oka, "A Theoretical Consideration of Pattern Trajectory on Gesture Spotting Recognition", 2nd International Conference on Automa Face and Gesture Recognition, Killington, VT, pp. 72 -77, October, 1996.

55 NALWA, V., "A guided Tour of Computer Vision", Addison-Wesley Publishing, York, 1993.

56 NAM, Y., K. Whon, and H. Lee-Kwang, "Modeling and Recognition of Hand Gest Using Colored Petri Nets", IEEE Transactions on Systems, Man and Cybernetics, vol. No. 5, Sept, pp. 514-521, 1999.

57 NISHIMURA, T. and R. Oka, "Spotting Recognition of Human Gestures from Varying Images", 2nd International Conference on Automate Face and Gest Recognition, Killington, VT, pp. 318-22, October, 1996.

58 OLIVER, N., B. Rosario and A. Pentland, "A Bayesian Computer Vision System Modeling Human Interactions", IEEE Transactions on Pattern Analysis and Machi Intelligence,, Vol. 22, No. 8, pp. 831-843, August 2000.

59 OLSON, T. and F. Brill, "Moving Object Detection and Recognition Algorithms Smart Cameras", Proc. DARPA Imaging Understanding Workshop, pp. 159-1 Monterey, Calif., San Francisco: Morgan, Kaufman, 1997.

60 OREN, M., C. Papageorgio, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian Detecti Using Wavelet Templates", Trans. IEEE Conf. Computer Vision and Patte Recognition, pp. 193-199.

61 PENTLAND, A., "Looking at People: Sensing for Ubiquitous and WearableComputing" IEEE Transactions on Pattern Analysis and Machine Intelligence,, Vol. 22, No. 107-119, January 2000.

62 PENTLAND, A., "Smart rooms", Scientific American, vol. 4, pp. 68-76, April, 1996.

63 POLANA, R. and R. Nelson, "Recognizing Activities", Proc. IEEE In Conf. Compu Vision, 1994.

64 PINHANEZ, C. and A. Bobick, "Human Action Detection using PNF Propagation Temporal Constraints", Proceedings of IEEE Computer Vision and Pattern Recognitio pp. 898-904, Santa Barbara , CA, 1998.

65 PUJOL, A., F. Lumbreras, X. Varona and J. Villanueva, " Locating People in Indoor Scenes for Real Applications", Proceedings of the Internacional Conference on Pattern Recognition (ICPR'00), pp. 201-204, 2000.

66 QUECK, F., "Eyes in the Interface", Image and Vision Computing, vol. 13, 1995.

67 RABINER, Lawrence R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", IEEE ASSP Magazine, vol. 3, no. 1, pp. 267-295, 1989.

68 RAO, C. and Mubarack Shah, "View-Invariant Representation and Learning of Human Action", IEEE, PAMI, vol. 22, pp. 747-757, 2001.

69 ROBERTS, L.G., "Machine Perception of Three-Dimensional Solids", Optical and Electro-Optical Information Processing, MIT, Press, Cambridge, Massachusetts, pp.159-197, 1965.

70 ROSENBERG, Y. and M. Werman, "Real-Time Object Tracking from a Moving Video Camera: A Software Approach on a PC", Proc. IEEE Workshop Applications of Computer Vision, pp. 238-239, Oct., 1998.

71 ROWLEY, H., S. Baluja, and T. Kanade, "Neural Network-Based Face Detection", Proc. IEEE Conf. computer Vision and Pattern Recognition, pp. 203-208, 1996.

72 RUI, Y. and P. Anandan, "Segmenting visual actions based on spatio-temporal motion patterns", IEEE Conference on Computer Vision and Pattern Recognition, pp. 111-118, 2000.

73 SELINGER, A. and L. Wixson, "Classifying mOving Objects as Rigid or Non-Rigid without Correspondences", DARPA Image Understanding Workshop, Monterey, Calif., San Francisco: Morgan Kauffman, Nov. 1998.

74 SHAWN, A. and J. Cooperstock, "Presenter Tracking in a Classroom Environment", AAAI Symposium on Intelligent Environments, pp. 102-109, Stanford, CA, 1999.

75 SCHLENZIG, J., E. Hunter and R. Jain, "Vision Based Hand Gesture Interpretation using Recursive Estimation", IEEE 28th Asilomar Conference on Signals, Systems, and Computers, vol. 2, pp.1267-1271, 1994.

76 SHIMADA, N., K. Kimura, Y. Shirai, and Y. Kuno, "Hand Posture Estimation by Combining 2-D Appearence-Based and 3-D Model-Based Approaches", In Proc. Conf. on Automatic Face and Gesture Recognition, pp. 268-273, 2000.

77 STARNER, T. and A. Pentland, "Visual Recognition of American Sign Language using Hidden Markov Models", Proceedings of the International Workshop on Automatic Face and Gesture Recognition", Zurich, Switzerland, 1995.

78 STAUFFER, C., and E.L. Grimson, "Learning Patterns of Activity Using Real-Time Tracking", IEEE Trans. on Pattern Analysis and machine Intelligence, vol. 22, no. 8, Aug., 2000.

79 STILLMAN, S., and I. Essa, "System for Tracking and Recognizing Multiple People", Proc. IEEE 2nd Int'l Audio and Video-Based Biometric Person Authentication, pp. 96-101, Washington D.C., Mar, 1999.

80 STURMAN, D., D. Zelter and S. Pieper, "Hands-on interaction with virtual environments", In Proc. of the ACM Symposium on User

Análisis de estabilidad para problemas dinámicos de satisfacción de restricciones

Manuel Iván Angles Domínguez y Hugo Terashima Marín

Centro de Sistemas Inteligentes, ITESM Campus Monterrey
Ave. Eugenio Garza Sada 2501 Sur, C.P. 64849
Monterrey, Nuevo Leon., Mexico, Tel. +52 (81) 8328 4379
iangles@consultant.com, terashima@itesm.mx

Resumen Existen problemas de diferentes tipos que pueden ser modelados como problemas dinámicos de satisfacción de restricciones, dónde las restricciones, las variables o el dominio cambian con el tiempo. El objetivo al resolver estos problemas es disminuir el número de variables cuya asignación cambia entre problemas consecutivos, concepto conocido como distancia. En la actualidad este problema de estabilidad ha sido estudiado solamente para variaciones en las restricciones del problema. En este trabajo de investigación se realiza un amplio análisis del problema de estabilidad, al modificar las variables, dominios, restricciones y combinaciones de estos elementos para el problema de Asignación de Recursos. Se presentan resultados experimentales en cuanto a eficiencia, distancia y un nuevo parámetro denominado estabilidad global; para las técnicas de reuso de solución, reuso de razonamiento y una combinación de ambas. Además, se muestra que el comportamiento de la distancia es lineal con respecto a las variaciones.

1. Introducción

Un problema de satisfacción de restricciones (CSP por sus siglas en inglés, *Constraint Satisfaction Problem*) es un problema compuesto por un conjunto finito de variables, cada una de las cuales tiene asociado un dominio finito de valores y un conjunto de restricciones que limitan los valores que las variables pueden simultáneamente tomar. Formalmente [1] un problema de satisfacción de restricciones es una terna $C = (X, D, K)$, donde $X = \{x_1, \ldots, x_n\}$ es un conjunto de variables; $D = \{D_{x_1}, \ldots, D_{x_n}\}$ es un conjunto de dominios de las variables, uno para cada variable y K es un conjunto de restricciones, cada una de las cuales es un par $k = (Y, R)$, donde $Y = \{x_{i1}, \ldots, x_{ik}\}$ es el conjunto de variables que intervienen en la restricción y R es una relación sobre los dominios de estas variables, es decir, $R \subset D_{x_{i1}} \times \ldots \times D_{x_{ik}}$

Una solución a un CSP es una asignación de un valor de su dominio para cada variable, de tal forma que todas las restricciones son satisfechas al mismo tiempo.

Un DCSP [2] es una serie de CSP estáticos que cambian intermitentemente sobre el tiempo, debido a la evolución de sus componentes (variables, dominios

Gelbukh, M. Hernández Cruz (Eds.) Avances la

restricciones) dado cambios producidos por el ambiente (cambios en el conjunto de tareas a ser ejecutadas y/o de sus condiciones de ejecución, como en problemas de asiganación de recursos), un usuario (tratándose de un problema interactivo) u otro proceso (el DCSP sólo representa una parte de un problema más grande y otra entidad involucrada en el proceso de resolver el problema global, cambia su pensamiento de cómo el proceso de solución debería ser hecho; por ejemplo, un problema multiagente distribuido). En su trabajo original Dechter y Dechter [2] consideran únicamente modificaciones en las restricciones de los CSPs. En total, existen seis posibles alteraciones entre CSPs de un DCSP: adición o disminución de variables, adición o disminución de valores en dominios, adición o disminución de restricciones. Jonsson y Frank [1] presentan una definición formal que considera todas las posibles alteraciones:

> Sea $C = (X, D, K)$ un problema de satisfacción de restricciones. Cualquier problema de la forma $C' = (X', D', K')$ tal que $X' \supseteq X$ (i.e. hay más variables), $D'_x \subseteq D_x$ para cada $x \in X$ (i.e. hay menores valores en el dominio) y $K' \supseteq K$, (i.e. más limitaciones entre las variables) es una restricción de C. Un problema de la forma $C' = (X', D', K')$ tal que $X' \subseteq X$ (i.e. hay menos variables), $D'_x \supseteq D_x$ para cada $x \in X$ (i.e. hay más valores en los dominios) y $K' \subseteq K$, (i.e. menos limitaciones entre las variables) es una relajación de C. Un Problema Dinámico de Satisfacción de Restricciones es una secuencia de problemas de satisfacción de restricciones $C_0, C_1, \ldots$, tal que cada problema C_i es una restricción o relajación de C_{i-1}.

Solucionar un DCSP consiste en encontrar la solución de cada uno de los CSPs que lo conforman. Una forma de hacer esto es resolver cada CSP, uno tras otro desde el principio, enfoque que presenta dos defectos: ineficiencia y falta de estabilidad. En muchos casos algo del trabajo hecho en el previo CSP podría ser de alguna forma reusado para resolver el nuevo CSP, ya que para aplicaciones en tiempo real, el tiempo para resolver el nuevo problema es limitado. En cuanto a estabilidad, si la solución del previo CSP representa una asignación actualmente bajo ejecución, cualquier nueva solución debería minimizar el esfuerzo necesario para aplicar la reciente asignación. Para problemas interactivos descentralizados es preferible tener la menor cantidad de modificaciones a la actual solución.

La falta de estabilidad en las soluciones (conocido también como distancia entre soluciones) es un tópico de suma importancia, debido al costo en dinero y esfuerzo, que representa modificar la actual asignación, a fin de cumplir con los nuevos requerimientos; especialmente cuando típicamente no hay modelo del futuro, es decir, no hay ningún conocimiento acerca de cuándo y qué modificaciones se presentarán, lo que dificulta el desarrollo de soluciones que sean robustas a los cambios.

Para de evitar estos problemas actualmente existen dos enfoques para aprovechar alguna característica de un CSP para resolver el siguiente CSP que se produzca [3]. El primero de ellos, denominado reuso de solución, utiliza la solución previa (si hay alguna) para determinar la nueva solución. Verfaillie y

Schiex [4] desarrollaron un algoritmo basado en el enfoque de reuso de solució denominado *LocalChanges*, que esta inspirado en técnicas de *Backjumping* la variable actual viola una restricción se revisa la asignación de la variable co la que se tiene el conflicto, no la asignación de la variable inmediata anterior) de reparación iterativa [5] (consiste en utilizar una asignación previa y repararla usando una secuencia de modificaciones locales, es decir, modificaciones de sól una variable asignada).

El segundo enfoque, denominado reuso de razonamiento, trata de guarda de una manera concisa, una descripción aproximada de la frontera del es pacio de solución y justificaciones de esta frontera en términos del conjunto de restricciones. Schiex y Verfaillie [6] desarrollaron un método denominado *NogoodBuilder*, que consiste en ir generando a partir de las restricciones del CSP un conjunto de *nogoods*, definidos como instanciaciones parciales de vari ables que no pueden ser extendidas a una solución de todo el conjunto de vari ables. Estos *nogoods* pueden ser adicionados como restricciones al siguiente CSP generado.

Estos enfoques han sido examinados para variaciones de las restricciones de los CSPs, pero actualmente, no existe un análisis detallado de dichos algoritmos. El interés de este trabajo es analizar el desempeño de estos dos enfoques, con respecto a eficiencia y estabilidad, para modificaciones en las variables, dominio y restricciones de los CSPs, y además, proponer y analizar nuevas maneras de integrar estos enfoques, de tal manera que se obtenga un mejor desempeño que los algoritmos actuales.

Este artículo se encuentra organizado de la siguiente manera. En la sección 2, se establece la metodología que se siguió para realizar un análisis de estabil idad sobre los enfoques para resolver DCSP. En la sección 3 se presentan los resultados del análisis de estabilidad. Finalmente, las conclusiones del trabajo de investigación son presentadas en la sección 4.

2. Metodología

El primer paso consistió en modelar el problema de Asignación de Recursos (AR) como un DCSP. Posteriormente se desarrolló un generador aleatorio uniforme para crear los problemas de prueba de acuerdo al modelo de AR. A partir del análisis de los algoritmos basados en los enfoques de reuso de solución y reuso de razonamiento se desarrollaron dos nuevos algoritmos que integran los dos reusos en el mismo procedimiento. Entonces el análisis de estabilidad se realizó sobre los cuatro algoritmos y consistió principalmente en dos subanálisis: un análisis de sensibilidad y un análisis de distancia.

2.1. El problema de asignación de recursos

La asignación de recursos, es conocida por ser un problema difícil de gran importancia práctica, ya que existen un amplio rango de dominios que pueden ser modelados como problemas AR (telecomunicaciones satélitales [7], ruteo dinámico en

redes computacionales [8]). El problema de asignación de recursos [9] consiste en destinar recursos a un conjunto de tareas, programadas en intervalos de tiempo dados, de tal forma que ningún recurso es asignado a dos diferentes tareas al mismo tiempo. Este problema, así como los CSPs y DCSP son NP completos 0], lo que significa que para dimensiones reales no pueden ser óptimamente resueltos en una razonable cantidad de tiempo. La complejidad de cualquier procedimiento para resolver el peor caso posible para estos tipos de problemas es exponencial. Formalizando el modelo tenemos: Variables: $\{x_1, \ldots, x_n\}$; donde x_i representa la tarea i de un total de n tareas. Dominios: $D_{x_i} = \{R_1, \ldots, R_m\}$; donde R_j representa el recurso j de un total de m recursos que a la tarea x_i se puede asignar. Restricciones: $x_i \neq x_j$ si $i \neq j$; establece que las tareas x_i y $_j$ no pueden ocupar el mismo recurso, debido a que las dos se están llevando a cabo simultáneamente. Con el fin de extender este modelo y que los resultados puedan ser aplicados a un mayor número de dominios se consideraron otros dos pos de restricciones: $>$ y $<$. Por lo tanto, el problema de asignación de recursos modelado como DCSP, consiste en una serie de CSPs definidos por el modelo anterior, los cuales sufren modificaciones en sus variables, dominios o restricciones través del tiempo.

Se desarrolló un generador de DCSP, que crea instancias aleatorias uniformes de CSPs binarios, basados en las características del problema de asignación de recursos. El generador está habilitado para aumentar o disminuir entre CSPs número de variables, el tamaño del dominio, el número de restricciones y cualquier combinación de estos tres conceptos, ya que cada uno de ellos es manejado independientemente. El tamaño de dominio más pequeño que puede tener una variable es de 2 elementos. Tal valor fue elegido con el fin de no tener demasiado restringido los dominios y permitir un mayor trabajo de los algoritmos. Los valores que pueden tomar van de 1 al tamaño máximo de dominio especificado.

2. Algoritmos analizados

unto a los algoritmos *LocalChanges* [4] y *NogoodBuilder* [6], se desarrollaron y analizaron dos nuevos algoritmos. El primero de ellos, denominado *Lopt* está basado en *LocalChanges* y junto con un método para generar *nogoods* implícito en la técnica de reparación iterativa, hace reuso de razonamiento también. Su método para generar *nogoods* poda más espacio de búsqueda que el de *NogoodBuilder*. El segundo algoritmo desarrollado, denominado *Nopt*, complementa a *NogoodBuilder* haciendo reuso de solución, aprovechando la naturaleza recursiva del algoritmo. Estos dos nuevos algoritmos utilizan el reuso de solución cuando se tienen relajaciones en el DCSP y el reuso de razonamiento en el caso de restricciones, permitiéndoles evitar iniciar desde cero la solución de cualquier CSP del DCSP.

3. Análisis de sensibilidad y análisis de distancia

La sensibilidad de los algoritmos fue medida con base en los siguientes tres parámetros:

a) Estabilidad (Distancia): Distancia entre dos soluciones sucesivas, es decir el número de variables con diferente asignación entre soluciones de 2 CSPs contiguos.
b) Generación nula: Número de veces que la distancia entre CSPs contiguos fue cero. Actualmente, el concepto de estabilidad considera únicamente el número de variables cuya asignación varía entre soluciones de dos CSPs contiguos, sin importar cuántas veces dicha variación se presenta durante la solución completa del DCSP. Por eso, junto a la evaluación de la estabilidad se propone una nueva métrica para medir su frecuencia durante la resolución del DCSP.
c) Verificaciones de consistencia: Número de veces en que un algoritmo determina si un valor potencial para una variable viola alguna restricción con alguna otra variable.

Para tener una perspectiva más amplia acerca del desempeño de los algoritmos y especializarlos de acuerdo al tipo de variaciones que puede tener un DCSP, su sensibilidad a los cambios fue evaluada para las siguientes modificaciones en el DCSP: número de variables (V), tamaño de los dominios (D), número de restricciones (R), número de variables y del tamaño del dominio (V,D), número de restricciones y del tamaño del dominio (R,D), número de variables y del número de restricciones (V,R), número de variables, número de restriccicones y tamaño del dominio (V,R,D). Considerando que en la práctica la mayoría de los dominios sufren cambios lentamente y sólo en ocasiones especiales los cambios pueden salirse del promedio, se estableció que el máximo número de cambios (número de variables o restricciones que aumentan o disminuyen, número de variables de dominio será modificado y número de elementos que son adicionados al dominio) entre un CSP y otro es de tres.

Con base en que los cambios entre los CSP's son aleatorios y uniformes se estableció un conjunto de 30 DCSP de prueba para cada tipo de modificación. Cada DCSP está formado por 20 CSPs binarios creados por el generador aleatorio uniforme de DCSP. El CSP inicial está formado por 40 variables, tamaño máximo de dominio igual a 5 y 14 restricciones. El número de variables se cogió para dar una mayor amplitud a la variación de este concepto, de acuerdo al máximo número de cambios permitidos, que es de 3. Como el generador uniforme, podemos considerar un cambio promedio de 2 variables y que durante 10 de los 20 CSP que conforman el DCSP se aumentan variables y durante otros 10 se disminuye. Con estas características se esta manejando una variación de variables de alrededor del 50 % de las variables iniciales. El tamaño máximo de dominio fue escogido arbitrariamente para que con base en él se pudieran cular el número de restricciones. Un número de 14 restricciones fue determinado experimentalmente, tratando de obtener aproximadamente un 50 % de solubilidad en el DCSP, es decir, que alrededor de 10 de los 20 CSPs tuvieran solución. Experimentalmente se observó que el tiempo de resolución de los problemas varíaba entre 0 y 850 milisegundos. Con base en esto se estableció que los algoritmos tuvieran un tiempo límite para resolver el CSP_i que varía aleatoriamente entre 1 y 1000 milisegundos antes de pasar a resolver el siguiente CSP_{i+1}.

El objetivo del Análisis de Distancia es determinar la forma en que varía la distancia entre dos CSPs. Se consideró el caso de restricciones entre CSPs, ya que como se mencionó en la sección 2.3.1, en caso de relajación, simplemente se utiliza la solución del anterior CSP, por lo que la distancia entre soluciones es cero. La distancia fue evaluada para las siguientes modificaciones en el DCSP: aumento en el número de variables en 2, 4, 6, 8, 10, 12, 14, 16, 18 y 20 variables. Aumento en el número de variables cuyo dominio será modificado en 2, 4, 6, 8, 10, 12, 14, 16, 18 y 20 variables. El número de valores que fueron disminuidos del dominio fué seleccionado aleatoriamente para cada variable. Aumento en el número de restricicones en 1, 2, 3, 4, 5, 6, 7 restricciones.

Se eligieron estas modificaciones para abarcar una variación total del 50% de los parámetros iniciales. Para cada una de las posibles modificaciones mencionadas anteriormente se evaluaron 300 DCSPs modelando el problema de Asignación de Recursos, lo que significa que las restricciones manejadas fueron: $>, <, \neq$. Cada DCSP estaba formado por 2 CSPs. El CSP inicial está formado por 40 variables, tamaño máximo de dominio igual a 5 y 14 restricciones. Las razones para la elección de estos parámetros se describen en el análisis de sensibilidad.

Con el fin de analizar cómo afectaba a la distancia el hecho de utilizar los dos enfoques para resolver DCSP con repecto a emplear uno solo, el análisis se aplicó a los algoritmos *Lopt* (que integra reuso de solución y de razonamiento) y *Localchanges* (solo emplea reuso de solución). No se consideró tiempo límite para la solución de los CSPs.

3. Resultados y discusión

En el análisis de sensibilidad se obtuvo el comportamiento de los algoritmos en cuanto a eficiencia, estabilidad (tambien conocida como distancia) y generación nula; al variar las variables, dominios, restricciones y combinaciones de estos conceptos. El segundo grupo de experimentos fue enfocado a un análisis de distancia que se realizó para determinar si existe alguna relación que permita calcular el número de asignaciones que cambiarán entre soluciones de 2 CSPs.

3.1. Análisis de sensibilidad

En la comparación de los parámetros de medición del análisis de sensibilidad se analiza el promedio del desempeño de los algoritmos sobre el conjunto de los 30 DCSP. En el caso de la distancia entre soluciones, para cada DCSP, se considera un promedio sobre el número de veces que la distancia no fue nula.

Eficiencia.- En la figura 1 se grafican el número de verificaciones de consistencia que realizaron los algoritmos para cada una de las modificaciones analizadas.

En general se observa que *NogoodBuilder* y *Nopt* realizan un mayor número de verificaciones que *LocalChanges* y *Lopt*. Esto es debido al proceso de eliminación de consistencia de cada algoritmo, *Backjumping* para *NogoodBuilder*

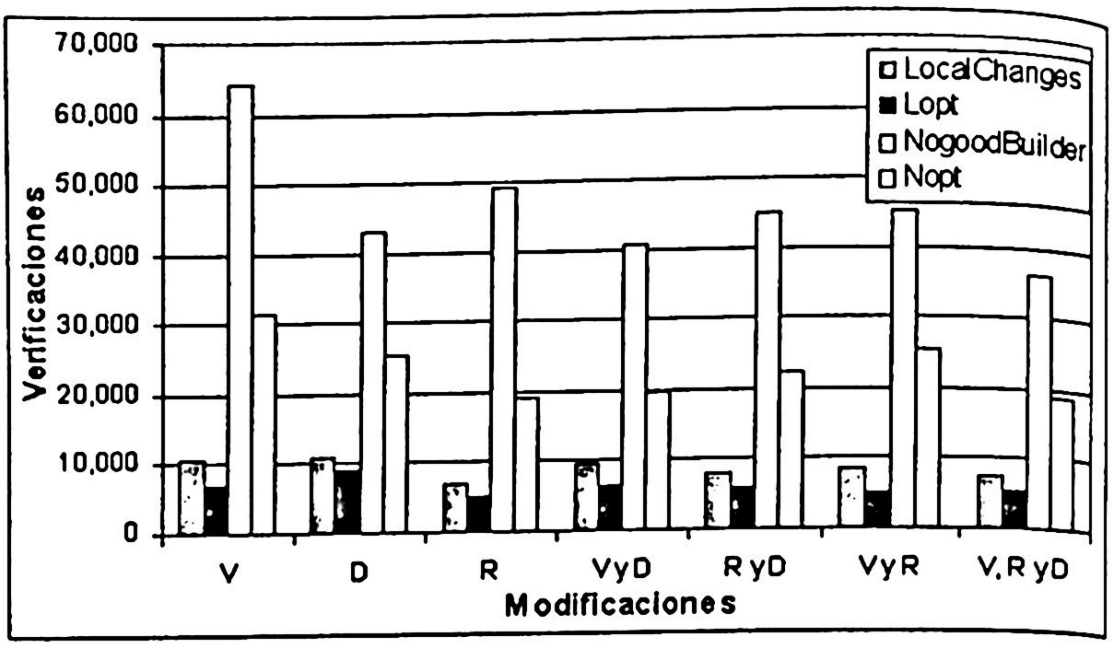


Figura 1. Resultados de eficiencia para el análisis de sensibilidad.

y *Nopt* y Reparación Iterativa para *LocalChanges* y *Lopt*, siendo este últim el que hace menos verificaciones de consistencia tal y como se observa en figura 1. Parte del ahorro de verificaciones es debido también a los enfoque de Reuso de Solución (RS) y Reuso de Razonamiento (RR). La diferencia entr *NogoodBuilder* y *Nopt* representa el trabajo que *Nopt* se ahorra por utilzar RS; es de aproximadamente el 50% de las verificaciones hechas por *NogoodBuilde Lopt* obtiene un ahorro de 30% sobre el trabajo hecho por *LocalChanges* debid al uso de RZ. El mayor ahorro que se presenta cuando se utiliza RR es debido que RZ solo poda ciertas ramas del árbol de búsqueda, mientras que RR elimin prácticamente todo el árbol cuando se tienen relajaciones entre los CSPs DCSP.

Distancia.- En la figura 2 se gráfican la distancia promedio que obtuviero los algoritmos para cada una de las modificaciones analizadas. En general observa que *LocalChanges* y *Lopt* generan una mayor distancia entre solucione contiguas. Esto puede deberse al proceso de búsqueda de cada algoritmo, para *Lopt* y *LocalChanges* es Reparación Iterativa (RI) y para *NogoodBuilde* es *Backjumping*. Entre *LocalChanges* y *Lopt* existe una pequeña diferencia cuál es producida por el reuso de razonamiento por parte de *Lopt*, ya que uso de *nogoods* implica no tomar en cuenta algunos valores de los dominios que obliga a asignar otros valores a las variables, aumentando la distancia entr las soluciones. En cambio, el reuso de solución que *Nopt* utiliza no afecta desempeño con respecto a *NogoodBuilder*, ya que este reuso evita precisamen que exista distancia entre las soluciones.

Generación Nula.- En la figura 3 se gráfican el número de CSPs cuy solución no sufrió modificaciones con respecto a la solución anterior. Entr *LocalChanges* y *Lopt* existe una pequeña diferencia la cuál es producida el reuso de razonamiento por parte de *Lopt*. Supongamos que el CSP_i no tie

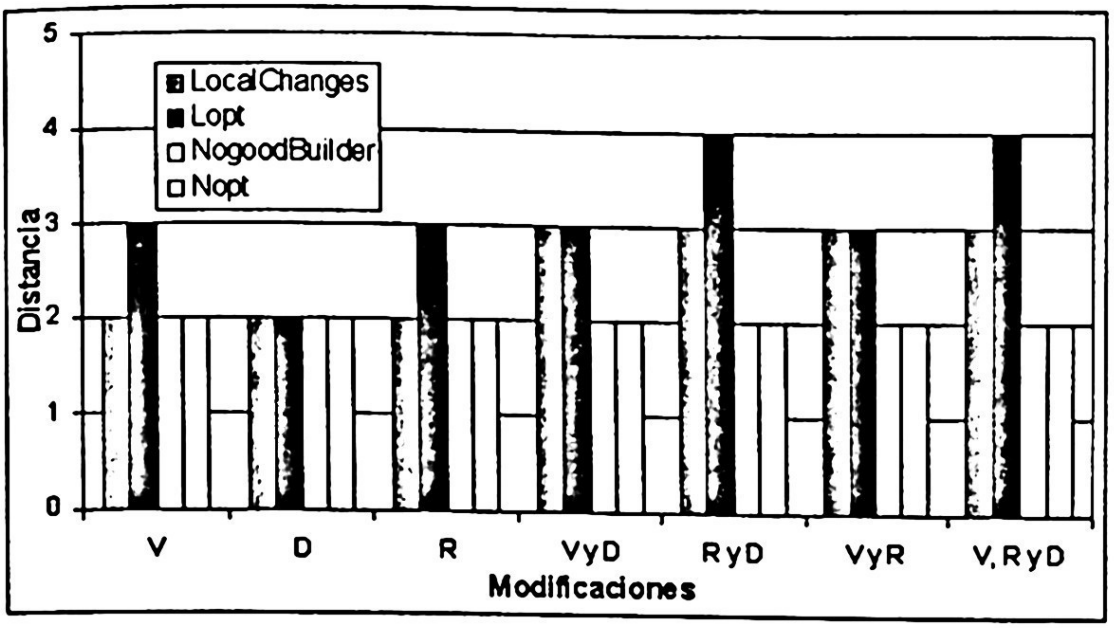


Figura 2. Resultados de distancia para el análisis de sensibilidad.

solución, pero se genera una instanciación parcial de variables de acuerdo al trabajo de procesamiento que llevaba el algoritmo antes de encontrar la falla. Para *LocalChanges* en CSP_{i+1} la nueva modificación puede permitir al algoritmo dar nuevos valores a las asignaciones antes de llegar a la inconsistencia. Al comparalas se encuentra que son diferentes disminuyendo la generación nula. En cambio, *Lopt* genera *nogoods* que le permiten desde el inico de solución de CSP_{i+1} determinar que no existe solución, por lo que no permite procesamiento que permita modificación de variables y se mantiene la generación nula. En promedio *Lopt* obtiene un aumento de 17% en la generación nula con respecto a *LocalChanges* y *Nopt* alrededor de un 14%, con respecto a *NogoodBuilder*, debido al reuso de solución.

El análisis de sensibilidad demostró que la utilización de integrar los enfoques de reuso de solución y reuso de razonamiento producen un mayor desempeño que utilizar un solo enfoque. Puesto que reuso de solución elimina verificaciones de consistencia y produce distancias cero entre soluciones aumentando la generación nula, es un enfoque que cualquier algoritmo para resolver DCSP debería utilizar. Esto conduce a enfocarnos a los casos de restricción de CSPs, en donde un algoritmo que utilice *nogoods*, con un proceso de reparación iterativa lo mas sistemático posible sería la mejor opción. Una propuesta para lograr esto es ajustar Reparación Iterativa para que empieze buscando soluciones cercanas a la establecida por una guía de *nogoods* y de ahí vaya ampliando su alcance de búsqueda. Los resultados obtenidos son independientes del tamaño del DCSP (en este caso fue de 20 CSPs), ya que las mediciones se hacen sobre CSPs contiguos, sin importar cuantos CSPs conforman el DCSP. Por otra parte, consideremos un DCSP cuyo CSP inicial tiene 100 variables. El comportamiento de los algoritmos al iniciar el CSP de 100 variables sería el mismo que si se considerara a este CSP como producto de otro DCSP que originalmente se inició con 40 variables (como

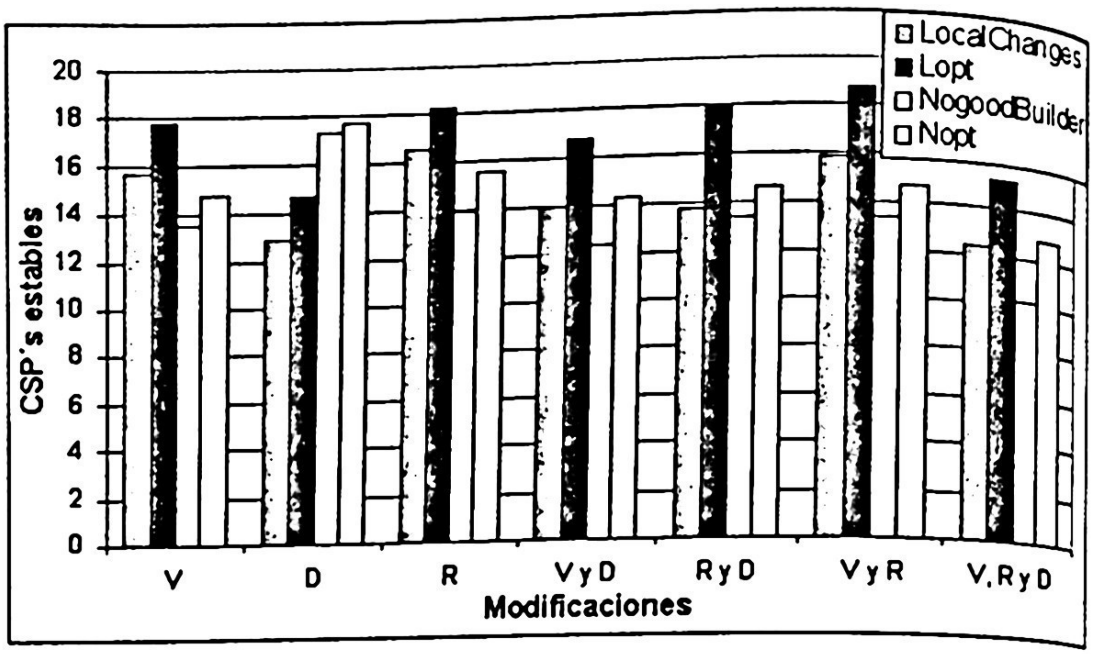


Figura 3. Resultados de generación nula para el análisis de sensibilidad

los examinados en este trabajo) y que ha sufrido modificaciones hasta llegar a variables. Por lo tanto, para CSPs con mayor número de variables se mantiene los resultados obtenidos en este trabajo. Un análisis semejante puede aplicar al tamaño del dominio y número de restricciones.

3.2. Análisis de distancia

En la figura 4 se muestra como aumenta la distancia al aumentar el núme de variables. Por ejemplo, si se aumenta el número de variables en un 40%, tiene un aumento de 20% en la distancia. De esta forma, para un CSP de variables, si una modificación produce un nuevo CSP con 16 nuevas variables, promedio, podría generarse una distancia de 8 variables con distinto valor entr las soluciones.

En la figura 5 se presenta el comportamiento de la distancia al variar número de variables a las que se les modifico su dominio. En este caso se obtie un mayor aumento en la distancia que el producido para la variación de variable En el caso de la figura 6, se ilustra la relación del incremento en la distancia modificar el número de restricciones.

Con base en las figuras 4, 5 y 6; se puede concluir que en general, el com portamiento de la distancia es lineal para los tres tipos de variaciones que analizaron. El análisis de distancia para restricciones en DCSP, demuestra la distancia entre soluciones varía de forma lineal con respecto a variacione en las variables, dominios y restricciones. Esta es la primera prueba de que posible determinar con base al número de elementos que varían, cual será la tancia que se produciría entre soluciones. Este comportamiento está acotado problema de Asignación de Recursos, por lo que futuros análisis para otros prob lemas modelados como DCSP y otros algoritmos son necesarios para verificar

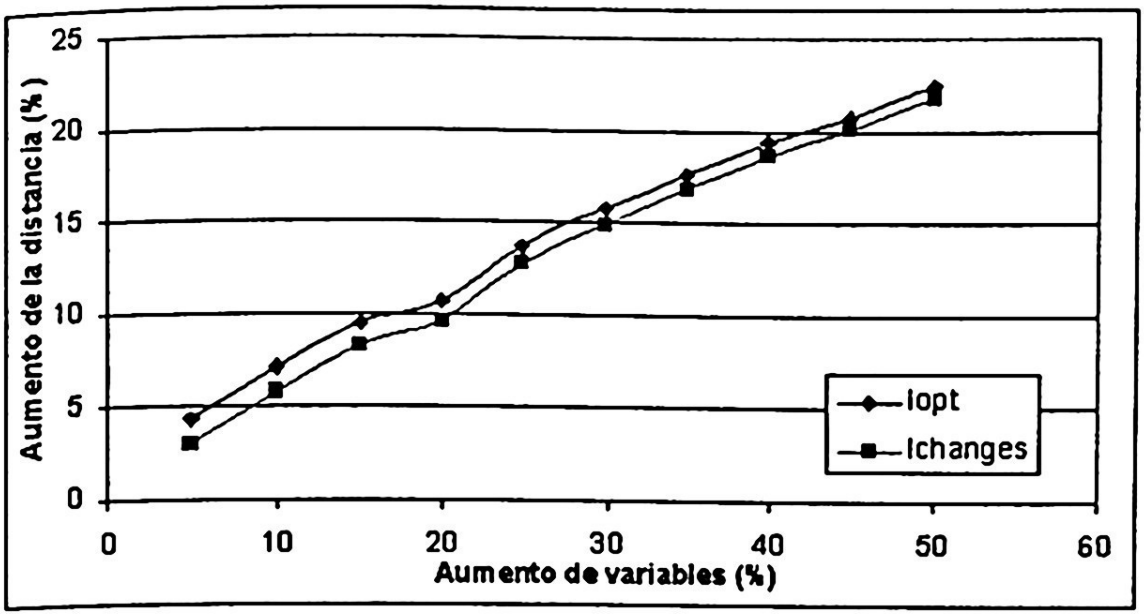


Figura 4. Comportamiento de la distancia al variar el número de variables.

la relación es siempre válida o bajo qué condiciones varía. El hecho de que para *LocalChanges* la relación sea lineal, implica que cualquier algoritmo que mejore *LocalChanges* tendrá que generar una relación con menor pendiente o en su mejor caso, una relación constante. Por otra parte, esta relación empírica podría ser utilizada como una heurística que permita minimizar la estabilidad local utilizando un algoritmo A* por ejemplo, el cuál minimiza una función objetivo utilizando información a través de una heurística.

4. Conclusiones

En este trabajo de investigación se desarrolló un análisis de estabilidad sobre cuatro algoritmos para resolver DCSP. Las pruebas realizadas sobre DCSP aleatorios han mostrado que ningún algoritmo obtiene el mejor desempeño en las tres medidas de comparación. En el caso de eficiencia, la técnica de reparación iterativa (*Lopt* y *LocalChanges*) obtiene un mejor desempeño que la ténica de *backjumping* de *NogoodBuilder* y *Nopt*. Por el contrario, *backjumping* obtiene menores distancias entre soluciones que reparación iterativa. En especial se ha observado que las características deseables en un algoritmo para resolver DCSP son: reuso de solución para el caso de relajaciones y reuso de razonamiento con un proceso de reparación iterativa sistemático para los casos de restricciones. Se ha verificado también la importancia del parámetro de medición propuesto, ya que los resultados indican que un algoritmo que produce buena estabilidad en el DCSP, no siempre producirá menor distancia entre soluciones. De esta manera, la generación nula permite elegir, de acuerdo al dominio de aplicación, que algoritmo es más conveniente de acuerdo a si se requiere mayor estabilidad durante todo el proceso de solución o menor estabilidad en pocas ocasiones. Con base en análisis de distancia se ha comprobado que existe una relación lineal entre el

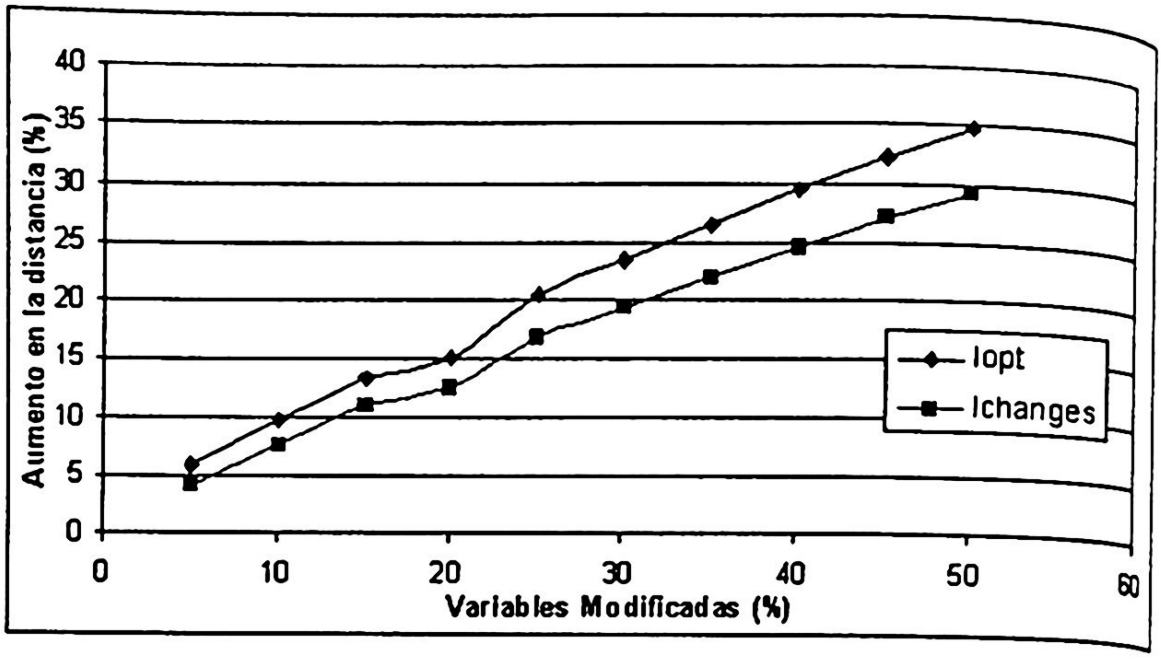


Figura 5. Comportamiento de la distancia al variar el tamaño del dominio.

incremento en la distancia de las soluciones con respecto a la variación individual del número de variables, tamaño de dominio y número de restricciones. La relación encontrada permite establecer, en forma promedio, el comportamiento de la distancia entre soluciones, lo que coadyuvará a una mejor toma de decisiones, en el caso de que las modificaciones puedan ser realizadas voluntariamente. El análisis de los algoritmos se realizó con el problema de asignación de recursos en su forma más general, es decir, sin especificar ningún dominio en particular; por lo que los resultados obtenidos pueden ser aplicados para cualquier dominio cuyo problema pueda ser presentado como asignación de recursos o en su caso a problemas que puedan ser modelados como CSPs binarios con las siguientes restricciones: $>, <, =$. Posibles extensiones a este trabajo incluyen realizar análisis de sensibilidad con otro tipos de problemas, que manejen otro tipo de restricciones o tamaño de dominio. Analizar si la relación de distancia se mantiene para otros tipos de problemas. Determinar límites inferiores o superiores al comportamiento de la distancia, así como desarrollar una teoría analítica de tal comportamiento. Este tipo de cuestiones requerirán respuesta con análisis más extensos en cuanto a formas del problema y tipos de modificaciones.

Agradecimientos

El primera autor agradece los apoyos recibidos del ITESM por la beca de Excelencia otorgada, del Centro de Sistemas Inteligentes del ITESM y de la Cátedra de Investigación CAT010.

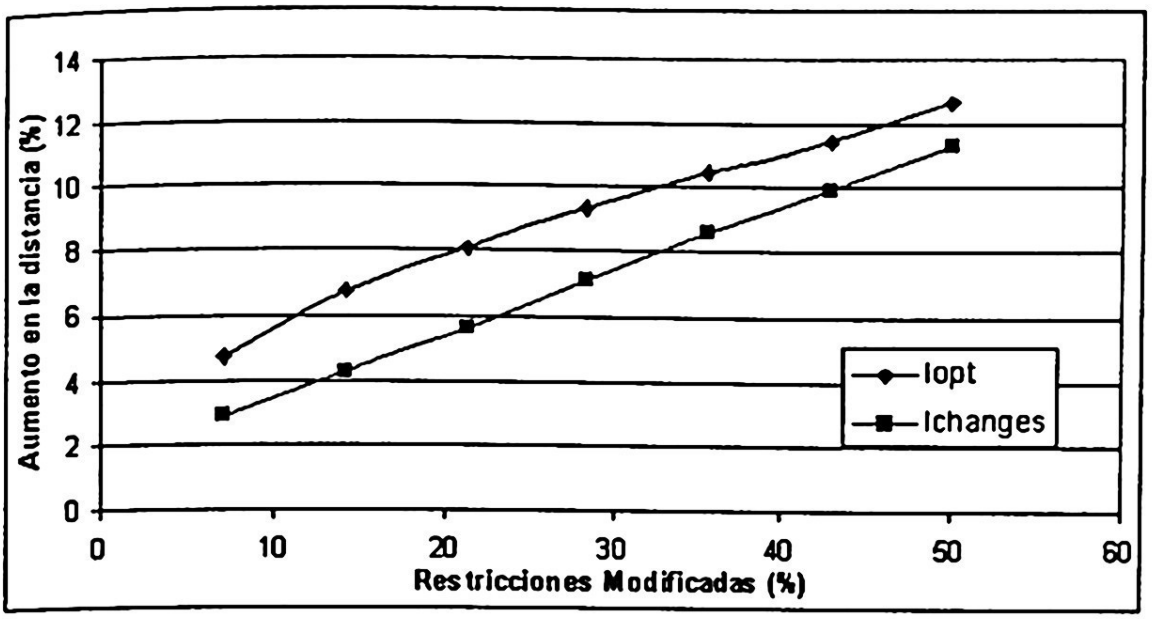


Figura 6. Comportamiento de la distancia al variar el número de restricciones.

Referencias

1. A.K. Jonsson and J. Frank. A framework for dynamic constraint reasoning using procedural constraints. In *European Artificial Intelligence Conference*, 2000.
R. Dechter and A. Dechter. Belief maintenance in dynamic constraint networks. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pages 37–42, 1988.
T. Schiex and G. Verfaillie. Two approaches to the solution maintenance problem in dynamic constraint satisfaction problems. 1993.
G. Verfaillie and T. Schiex. Solution reuse in dynamic constraint satisfaction problems. In *AAAI, Vol. 1*, pages 307–312, 1994.
S. Minton, M. D. Johnston, A. B. Philips, and P. Laird. Minimizing conflicts: A heuristic repair method for constraint satisfaction and scheduling problems. *Artificial Intelligence*, 58(1-3):161–205, 1992.
T. Schiex and G. Verfaillie. Nogood recording for static and dynamic constraint satisfaction problems. 1993.
. C. Plaunt, A.K. Jonsson, and J. Frank. Run-time satellite tele-communications call handling as dynamic constraint satisfaction. In *Proceedings of the IEEE Aerospace Conference 1999*, pages 165–174, 1999.
Murat Alanyali and Bruce Hajek. On simple algorithms for dynamic load balancing. In *INFOCOM (1)*, pages 230–238, 1995.
B. Y. Choueiry, G. Noubir, and B. Faltings. Blending AI and mathematics: the case of resource allocation. In *Fourth International Symposium on Artificial Intelligence and Mathematics*, pages 32-37, Fort Lauderdale, Florida, 1996.
C. P. Gomes and J. Hsu. ABA: An assignment based algorithm for resource allocation. *SIGART Bulletin*, 7(1):2–8, 1996.

Análisis de terminación de reglas, un enfoque con red de Petri

Joselito Medina Marín y Xiaoou Li

Sección de Computación
Departamento de Ingeniería Eléctrica
CINVESTAV-IPN
A.P. 14-740, Av.IPN 2508, México D.F., 07360, México
jmedina@computacion.cs.cinvestav.mx, lixo@cs.cinvestav.mx

Resumen El análisis de reglas en bases de datos activas es muy importante, ya que pueden detectarse problemas como el de No-terminación. Este problema se origina porque la relación que guardan las reglas, la acción de una regla es el evento que activa a otra, genera cadenas entre ellas y si una es disparada, entonces se produce un disparo consecutivo de reglas hasta volver con la primera regla disparada, provocando un disparo inifinito de reglas. El problema de No-terminación debilita al sistema manejador de la base de datos, por muy robusto que éste sea. En este artículo presentamos el análisis de la no-terminación en el disparo de reglas, utilizando los fundamentos teóricos de la Red de Petri Coloreada Condicional. Se propone como método de análisis la aplicación de la matriz de incidencia en la red de Petri, para detectar la no terminación del disparo de reglas en bases de datos activas.

1. Introducción

Las bases de datos activas (BDA), a diferencia de las bases de datos (BD) tradicionales, tienen la capacidad de reaccionar ante la ocurrencia de eventos interés y, si se presentan las condiciones apropiadas, se ejecutan una o varias acciones en la BD [1] [2]. Esta capacidad que tienen las BDA se logra mediante la definición de una base de reglas *evento-condición-acción* (reglas ECA ó reglas activas), cuya forma general es:

on evento
if condición
then acción

En las cuales se especifica el evento que se debe detectar, la condición que se evaluará y la acción que será ejecutada en la BD. El desarrollo de una base reglas ECA es una tarea sencilla cuando se define un reducido número de reglas sin embargo, si se desarrolla una base de reglas con un número considerable elementos, es posible perder el control que se tenga sobre las relaciones existentes entre las reglas, ocasionando dificultades para detectar la presencia de ciclos infinitos durante el procesamiento de reglas y, en consecuencia, el problema

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 56-67, 2003. © Centro de Investigación en Computación, IPN, México.

no-terminación. La no terminación del disparo de reglas conduce al sistema manejador de BD a un estado inconsistente, provocando caídas al sistema y pérdida de información, entre otras cosas.

Un modelo estructural para modelar las reglas ECA fue propuesto por los mismos autores de este artículo, denominado Red de Petri Coloreada Condicional (CCPN) [3],[4]. Una CCPN es una extensión de una red de Petri (PN) tradicional, con capacidades para modelar reglas ECA. En este artículo se analiza la información almacenada en la matriz de incidencia de la CCPN, la cual nos ayuda a determinar si el disparo de reglas activas termina o no. Además, en la CCPN se realiza el análisis de terminación de una base de reglas activas. Para llevar a cabo este análisis, se propone un algoritmo para encontrar todas las rutas cíclicas de la CCPN a partir de la matriz de incidencia respectiva. A continuación se muestra la definición de la CCPN, presentada en [5].

1. Red de Petri coloreada condicional

na PN es un tipo particular de gráfica dirigida bipartita, compuesta por dos tipos de objetos. Estos objetos son **lugares y transiciones**, los arcos que los unen están dirigidos de lugares a transiciones o de transiciones a lugares. Gráficamente, los lugares se representan por círculos y las transiciones por barras rectángulos. En su forma más simple, una PN se representa por una transición unida con sus lugares de entrada y lugares de salida. Esta red básica se utliza para representar varios aspectos de un sistema que se pretende modelar. Con la finalidad de estudiar el comportamiento dinámico de los sistemas modelados, cada lugar puede tener cero o más tokens. Los tokens se representan gráficamente por pequeños círculos rellenos. La presencia o ausencia de un token dentro de un lugar indica si una condición asociada con este lugar es falsa verdadera, ó también nos indica si un dispositivo que se está modelando se encuentra disponible o no. Una extensión de PN para modelar reglas ECA y a vez llevar a cabo la simulación del comportamiento de las mismas es la Red Petri Coloreada Condicional (CCPN) [3] [4]. La extensión CCPN se propuso para ofrecer un modelo estructural de representación de reglas activas. El modelo original de PN no cubre el manejo de información (colores) en los tokens. Para modelación de sistemas donde es necesario manejar datos se crearon las redes Petri coloreadas (CPN). La CCPN, además de manejar información de la BD tal como una CPN modela un sistema con manejo de datos), también evalúa parte condicional de la regla. La condición de cada regla se almacena en las transiciones, donde se lleva a cabo su evaluación, auxiliándose de la información que contienen los tokens.

En la figura 1 se muestra un ejemplo con tres reglas activas. Las dos primeras reglas tienen el mismo evento que las activa (insert_VENTAS), por lo tanto, se crea un lugar p_0 para este evento, una transición copy t_0 para crear dos salidas p_0 y evaluar, por separado, a las reglas. El evento de la regla 3 es la acción la regla 1, por lo que el lugar de salida p_4 pasa a ser el lugar de entrada de la tercera regla, concatenndo así la primer y tercer regla. Cuando ocurre el evento insert_VENTAS, la información del registro agregado se almacenará en un token

que se desplazará de p_0 a t_0 y, a su vez, de t_0 se desplazará un token hacia p_1 y otro hacia p_2 (p_1 y p_2 representan copias del lugar p_0). Con la información de los tokens se evalúan las condiciones almacenadas en t_1 y t_2, y si éstas cumplen, entonces se generan tokens con información sobre la acción de la regla. Los tokens que lleguen a p_4 serán enviados a t_3 para evaluar la condición que tiene almacenada, y si el resultado es positivo, entonces se envía a p_5 el token referente a la acción de la regla. Antes de dar una definición formal de la CCPN

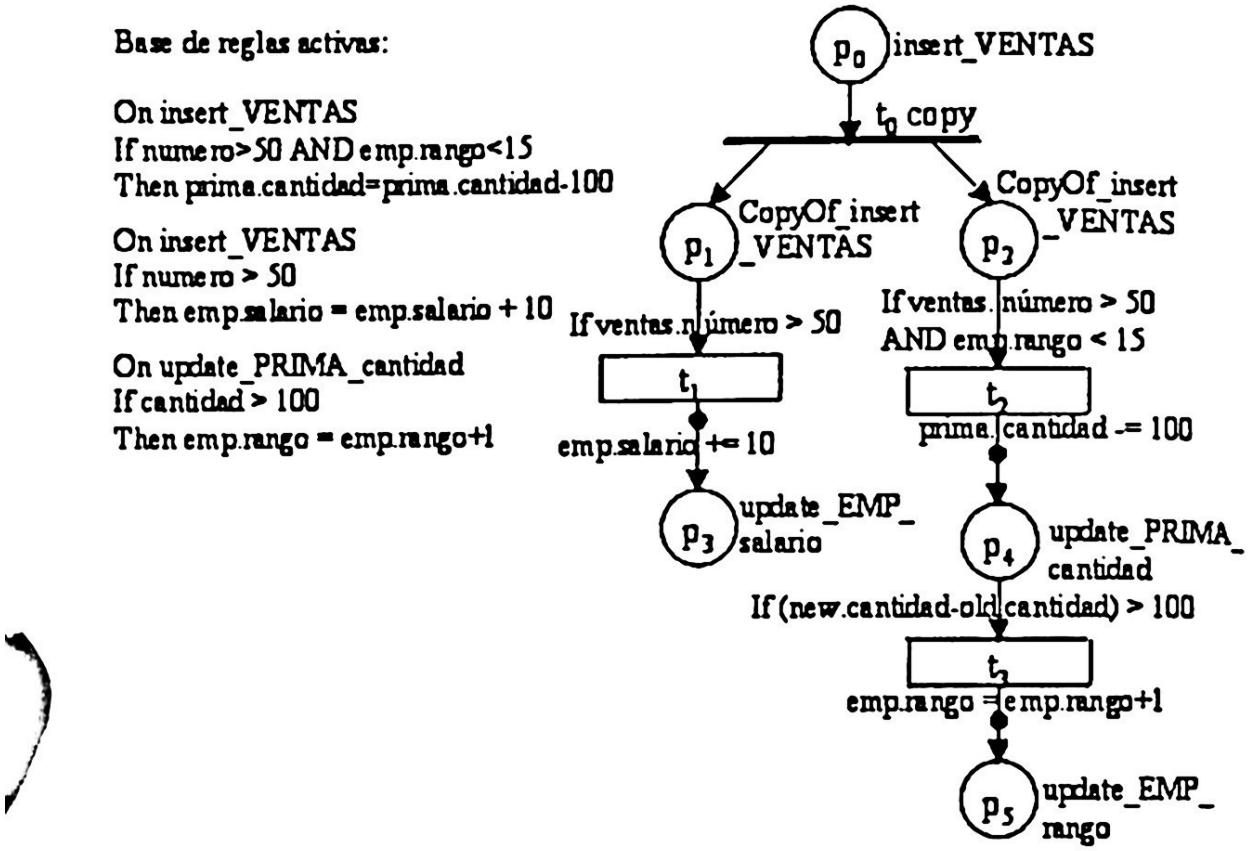


Figura 1. Ejemplo de una red de Petri Coloreada Condicional.

definiremos algunos elementos necesarios en la definición.

Definición 1. *Se le llama "color" a la estructura que tiene el token acorde lugar donde se coloca.*

Definición 2. *La CCPN tiene cuatro tipos de lugares:*
$P_{insert_tabla} = \{p_{it} \mid p_{it}$ *representa un lugar de tipo "INSERT"*$\}$,
$P_{update_tabla_columna} = \{p_{utc} \mid p_{utc}$ *representa un lugar de tipo "UPDATE"*$\}$,
$P_{delete_tabla} = \{p_{dt} \mid p_{dt}$ *representa un lugar de tipo "DELETE"*$\}$,
$P_{copy} = \{p_c \mid p_c$ *representa un lugar de tipo "COPY"*$\}$

Definición 3. *La CCPN tiene dos tipos de transición:*
$T_{regla} = \{t_r \mid t_r$ *representa la transición que almacena la parte condicional de regla ECA*$\}$ *y*
$T_{copy} = \{t_c \mid t_c$ *representa la transición que genera duplicados del token proveniente de su lugar de entrada*$\}$

Las transiciones $t \in T_{regla}$ se representan por un rectángulo. Mientras que las transiciones $t \in T_{copy}$ se representan por una línea gruesa.

Definición 4. *La CCPN, al igual que las PN, tiene arcos de entrada* $A_{in} = \{(p,t) \mid p \in P,\ t \in T,\ (p,t) \in \{PxT\}\}$ *y de salida* $A_{out} = \{(t,p) \mid t \in T,\ p \in P,\ (t,p) \in \{TxP\}\}$

Formalmente una CCPN se define de la siguiente manera:

Definición 5. *Una PN Coloreada Condicional (CCPN) es una 9-tupla:*

$$CCPN = \{\Sigma, P, T, A, N, C, Cond, Action, I\}$$

donde:

Σ *Conjunto finito de tipos de dato, los cuales son soportados por los lugares, llamado conjunto de colores.*

$P = \{p_1, p_2, \ldots, p_m\}$ *es un conjunto finito de lugares. Este conjunto está compuesto por* $P = P_{insert_tabla} \cup P_{update_tabla_columna} \cup P_{delete_tabla} \cup P_{copy}$

$T = \{t_1, t_2, \ldots, t_n\}$ *es un conjunto finito de transiciones. Este conjunto está compuesto por dos subconjuntos, es decir,* $T = T_{regla} \cup T_{copy}$

A *: conjunto finito de arcos dirigidos de* P *a* T *o de* T *a* P*. A está formado por los conjuntos* A_{in} *y* A_{out}*,* $A = A_{in} \cup A_{out}$

$N : A \rightarrow \{PxT\} \cup \{TxP\}$ *es una función nodo que asocia a cada arco un par ordenado* $(p,t) \in A_{in}$ *ó* $(t,p) \in A_{out}$*, donde el primer elemento del par ordenado es el nodo origen y el segundo es el destino.*

$C : P \rightarrow \Sigma$ *es una función color, que relaciona a cada lugar* $p \in P$ *un tipo de dato* $Type(p) \in \Sigma$*.*

$Cond : T_{regla} \rightarrow Bool$*, es una función condición que evalúa la parte condicional de la regla ECA almacenada en la transición* $t \in T_{regla}$*.*

$Acción : T_{regla} \rightarrow C(p)_{MS}$ *es una función acción que asocia cada transición* $t \in T_{regla}$ *con un multi-conjunto de un tipo de dato* $C(p)$, donde p es el lugar de salida de t.

$I : P \rightarrow C(p)_{MS}$ *es una función de inicialización que relaciona cada lugar* $p \in P$ *con un multi-conjunto de un tipo de dato* $C(p)$*.*

Regla de disparo de transiciones Habiendo dado una definición formal de la CCPN, ahora describiremos su comportamiento. Uno de los elementos más importantes que rigen el comportamiento de la PN son los tokens. Los tokens de CCPN no son tokens simples, sino que tienen la capacidad adicional de almacenar información de un cierto tipo de dato. Su definición es la siguiente:

Definición 6. *Un elemento token es un par* (p,c) *donde* $p \in P$ *y* $c \in C(p)$*. El conjunto de todos los elementos token está denotado por* TE*. Una marca* M *es un multi-conjunto sobre* TE*. La marca inicial* M_0 *es obtenida al evaluar las expresiones de inicialización:*

$$\forall (p,c) \in TE : M_0(p,c) = I(p).$$

El conjunto de todas las marcas es expresado por M. Otro elemento mental que rige el comportamiento de la CCPN es la transición. Una transici $t \in T_{copy}$ se dispara en una marca M_x si y sólo si:

$$\forall p \in {}^{\cdot}t : M_x(p) > 0$$

Una transición $t \in T_{regla}$ se dispara en una marca M_y si y sólo si se satisfac siguiente:

$$\forall p \in {}^{\cdot}t : M_y(p) > 0, Type(Cond(t)) = verdadero$$

La expresión ${}^{\cdot}t$ representa al conjunto de los lugares de entrada $p \in P$ de transición $t \in T$ y el conjunto de los lugares de salida $p \in P$ de una transic $t \in T$ se representa por la expresión $t^{\cdot}$.

1.2. Trabajos relacionados

Se han desarrollado diferentes enfoques para realizar el análisis de bases glas ECA. En [6] se presenta un análisis de bases de reglas activas basado algoritmo de "propagación", el cual utiliza un álgebra relacional extendida determinar cuando la acción de una regla afecta a la condición de otra y así tectar la formación de cadenas entre las relaciones de reglas. En la metodolo que se está proponiendo, la CCPN muestra gráficamente, y a simple vista, relación existente entre la acción de una regla y el evento de otra, además, bido a que la parte condicional de cada regla se almacena en transiciones transición por cada regla), es fácil analizar si la acción de una regla afecta condición de la otra regla con quien se encuentra conectada. Fraternali, boschi y Tanca en [7], construyen una "Hipergráfica de disparo". En esta gráfi los nodos son restricciones y los arcos son reglas, que viajan desde una restricci hacia un conjunto de restricciones. Los ciclos en la gráfica se interpretan secuencias infinitas de reglas. Las reglas que son parte de los ciclos se elimi de la gráfica, pero no se eliminan en el sistema, solamente se marcan para monitoreo especial en tiempo de ejecución y prevenir ciclos infinitos. En trabajo de investigación, los ciclos presentes en la gráfica de la CCPN consideran ciclos infinitos en el disparo de reglas, porque la acción de una del ciclo puede provocar que la condición de la regla que le precede sea falsa.

El artículo [8] describe la implementación del método de la Gráfica de paro Refinado (RTG) para el análisis de terminación de reglas activas. El mé do RTG estudia el contenido de ciclos de reglas en una gráfica de disparo. análisis demuestra que una regla no puede disparar a otra, el arco que conect estas dos reglas en la gráfica de disparo se elimina, hasta obtener una gráfica disparo refinada. Al igual que en nuestra investigación, se realiza un análisis haustivo para determinar si una regla dispara a otra; pero a diferencia de [8], nuestro caso solamente analizamos aquellas reglas que se encuentran dentro rutas cíclicas. En [9] y [11] se utilizan métodos de análisis de PN para detectar presencia de ciclos en el disparo de reglas. En [9] utilizan el árbol de alcanzab idad ("reachability tree") y en [11] aplican el árbol de cobertura ("coverabili tree"). En estos métodos concluyen que con la presencia de ciclos en la PN,

probable que ocurra el problema de no terminación del disparo de reglas. En el presente documento, también se propone la aplicación de un método de análisis de PN para detectar el problema de no terminación: la matriz de incidencia, donde, además de encontrar los ciclos en la CCPN, se estudia la posibilidad del disparo infinito de las reglas que forman parte de los ciclos.

Otro trabajo interesante es un método, basado en redes de Petri, para analizar las reglas de bases de datos activas y determinar si las reglas (1) presentan ciclos, (2) son inconsistentes y (3) entran en contradicciones. Los autores proponen una Red de Petri con Restricciones (Constraint Petri Net, CPN), la cual es una sistema de red Predicado/Transición y almacena conjuntos de restricciones en cada transición [10]. Sin embargo, este método no realiza una búsqueda exhaustiva para detectar las contradicciones de las reglas. Finalmente, en [12], Detlef Zimmer, Rainer Unland y Axel Meckenstock, proponen a las redes de Petri como la base para un análisis de reglas en tiempo de compilación. Argumentan que el análisis basado en redes de Petri es un enfoque que puede utilizarse para una especificación detallada del comportamiento de las reglas. Precisamente, en la CCPN se ha observado que, muchas de las características de las reglas activas, pueden ser representadas.

2. Matriz de incidencia de la CCPN

El comportamiento dinámico de muchos sistemas estudiados en ingeniería pueden describirse mediante ecuaciones diferenciales o ecuaciones algebraicas. El comportamiento dinámico de las PN también se describen por medio de ecuaciones. Uno de los métodos de análisis del comportamiento de un sistema modelado con redes de Petri es la *matriz de incidencia.* [13], la cual almacena información relevante sobre la relación existente entre los lugares (representados por las columnas de la matriz, y las transiciones (representados por los renglones). La definición formal de la matriz de incidencia es la siguiente.

Definición 7. *Para una red de Petri N, con n transiciones y m lugares, la matriz de incidencia $A = [a_{ij}]$ es una matriz de números enteros de $n \times m$. El valor para cada elemento de la matriz está dado por:*

$$a_{ij} = a_{ij}^{+} - a_{ij}^{-}$$

donde $a_{ij}^{+} = w(i,j)$ es el peso del arco que conecta una transición $t_i \in T$ con su lugar de salida $p_j \in P$ y $a_{ij}^{-} = w(j,i)$ es el peso del arco que conecta una transición $t_i \in T$ con su lugar de entrada $p_j \in P$.

En la matriz de incidencia se almacena suficiente información sobre la estructura completa de una PN, y es posible reconstruir la PN original a partir de los valores almacenados en la matriz de incidencia. Para el caso particular de este enfoque, la matriz de incidencia almacena solamente valores -1, 0 y 1, porque los pesos de los arcos en la CCPN siempre es 1. El valor a_{ij} de A, que representa la relación entre una transición $t_i \in T$ y su lugar de entrada $p_j \in P$ es -1, porque

$a_{ij}^{+} = 0$ y $a_{ij}^{-} = 1$, por lo tanto $a_{ij} = 0 - 1 = -1$. De manera similar, el valor de A que representa la relación entre una transición $t_i \in T$ y su lugar de $p_j \in P$ es 1, porque $a_{ij}^{+} = 1$ y $a_{ij}^{-} = 0$, por lo tanto $a_{ij} = 1 - 0 = 1$. El de a_{ij} es 0 cuando t_i y p_j no tienen relación alguna, es decir, que p_j no lugar de entrada ni de salida para t_i. Lo anterior puede resumirse de la siguiente manera:

$$a_{ij} = \begin{cases} -1 \text{ El lugar } p_j \in P \text{ es un lugar de entrada a la transición } t_i \in T \\ 0 \text{ No existe un arco que conecta al lugar } p_j \in P \text{ con la transición } t_i \in T \text{ y viceversa} \\ 1 \text{ El lugar } p_j \in P \text{ es un lugar de salida de la transición } t_i \in T \end{cases}$$

En cada una de las columnas de A aparecerá, a lo más, un valor $a_{ij} =$ Si el lugar $p_j \in P$ es un lugar de entrada, entonces en su respectiva columna de A aparecerá un valor de -1. En cambio, si el lugar p_j solamente es lugar salida, entonces no existirán valores de -1 en la columna j de la matriz en la matriz no existen columnas donde solo aparezca un valor de -1 y el 0's, entonces todos los lugares son lugares de entrada y salida a la vez.

En base a las propiedades particulares de la matriz de incidencia para CCPN, se presentan las siguientes definiciones.

Definición 8. *Un lugar es un nodo inicial NI si la columna presenta solamente valores 0's y un valor de* -1.

Definición 9. *Un nodo terminal NT es un lugar que representa la acción, no el evento, de una o varias reglas. En la matriz de incidencia, este lugar caracteriza por almacenar solamente valores 0's y 1's en la columna correspondiente. La ausencia de valores* -1*'s es debido a que el lugar no representa lugar de entrada para alguna transición.*

3. Análisis de terminación

Un tópico importante en diseño de bases de reglas activas es el problema No-terminación. El problema de no-terminación se presenta cuando la acción de una regla r_0 provoca el disparo de una regla r_1 y, la regla r_1, provoca disparo de la regla r_2 y así sucesívamente, hasta llegar a una regla r_n, donde esta última provoca el disparo de la regla r_0, produciendo el disparo infinito reglas. Es necesario, determinar la Terminación o No-terminación del disparo reglas en una BDA para evitar estados incosistentes en la BD. A continuación presentamos algunas definiciones necesarias en la demostración del teorema Terminación, basado en CCPN:

Definición 10. *Una ruta R es una secuencia de pares ordenados* (i, j)*, obtenida a partir de la matriz de incidencia de la CCPN. La secuencia de pares ordenados* (i, j) *es el desplazamiento sobre la matriz de incidencia que describe la conexión que existe entre los lugares y las transiciones. Las parejas ordenadas de una ruta están formadas de la siguiente manera:*

$$(a, b), (a, c), (d, c), \cdots, (w, x), (y, x), (y, z)$$

Los valores de i y j son los índices en la matriz de incidencia. El primer par ordenado está compuesto por la transición t_a y su lugar de entrada p_b, continuando con la ruta, el segundo par ordenado tiene la misma transición t_a y su lugar de salida p_c, el tercer par ordenado (d, c) tiene el mismo lugar p_c pero con la transición t_d, de la cual es su lugar de entrada. El órden de los valores de cada par ordenado debe ser -1 y 1, de forma alternada, como se muestra en la siguiente tabla:

(a,b)	(a,c)	(d,c)	$\cdots$	(w,x)	(y,x)	(y,z)
-1	1	-1	$\cdots$	-1	1	-1

Para formar una ruta, primero se debe localizar, en las columnas, el lugar que represente a un nodo inicial; si no existe ningún nodo inicial (existen ciclos) tomamos el lugar de la primer columna. El lugar de la primer columna siempre almacenará un valo -1, porque, de acuerdo al algoritmo para formar una CCPN [5], el lugar p_0 es el evento de la primera regla. En la columna seleccionada localizamos el valor -1, siendo la coordenada de -1 el primer elemento de R. Posteriormente, en el renglón donde se localizó el valor -1 se busca el valor 1, y la coordenada donde se ubique será el segundo elemento de R. Sucesivamente se busca el siguiente -1 en las columnas y el siguiente 1 en los renglones, hasta que no se encuentre uno de éstos valores. Si alguna de las coordenadas ya existe dentro de la secuencia, entonces se trata de una ruta cíclica. A continuación se presenta un algoritmo recursivo para encontrar las parejas ordenadas que forman una ruta:

```
siguienteNodo(int valor, int indice)
  if(valor == -1)
    boolean flag = false;
    for(i=0; i<n; i++)
      if(A[i][indice] == valor)
        flag = true;
        if(noExisteNodo(i,indice))
          agregaNodo(i,indice);
          siguienteNodo(1,i);  // busca el valor 1 en el rengl\'{o}n
          eliminaUltimoNodo();
        else  // Se encontro una ruta c\'{i}clica
          agregaNodo(i,indice);
          imprimeRuta();
          eliminaUltimoNodo();
    if(!flag)  // Se encontro una ruta ac\'{i}clica
      imprimeRuta();
  else
    for(j=0; j<m; j++)
      if(A[indice][j] == valor)
        agregaNodo(indice,j);
        siguienteNodo(-1,j);  // busca el valor -1 en la columna j
        eliminaUltimoNodo();
```

Definición 11. *Una ruta cíclica RC es una ruta R donde el último par ordenado (y, z) ya se encuentra listado en R, es decir, $y = q_k$ y $z = r_k$, $k = 1, 2, ..., |R| - 1$.*

Definición 12. *Una ruta acíclica RA es aquella donde la última pareja ordenada (i, j) de la ruta es diferente de sus antecesoras y le corresponde una valor de en su matriz de incidencia respectiva, es decir, que RA tiene un nodo terminal*

Dadas estas definiciones, presentamos el siguiente teorema de terminación:

Teorema 1. *Si todas las rutas R de una CCPN son acíclicas, entonces el disparo de las reglas termina.*

Demostración. Dado que las rutas R de una CCPN son acíclicas, por definición de R no existen nodos en las rutas que se repitan, todos los nodos, en cada ruta, son diferentes, por lo que cada ruta tiene un nodo inicial y un nodo terminal. Suponiendo que la pareja ordenada (i_t, j_t) es un nodo terminal de la ruta R, el lugar j_t es un lugar que representa la acción de una o mas reglas, mas no el evento activador de otras, terminando en este punto con la ejecución consecutiva de reglas, por lo tanto el disparo de reglas ECA termina.

A partir del teorema presentamos el siguiente corolario:

Corolario 1. *Si existen rutas cíclicas RC en la CCPN, y existe al menos una transición $t \in T_{rule}$, donde $Cond(t) = false$ dentro de cada RC, entonces el disparo de las reglas ECA termina.*

Demostración. Dado que en la CCPN existen rutas cíclicas RC entonces existen parejas ordenadas (i, j) que se repiten según la definición de RC. Se presenta un disparo consecutivo de reglas cuando cada acción de reglas, dentro de la ruta cíclica, provoca que la condición de la siguiente regla sea verdadera, sin embargo, de acuerdo a los argumentos del corolario, existe al menos una transición, tal que $Type(Cond(t)) = falso$, truncando de esta manera la activación de una siguiente regla, convirtiendo así la última pareja ordenada (i, j), con valor de -1, en un nodo terminal, y con la existencia de un nodo terminal en la ruta R entonces el disparo de las reglas ECA termina.

4. Caso de estudio

Para mostrar la viabilidad que ofrece nuestro enfoque en la detección de ciclos infinitos, tomamos la base de reglas que se presenta en [10], la cual se describe a continuación:

Regla 1: Cuando la prima de un empleado se modifica, si la cantidad es mayor que \$100.00, entonces el rango del empleado se incrementa en uno.

Regla 2: Cuando el rango de un empleado se modifica (supongamos que se incrementa), entonces la prima del empleado se incrementa diez veces el nivel del nuevo rango.

Regla 3: Cuando se obtienen las ventas del mes y el número de éstas es superior a 50, entonces se incrementan $10.00 al salario del empleado.

Regla 4: Cuando se obtienen las ventas del mes y el número de éstas es superior a 100, entonces el rango del empleado se incrementa en un nivel.

Regla 5: Cuando el nivel del rango de un empleado se modifica y el rango alcanzó el nivel 15, entonces el salario del empleado se incrementa en un 10%.

Regla 6: Cuando se obtienen las ventas del mes y el número de éstas es superior a 50 y el rango de empleado que las obtuvo es menor de del nivel 15, entonces el monto de su prima se decrementa $100.00.

Las tablas que forman parte de la BD son:

`EMPLEADO(emp_id, nombre, rango, salario)`

`PRIMA(emp_id, cantidad)`

`VENTAS(emp_id, mes, numero)`

La CCPN generada a partir de éstas cuatro reglas se muestra en la figura 2.

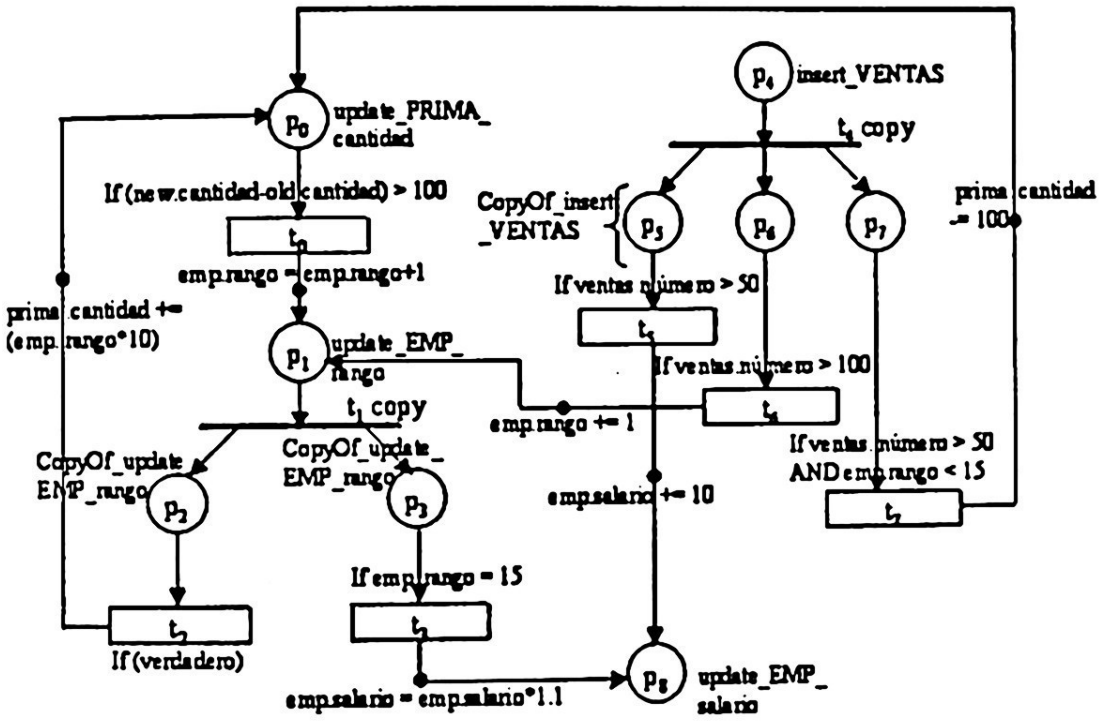


Figura 2. CCPN obtenida a partir de la base de reglas.

La matriz de incidencia A correspondiente a la CCPN es la siguiente

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

Las rutas R que se obtienen de A son las siguientes:

R[0]: (4,4), (4,5), (5,5), (5,8)
R[1]: (4,4), (4,6), (6,6), (6,1), **(1,1)**, (1,2), (2,2), (2,0), (0,0), (0,1), (1,1)
R[2]: (4,4), (4,6), (6,6), (6,1), (1,1), (1,3), (3,3), (3,8)
R[3]: (4,4), (4,7), (7,7), (7,0), **(0,0)**, (0,1), (1,1), (1,2), (2,2), (2,0), (0,0)
R[4]: (4,4), (4,7), (7,7), (7,0), (0,0), (0,1), (1,1), (1,3), (3,3), (3,8)

Se puede observar que son cinco rutas diferentes que se pueden generar el disparo de las reglas, de las cuales, dos de ellas contienen la misma secuencia cíclica de nodos (R[1] y R[3]). Dado que existen rutas cíclicas, es posible que el disparo de reglas sea infinito, sin embargo, de acuerdo a la descripción del Corolario 1, debe existir al menos una transición, dentro de la ruta cíclica, que condición sea falsa. Dentro de la ruta se encuentran las transiciones $t_0, t_2 \in T_{rule}$, pero la condición que se evalúa en t_2 siempre es verdadera. La condición de verifica si el campo cantidad de la tabla PRIMA se incrementó en más de $100, el lugar p_0, que representa al evento de actualizar el campo cantidad en la tabla PRIMA, recibe información de t_2 y de t_7. La transición t_7 no provoca el disparo de t_2 porque la actualización del campo cantidad es un decremento. La transición t_2 actualiza el valor de prima.cantidad incrementándole en diez veces el valor del rango, por lo que si el valor del rango es igual a un número mayor que 10 entonces el valor prima.cantidad se incrementará en más de $100, evaluando en verdadero la condición de t_0, incrementando a la vez el valor de rango y produciendo una vez más un incremento mayor de $100 para prima.cantidad, generando de esta manera en un disparo infinito de reglas. Se puede concluir de que si el valor de emp.rango es menor de 10, entonces el disparo de las reglas termina.

5. Conclusiones y trabajo futuro

En este artículo se presentó un enfoque para atacar el problema de No-terminación en BDA, diferente a los desarrollados en la actualidad. En este enfoque se utiliza la CCPN para analizar la base de reglas ECA, con el apoyo de la matriz de incidencia, herramienta de análisis en la teoría de PN. La matriz de incidencia de una CCPN presenta propiedades importantes en la detección de ciclos. Al igual que en muchos trabajos relacionados, se parte de que la ausencia de ciclos garantiza la terminación del disparo de las reglas ECA. En nuestro caso, además de probar la terminación del disparo de reglas debido a la ausencia de ciclos, probamos que, aún con la presencia de ciclos, el disparo de reglas termina, si existe al menos una regla cuya condición sea falsa y, de esta manera, romper el ciclo infinito en que pueda caer el disparo de las reglas.

En el caso de estudio presentado, la estructura de la CCPN ofrece un número menor de lugares y transiciones que los trabajos relacionados donde se aplican las redes de Petri. En otros enfoques solamente verifican la no existencia de ciclos en la gráfica de disparo de las reglas, para determinar que el disparo de reglas termina (como es el caso de donde se tomó la base de reglas), mientras que en este trabajo se verifica que las condiciones presentes en el ciclo sean verdaderas.

para asegurar que el disparo de las reglas no va a terminar. Por el momento, la CCPN solo detecta eventos simples. Se tiene propuesto, como trabajo a seguir, incorporar al modelo de la CCPN la detección de eventos compuestos, y a la vez, considerarlos en el análisis de No-terminación.

Referencias

1. Paton, N.W., Diaz O.: Active Database Systems. In: ACM Computing Surveys, Vol. 31, No. 1,1999, pp. 64-103
2. Chakravarthy, S.: Early Active Database Efforts: A Capsule Summary. In: IEEE Transactions on Knowledge and Data Engineering, Vol.7, No.6,1995, pp.1008-1010
3. Li, X., Medina Marín, J., and Chapa, S. V.: A Structural Model of ECA Rules in Active Database. In: Mexican International Conference on Artificial Intelligence (MICAI'02), Mérida, Yucatan, México, April 22-26, 2002
4. Medina Marín, J., y Li, X.: Red de Petri Coloreada Condicional y su Aplicación en Sistemas de Bases de Datos Activas. En: Octava Conferencia de Ingeniería Eléctrica 2002 (CIE'02), México, D.F., México, Septiembre 4, 5 y 6, 2002.
5. Medina Marín, J.: Red de Petri Coloreada Condicional (CCPN) y su Aplicación en Bases de Datos Activas. En: Tesis de Maestría de la Sección de Computación, Depto. de Ingeniería Eléctrica, CINVESTAV-IPN, México. Septiembre de 2002.
6. Baralis, E. and Widom, J.: An Algebraic Approach to Static Analysis of Active Database Rules. In: ACM Transactions on Database Systems, Vol. 25, No. 3, September 2000, pp. 269-332.
7. Fraternali, P., Paraboschi, S., and Tanca, L.: Automatic Rule Generation for Constraint Enforcement in Active Databases. In: U.W. Lipeck and B. Thalheim (Eds.), Modelling Database Dynamics. London: Springer-Verlag, 1993. Selected Papers from the Fourth International Workshop Foundations of Models and Languages for Data and Objects, pp. 153-173.
8. Urban, S.D., Tschudi, M.K., Dietrich, S.w. and Karadimce, A.P.: Active Rule Termination Analysis: An Implementation and Evaluation of the Refined Triggering Graph Method. In: Journal of Intelligent Information Systems, No. 12, 1999, pp. 27-60.
9. Kokkinaki, A.I.: On using Multiple Abstraction Models to Analyze Active Databases Behavior. In: Biennial World Conference on Integrated Design and Process Technology, Berlin, Germany, June, 1998.
10. Guisheng, Y., Qun, L., Jianpei, Z., Jie, L., Daxin, L.: Petri Based Analysis Method For Active Database Rules. In: IEEE International Conference on Systems, Man and Cybernetics, vol. 2, 1996, pp. 858-863
1. Zimmer, D.: Rule Termination Analysis based on a Rule Meta Model. In: Cadlab Report 2, Cadlab, Bahnhofstr. 32, 33102 Paderborn, Germany, April 1995
12. Zimmer, D., Unland, R., Meckenstock, A.: Using petri nets for rule termination analysis. In: Proceedings of the Workshop on Databases: Active and Real-Time 1996 (CIKM'96), Rockville, Maryland, 1996.
13. Murata, T.: Petri Nets: Properties, analysis, and applications". In: Proceedings of the IEEE, 77(4):541-580, 1989.

Iterative n-Literal Rule for Solving Satisfiability Problems

Rafael Rivera-Lopez [1], Olga Leticia Prado-de-la-Paz [2], and Juan Frausto-Solis

[1] Deparment of Systems and Computation, Technological Institute of Veracruz
Miguel A. de Quevedo 2779, 91860, Veracruz, Veracruz, Mexico
rrivera@itver.edu.mx

[2] Temic de Mexico, S.A. de C.V.
Av. Ignacio Allende Lote 20, Parque Industrial Cuautla, Cd. Ayala, Morelos, Mexico
olga.prado@temic.com

[3] Department of Computer Science, ITESM, Campus Cuernavaca
Av. Paseo de la Reforma 182-A, 62289, Temixco, Morelos, Mexico
juan.frausto@itesm.mx

Abstract. In this paper a new rule applicable to the process of resolution of satisfiability (SAT) problems is presented. This approach, called the n-literals rule (nLR), is an extension of the one-literal rule (1LR) proposed by Davis, Logemann and Loveland that eliminates a set of literals in each step of one iterative resolution process of a SAT problem. This document describes the elements that define the nLR and presents some experimental results showing an improvement by using this new rule.

1 Introduction

In recent years, many approaches have emerged for solving the satisfiability (SAT) problem. These approaches include local search ([1], [2]), backtracking ([3], [5]), Binary Decision Diagrams [6], and so on. Gu describes in several papers [8] and [9]) many methods for solving instances of SAT problems and proposes various classifications for them. The great amount of SAT methods, are derived from the fact that the SAT problem is one of the most important problems in complexity theory, artificial intelligence and other disciplines. In particular, the method developed in 1960 for Davis and Putnam, is considered one of the major practical approaches for solving SAT problems, because it is a complete method. The original DP method is based on unit resolution [10] and their revised version (DPL by Davis, et al. in [11]) uses splitting for avoiding memory explosion. method uses four rules for testing the unsatisfiability of a set of clauses: tautology rule (TR), one-literal rule (1LR), pure-literal rule (PLR) and splitting rule (SR).

The most important rules in the DPL method are the 1LR and the SR; the permits the elimination of unit clauses in the original logical formula and SR divides the problem into two smaller sub-problems. In [12], Frausto-Solis and Sanchez-Arr

propose two variants of the 1LR in aims to improve the performance of DPL method: the structured 1LR with support set and the structured n-literal rule and in 3] one generalization of 1LR for two-literals is proposed using a recursive method. This paper presents one rule based on the structured n-literal rule, that eliminates a set of literals in each step of an iterative process of resolution. This document describes the elements that define the nLR and presents some experimental results that indicate the existence of an improvement in the resolution of problems using this rule.

Background

This section describes the concepts used for the developed of this approach: The SAT problem, the DPL method and the 1-literal rule.

2.1 Satisfiability (SAT) Problem

SAT problem was the first known NP-complete problem [14] and it has a fundamental role in complexity theory (this is the engine that impulses the search of an efficient method for solving it). The general question for the SAT problem is to decide if one assignment for the literals of a logical formula that makes it true exists or not. A SAT formula is in its conjunctive normal form (CNF) as in (1), where one clause is a disjunction of literals, as in (2), and a literal is a Boolean variable negated or not).

$$F = Clause_1 \wedge Clause_2 \wedge \ldots \wedge Clause_m \tag{1}$$

$$Clause_i = x_1 \vee x_2 \vee \ldots \vee x_n \tag{2}$$

2.2 DPL Method

Herbrand method (HM) is based on a refutation procedure and is the basis for the modern automatic theorem proving. Several researchers, as Gilmore, tried to applied HM [16], but their methods were inefficient. Davis and Putnam ([10], [11]) introduce a more efficient method for deciding the satisfiability of a set of clauses as 1). DPL method consists of four rules showed in Table 1, where S, S' and S'' are set of clauses, L is a literal and $\sim L$ is its negation.

In the implementation of the DPL method, both the tautology and pure-literal rules are often deleted. The tautology rule can applied before of start the DPL and the pure-literal rule can be considered the splitting rule with the set S_2 without clauses B_1. Figure 1 shows the algorithm for the DPL method.

Table 1. The rules of the DPL method

Rule	Description
Tautology	All tautologies are deleted from S and the set S' is constructed the remaining clauses. If S' is unsatisfiable then the original set unsatisfiable too.
One-Literal	If a unit clause L exists in S, all clauses containing L are deleted the set S' is constructed with the remaining clauses. S is satisfiable S' is empty. Otherwise, the set S'' is obtaining from S' by dele the literals $\sim L$ from all clauses in S' (S' is simplified in $\sim L$). If clause in S'' is the null clause ($\bullet$), then S'' and S are unsatisfiable.
Pure-Literal	A literal L is pure in S if the literal $\sim L$ does not appear in any in S. If L is pure, all clauses containing L can be deleted. If maining set S' is unsatisfiable, then S is unsatisfiable too.
Splitting	If S can be written as $(A_1 \vee L) \wedge \ldots \wedge (A_m \vee L) \wedge (B_1 \vee \sim L) \wedge \ldots \wedge \sim L) \wedge R$, then the sets S_1 and S_2 can be constructed as $S_1 = A_1 \ldots A_m \wedge R$ and $S_2 = B_1 \wedge \ldots \wedge B_n \wedge R$), respectively. If both S_1 and unsatisfiables then the original set S is unsatisfiable too.

```
procedure DPL(S);
{ S is a set of clauses in CNF }
begin
  if unit clause exists in S
  then begin { 1-literal rule }
    L = L∈S and L is a unit clause;
    S' = S \clauses with L;
    if S' = ∅ then return SAT;
    else begin
      S" = S simplified in ~L;
      if •∈ S" then return UNSAT;
      else return DPL(S");
    end
  end
  else begin { Splitting rule }
    L is a selected literal of S using some heuristic;
    if DPL(S ∪ L) is SAT
    then return(SAT);
    else
      if DPL(S ∪ ~L) is SAT
      then return(SAT);
      else return UNSAT;
  end
end.
```

Fig. 1. The DLP method

3 The n-Literals Rule (nLR) Method

The DPL method is based in the determination of one literal for testing the satisfiability of a logical formula because both the 1LR and the SR need one literal for evaluating the set of clauses. The nLR method uses n literals, instead of only one, in each step of the resolution procedure. This approach produces that the both 1-literal rule and the splitting rule can be enunciated in one rule called the n-literals rule (nLR) and the resolution method can be called nLR method. Table 2 shows the description of the n-literals rule.

Table 2. The n-literals rule

Rule	Description
n-literals	If n literals are selected of S, these n literals are assigned with some truth-value and the clauses containing some of these n literals are removed of S for constructing the new set S'. S is satisfiable if S' is empty. Otherwise, the set S'' is obtaining from S' by deleting the complement of the n literals from all clauses in S'. If for any combination of truth-values for these n literals, some clause in S'' is the null clause (•), then S'' and S are unsatisfiable.

3.1 Description of the Method

The nLR method needs to know the number of literals to eliminate in test and, once these literals are selected of the original set, they are assigned by some truth-value (one interpretation is assigned to the set of n literals). The method eliminates all the clauses that contain some of these literals. If the new set of clauses is empty, then the problem is satisfiable and the method finalizes; otherwise, the remaining clauses are simplified removing the complement of these n literals for them. If the result is unsatisfiable, then one new interpretation is assigned to the set of literals and newly is applied the procedure. If all possible interpretations produce that the set is unsatisfiable, then the original formula is unsatisfiable.

If when removing the complement of the n literals of the original set does not take place a null clause, then it is necessary to select another set of n literals and the procedure is repeated. When the null clause is founded in the reduced set (the remaining clauses simplified removing the complement of the n literals) it is necessary test all the possible interpretations for the set of n literals. The maximum number of interpretations for n literals is 2^n. Figure 2 shows the search three for the nLR when n=3 and Figure 3 shows the algorithm for the recursive nLR method.

In this paper we use the iterative version of the nLR method due to the recursive version of any algorithm needs additional time and memory space that the iterative version [17]. The iterative version of the nLR method is shown in figure 4.

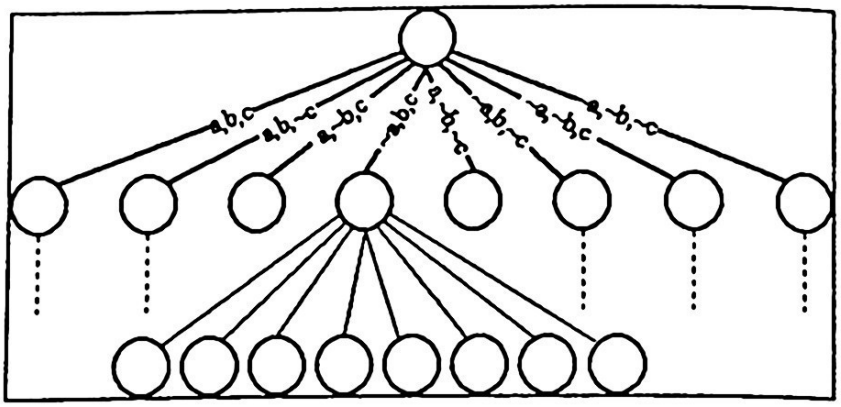


Fig. 2. Partial search three for the nLR method when n=3

```
procedure recursive_nLR(S, N, i)
{ S is a set of clauses in CNF }
{ N is a set of n literals (L) selecting of S using some heuristic}
{ i is the number of interpretation, the maximum value of i is 2^n }
begin
  ∀L_k ∈ N (L_k is assigned with truth-value, not assigned previously);
  S' = S \ clauses in N;
  if S' = ∅ then return SAT;
  else begin
    S" = S simplified with the complement of clauses in N;
    if • ∈ S"
    then
      if i = 2^n then return UNSAT;
      else return recursive_nLR(S, N, i+1);
    else begin
      { The set S is reduced and other n literals are selected}
      S = S";
      N = {L_k | L_k ∈ S and k : 1, ..., n};
      i = 0;
      recursive_nLR(S, N, i+1);
    end
  end
end.
```

Fig. 3. The recursive nLR method.

3.2 Selection of the Set of Literals

The selection of n literals is another important aspect in the nLR method. In simi
form of the splitting rule on the DPL method, the n literals selected for testi
satisfiability of a logical formula can be deterministic or random. These heuristi

can be based over the number of occurrences of the literals in the original formula, selecting the most frequently or the less frequently, or if the literals are independents or not, or by column (selecting the literals in one clause), or in lexicographic al order and so on. Figure 5 shows several heuristics that can be applied in the nLR method.

```
procedure iterative_ nLR(S, n);
{S is a set of clauses in CNF }
{ n is the number of literal to eliminate }
{ N is a set of n literals (L) selecting from S using some heuristic }
begin
 i = 1; j = 1;
 N_i = {L_k | L_k ∈ S and k : 1, ..., n};
 repeat
   ∀L_k ∈ N_i (L_k is assigned with truth-value, not assigned previously);
   S' = S \ clauses in N_i;
   if S' = ∅ then S is SAT;
   else begin
     S" = S simplified in the complement of clauses in N_i;
     if • ∈ S"
     then
      if j = 2^n
      then S is UNSAT;
     else begin
       { other interpretation of literals in N_i is necessary }
       j = j + 1;
     end
     else begin
       { The set S is reduced and otherf n literals N_i is selected }
       S = S";
       i = i + 1; j = 1;
       N_i = {L_k | L_k ∈ S and k : 1, ..., n};
     end
   end
 until S is SAT or UNSAT;
end.
```

Fig. 4. The iterative n-literal rule.

Experimentation and Results

For the test of this method several problems were resolved. Several hard instances of the problem were generated: instances of satisfiable formulas with 50 variables in

formulas with 50 or 100 clauses and instances of unsatisfiable formulas for lq variables in formulas with 403 clauses.

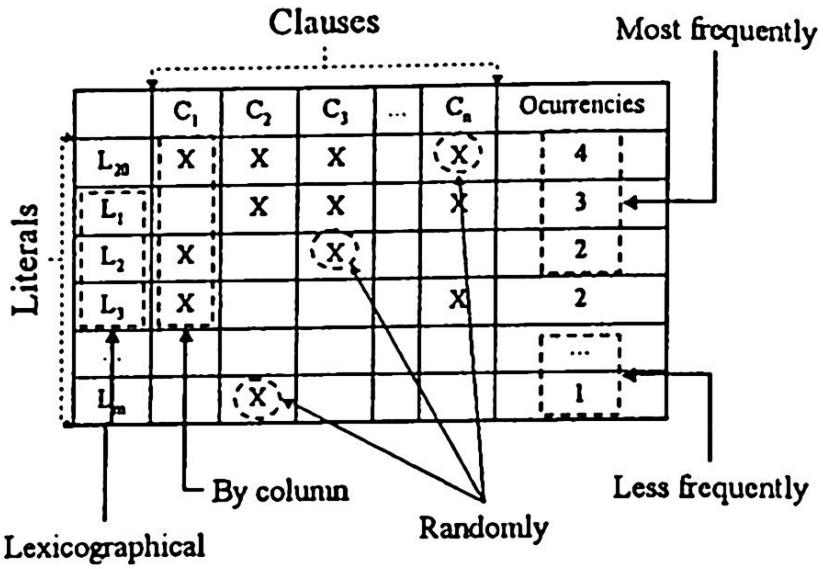


Fig. 5. Several forms for selecting the n literals in the nLR method

In Figure 6 is showed the behavior of nLR for n=1,2 and 3 in instances the satisfiable formulas. In this case the nLR is more efficient when n is increased because the number of iterations has a decremented behavior. The nLR have an improvement of 58% in the case of 50 clauses and 65% in the case of 100 clauses.

When the approach was probed with unsatisfiable hard instances (figure 7), the performance of nLR is degraded due to the combinatorial explosion in the number of interpretations that must be tested.

Another group of tests was made modifying the number of literals. Table 3 shows these test. In this case is observed that the number of iterations for the nLR is better than the 1LR.

5 Conclusions

The nLR method is a generalization of the rules proposed for Davis, Putnam and Loveland for solving satisfiability problems; when n = 1 the nLR is the original DPL method. In this paper is used the iterative approach for nLR due to its uses less memory and time for resolution of SAT problems.

For the results of the experimentation is showed that the effectiveness of the nLR has a better performance when n is increased. In the case of satisfiable formulas is clear the improvement in the number of iterations when n is increased. Is obvious that in the case for unsatisfiable formulas the nLR has not better performance due the increasing in the number of combinations in the interpretation of the set of literals.

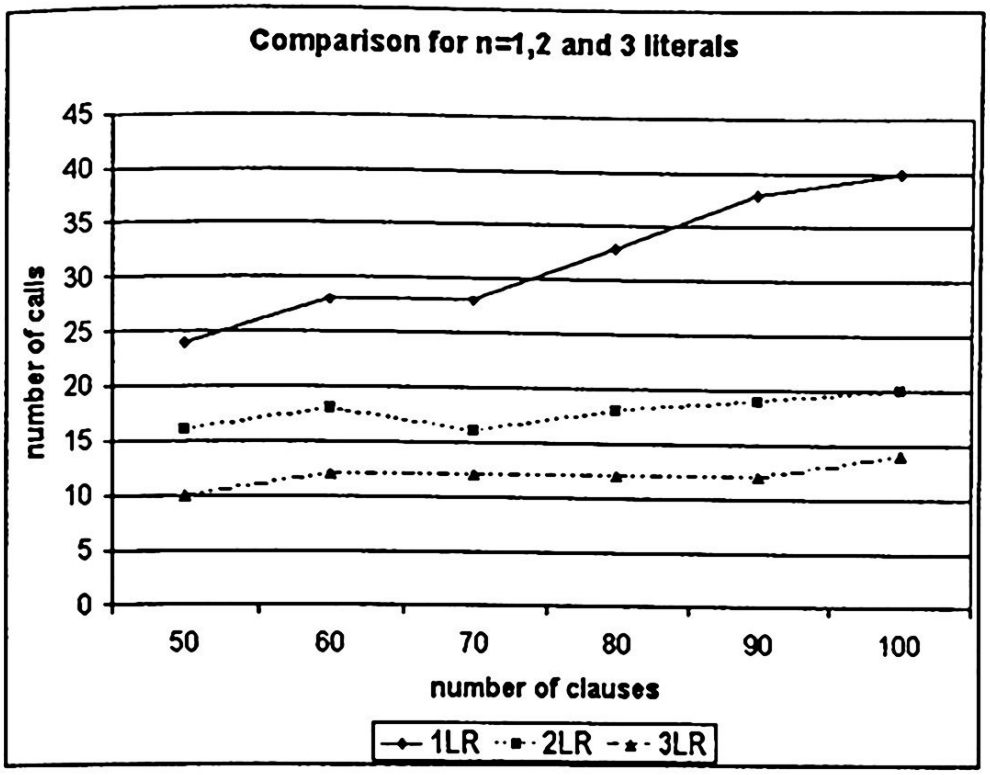


Fig. 6. Number of iterations for the nLR method when n=1, 2 and 3 and hard instances of 50-100 clauses with 50 literals.

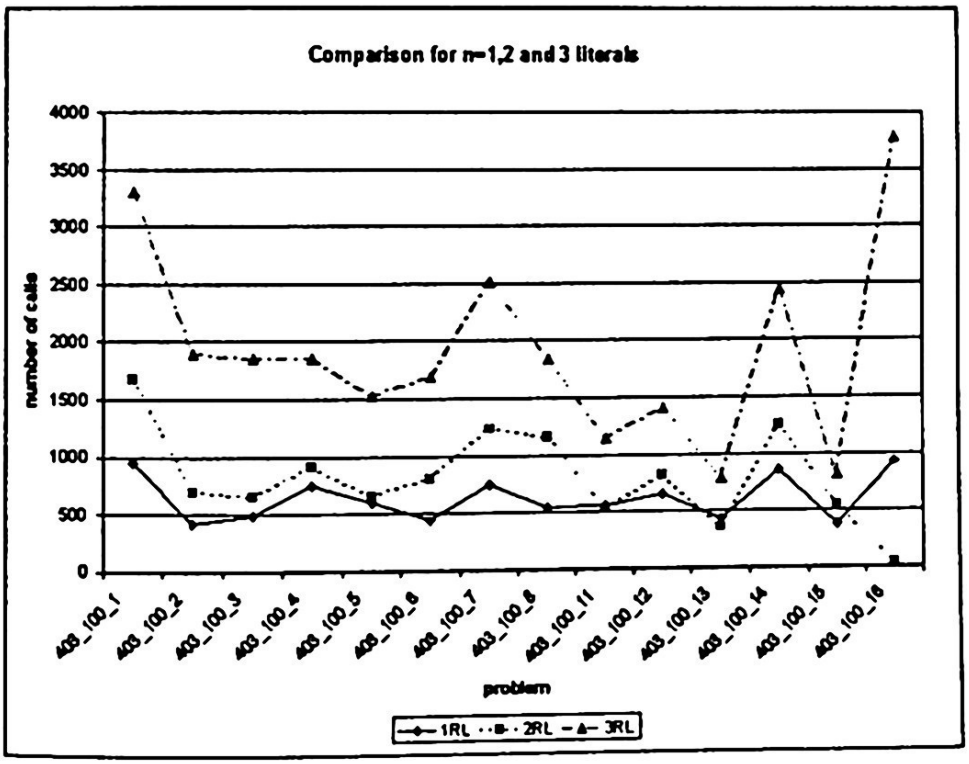


Fig. 7. Number of iterations for the nLR method when n=1, 2 and 3 and hard instances of 403 clauses with 100 literals

Table 3. Set of tests for several values of n

Clauses	Literals	n	Iterations	
			1LR	nLR
400	100	10	9	2
380	102	5	8	3
360	50	3	1	1
450	80	18	9	2
450	150	27	8	1
492	140	2	9	6
300	50	2	9	5
420	110	5	8	3

References

1 Selman, B., Levesque, H. and Mitchell, D.: A new method for solving hard satisfiabi problems, in Proc. of the 10th National Conference on Artificial Intelligence (AAAI San Jose, USA (1992) 440-446.

2 Selman, B. and Kautz, H.: Domain-independent extensions of GSAT: Solving structured satisfiability problems, in Proc. of the Int. Joint Conf. on Artificial Intellige (IJCAI-93), Chambéry, France, (1993) 290-295.

3 Larrabee, T.: Test pattern generation using Boolean satisfiability, in IEEE Trans Computer Aided Design, Vol. 11, No. 1 (1992) 4-15.

4 Bayardo, R. and Schrag, R.: Using CSP look-back techniques to solve real world instances, in Proc. of the 14th Nat. Conf. on Artificial Intelligence, (AAAI-, Providence, USA (1997), 203-208.

5 Sosic, R. and Gu, J.: Fast search algorithms for the n-queens problem, in IEEE Trans Systems, Man and Cybernetics, Vol. 21, No. 6 (1991) 1572-1576.

6 Ashar, P., Ghosh, A. and Devadas, S.: Boolean satisfiability and equivalence check using general Binary Decision Diagrams, in Proc. of IEEE Int. Conf. on Comp Design: VLSI in Computers and Processors, Cambridge, USA (1991) 259-264.

7 Gu, J.: Local search for satisfiability (SAT) problem, in IEEE Trans. on Systems, and Cybernetics, Vol. 23, No. 4 (1993) 1108-1129.

8 Gu, J.: Global Optimization for Satisfiability (SAT) problem, in IEEE Trans. Knowledge and Data engineering, Vol. 6, No. 3 (1994) 361-381.

9 Gu, J., Purdom, O.W., Franco, J. and Wah, B. W.: Algorithms for the satisfiability problem: A Survey, in DIMACS Series on Discrete Mathematics and Theore Computer Science: Satisfiability (SAT) Problem, Vol. 35 (1997) 19-152.

10 Davis, M. and Putnam, H.: A computing procedure for quantification theory, in Journal the ACM, Vol. 7, No. 3 (1960) 201-215.

11 Davis, M., Logemann, G. and Loveland, D.: A machine program for Theorem Proving. Communications of the ACM, Vol 5, No. 7 (1962) 394-397.

12 Frausto-Solis, J. and Sanchez-Ante, G.: Fundamentos de Lógica Computacional, 1st. edition. Trillas, México, 2000.

13 Ruiz-Cuevas, A., Torres-Jimenez, J. and Frausto-Solis, J.: Two literals Rule (2LR): A New Algorithm for Solving Satisfiability (SAT), in Int. Symp. in Intelligent Technologies, Apizaco, Mexico (2000).

14 Cook, S.A.: The complexity of theorem-proving procedures, in Proc. of the 3rd IEEE Symp. on the Foundations of Computer Science, Shaker Heights, USA (1971) 151-158.

15 Gent, I. P. and Walsh, T.: The search for satisfaction, in http://dream.dai.ed.ac.uk/group/tw/sat/sat-survey2.ps (1999).

16 Gilmore, P.C.: A proof method for quantification theory, in Journal fo the ACM, vol. 7 (1960) 201-215.

17 Stojmenovic, I.: Recursive Algorithms in Computer Science Courses: Fibonacci Numbers and Binomial Coefficients, in IEEE Trans. On Education, Vol. 43, No. 3 (2000), 273-276.

Ambiente web para la construcción, simulación y visualización 3D de autómatas finitos y lenguajes regul

Claudia Noguerón, Manuel Chim, Juan Carlos Tovilla, Leandro Balladares

Centro de Investigación en Computación del Instituto Politécnico Nacional,
Unidad Profesional "Adolfo López Mateos", Zacatenco, C.P. 07738, México, México
cnogo@hotmail.com, manuelchi@hotmail.com,
jsarmiento@sagitario.cic.ipn.mx, ballad@cic.ipn.mx

Resumen. En las ciencias computacionales se usan frecuentemente modelos abstractos. El aprendizaje de estos modelos es difícil y muchas veces requiere de un esfuerzo mental mayor que en otras áreas del conocimiento por parte los estudiantes cuando se usan los métodos de aprendizaje tradicionales. Lo anterior, demanda la búsqueda de métodos que asistan al estudiante durante el aprendizaje de dichos modelos. Hemos desarrollado un software de aprendizaje basado en el Web que busca ayudar al estudiante a comprender mejor los principios de construcción de compiladores, en particular la etapa de análisis léxico. El software ofrece un ambiente en el que son presentados de manera gráfica y animada las definiciones y algoritmos más importantes. Las animaciones presentadas, muestran como se construyen los autómatas finitos a partir de expresiones regulares, así como la operación del autómata en tres dimensiones (3D). Se presentan los principios de diseño e implementación del software desarrollado, así como el trabajo que aún se está desarrollando (trabajo futuro). Finalmente, se describen algunos trabajos relacionados.

1 Introducción

La tarea diaria de un profesor / difundidor de las ciencias de la computación señar conocimiento abstracto y promover el entendimiento correcto de este miento entre sus estudiantes. Por ejemplo, cuando un profesor quiere descri operación de los autómatas de pila o de los analizadores sintácticos ascendent lo único que tiene a la mano en el salón de clases es un pizarrón y tal vez plumones de distintos colores. El reto para el profesor, es explicar el funcion de dichos modelos mediante un ejemplo, haciendo uso únicamente de los mate que tiene al alcance y de su habilidad para explicar este tipo de conceptos. de tres o cuatro pasos el profesor comienza a borrar símbolos (en el caso del zador LR) y estados para agregar otros (en el caso del autómata de pila) de Los estudiantes, tienen que esforzarse más por reconstruir la secuencia de ope- nes complicadas de escritura y borrado y descubrir un sentido en el desorden, entender la funcionalidad de los modelos que el profesor intenta explicar. La

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en Mé
CORE-2003, México.

lización y simulación gráfica de los autómatas de pila y de los analizadores sintácticos es una posible solución a este dilema.

En este trabajo, se describe un software de aprendizaje basado en el Web, desarrollado en el Centro de Investigación de Computación del IPN por alumnos del curso de Compiladores, que busca ayudar a los estudiantes a comprender mejor los principios de construcción de compiladores, en particular la etapa de análisis léxico. El software ofrece un ambiente en el que son presentados de manera gráfica las definiciones y algoritmos más importantes. Las animaciones presentadas, muestran gráficamente como se construyen los autómatas finitos a partir de expresiones regulares, así como la operación del autómata. Este software, formará parte de un libro electrónico basado en el Web del curso de Autómatas y Compiladores que actualmente está siendo desarrollado en el centro y que pretende integrar animaciones gráficas 3D de los algoritmos de cada una de las fases de un compilador.

2 Animación del algoritmos

La animación de algoritmos se refiere a ilustrar el comportamiento de un programa visualizando sus operaciones fundamentales mientras está en ejecución. Se ha probado, que estas visualizaciones son de utilidad en los campos de investigación y de educación en el diseño y análisis de algoritmos [5], [27], [28], [29], [34], [35]. Los primeros sistemas de animación de algoritmos eran dependientes de la plataforma [2–4], [21]. Con el advenimiento de las nuevas tecnologías, la popularidad del Web y la evolución del lenguaje de programación Java, los desarrolladores se han inclinado por construir sistemas de animación de algoritmos en línea, siendo de gran utilidad para educación a distancia [6], [7], [10–12], [16], [20], [26], [32], [33].

Los sistemas de animación de algoritmos han sido extensamente utilizados como soporte de enseñanza en las ciencias de la computación. Se han llevado a cabo diversos estudios para evaluar su efectividad en la educación, pero los resultados varían. Por ejemplo Brown [4], usó un sistema llamado BALSA-I para enseñar en cursos de Introducción a la Programación y Estructuras de Datos. El sistema fue utilizado como un visualizador de programas en el primer curso y como animador de algoritmos de alto nivel en el segundo. Brown, reportó que el uso de animaciones como complemento en algunos temas, incremento la rapidez de comprensión del conocimiento por parte de los alumnos en comparación con los temas en dónde no eran usadas animaciones. Otros estudios que han reportado resultados positivos en la utilidad de la animación de algoritmos para enseñanza son [13], [15], [24], [26], [32], [33]. Algunos de estos reportes indican que las animaciones deben de presentarse a los alumnos como material complementario y que debe de haber una explicación del funcionamiento de las animaciones por parte del instructor antes de que los estudiantes las utilicen, lo anterior con la finalidad de facilitar el aprendizaje. Otros trabajos, reportan que la animación de algoritmos debe de utilizarse únicamente como soporte para realización de tareas por parte de los estudiantes. Los trabajos anteriores, concluyen y confirman la utilidad pedagógica de enseñar con

animación de algoritmos en lugar de utilizar los métodos tradicionales (texto, genes).

Algunos autores [8], [9], [13–15], [19], [28], coinciden en que para que los sistemas de animación de algoritmos sean efectivos y benéficos para los usuarios, debe cuidarse su diseño y la utilización de las animaciones, además de respetar los siguientes lineamientos:

- *Acceso abierto.* Los usuarios deben tener acceso abierto a las animaciones, es decir, además de tener acceso al sistema en la escuela, deben poder accesarlo desde su casa o desde cualquier lugar.
- *Control de la información.* El *sistema* de animación debe estar parametrizado, es decir, debe permitir a los usuarios crear sus propios conjuntos de datos de entrada para las animaciones.
- *Interactividad.* El sistema de *animación* debe proporcionar interacción entre usuarios y el sistema. La interacción debe contemplar animar paso a paso, hacer un paso o toda la animación, adelantar o regresar la animación a un estado determinado de manera rápida, *etc.*
- *Historia*: El sistema debe *permitir* al usuario regresar y ver estados previos de animación en ejecución.
- *Retroalimentación*: Debe *obtenerse* de alguna manera la retroalimentación de estudiantes para evaluar la efectividad del sistema, de tal manera que sea posible mejorarlo.

El software desarrollado considera algunos de estos lineamientos, sin embargo algunos de ellos faltan por ser considerados y queda como trabajo futuro.

3 Animación del análisis léxico

El análisis léxico es la primera fase de un compilador. Su principal función consiste en leer los caracteres de entrada y elaborar como salida una secuencia de componentes léxicos que utiliza el analizador sintáctico para hacer el análisis. El software aprendizaje que hemos desarrollado cubre la descripción de lenguajes regulares través de expresiones regulares, la teoría de autómatas finitos, la generación autómata finito determinista mínimo a partir de una expresión regular. Todos algoritmos fueron implementados como se describen en [1]. La idea de este software es brindar a los alumnos de los cursos de Compiladores y de Teoría de Autómatas Lenguajes Formales material de apoyo complementario que facilite el entendimiento de modelos abstractos.

Concretamente, la versión actual del software permite especificar una expresión regular (siguiendo la notación descrita en [1]) y a partir de ella genera de manera automática:

- el autómata finito no determinista (AFN),
- el autómata finito determinista (AFD), o

el autómata finito determinista con estados mínimos que acepta el lenguaje descrito por la expresión regular.

Además, el software permite dar como entrada cadenas de símbolos y realizar la simulación automática o paso por paso para ver si la cadena es aceptada o no por el autómata. El software realiza la animación gráfica de los algoritmos de construcción de simulación de los autómatas finitos AFDs y AFNs (ver Fi g. 1).

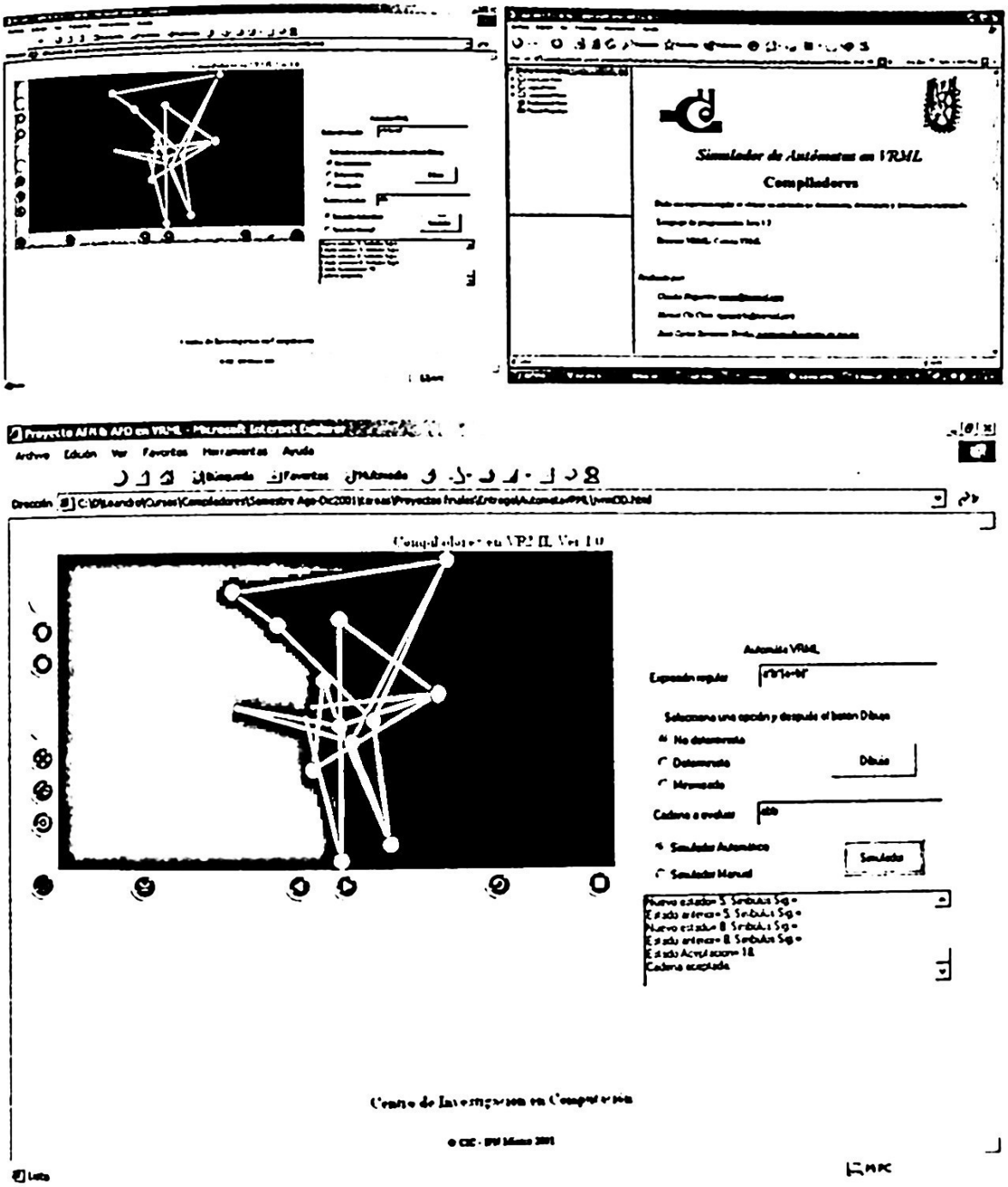


Fig 1. Animación gráfica de la generación y simulación de autómatas finitos.

El autómata que aparece en la Fig. 1 por ejemplo, corresponde a la expresión regular `a*b* (a+b) *`. El autómata que aparece en la figura corresponde al AFN que

acepta el lenguaje descrito por la expresión. En el mundo VRML los estados son representados por esferas grises, el estado inicial por una esfera azul y el final por una esfera roja. Dada una cadena de símbolos, el software permite elegir realizar simulación del autómata manualmente o automáticamente. En el caso de la simulación manual el usuario tendrá que hacer clic sobre el botón Simulador tantas veces como sea necesario hasta que se llegue a un estado final o se rechace la cadena. En el caso de la simulación automática no es necesario ir haciendo clic. En ambas opciones, el software despliega el estado en que se encuentra el autómata así como símbolo de entrada actual en cada paso de la simulación y al final un mensaje indicando que la cadena fue aceptada o rechazada. Durante la animación el estado actual se ilumina de color amarillo, lo cual permite apreciar en que estado se encuentra el autómata en cualquier momento.

4 Diseño e implementación

El software se diseñó e implementó en base al paradigma orientado a objetos. Para el modelado de los diagramas de clases se utilizó la herramienta Rational Rose®. Fig. 2 ilustra el diagrama de clases principal de la aplicación y en la Tabla 1 describe brevemente cada una de las clases que lo componen.

Tabla 1. Principales clases de la aplicación.

Clase	Descripción
`AutomataVRML`	Integra los distintos componentes para llevar a cabo la animación gráfica de los algoritmos.
`Automatax`	Es una clase abstracta que define los elementos comunes un autómata finito: estado inicial, transición, conjunto símbolos, estado final, conjunto de estados.
`Anasin`	Define e implementa el analizador sintáctico para las expresiones regulares.
`Analex`	Define e implementa el analizador léxico para las expresiones regulares.
`MatrizTransAFD`	Define e implementa la estructura de datos para almacenar la descripción del AFD.
`MatrizTransAFDM`	Define e implementa la estructura de datos para almacenar la descripción del AFD mínimo.
`jVRMLaux`	Clase utilitaria para el dibujo de las geometrías que integran el modelo 3D del autómata.

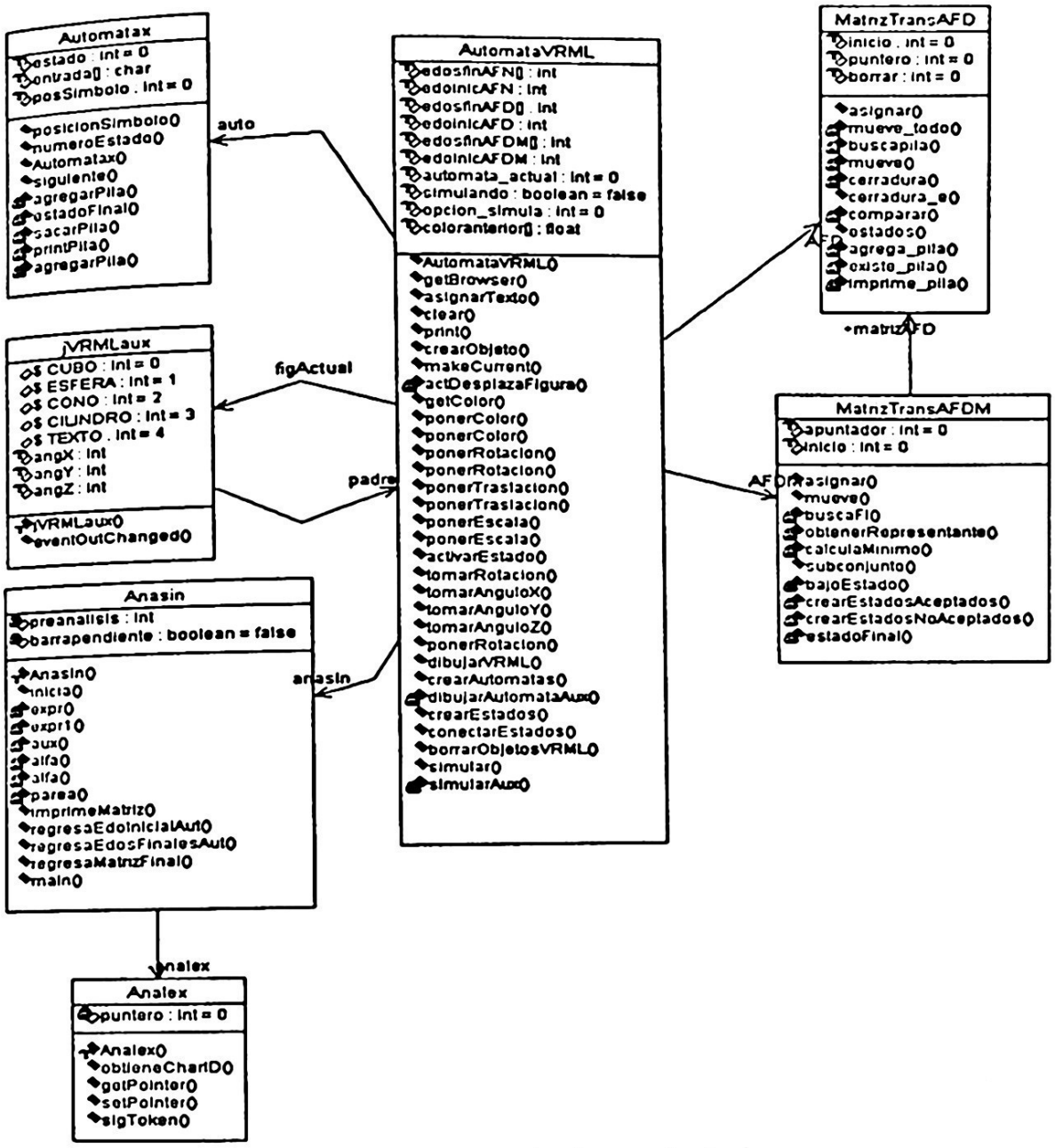


Fig. 2. Diagrama de clases principal.

La utilización de la herramienta Rational Rose para el diseño del software tuvo la ventaja de que permitió documentar el diseño y generar de manera automática dicha documentación en formato Web. Esta documentación ha sido generada con la finalidad de que quienes deseen modificar o extender la funcionalidad del software puedan llevarlo a cabo con relativa facilidad. Se han documentado cada una de las clases, junto con sus atributos, métodos y parámetros de los métodos. La Fig. 3 ilustra parte de la documentación de la clase AutomataVRML y Anasin.

El software es compatible con el Web y para su desarrollo se utilizó los lenguajes Java, VRML [37] y HTML. Java fue utilizado para la implementación de los algoritmos, así como para la implementación del Applet que genera la interfaz 2D. VRML es utilizado para generar la vista 3D de los autómatas y HTML para presentar dicho contenido (ver Fig. 4).

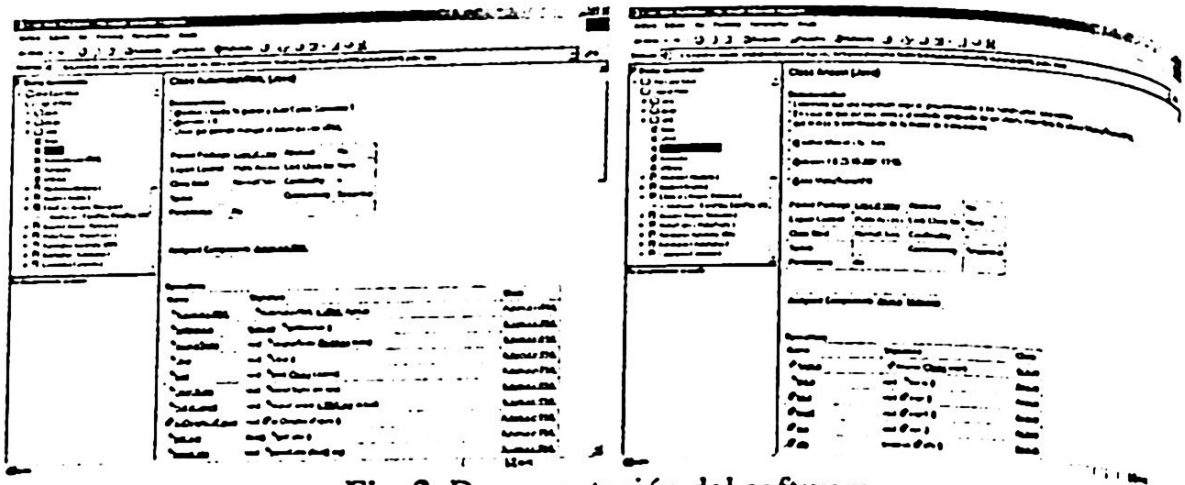


Fig. 3. Documentación del software.

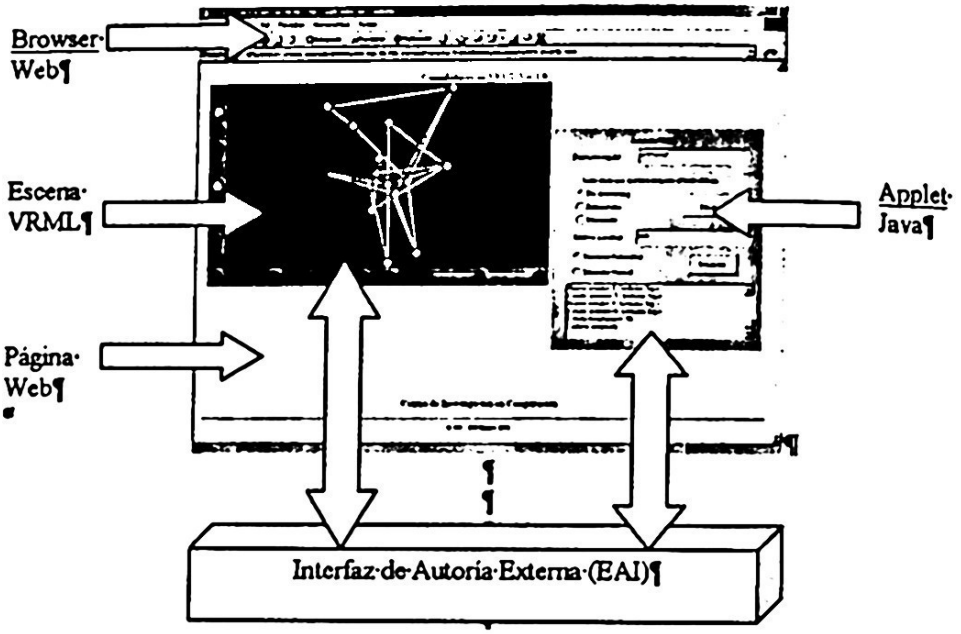


Fig. 4. Tecnologías utilizadas para la implementación del software.

Para lograr la interación del Applet Java con el mundo VRML en el que se construye el autómata, se utilizó la Interfaz de Autoría Interna (External Authoring Interface – EAI) [38]. La EAI es una interfaz de comunicación bidireccional entre VRML y programas externos (por lo general Applets). En nuestro caso, la EAI nos permite modificar dinámicamente el contenido de la escena VRML a través del programa Java; en este caso, el programa Java modifica la escena VRML cuando se generan los autómatas finitos o cuando se realiza la simulación.

6 Evaluación

El software que hemos desarrollado está destinado a estudiantes de los cursos de Compiladores y Lenguajes y Autómatas de Postgrado y Licenciatura. Actualmente esta siendo pre-evaluado por estudiantes del 1er. semestre de la Maestría en Cien-

cias de la Computación del curso de Lenguajes y Autómatas. A pesar de que el software no está terminado al 100%, los estudiantes se han familiarizado fácilmente con las visualizaciones y animaciones y han sido capaces de establecer una conexión de éstas con la teoría (definiciones, algoritmos, *etc.*). Algunos estudiantes han utilizado el software como apoyo para la comprobación de ejercicios previamente realizados con lápiz y papel por ellos mismos, logrando al final un mejor entendimiento de los conceptos y algoritmos. Hemos visto, que la vista 3D generada del autómata resulta atractiva para los estudiantes y éstos han manifestado que les resulta más atractiva esta vista que una en 2D. Además, el hecho de haber utilizado VRML en particular para la generación de la vista 3D agrega al software las capacidades de interacción del visualizador VRML, permitiéndoles a los estudiantes apreciar tanto la construcción como la simulación del autómata desde cualquier punto del espacio tridimensional; los estudiantes pueden rotar el modelo en cualquiera de los tres ejes, acercarse, alejarse, *etc.* Este aspecto ha sido un punto importante e innovador de nuestro software comparado con otros que realizan el mismo tipo de animaciones pero en dos dimensiones. Se ha visto que entre mayor interactividad permita el software a los estudiantes más atractivo resulta para ellos.

Trabajos relacionados

En años recientes, en la Universidad de Saarland en Alemania se han desarrollado visualizaciones en el contexto del diseño de compiladores , incluyendo visualizaciones de máquinas abstractas para lenguajes de programación funcionales, lógicos e imperativos [11], [6], [7]. Estas visualizaciones fueron implementadas en UNIX bajo X-Windows. Las visualizaciones muestran el efecto de la ejecución de instrucciones de máquina sobre el montículo y la pila en tiempo de ejecución, sin embargo, contienen pocas animaciones. Además se ha desarrollado una herramienta llamada VCG (Visualization of Compiler Graphs) para la visualización de grafos que tienen que ver con el diseño de compiladores. Esta herramienta esta disponible para distintas plataformas, incluyendo Microsoft Windows [17], [18], [12]. Además, actualmente existen un gran número de animaciones de algoritmos, sin embargo, se ha desarrollado poco trabajo de importancia en el campo. En [2], [4] se describen muchos de los principios y sistemas existentes para la animación de algoritmos.

En la Universidad de Saarland (Alemania), existe una propuesta similar a la que estamos desarrollando llamada GANIMAL [10]. La meta de este proyecto es crear un software de aprendizaje de exploración para el diseño de compiladores, en el que generen de manera automática a partir de especificaciones las visualizaciones y animaciones de cada una de las fases de un compilador. Sin embargo, los ejemplos que hasta ahora han desarrollado no están basado en el Web y utilizan para su desarrollo una herramienta multimedia llamada ToolBook 3.0 para Windows, a pesar de que se dice que están trabajando en una versión compatible con el Web. Lo anterior hace que la aplicación únicamente pueda ejecutarse sobre plataforma Windows siempre y cuando se cuente con el plugin apropiado para la ejecución del visualizador.

Otros proyectos relacionados al que pretende realizarse son [12], [17], [18], En México, no sabemos de otras Universidades que estén trabajando en proyectos similares.

8 Conlcusiones y trabajo futuro

Hemos desarrollado un software de aprendizaje basado en el Web "Animación Análisis Léxico" que ayuda a los estudiantes a entender mejor los principios construcción de compiladores, en particular de la fase de análisis léxico. El software ofrece, por un lado una introducción interactiva a los problemas de análisis léxico, en el que las definiciones y algoritmos más importantes se presentan de manera gráfica. Las animaciones muestran como es creado un autómata finito a partir una expresión regular, así como su operación.

Una de las aportaciones más importantes de este trabajo en comparación a que realizan el mismo tipo de animaciones, es la utilización de vistas 3D y en particular del uso del lenguaje VRML para la generación de éstas. VRML añade a nuestro software características de realidad virtual, y permite añadirle capacidades interacción en tiempo real adicionales (navegación, caminar, volar, cambiar punto de vista, acercar, alejar, rotar, *etc.*) que lo hacen más atractivo para los diantes.

Aún estamos trabajando en la conclusión del software. Actualmente, estamos bajando en el desarrollo del contenido teórico y las animaciones de algunos de algoritmos de las fases de Análisis Sintáctico y Análisis Semántico. Como trabajo futuro, hemos considerado agregar a nuestro software capacidades multiusuario colaboración, de tal manera que varios estudiantes puedan apreciar la animación los algoritmos e interactuar sobre los modelos al mismo tiempo de manera coordinada . Lo anterior, hará posible la utilización del software para fines de educación distancia. Junto con otros colegas del Centro, hemos desarrollado ideas para construcción de mundos virtuales distribuidos colaborativos basados en el Web [23], que nos permitirán llevar a cabo lo descrito anteriormente.

Referencias

1. Aho, Sethi and Ullman. Addison-Wesley Iberoamericana, 1990.
2. Hausner, D.P. Dobkin. *Making Geometry visible: an Introduction to the Animation Geometric Algoritms*. Computer Science Department, Princeton University, 1996.
3. Braune, S. Diehl, A. Kerren and R. Wilhelm. *Animation of the Generation and Computation of Finite Automata for Learning Software*. In Proceedings of Workshop of plementing Automata WIA'99, Potsdam, Germany, July, Springer Verlag, LNCS 2214. 2001.
4. Brown, Marc H. *Algorithm Animation*. An ACM Distinguished Dissertation: 1987. Cambridge, Massachusetts: The MIT Press, 1987.

5. B. Shniderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* 3er. Edition, Addison-Wesley, 1997.
6. B. Steiner. *Visualization of the abstract Machine MaMa.* Master's Thesis (in German), University of Saarland, Saarbrüken (Germany), 1992.
7. B. Steiner. *Visualization of the abstract Machine WiM.* Master's Thesis (in German), University of Saarland, Saarbrüken (Germany), 1995.
8. Byrne, M.D., Catrambone, R., Stasko, J.T. *Do Algorithm Animations Aid Learning?.* Technical Report GIT-GVU-96-18, August 1996.
9. Duane J. Jarc , Michael B. Feldman , Rachelle S. Heller. *Assessing the benefits of interactive prediction using Web-based algorithm animation courseware.* ACM SIGCSE Bulletin , Proceedings of the thirty-first SIGCSE technical symposium on Computer science education March 2000 Volume 32 Issue 1
10. *GANIMAL Project*, URL: http://rw4.cs.uni-sb.de/~ganimal/
11. G. Kohlmann. *Visualization of the abstract P-Machine.* Master's Thesis (in German), Univeersity of Saarland, Saarbrucken (Germany), 1995.
12. G. Sander. *Visualization of Compiler Graphs.* Technical Report, University of Saarland, Saarbrüken (Germany), 1995.
13. Guido Rößling , Bernd Freisleben. *Experiences in using animations in introductory computer science lectures.* ACM SIGCSE Bulletin , Proceedings of the thirty-first SIGCSE technical symposium on Computer science education March 2000. Volume 32 Issue 1.
14. Guido Rößling , Thomas L. Naps. *Visualisation: A testbed for pedagogical requirements in algorithm visualizations.* Proceedings of the 7th annual conference on Innovation and technology in computer science education June 2002.
15. Hansen, S.R., Narayanan, N.H., Schrimpsher D. *Helping Learners Visualize and Comprehend* Algorithms. Interactive Multimedia Electronic Journal of Computer-Enhanced Learning
16. Herbert L. Dershem , Ryan L. McFall , Ngozi Uti. *Animation of Java linked lists.* ACM SIGCSE Bulletin , Proceedings of the 33rd SIGCSE technical symposium on Computer science education February 2002 Volume 34 Issue 1.
17. I. Lemke. *Development and Implementation of a Visualization Toolkit for Applications in Compiler Construction.* Master's Thesis (in German), University of Saarland, Saarbrüken (Germany), 1994.
18. I. Lemke, G. Sander. *Visualization of Compiler Graphs.* User Documentation, 1994.
19. Kohoe, C., Stasko, J., Taylor, A. *Rethinking the Evaluation of Algorithm Animations as Learning Aids: An Observational Study.* Technical Report GIT-GVU-99-10 March 1999
20. Marc Najork. *Web-based algorithm animation.* Proceedings of the 38th conference on Design automation June 2001.
21. M. H. Brown. *A System for Algorithm Animation.* In Proc. of ACM SIGGRAPH'84, pp. 177-186, July 1984.
22. M. H. Brown. *Algorithm Animation.* MIT Press, 1987.
23. Menchaca R. & Quintero. "*Distributed Virtual Worlds for Collaborative Work based on Java RMI and VRML*", Proceedings of the IEEE 6th International Workshop on Groupware CRIWG 2000, Madeira, Portugal.

24. Peter Gordon McDonald. *Using algorithm animations to assist teaching state space search: an empirical evaluation.* Proceedings of the on Australasian computing education conference December 2000.
25. Quintero-Tellez, R., Menchaca-Mendez, R., Estrada-Segovia. G. M. *Implementation Aspects of Synchronous Collaborative Distributed Virtual Worlds*, 5th IASTED International Conference, Computer Graphics and Imaging, 2002.
26. Rachel Sturm-Beiss , Deborah Sturm. *CS1 concepts using simple animation in Java* The Journal of Computing in Small Colleges , Proceedings of the fifth annual CCSC northeastern conference on The journal of computing in small colleges April 2000. ume 15 Issue 5.
27. Stasko, J.T. *Tango: A Framework and System for Algorithm Animation.* IEEE Computer 23, September 1990, pp.27-39.
28. Stasko, J.T. *Using Student-Built Algorithm Animations as Learning Aids.* ACM SIGCSE, 1997, pp.25-29.
29. Stasko J., Bradre, A., Lewis, C. *Do Algorithm Animations Assist Learning? An Empirical Study and* Analysis. ACM INTERCHI, April 1993, pp.61-66.
30. Stasko J.T., Wehrli, J.F. *Three-Dimensional Computation Visualization.* Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, Technical Report GIT-GVU-92-20, September 1992.
31. Stephan Diehl. *Specializing Visualization Algorithms.* In F.H. Post, G.P. Bonneau, G.M. Nielson, eds., Proceedings of the Dagstuhl Scientific Visualization Seminar 2001, Kluwer Academic Publishers, May 2002.
32. Stephan Diehl. *Web3D (3D on the Web: From Interactive Simulations to Networked Virtual Learning Environments).* In Handbook on Information Technologies for Education & Training, H.H. Adelsberger, B. Collis, J.M. Pawlowski (Eds.), International Handbook on Information Systems Series, Springer, 2002.
33. Stephan Diehl, Andreas Kerren, and Torsten Weller. *Visual Exploration of Generation Algorithms for Finite Automata on the Web.* In "Implementation and Application Automata", S. Yu, A. Paun (Eds.), 5th International Conference, CIAA 2000, London Ontario, Canada, July 24-25, 2000, Springer Verlag, LNCS 2088, 2001.
34. Stephan Diehl, Carsten Görg and Andreas Kerren . *Animating Algorithms Live and Post Mortem.* In "Software Visualization", State-of-the-Art Survey, Springer LNCS 2269, 2002.
35. Thomas Naps. *Instructional interaction with algorithm visualizations.* The Journal Computing in Small Colleges October 2000. Volume 16 Issue 1.
36. Vic Ciesielski , Peter McDonald. *Using animation of state space algorithms to overcome student learning difficulties.* ACM SIGCSE Bulletin, Proceedings of the 6th annual conference on Innovation and technology in computer science education June 2001. Volume 33 Issue 3.
37. VRML. *Virtual Reality Modeling Language.* URL: http://www.web3d.org/technicalinfo/specifications/vrml97/index.htm
38. VRML-EAI. *External Authoring Interface.* URL: http://www.web3d.org/technicalinfo/specifications/eai/index.html

Análisis, diseño e implementación de un simulador de combate aéreo basado en lógica difusa

Francisco Cano [1], Fernando Juárez [1], Servando Ortega [1], Edmundo Camarillo [1], Leandro Balladares Ocaña [2]

[1] Escuela Superior de Computo del Instituto Politécnico Nacional, Unidad Profesional "Adolfo López Mateos" Zacatenco, C.P. 07738, México, México
duran_francisco@hotmail.com, son_fernando@hotmail.com, sislas2012@hotmail.com, edmundordc@hotmail.com

[2] Centro de Investigación en Computación del Instituto Politécnico Nacional, Unidad Profesional "Adolfo López Mateos" Zacatenco, C.P. 07738, México, México
ballad@cic.ipn.mx

Resumen. En este trabajo se describe el análisis, diseño y la implementación de un sistema de software de simulación de combate aéreo entre dos aeronave, llamado JIBA3D. Las aeronaves son controladas por el usuario y la computadora, respectivamente. El combate se lleva a cabo dentro de un espacio o mundo virtual tridimensional (3D) de dimensiones finitas, en el que hay obstáculos que las naves deben evitar. Se describe el análisis, diseño e implementación de cuatro módulos basados en lógica difusa, utilizados para controlar automáticamente la aeronave de la computadora, a saber: control de navegación, control de evasión / ataque, control de sistema de mira y control para proyectiles teledirigidos. El simulador opera y soporta interacción en tiempo real. La implementación fue realizada siguiendo la metodología de la programación orientada a objetos en el lenguaje C++.

1 Introducción

Actualmente, es innegable que una de las principales áreas de aplicación de las Ciencias Computacionales son los juegos interactivos basados en computadora. Dos de las áreas de las que más se han beneficiado los videojuegos actuales son: la graficación por computadora (en especial la graficación 3D en tiempo real) y la inteligencia artificial (IA). Tanto la IA, como las graficas 3D, tienen funciones importantes dentro del área computacional. La primera, como técnica para solucionar problemas en los que tiene que ver la experiencia humana y la segunda dentro del diseño y modelado de sistemas reales como virtuales [2–5], [7–12].

En la actualidad, existen un sinnúmero de sistemas de entretenimiento que hacen uso de graficas tridimensionales (3D), tales sistemas van desde juegos relativamente mples como el "Block Out" [5], hasta avanzados juegos de realidad virtual que hacen uso de dispositivos complejos como cascos, guantes para navegación o lentes

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 89–106, 2003. © Centro de Investigación en Computación, IPN, México.

para proyección estereoscópica [11], [12], [14]. Poro otro lado, la IA permite crear personajes o entidades que den la impresión de tener identidad, comportamiento, "inteligencia", haciendo que los juegos sean más reales, atractivos y presenten yor reto para los usuarios [9]. Para nuestro propósito, la conjunción de las gráficas 3D, la IA y técnicas de interacción, dará como resultado un sistema que involucra tanto la simulación del proceso de toma de decisiones humano simulado con lógica difusa [6], [13], como aspectos de graficación, como son la animación y la detección de colisiones en tiempo real. El desarrollo de este trabajo permitió comprobar utilidad y el buen desempeño de los sistemas difusos en la toma de decisiones ambientes dinámicos. El trabajo aquí presentado bien puede servir de base para desarrollo de simuladores de combate profesionales o para el desarrollo de sistemas de entretenimiento (videojuegos).

2 JIBA 3D

JIBA 3D es un sistema de simulación de combate aéreo basado en software entre dos naves dentro de un ambiente 3D. El juego consta de cinco tipos de naves a gir por el usuario y las naves restantes serán operadas por la computadora. El tipo batallas serán uno a uno, es decir cada quien sólo dispondrá de una nave (usuario y computadora). La vista que tendrá el usuario durante el juego es de tercera persona ya que este tipo de vista tiene un rango más amplio de visión. El ganador será aquel que logre eliminar a la otra nave primero, y ello lo conseguirá por medio del armamento con el que cuentan las naves, a saber: *rayos láser*, cuya trayectoria está definida por una línea recta que se prolonga hacia el punto donde apuntaba la mira de nave; *mísiles*, que pueden ser de *dirección fija* (son los mísiles que van en línea recta hacia el punto donde apuntaba la mira de la nave al momento de dispararlos) y *"teledirigidos"* (son aquellos mísiles que al ser disparados rastrean al enemigo dentro del campo de visión del misil y lo persiguen para tratar de hacer contacto.

Además, cada nave cuenta con un sistema de mira automática que permite ubicar al enemigo y dispararle con precisión, siempre y cuando se encuentre en el rango su campo de visión. Las naves se desplazan por un escenario de dimensiones finitas en forma de cubo. El usuario controla a su nave mediante entradas de teclado o *joystick*, mientras que la computadora, utiliza un sistema de toma de decisiones basado en lógica difusa.

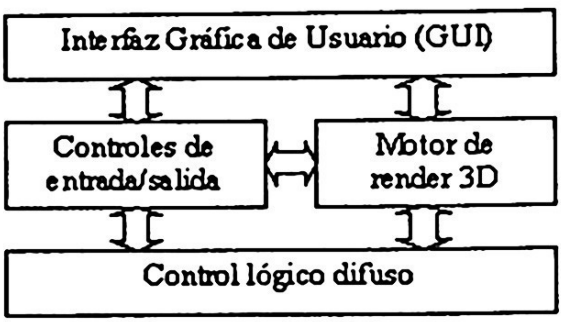


Fig. 1. Diagrama general de JIBA 3D.

En la Fig. 1 se presenta la arquitectura general de JIBA3D. En las siguientes secciones se describe brevemente cada uno de los módulos que la integran.

3 Modelado del sistema

La lógica difusa es una técnica que permite modelar sistemas expertos difusos, los cuales dan grandes ventajas sobre los sistemas expertos comunes. Una de las ventajas más importantes de tales sistemas es que permiten emplear el *razonamiento* aproximado, lo que hace al sistema más robusto y le da capacidad de tomar decisiones de forma muy parecida a como lo hace el hombre [6], [13]. Para desarrollar JIBA3D, se diseñaron e implementaron cuatro sistemas de control difuso:

- *sistema de navegación*,
- *sistema de evasion / ataque*,
- *sistema asistente de mira*, y
- *sistema guía de mísiles*.

Los pasos a seguir para el modelado de los sistemas fueron: 1) Establecer el conjunto de variables de entrada / salida. 2) Determinar los universos de discurso de cada una de estas variables y los conjuntos difusos para las mismas. En JIBA. 3D, se eligieron universos de discurso para la posiciones definidos en unidades de distancia y las direcciones en porcentaje de desplazamiento. La elección de porcentajes de desplazamiento, es debido a que estos podrán determinar más concretamente la razón dirección-velocidad con un patrón máximo de 1 adicional al estándar. 3) Definir las reglas, que determinarán el comportamiento del "ente-máquina" y que será esencial en su capacidad de respuesta. Las reglas, modelan la máquina de inferencia del sistema experto y se definen como del tipo `si ... entonces`. Enseguida, se presentan los sistemas de control modelados como casos aislados y se desarrolla en cada uno de ellos los tres pasos anteriores

3.1 Obtención de las entradas

Las entradas a los sistemas representan las posiciones relativas de los objetos del ambiente o de la nave del usuario, en cuyo caso también se incluye su vector de desplazamiento, y en base a las cuales cada uno de los sistemas difusos generará una acción. Ahora, se explicará como se obtiene cada una de ellas, ya que los parámetros que conocemos corresponden a valores del mundo en el que se desenvuelven y como tales proveen información útil para el buen funcionamiento de los sistemas difusos (ver Fig. 2a).

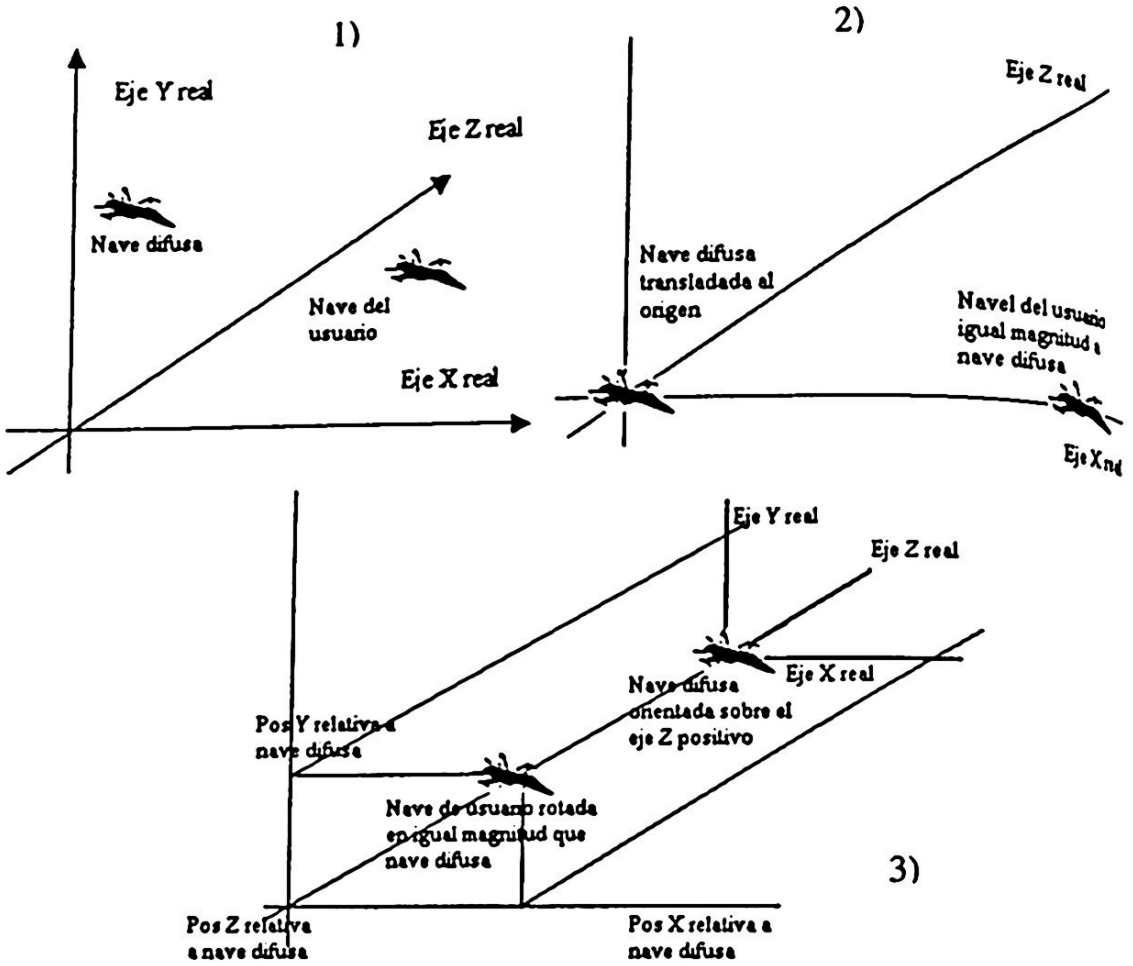


Fig. 2. 1) Posiciones iniciales de la nave difusa y del usuario en el mundo virtual, 2), 3) Transformaciones aplicadas a las posiciones de la nave difusa y del usuario en el mundo virtual.

El primer paso, antes de poder usar estos valores para que los sistemas difusos puedan interpretarlos de manera correcta, es llevar el espacio de coordenadas "universales", a coordenadas relativas a la nave difusa o al misil según sea el caso. Para llevar a cabo esa tarea, primero se debe trasladar a la nave o misil difusos, al "origen" del mundo, y trasladar, en la misma magnitud al objeto que se va a someter consideración por los sistemas difusos. Con esto, solo podemos conocer de manera aproximada, la posición relativa que guarda un elemento del escenario, o el usuario respecto a la nave o misil difusos (ver Fig. 2b).

El segundo paso, requiere hacer una transformación de rotación de cada una las componentes de posición relativa de la nave del usuario o del elemento del escenario, a fin de conocer la orientación de la posición de los mismos. Esta transformación es esencial, ya que todos los elementos del escenario mantienen una orientación "universal", la cual, dependiendo del tipo del elemento puede cambiar (tal el caso de las naves, ya sea la difusa o la del usuario, y la de los mísiles). Así pues la orientación de uno respecto a otro no es la misma en todo momento, y es necesario conocerla para determinar si estos se acercan, alejan o conservan la misma distancia (ver Fig. 2c).

El tercer paso, es interpretar cada uno de los valores obtenidos para que el sistema difuso pueda usarlos de manera correcta y este pueda generar una respuesta

satisfactoria. La interpretación de cada uno de los valores, se basó en la representación de un mapa cartesiano, así pues, para valores del eje de las X relativo, si estos son positivos, su significado semántico, es el de una posición relativa hacia la derecha del vector de desplazamiento de la nave o el misil difusos y un valor negativo significa una posición relativa hacia la izquierda del vector de desplazamiento; para el eje de la Y, un valor positivo significa una posición por sobre el vector de desplazamiento y un valor negativo una posición por debajo del vector de desplazamiento; y para el eje de la Z, un valor positivo significa una posición al frente del vector de desplazamiento y un valor negativo una posición detrás del mismo.

El cuarto paso, consiste en validar los valores obtenidos para determinar que control difuso es el que va a actuar. Para ese propósito, y en base a los universos de discurso de las variables de cada sistema, se someten a consenso en una primera instancia los elementos fijos del "mundo", esto es, todos aquellos elementos que podrían representar un obstáculo dentro del entorno virtual. Si alguno de ellos, representa en forma potencial un obstáculo dentro de nuestro espacio inmediato de desplazamiento, se pondrá en marcha el sistema difuso de navegación, el cual devolverá como respuesta componentes de desplazamiento que permitan a la nave difusa librar de la mejor manera dicho obstáculo. Si por el contrario, ningún elemento fijo cae en la situación anterior, el sistema difuso que entrará en funcionamiento será el de evasión ataque, que como se dijo, tiene la capacidad de generar una respuesta que permita a la nave difusa ganar ventaja sobre la nave del usuario.

3.2 Uso de variables de entrada en los sistemas de navegación – evasión / ataque

Modelado del sistema de navegación

El sistema de navegación, tiene como función librar cualquier obstáculo que se presente enfrente de la nave difusa, para ello, toma las entradas del sistema y efectúa una evaluación de ellas para determinar la posición relativa del obstáculo respecto de la nave. Si el obstáculo se encuentra en un rango dentro del espacio de desplazamiento inmediato de la nave difusa, se procede a hacer una "medición" de éste, tomando como datos las posiciones relativas de sus vértices, ya que todos los objetos inanimados del escenario son poligonales. Para obtener de manera adecuada esta información, se hace uso de tres variables temporales, las cuales van a almacenar las posiciones relativas de los vértices del objeto que se encuentran más alejadas la izquierda, a la derecha y hacia arriba respectivamente, ya que todos los objetos son cerrados y están fijos al piso del ambiente virtual. Habiendo realizado la medición del objeto, se someten a consenso las tres variables auxiliares, para obtener la de menor magnitud, ya que ésta nos permitirá encontrar la trayectoria menos forzada para evadir al obstáculo. Habiendo realizado el proceso anterior, el valor obtenido se envía como entrada al sistema difuso y este genera como salida una componente de desplazamiento ya sea para X, o Y, que junto con las otras dos, nos darán un desplazamiento real dentro de nuestro mundo virtual (ver Fig. 3). En el Anexo A.1 se lista la definición de las variables de entrada y matrices de reglas del sistema de navegación.

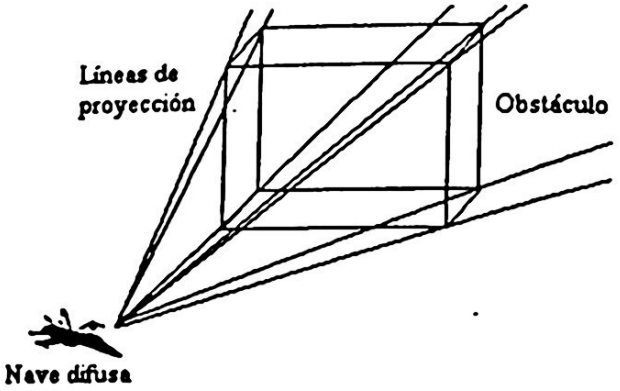


Fig. 3. Detección de obstáculos.

Modelado del sistema de evasión / ataque

En este sistema se detectaron dos variables generales de entrada y una de salida, las cuales se parten en tres cada una. La primera describe la posición relativa del objeto controlado por el usuario respecto del controlado por la máquina. La segunda, describe la dirección que el objeto controlado por el usuario, toma respecto del controlado por la máquina. La tercer variable, expresa o representa la dirección que debe tomar el objeto controlado por la máquina para obtener una ventaja sobre el usuario.

La primer variable, correspondiente a la posición relativa se divide en componentes relativas para los ejes X, Y, y Z respectivamente, y tomando siempre como origen a la nave difusa con dirección sobre el eje Z positivo. En el Anexo A.2, se muestran la definición del universo de discurso. La segunda variable, que describe la dirección relativa, se divide en componentes simples de desplazamiento relativo sobre los ejes X, Y, y Z como en el caso anterior, y que permitirán al control difuso "saber" como se comporta el usuario y que le permitirá definir su respuesta, ya sea de evasión o de ataque. En el Anexo A.2, se muestran las definiciones de sus universos de discurso. La tercer variable, que describe las variables de desplazamiento del objeto controlado por la máquina, se divide al igual que la segunda en componentes de desplazamiento relativo para cada uno de los ejes, y se obtendrán como salida del proceso de inferencia difuso del sistema de control. Los valores reales de estas variables, definirán el comportamiento de evasión o ataque de la nave difusa, por lo que la correcta interpretación de las mismas se traducirá en un desempeño óptimo de la nave difusa. En el Anexo A.2 se muestra la definición de ésta.

Habiendo definido los universos de discurso y los conjuntos, la tarea más importante es ahora modelar las reglas difusas y la máquina de inferencia. La definición de las reglas difusas, se hizo empleando el modelo difuso del sistema que se quería controlar y en base a un análisis del comportamiento del mismo. En el Anexo A.2 se muestran solo los casos generales en forma de matriz de reglas que se definieron para el modelado del sistema de evasión / ataque. En estas matrices de reglas, se pueden observar algunas casillas en blanco, debido a que estas se consideraron cruciales para la toma de decisiones y para la que se definió alrededor de 750 reglas específicas, que por cuestión de formato, no se incluyen en el artículo

El sistema de evasión / ataque, basa su funcionamiento en la posición relativa que el usuario representa respecto a la nave difusa, y dependiendo de esta, se considerará

o no el parámetro de desplazamiento relativo, que determinará si la acción a emprender es de evasión o ataque. Al validar la posición relativa del usuario respecto de la nave difusa, se puede determinar si esta se encuentra dentro de nuestro espacio de visión, es decir, si puede ser captada por nuestro "radar" virtual, con lo que se pondrá o no en marcha el sistema difuso. Si el usuario se encuentra visible a nuestro radar, entonces se tomarán sus parámetros de posición y desplazamiento relativos para enviarlos al control difuso. En caso contrario se considerará una visión parcial para cada una de las componentes, las cuales, en cada caso serán requeridas por la función difusa. Si ninguno de los casos anteriores se cumple, se efectuarán acciones que permitan a la nave difusa encaminarse al encuentro de la nave del usuario (ver Fig. 4).

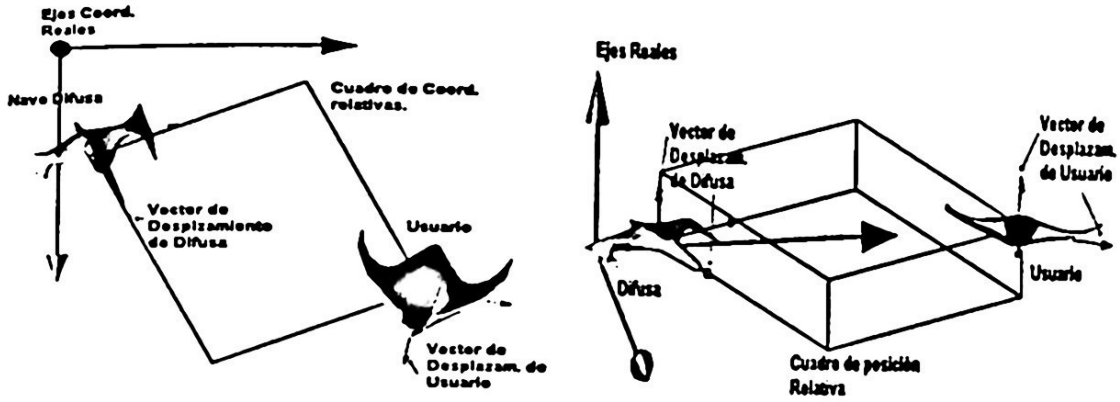


Fig. 4. Vista XZ y de perspectiva de situación inicial de evasión / ataque.

En cualquiera de los tres casos anteriores, la salida generada por el sistema simple o el difuso, será convertida a desplazamientos reales del mundo virtual que permitan a la nave difusa generar una respuesta que le beneficie, y permita obtener ventaja sobre el usuario (ver Fig. 5).

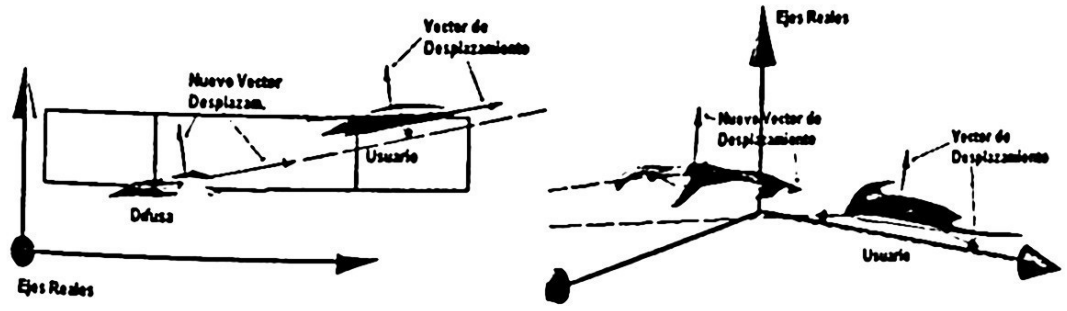


Fig. 5. Vistas auxiliares y de perspectiva de situación final de evasión / ataque.

Un caso especial se debe considerar cuando la nave difusa se acerca a los límites permitidos del mundo virtual, ya que no debe por ningún motivo abandonar al escenario, independientemente de donde se encuentre el usuario. En ese caso, se debe emprender una acción temporal que obligue a la nave difusa a regresar al espacio

permitido de acción. En un principio, se pensó en crear un sistema que controlara esa situación, pero con el modelado del sistema de evasión / ataque, se observó que se podía prescindir del mismo, pues el problema se solucionaba con hacer, por medio del sistema de evasión / ataque, que la nave difusa considerará un enemigo virtual situado en el centro del escenario. Así, en cualquier caso en que la nave difusa se acercara a los límites externos del mundo virtual, emprendería por un tiempo muy limitado un ataque contra ese enemigo virtual simulando el regreso al escenario eliminando así la posibilidad de dejar al usuario sin enemigo.

La puesta en funcionamiento de uno u otro sistema (navegación o evasión / ataque), dependerá de la situación que se presente, así pues, como se explicó en último punto de la sección anterior, si se tiene un obstáculo dentro del espacio inmediato de desplazamiento, el sistema que se pondrá en marcha será el de navegación.

3.3 Uso de variables de entrada en los sistemas de asistente de mira y guía de misiles

Los sistemas de asistente de mira y de guía de misiles son muy parecidos en cuanto a su funcionamiento, ya que ambos tienen una región ciega en el espacio inmediato de desplazamiento, que permite al usuario y a la nave difusa según sea el caso, evitar ser "detectados" por el mecanismo estricto de ataque, es decir, por la ejecución de una acción de disparo de misiles difusos, implícita por la detección del objetivo en la mira. En el espacio inmediato a esta región ciega, se encuentra una pirámide que representa el espacio de visión de dichos sistemas. Como dicho espacio no uniforme, debido a su forma piramidal creciente, se detectó la ventaja de poder aumentar la precisión en la localización de las naves, y en la estimación de su posición siguiente inmediata por lo que se empleó un factor de escalamiento que permite hacer dicha transformación independientemente de la posición sobre el eje de desplazamiento inmediato relativo de la nave que haga uso de dicho sistema.

El sistema de guía de misiles, emplea una representación bidimensional de la posición y desplazamiento relativos del objetivo para determinar si tiene que cambiar o no sus componentes de desplazamiento, situarse sobre el eje de desplazamiento inmediato y poder hacer contacto con este. Por lo tanto, la salida generada está compuesta por un vector tridimensional de desplazamiento real que define la trayectoria inmediata del misil.

Sistema asistente de mira

Para que el usuario no se preocupe por apuntar hacia las naves enemigas, se diseñó una mira "inteligente" capaz de seguir, en un cierto rango de visión, a la nave enemiga más cercana. Para ello se definieron los conjuntos de entrada y salida respectivos, así como sus universos de discurso, conjuntos difusos, tipos de funciones membresía y reglas difusas (ver Anexo A.3).

Conjuntos de entrada y salida

Para poder conocer como se va a desplazar la mira es necesario conocer la posición de la nave enemiga y su desplazamiento. Lo anterior permitió concluir que se necesitaban los siguientes conjuntos de entrada: posición en X, Y, Z de la nave enemiga, así como su desplazamiento sobre cada eje. Como salida era necesario saber la dirección sobre los ejes X, Y, Z en la que se desplazará la mira, es decir, sus nuevos desplazamientos.

Definición de conjuntos difusos

Antes de poder definir los conjuntos difusos de las variables de entrada y salida, es necesario definir primero el universo de discurso de las variables, así como la forma en que la mira va a ver al enemigo. Primero definiremos como es el campo de visión de la mira o región de visión. La mira va a poder localizar al enemigo siempre y cuando este se localice en el campo de visión especificado para la mira. Para nuestro caso el campo de visión será una pirámide. Se escoge la pirámide porque es muy similar a la vista que se tendría en la realidad (entre más cerca menos se ve, o menor rango, y entre más lejos es mayor la visión). La punta de la pirámide estará en las coordenadas de la nave, pero, como es de suponer, la mira no podrá desplazarse hasta este punto, ya que esto querría decir que: (1) podemos dejar que el enemigo se acerque demasiado; (2) el espacio con el que se trabaja es del tamaño de un punto (ver Fig. 6).

En vez de eso, consideraremos que la mira comienza a funcionar en una parte delante de la nave. Otra consideración que hay que realizar antes de comenzar, es que la mira tomará como origen a la nave en la que está implementada, así como las coordenadas relativas con respecto a ella de la nave enemiga.

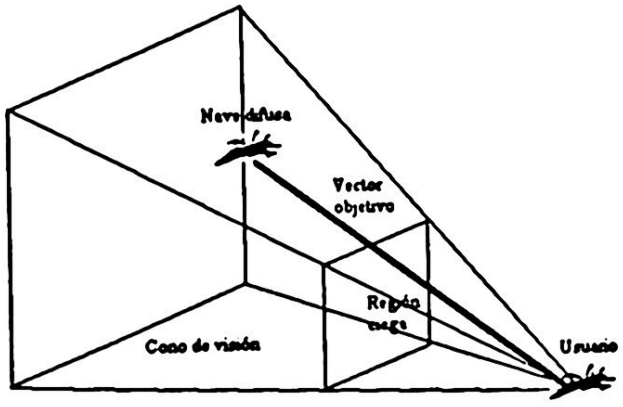


Fig. 6. Definición de la pirámide de visión de las naves.

Conjuntos difusos de entrada

Con esto podemos empezar a definir el universo de discurso. Para `XPosU` y `YPosU` tendremos conjuntos idénticos (sólo cambiarán los nombres de la etiquetas) así como universos de discurso, ya que como se puede apreciar en el Anexo A.3 siempre se tendrán cuadros en lo que a X y Y se refiere. En el caso del eje Z no es igual, ya que en este no podemos considerar posiciones negativas, sino que, es desde una

posición delante del origen hasta otra. Los desplazamientos de XDesU y YDesU como en los casos de las posiciones, son iguales (ver Anexo A.3).

Conjuntos difusos de salida
Los conjuntos de salida quedan de la forma mostrada en el Anexo A, que es idéntica a los desplazamientos en los ejes.

Definición de la base de reglas difusas
Una vez definidos los conjuntos de entrada y salida, sus universos de discurso, menzaremos a realizar la base de reglas para la mira inteligente. Para ello realizaremos módulos que controlen el movimiento en cada eje y simplificar así la base reglas. Comenzaremos con el movimiento sobre el eje X (cabe señalar que iguales los conjuntos X y Y, su base de reglas es idéntica solo cambiando las etiquetas correspondientes). Ver las reglas en el Anexo A.

Sistema de guía de mísiles
Cuando cualquiera de las naves (del usuario y difusa) disparen mísiles, tendrán posibilidad de que sean de dos tipos: misil directo y misil guiado. El misil directo tiene como característica que siempre va en una sola dirección y nunca cambia; cambio los mísiles guiados cuentan con un sistema difuso que les permite local la nave enemiga e ir directo hacia ella para hacer impacto. En esta sección se hablará del sistema difuso que se modeló para ello.

Conjuntos de entrada y salida
Para poder conocer como se va a desplazar el mira necesitamos conocer la posición de la nave enemiga y su desplazamiento. Lo anterior nos lleva a concluir que necesitamos los siguientes conjuntos de entrada: posición en X, Y, Z de la nave enemiga (XPosU, YPosU, ZposU respectivamente), así como su desplazamiento sobre eje (XDesU, YDesU y ZDesU). Como salida necesitamos saber la dirección los ejes X, Y, Z en la que se desplazará el misil, es decir, sus nuevos desplazamientos: XNDes, y YNDes (el desplazamiento sobre el eje Z no se considera porque siempre es hacia delante y constante)

Definición de conjuntos difusos
Los conjunto difusos utilizados para el sistema difuso del misil, son los mismos en la mira, solo se elimina el sistema que controla el movimiento sobre el eje También cabe mencionar que el misil contará con una región ciega al igual que mira, solo que este es más amplio. Otra consideración que hay que realizar antes comenzar, es que el misil tomará como origen a la nave en de la que saldrá para implementada, y después el control difuso modificará sus coordenadas.

4 Motor de *rendering* 3D

El proceso de síntesis de la escena se dividió en dos procesos; el procesamiento geométrico y el proceso *raster*. En el primero de ellos se involucró a los procesos de transformación de coordenadas reales a coordenadas de visión, y fue donde se empleó los procesos de recorte o *clipping* y proyección de la escena. La Fig. 7 muestra los módulos de los que se compone el motor de *render* 3D de JIBA3D.

En JIBA3D se utilizó proyección de perspectiva, en la que se emplea un cono cuadrado de visión, que permite crear un efecto de profundidad en la imagen generada y da mayor realismo a la escena sintetizada, además de que este tipo de proyección puede ser realizada en tiempo real[1].

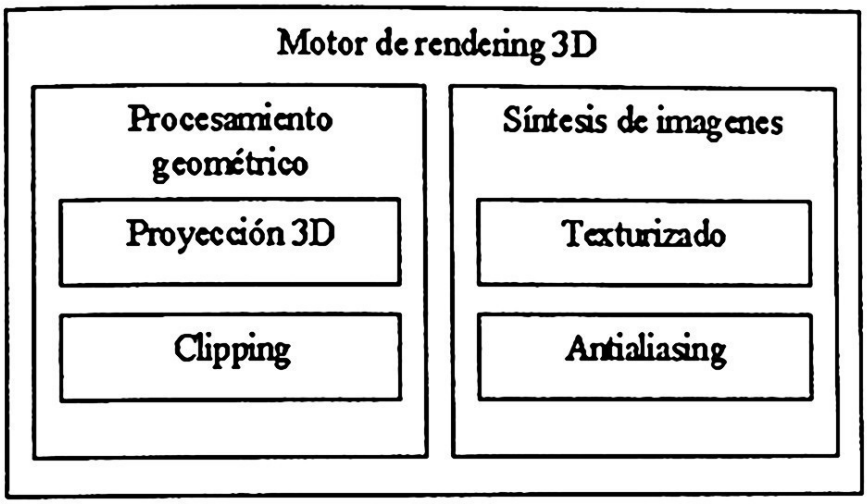


Fig. 7. Diagrama a bloques del motor de render 3D.

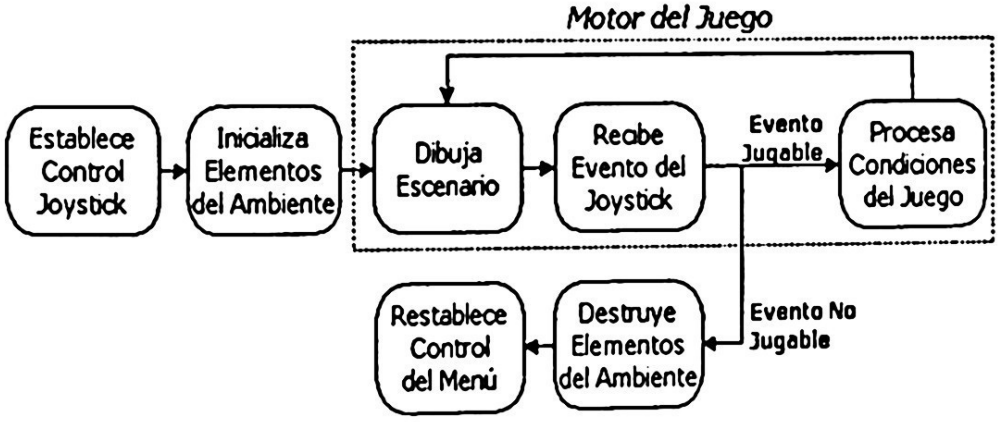


Fig. 8. Motor del juego.

[1] Existen otras técnicas que utilizan algoritmos de iluminación como el *ray tracing* - trazado de rayos - que permiten dar gran calidad a las imágenes de las escenas, pero son muy costosas en cuanto al tiempo de procesamiento y solo son usadas para crear animaciones que no requieren ser generadas en tiempo real.

5 Motor del juego

El motor del juego es el núcleo de JIBA3D y quien integra la funcionalidad de módulos descritos anteriormente para dar vida al sistema, además implementa lógica del mismo.

En la Fig. 8 se ilustra el contexto del motor de juego y su papel en JIBA3D.

6 Tecnologías de desarrollo

JIBA3D fue implementado sobre Windows 98 utilizando el lenguaje de programación C++. Se utilizó la herramienta Borland C++ Builder 5.0. El motor de render implementó utilizando el API GLIDE de la empresa 3dfx Interactive.

Para la implementación de los controles difusos, se utilizó la herramienta FIDE [1], que permite definir de manera muy sencilla y con reglas sintácticas simples un sistema difuso completo, además de generar el código fuente en ANSI C sistema modelado, en forma de una función que recibe como parámetros los valores de las variables de entrada sin fusificar y retorna los valores de las variables de salida defusificadas. El aspecto anterior dio gran apoyo a la tarea de diseño y parte de implementación de nuestros sistemas de toma de decisiones, ya que no fue necesario desarrollar los procesos de fusificación, la máquina de inferencia y defusificación. Un punto importante, es el hecho de que con esta herramienta se pudo observar aunque a un nivel muy básico el comportamiento de nuestros sistemas, lo que nos permitió hacer estimaciones de comportamiento en aplicaciones reales.

Referencias

1. Aptronix, Inc. (1994), "*FIDE: Fuzzy Inference Development Environment*".
2. Watt (1993), "*3D Computer Graphics*", Addison-Wesley.
3. Watt & F. Policarpio (2001), "*3D Games Real-time Rendering and Software Techonology*", Volume One, Addison-Wesley.
4. Watt, S. Maddock (2000), "*Computer Games Techonolgy and Higher Education*", Virtual Reality: Research, Development and Applications, Volume 5, Number 4, p. 185-194.
5. "*Block Out*", URL: http://www.xdgames.com/games/borx/
6. Chin-teng Lin & C.S. George Lee (1996), "*Neural Fuzzy Systems*", Prentince Hall.
7. Durán C. Fco, et al (2001), "*Manual de Referencia Técnica de JIBA3D*", Trabajo Terminal TTR0033 de la ESCOM -IPN.
8. John E. Laird (2002), "*Research in Human – Level AI Using Computer Games*", Communications of the ACM, January 2002, Vol. 45, Number 1, p. 32-35.
9. M. Cavazza (2000), "*AI in Computer Games: Survey and Perspectives*", Virtual Reality: Research, Development and Applications, Volume 5, Number 4, p. 223-235.

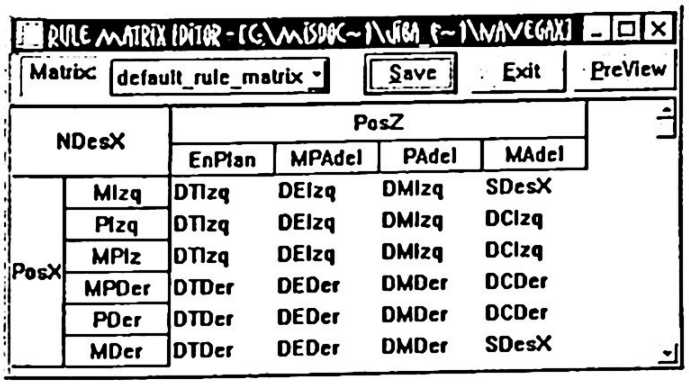
10. Michael Lewis & Jeffrey Jacobson (2002), "*Game Engines in Scientific Research*", Communications of the ACM, January 2002, Vol. 45, Number 1, p. 27-31.
11. Scientific American (2002), "*Information Techology: Augmented Reality: A New Way of Seeing*", April 2002, p. 34.
12. Wayne P. & Bruce T. (2002), "*ARQuake: The Outdoor Augmented Reality Gaming System*", Communications of the ACM, January 2002, Vol. 45, Number 1, p. 36-38.
13. Witold Pedrycz (1993), "*Fuzzy Control and Fuzzy Systems*", Research Studies Press.
14. Wolgang B., et al (2001), "*Projects in VR*", IEEE Computer Graphics and Applications, November/December 2001, p. 14-17.

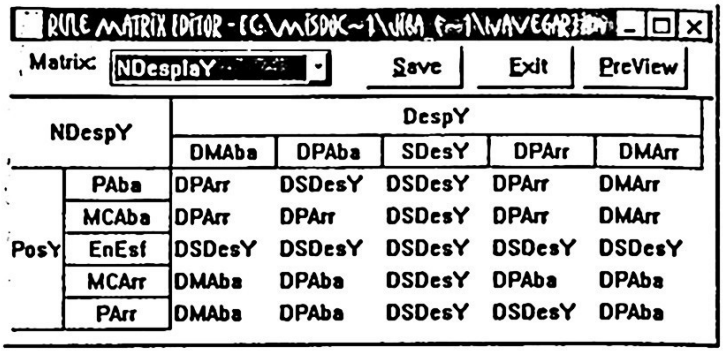
Anexo A – Modelado de los sistemas difusos

A.1 Variables de entrada y matrices de reglas del sistema de navegación

```
invar PosX "" :-20.0 (  ) 20.0 [
   MIzq (@-20.0, 1, @-15.0, 1, @-10.0, 0),
   PIzq (@-15.0, 0, @-10.0, 1, @-5.0, 0),
   CIzq (@-10.0, 0, @-5.0, 1, @0.0, 0),
   CDer (@0.0, 0, @5.0, 1, @10.0, 0),
   PDer (@5.0, 0, @10.0, 1, @15.0, 0),
   MDer (@10.0, 0, @15.0, 1, @20.0, 1)
];
invar PosZ "" :0.0 (  ) 30.0 [
   Colision(@0.0, 1, @5.0, 0),
   EnFrente(@0.0, 0, @5.0, 1, @10.0, 0),
   MCerca(@5.0, 0, @10.0, 1, @15.0, 0),
   PCerca(@10.0, 0, @16.0, 1, @22.0, 0),
   Normal (@16.0, 0, @22.0, 1, @30.0, 1)
];
invar PosY "" :0.0 (  ) 25.0 [
   Arriba(@0.0, 0, @10.0, 1, @15.0, 1),
   PArr (@10.0, 0, @15.0, 1, @20.0, 0),
   MArr (@15.0, 0, @20.0, 1, @25.0, 1 )
];
```

donde: `PosX`, `PosY` y `PosZ` son las componentes de posición relativas del objeto ambiental respecto a la nave difusa, Las palabras al inicio de cada línea son las etiquetas de los conjuntos difusos, los valores precedidos por arriba definen los rangos de cada conjunto difuso, seguido de su valor de pertenencia.





A.2 Definición de variables de entrada de posición y desplazamiento del sistema de evasión / ataque

```
invar PosZ "Pix" :-240.0 (  ) 240.0 [
   HAtras (@-240.0, 1, @-180.0, 1, @-120.0, 0),
   MuAtras (@-180.0, 0, @-120.0, 1, @-80.0, 0),
   MeAtras (@-120.0, 0, @-80.0, 1, @-40.0, 0),
   PAtras (@-80.0, 0, @-40.0, 1, @0.0, 0),
   EnPlan (@-10.0, 0, @0.0, 1, @10.0, 0),
   PAdel (@0.0, 0, @40.0, 1, @80.0, 0),
   MeAdel (@40.0, 0, @80.0, 1, @120.0, 0),
   MuAdel (@80.0, 0, @120.0, 1, @180.0, 0),
   HAdel (@120.0, 0, @180.0, 1, @240, 1)
];
invar PosZ "Pix" :-240.0 (  ) 240.0 [
   HAtras (@-240.0, 1, @-180.0, 1, @-120.0, 0),
   MuAtras (@-180.0, 0, @-120.0, 1, @-80.0, 0),
   MeAtras (@-120.0, 0, @-80.0, 1, @-40.0, 0),
   PAtras (@-80.0, 0, @-40.0, 1, @0.0, 0),
   EnPlan (@-10.0, 0, @0.0, 1, @10.0, 0),
   PAdel (@0.0, 0, @40.0, 1, @80.0, 0),
   MeAdel (@40.0, 0, @80.0, 1, @120.0, 0),
   MuAdel (@80.0, 0, @120.0, 1, @180.0, 0),
   HAdel (@120.0, 0, @180.0, 1, @240, 1)
];
invar PosY "Pix" :-240.0 (  ) 240.0 [
   HAba (@-240.0, 1, @-180.0, 1, @-120.0, 0),
   MuAba (@-180.0, 0, @-120.0, 1, @-80.0, 0),
   MeAba(@-120.0, 0, @-80.0, 1, @-40.0, 0),
   PAba (@-80.0, 0, @-40.0, 1, @0.0, 0),
   EnNiv (@-50.0, 0, @0.0, 1, @5.0, 0),
   PArr (@0.0, 0, @40.0, 1, @80.0, 0),
   MeArr (@40.0, 0, @80.0, 1, @120.0, 0),
   MuArr (@80.0, 0, @120.0, 1, @180.0, 0),
   Harr (@120.0, 0, @180.0, 1, @240.0, 1)
];
```

donde: PosX, PosY y PosZ representan las componentes de posición relativas del usuario respecto a la nave difusa, Las palabras al inicio de cada línea son las etiquetas de los conjuntos difusos, los valores precedidos por arriba definen los rangos de cada conjunto difuso seguido de su valor de pertenencia.

```
invar DesX "" :-1.000000 (  ) 1.000000 [
   DTIzq (@-1.0, 1, @-0.75, 0),
   DEIzq (@-1.0, 0, @-0.75, 1, @-0.5, 0),
   DMIzq (@-0.75, 0, @-0.5, 1, @-0.25, 0),
   DCIzq (@-0.5, 0, @-0.25, 1, @0.0, 0),
   SDesX (@-1.0, 0, @0.0, 0, @1.0, 0.0),
   DCDer (@0.0, 0, @0.25, 1, @0.5, 0),
   DMDer (@0.25, 0, @0.5, 1, @0.75, 0),
   DEDer (@0.5, 0, @0.75, 1, @1.0, 0),
   DTDer (@0.75, 0, @1.0, 1)
];
invar DesZ "" :-1.0 (  ) 1.000000 [
   DTAtras (@-1.0, 1, @-0.75, 1,@-0.55, 0),
   DEAtras  (@-0.75, 0, @-0.55, 1, @-0.35, 0),
   DMAtras (@-0.55, 0, @-0.35, 1, @-0.15, 0),
   DCAtras  (@-0.35, 0, @-0.15, 1, @0.0, 0),
   SDesZ (@-0.15, 0, @0.0, 1, @0.15, 0),
   DCAdel (@0.0, 0, @0.15, 1, @0.35, 0),
   DMAdel (@0.15, 0, @0.35, 1, @0.55, 0),
   DEAdel (@0.35, 0, @0.55, 1, @0.75, 0),
   DTAdel (@0.55, 0, @0.75, 1, @1.0, 1)
];
invar DesY "" :-1.0 (  ) 1.0 [
   DTAba (@-1.0, 1, @-0.75, 1,@-0.55, 0),
   DEAba (@-0.75, 0, @-0.55, 1, @-0.35, 0),
   DMAba (@-0.55, 0, @-0.35, 1, @-0.15, 0),
   DCAba (@-0.35, 0, @-0.15, 1, @0.0, 0),
   SDesY (@-0.15, 0, @0.0, 1, @0.15, 0),
   DCArr (@0.0, 0, @0.15, 1, @0.35, 0),
   DMArr (@0.15, 0, @0.35, 1, @0.55, 0),
   DEArr (@0.35, 0, @0.55, 1, @0.75, 0),
   DTArr (@0.55, 0, @0.75, 1, @1.0, 1)
];
```

donde: `DesX`, `DesY` y `DesZ` representan las componentes relativas de desplazamiento del usuario respecto a la nave difusa, Las palabras al inicio de cada línea son las etiquetas de los conjuntos difusos, los valores precedidos por arriba definen los rangos de cada conjunto difuso, seguido de su valor de pertenencia.

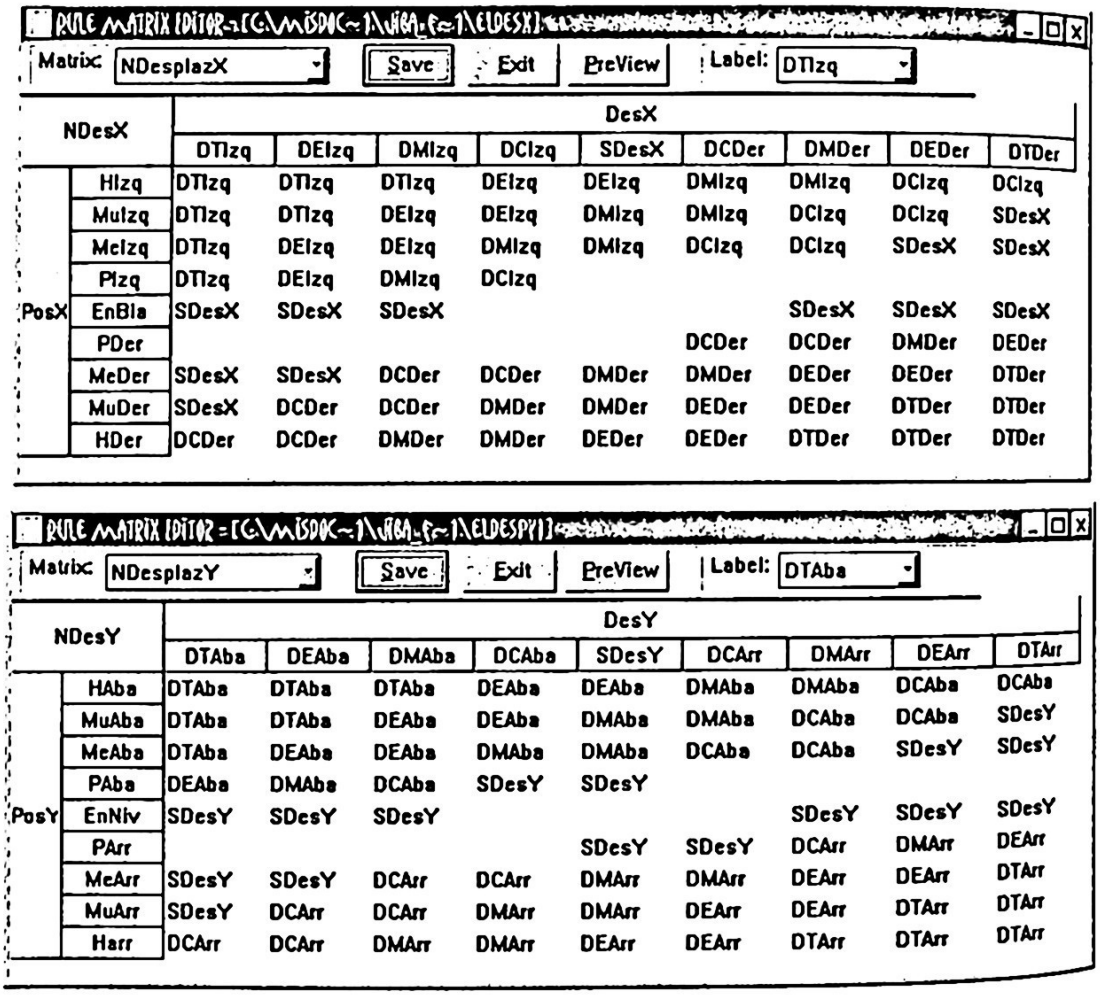
Variable de salida de desplazamiento del sistema de evasión / ataque

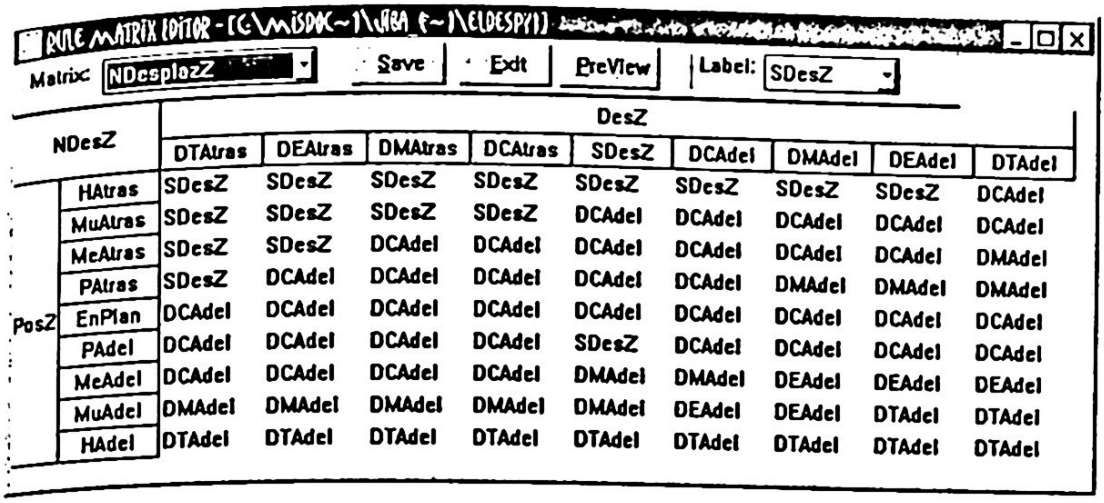
```
outvar NDesX "" :-0.25 ( ) 0.25 centroid [
   DTIzq (@-0.25, 1, @0.15, 0 ),
   DEIzq (@-0.20, 0, @-0.15, 1, @-0.10, 0),
   DMIzq (@-0.15, 0, @-0.10, 1, @-0.05, 0),
   DCIzq (@-0.10, 0, @-0.05, 1, @0.0, 0),
   SDesX (@-0.05, 0.00, @-0.0, 1, @0.05, 0),
   DCDer (@0.0, 0, @0.05, 1, @0.10, 0),
   DMDer (@0.05, 0, @0.10, 1, @0.15, 0),
   DEDer (@0.10, 0, @0.15, 1, @0.20, 0),
   DTDer (@0.15, 0, @0.25, 1)
];
outvar NDesZ "" :0.0 ( ) 1.0 centroid [
   SDesZ (@0.000000, 1, @0.25, 0),
   DCAdel (@0.0, 0, @0.25, 1, @0.50, 0),
```

```
    DMAdel (@0.25, 0, @0.50, 1, @0.750, 0),
    DEAdel (@0.50, 0, @0.75, 1, @1.0, 0),
    DTAdel (@0.75, 0, @1.0, 1)
]
outvar NDesY "" :-0.25 ( ) 0.25 centroid [
    DTAba (@-0.25, 1, @-0.20, 1, @-0.15, 0),
    DEAba (@-0.20, 0, @-0.15, 1, @-0.10, 0),
    DMAba (@-0.15, 0, @-0.10, 1, @-0.050, 0),
    DCAba (@-0.10, 0, @-0.050, 1, @0.0, 0),
    SDesY (@-0.05, 0, @0.0, 1, @0.05, 0),
    DCArr (@0.0, 0, @0.05, 1, @0.10, 0),
    DMArr (@0.05, 0, @0.10, 1, @0.15, 0),
    DEArr (@0.10, 0, @0.15, 1, @0.20, 0),
    DTArr (@0.15, 0, @0.20, 1, @0.25, 1)
];
```

donde: NDesX, NDesY y NDesZ representan las componentes relativas de desplazami que deberá tomar la nave difusa respecto del usuario, Las palabras al inicio de cada línea las etiquetas de los conjuntos difusos, los valores precedidos por arriba definen los rango cada conjunto difuso, seguido de su valor de pertenencia.

Matrices de reglas




RULE MATRIX EDITOR - [C:\MISDOC~1\JUEGA~1\ELDESPY1]

Matrix: NDesplozZ | Save | Exit | PreView | Label: SDesZ

NDesZ	DesZ								
	DTAtras	DEAtras	DMAtras	DCAtras	SDesZ	DCAdel	DMAdel	DEAdel	DTAdel
HAtras	SDesZ	SDesZ	SDesZ	SDesZ	SDesZ	SDesZ	SDesZ	SDesZ	DCAdel
MuAtras	SDesZ	SDesZ	SDesZ	SDesZ	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel
MeAtras	SDesZ	SDesZ	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DMAdel
PAtras	SDesZ	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DMAdel	DMAdel	DMAdel
PosZ EnPlan	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel	DCAdel
PAdel	DCAdel	DCAdel	DCAdel	DCAdel	SDesZ	DCAdel	DCAdel	DCAdel	DCAdel
MeAdel	DCAdel	DCAdel	DCAdel	DCAdel	DMAdel	DMAdel	DEAdel	DEAdel	DEAdel
MuAdel	DMAdel	DMAdel	DMAdel	DMAdel	DMAdel	DEAdel	DEAdel	DTAdel	DTAdel
HAdel	DTAdel	DTAdel	DTAdel	DTAdel	DTAdel	DTAdel	DTAdel	DTAdel	DTAdel

A.3. Sistema asistente de mira

Definición de variables de entrada de posición X, Y y Z

```
invar XPosU "pix" :-125 (1) 125 [
   DIzq (@-125, 1,@-91,1, @-62, 0),
   MIzq (@-91, 0, @-62, 1, @-31, 0),
   PIzq (@-50, 0, @-20, 1, @0, 0),
   EnplaX (@-20, 0, @0, 1, @20, 0),
   PDer (@0, 0, @20, 1, @50, 0),
   MDer (@31, 0, @62, 1, @91, 0),
   DDer (@62, 0,@91,1, @125, 1)
];
invar YPosU "pix" :-125 (1) 125 [
   DAba (@-125,1,@-91,1, @-50, 0),
   MAba (@-91, 0, @-62, 1, @-31, 0),
   PAba (@-62, 0, @-20, 1, @0, 0),
   EnplaY (@-20, 0, @0, 1, @20, 0),
   PArr (@0, 0, @20, 1, @50, 0),
   MArr (@31, 0, @62, 1, @91, 0),
   DArr (@62, 0,@91,1, @125, 1)
];
invar ZPosU "pix" :100 ( 1 ) 350 [
   Enfrente (@100, 1, @135, 0),
   MCerca (@100, 0, @135, 1, @170, 0),
   Cerca (@135, 0, @170, 1, @205, 0),
   MeCerca (@170, 0, @205, 1, @240, 0),
   MLejos (@205, 0, @240, 1, @275,0),
   Lejos(@240, 0, @275, 1, @310,0 ),
   MuLejos( @275, 0, @310, 1, @350,1)
];
```

Definición de variables de entrada de desplazamiento

```
invar XDesU "" :-1.000000 (  ) 1.000000 [
   Izq (@-1, 1, @-0.75, 1,@-0.5,0),
   PIzqu (@-0.75, 0, @-0.25, 1, @0, 0),
   SinDesX (@-0.25, 0, @0, 1, @0.25, 0),
   PDere (@0, 0, @0.25, 1, @0.75, 0),
   Der (@0.5, 0, @0.75, 1, @1,1)
];
```

```
invar YDesU "" :-1.000000 (  ) 1.000000 [
   Abajo (@-1, 1, @-0.75, 1,@-0.5,0),
   PAba (@-0.75, 0, @-0.25, 1, @0, 0),
   SinDesY (@-0.25, 0, @0, 1, @0.25, 0),
   PArri (@0, 0, @0.25, 1, @0.75, 0),
   Arr (@0.5, 0, @0.75, 1, @1,1)
];
invar ZDesU "" :-1.000000 (  ) 1.000000 [
   Atras (@-1, 1, @-0.75, 1,@-0.5,0),
   PAtras (@-0.75, 0, @-0.25, 1, @0, 0),
   SinDesZ (@-0.25, 0, @0, 1, @0.25, 0),
   PAdel (@0, 0, @0.25, 1, @0.75, 0),
   Adel (@0.5, 0, @0.75, 1, @1,1)
];
```

Conjuntos difusos de salida

```
invar XNDesU "" :-1.000000 (  ) 1.000000 [
   Izq (@-1, 1, @-0.75, 1,@-0.5,0),
   PIzqu (@-0.75, 0, @-0.25, 1, @0, 0),
   SinDesX (@-0.25, 0, @0, 1, @0.25, 0),
   PDere (@0, 0, @0.25, 1, @0.75, 0),
   Der (@0.5, 0, @0.75, 1, @1,1)
];
invar ZNDesU "" :-1.000000 (  ) 1.000000 [
   Atras (@-1, 1, @-0.75, 1,@-0.5,0),
   PAtras (@-0.75, 0, @-0.25, 1, @0, 0),
   SinDesZ (@-0.25, 0, @0, 1, @0.25, 0),
   PAdel (@0, 0, @0.25, 1, @0.75, 0),
   Adel (@0.5, 0, @0.75, 1, @1,1)
];
invar YNDesU "" :-1.000000 (  ) 1.000000 [
   Abajo (@-1, 1, @-0.75, 1,@-0.5,0),
   PAba (@-0.75, 0, @-0.25, 1, @0, 0),
   SinDesY (@-0.25, 0, @0, 1, @0.25, 0),
   PArri (@0, 0, @0.25, 1, @0.75, 0),
   Arr (@0.5, 0, @0.75, 1, @1,1)
];
```

Reglas difusas

```
$Base de reglas para el movimiento en X
if XPosU is DIzq and XDesU is Izq then XNDes is PIzqu;
if XPosU is DIzq and XDesU is PIzqu then XNDes is PIzqu;
if XPosU is DIzq and XDesU is SinDesX then XNDes is Izq;
.
.
.
```

Para el eje Z se tienen las siguientes reglas:

```
if ZPosU is MuLejos then ZNDes is Ade;
if ZPosU is Lejos then ZNDes is Ade;
if ZPosU is MLejos and ZDesU is Atras then ZNDes is PAde;
...
```

Implementación de protocolos de interacción FIPA compatibles para agentes con la herramienta CAP-AgentTool

Ernesto Germán Soto [1], Rigoberto Velásquez Elenes [2], Omar Olivas Zazueta [2]

[1] Instituto Mexicano del Petróleo, Eje Central Lázaro Cárdenas 152, Col. San Bartolo Atepehuacan, 07730, México, D.F.
egerman@imp.mx

[2]Universidad Autónoma de Sinaloa, Angel Flores y Riva Palacio S/N, Colonia Centro, Culiacán Sinaloa.
verxx@info.uasnet.mx, oaoz@hotmail.com

Resumen. Los protocolos de interacción constituyen un mecanismo que ayuda a la implementación de conversaciones y patrones de mensajes comunicativos entre agentes. En este trabajo se presenta la forma en que estos protocolos pueden ser incorporados en el tiempo de diseño de los agentes que se crean con la herramienta CAP-AgentTool y se muestra el papel que juega el programador en las diferentes etapas de comunicación que conforman a un protocolo. Además, se explica la implementación de tres protocolos especificados por FIPA: Query, Request y Contract-Net. Por último, se incluye una pequeña aplicación con varios agentes que intercambian mensajes por medio del protocolo Contract-Net. La infraestructura utilizada para el despliegue de estos agentes es la plataforma de agentes CAP.

1 Introducción

Las conversaciones que se llevan a cabo entre los agentes caen en determinados patrones regularmente. En tales casos, se espera que ocurra cierta secuencia de mensajes y en algún momento de la conversación, se espera que sigan otros mensajes. Estos patrones típicos de intercambio de mensajes son llamados protocolos de interacción (PI). En principio, un agente es visto como aquella entidad de software que permite la colaboración entre aplicaciones por medio de un lenguaje de agentes común y que utiliza los servicios de una plataforma compatible con FIPA.

Un diseñador de aplicaciones de agentes puede decidir hacer a los agentes lo suficientemente conscientes del significado de los mensajes de tal forma que el proceso de planeación cause que los PI emerjan espontáneamente de las decisiones del agente. Esto sin embargo, implica una gran carga de capacidades y complejidad en la implementación del agente. Una forma alternativa y muy pragmática es pre-definir los PI para que agentes implementados en forma sencilla puedan involucrarse en conversaciones que tengan sentido con otros agentes, simplemente apegándose al protocolo conocido. Tal es el caso de las aplicaciones de comercio electrónico, mercados electrónicos y de subastas que están siendo desarrolladas con agentes y mecanismos de negociación específicos para su interacción [1], [2].

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 107–117, 2003. © Centro de Investigación en Computación, IPN, México.

El apegarse al uso de protocolos trae ventajas a los desarrolladores de agentes, pues pueden enfocarse a implementar sólo aquellos actos comunicativos involucrados los protocolos en lugar de considerar todo el espectro de actos. La forma de especificar el protocolo en una conversación es poniendo su nombre en el parámetro *protocol* del mensaje. FIPA es una organización internacional que trabaja en la especificación y estandarización de agentes y ha establecido un conjunto de protocolos estándar través de sus especificaciones en la biblioteca de protocolos de interacción [3]. ventaja que esto trae es que se logra cierto grado de interoperabilidad entre aplicaciones distribuidas con agentes heterogéneos.

Por otra parte, los agentes componentes que se generan con la herramienta CAP-AgentTool tienen la posibilidad de incorporar una serie de PI que les permiten comunicarse de manera predefinida con otros agentes siguiendo la secuencia de mensajes establecida en cada uno. Los protocolos que aquí se presentan son: Fipa-Query, Fipa-Request y Fipa-ContractNet y están basados en la especificación que FIPA establece para cada uno. Además de seguir la secuencia de mensajes predefinida en cada protocolo, la implementación que se propone en este trabajo considera el uso del lenguaje de comunicación de agentes FIPA-ACL. La sintaxis y semántica del contenido de mensajes están basadas en la representación de objetos de contenido definidos en lenguaje de contenido FIPA-RDF0. La ventaja de utilizar esta herramienta radica que los programadores pueden implementar los PI mucho más fácil, rápida y consistentemente, sin tener que preocuparse por estudiar y entender las especificaciones respectivas.

Después de explicar el funcionamiento de la implementación, se muestra la forma de utilizar los protocolos en una aplicación de demostración con varios agentes guiendo el protocolo Contract-Net. La finalidad es indicar las partes del protocolo que deben ser implementadas en los agentes y de acuerdo a la aplicación particular que los utilice. El despliegue de esta aplicación se realiza sobre la plataforma de agentes CAP.

2 Infraestructura utilizada

2.1 Component Agent Platform (CAP)

La plataforma CAP está basada en el modelo de referencia de FIPA (*Foundation for Intelligents Physical Agents*). Los servicios básicos y obligatorios tienen la característica de que se han implementado sobre un servidor DCOM a través de los componentes AMS (*Agent Management System*), DF (*Directory Facilitator*) y ACC (*Agent Communication Channel*).

El lenguaje de comunicación entre agentes de CAP es FIPA-ACL debido a que FIPA impone que la comunicación entre agentes debe ser de alto nivel refiriéndose la manipulación e intercambio de conocimiento entre los agentes. En un ambiente distribuido, un agente suele cumplir con sus objetivos a través de la interacción con otros agentes y ejecutando acciones, lográndolo precisamente a través del uso de su lenguaje de comunicación entre agentes [4], [5].

2.2 Agentes

Un agente es el actor fundamental en una plataforma que combina una o más capacidades de servicios en un modelo de ejecución integrado y unificado que puede incluir acceso a software externo, usuarios humanos y facilidades de comunicación. Definido por FIPA, un agente es todo aquel proceso computacional que implementa la funcionalidad autónoma y de comunicación de una aplicación.

En la teoría de agentes colaborativos heterogéneos un agente es todo aquel proceso computacional que se comunica a través de un lenguaje de comunicación de agentes [6].

2.3 Herramienta CAP-AgentTool

CAP-AgentTool es una herramienta que tiene por finalidad principal ayudar a los programadores en el proceso de creación de controles agente que corran sobre la plataforma de agentes CAP. A través de ésta es posible ir configurando paso a paso los aspectos que permiten la creación de nuevas clases de agentes como controles ActiveX [7]. En particular, uno de los aspectos que se puede configurar es la incorporación de algunos protocolos de interacción que ya han sido pre-definidos y cuya secuencia de mensajes es incorporada y controlada al momento de diseñar el agente [8].

Solución

3.1 Lenguaje de comunicación de agentes

El lenguaje de comunicación usado por los agentes creados por CAP-AgentTool es FIPA-ACL, dado que están diseñados para ejecutarse en una plataforma CAP. Este es un lenguaje de comunicación entre agentes que está definido a través de la teoría de actos del habla o *speech acts*. Por consiguiente, FIPA ofrece una semántica para diferentes tipos de actos comunicativos [9] y una forma de construirlos [10]. Todos los agentes creados con CAP-AgentTool tienen un mecanismo interno para enviar y recibir mensajes, por medio de una plataforma CAP, hacia otros agentes. De esta manera, también la implementación de los PI en agentes que aquí se presenta utiliza este lenguaje.

3.2 Selección de protocolos en CAP-AgentTool

En la herramienta para crear controles de agente CAP-AgentTool se permite al programador que especifique los protocolos que el agente va a usar. En [8] se presenta el trabajo que da soporte a esta herramienta para programar agentes. Algunas extensiones fueron realizadas a CAP-AgentTool para permitir que los agentes utilicen los PI, y a continuación se explican. En general, con este trabajo se trata de dotar al agente con cierto grado de flexibilidad con respecto a sus capacidades internas. Al indicarle los protocolos que el agente va a usar, la herramienta está diseñada para incrustar el código fuente necesario para controlar la implementación y ejecución del protocolo.

Con esto se trata de hacer la implementación generalizada de la funcionalidad de protocolos lo cual conlleva a que los detalles de la implementación específicos de aplicación que quiera utilizar los protocolos en los agentes estén abiertos a las necesidades de cada una.

La implementación de los protocolos que aquí se da se concentra en especificar y controlar la secuencia de mensajes que cada protocolo tiene definidos, pero en cada uno de ellos existen aspectos que deben ser dejados abiertos. En este sentido, agentes que los incorporen a su funcionalidad interna están obligados a ofrecer la cionalidad específica de la aplicación que los utilice o del dominio al que corresponde el agente. En la figura 1 se aprecia la interfaz que existe en la herramienta CAP-AgentTool para especificar los protocolos que un nuevo tipo de agente va a incorporar a su funcionalidad.

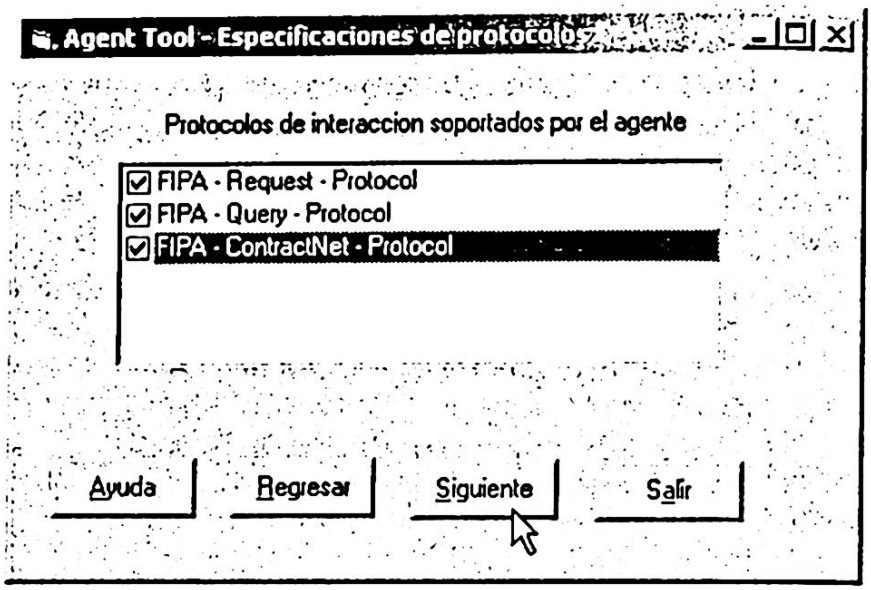


Fig. 1. Selección de protocolos de interacción en CAP-AgentTool

Cuando se procede a la etapa de generación del control de agente en la herramienta, se verifica si el programador seleccionó PI de la lista y si fue así, entonces, incrusta en el código fuente del control de agente las instrucciones necesarias para permitir seguimiento al protocolo. Como parte de este código está la sección en la que se verifican los diferentes mensajes que pueden pertenecer a una conversación y algunos métodos internos que se requieren para tomar las decisiones que son necesarias en gunos puntos de la interacción y que quedan abiertos a la implementación específica de la aplicación. Para cada uno de los protocolos considerados, se crearon plantillas genéricas en archivos de texto, los cuales contienen el código necesario mencionado anteriormente para el funcionamiento de los protocolos. Estos archivos se van a usar dependiendo del protocolo que se añada en tiempo de diseño de cada clase de agente.

3.3 Implementación del protocolo FIPA-Request

El protocolo de interacción *FIPA-Request* [11] permite a un agente que solicite a otro la realización de una acción y el agente receptor realiza la acción o responde que no puede realizarla. La representación de este protocolo se da en la figura 2.

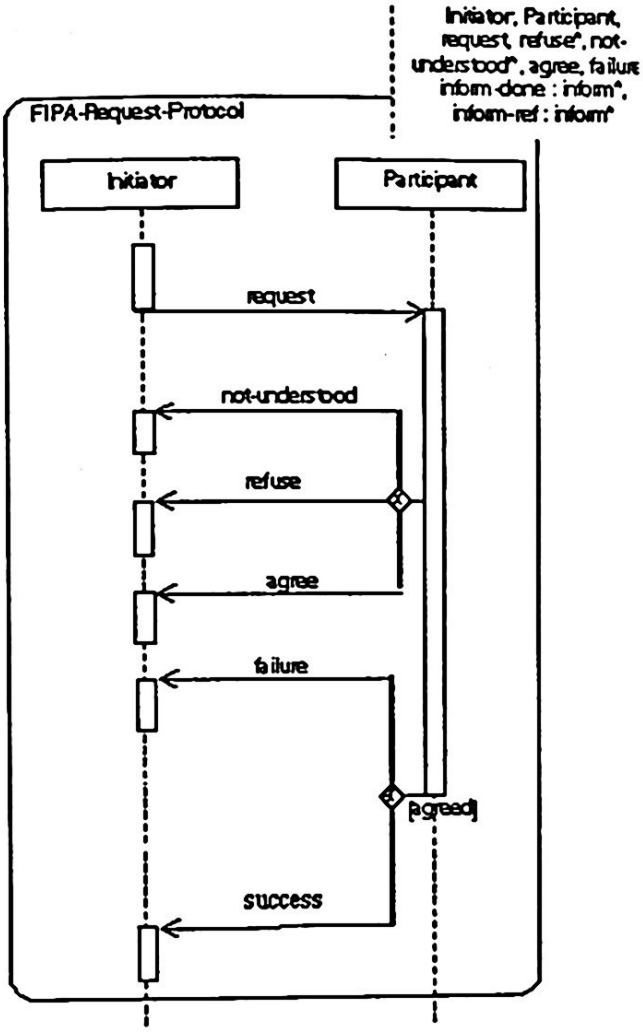


Fig. 2. Diagrama del protocolo Fipa-Request

El agente que incluye este protocolo verifica el contenido del mensaje *request* cada vez que recibe un mensaje de este tipo. Si determina que el mensaje trae consigo en el campo *protocol* la cadena "FIPA-Request-Protocol" entonces el agente entiende que este mensaje debe ser tratado de forma consistente con el protocolo *request*. Para lograr esto, debe utilizar los objetos de contenido que existen de acuerdo con la sintaxis de cada acto comunicativo. En el caso específico de *request*, el contenido del mensaje es una expresión de acción que denota la acción solicitada. Para manipular esta expresión de acción se puede utilizar la clase *Accion* que permite utilizar las funciones de crear y leer los elementos que forman parte de una acción, según se especifica en el lenguaje de contenido FIPA-RDF0 [12], el cual está disponible para estos agentes como parte de las herramientas de programación de CAP-AgentTool. Si la estructura del mensaje o el contenido del mensaje no están bien formados (su sintaxis o semánti-

ca no es correcta), el agente contesta con un mensaje del tipo *not-understood* para indicarle al agente que solicita la acción que no pudo entender el mensaje.

Por el contrario, si el agente puede procesar el mensaje y la acción expresada en el contenido, entonces procede a buscar dicha acción en la base de acciones que cada agente posee. Si la acción solicitada es encontrada, entonces el agente debe decidir si ejecuta la acción o se rehúsa. Para decidir cuándo se realiza una acción o no, se creó una función llamada "Decidir_Accion" recibiendo como parámetro una variable de tipo vector para indicar las condiciones de la realización de la acción expresadas como hechos o proposiciones del dominio, de acuerdo al lenguaje de contenido. En el listado de código siguiente se muestra la implementación del método "ProtocoloRequest" que los agentes poseen para controlar el seguimiento del protocolo de interacción *FIPA-Request*.

```
Public Sub ProtocoloRequest()
  'preguntar si utiliza el protocolo FIPA-Request-Protocol
  Dim accionr As requestContent
  Set accionr = New requestContent
  If parser.protocol = "FIPA-Request-Protocol" Then
  'ver si entiende el mensaje
      If (accionr.deparse(parser.content) = True) Then
      'buscar la accion
            If (buscarAccion(accionr.accion) = True) Then
                  'llamar al procedimiento que decide
                  '       si se realiza la accion o no
                Dim Requisitos As Collection
                Set Requisitos = New Collection
                  'agregar los parametros necesarios para tomar
                  '     la decisión de realizar o no una acción
                If (Decidir_Accion(Requisitos) = True) Then
                     'si la encontró invocar la accion
                      InvocarAccion accionr.accion
                Else
                   ' se reusa hacer la accion y manda un refuse
                  ExecRefuse 1, "FIPA-Request-Protocol"
                End If
            Else
               ExecNotUnderstood 1, "FIPA Request-Protocol"
            End If
       Else
            'si no es que no entiende el mensaje llamar
            '   al notunderstood
         ExecNotUnderstood 2, "FIPA-Request-Protocol"
       End If
     End If
End Sub
```

Si la función "Decidir_Accion" se evalúa a verdadero, entonces el agente invoca la acción por medio de su función interna "InvocarAccion". Dentro de esta función el agente determina si la acción invocada fue realizada con éxito. En este caso, envía un mensaje *success* al agente que solicitó la acción, de lo contrario envía un mensaje *failure*.

3.4 Implementación del protocolo FIPA-Query

En el protocolo de interacción *FIPA-Query* [13], el agente receptor es solicitado a realizar un tipo de acto *inform*. El acto *inform* es una consulta, y hay dos tipos de actos *Query: query-if y query-ref* y cualquiera de estos actos pueden ser usados para iniciar este protocolo. En cualquier caso, un *inform* es usado en respuesta, aunque el contenido del *inform* dado en respuesta a un *query-ref* debería ser una expresión de referencia.

En este protocolo, al agente destinatario se le pide que efectúe una acción de carácter informativo. El remitente puede contestar con un *not-understood*, un *failure*, un *refuse* o un *inform*.

Según las especificaciones de FIPA el Protocolo de interacción *FIPA-Query* funciona de la siguiente manera:

1.- El agente iniciador envía un *query-if* o un *query-ref* al agente participante que lleva la proposición a comprobar.
2.- El agente participante debe contestar con *not-understood*, *refuse*, *failure* o *inform*.
3.- Responde con *not-understood* cuando no entiende el mensaje.
4.- Responde con *refuse* cuando se rehúsa a comprobar el hecho o proposición.
5.- Responde con *failure* cuando el agente no tiene información sobre el hecho o proposición.
6.- Responde con *inform* cuando encontró la proposición sin ningún problema.

Cuando se crea un agente y se selecciona que va a utilizar el protocolo *FIPA-Query* entonces se le añade una porción de código fuente para hacer la invocación del método "ProtocoloQueryIf" que es el que se encarga de verificar el seguimiento del protocolo. En el funcionamiento de este método se realizan las siguientes cosas:

Cada vez que le llega un mensaje de tipo *query* al agente, éste invoca el método que le da seguimiento al protocolo. Internamente en el método, se verifica que el campo *protocol* del mensaje recibido lleve la cadena "FIPA-Query-Protocol". Si es así, se procede a obtener la proposición que denota el hecho que se está solicitando para comprobar en la base de conocimientos del agente; esto se hace través de un objeto de contenido *QueryIfContent*, para manipular el contenido de los actos *query*.

Si el mensaje es correcto, entonces se manda buscar el hecho solicitado con los mecanismos internos que cada agente posee para consultar su base de conocimientos. Si encontró la proposición solicitada, debe decidir si se genera la respuesta. Para esto, el programador del agente tiene que implementar la función de decisión llamada "Decidir_Proposicion" que está abierta para que se establezcan las condiciones bajo las cuales el agente va a realizar la acción de informar o se va a rehusar a hacerlo. Si ésta función devuelve verdadero significa que decide generar el *inform* y genera el mensaje de respuesta; si devuelve falso significa que decide no generar el *inform* por lo tanto invoca al método "ExecRefuse" indicando que se rehúsa a generar el mensaje de respuesta.

Si al buscar el hecho solicitado el agente no lo encuentra, entonces genera un mensaje ACL de tipo *failure* que indica que el agente no tiene información sobre la proposición dada y por lo tanto se considera un fallo.

En caso de que la estructura del mensaje recibido originalmente este mal construido se genera un mensaje de tipo *not-understood* y termina la conversación.

3.5 Implementación del protocolo FIPA-Contract-Net

En el protocolo de interacción *FIPA-ContractNet* [14], un agente toma el rol de ciador de la interacción en busca de que una tarea sea realizada por uno o más de grupo de otros agentes y trata optimizar una función que caracteriza a la acción. Esta característica es comúnmente expresada como el precio, en alguna forma específica del dominio, pero podría ser también el tiempo más corto en el que se completa la ción o una aceptable distribución de tareas. Los agentes que utilizan este protocolo pueden ocupar dos roles: iniciador y participante según sea el caso.

El iniciador del protocolo solicita propuestas de otros agentes por medio del envío de un mensaje *call for proposals* o *cfp* en el que especifica la acción y algunas condiciones para su ejecución. Los agentes que reciben el *cfp* son vistos como participantes potenciales y son capaces de generar propuestas para realizar la acción a través mensajes *propose*. Las propuestas del participante incluyen sus precondiciones para realización de la acción, las cuales pueden ser el precio o el tiempo en el que la tarea será realizada, etc. Alternativamente, el participante se puede rehusar a generar y enviar propuesta. Una vez que el tiempo límite especificado ha pasado, el iniciador evalúa las propuestas recibidas y selecciona a los agentes para que realicen la acción. Uno, varios o ningún agente puede ser seleccionado. A los agentes de las propuestas seleccionadas se les envía un mensaje *accept-proposal* y los demás recibirán un *reject-proposal.* Las propuestas están ligadas al participante, así que, una vez que el iniciador acepta la propuesta, el participante adquiere un compromiso de realizar la acción. Una vez que el participante ha realizado la acción, este envía un mensaje terminación al iniciador.

Para la implementación de este protocolo se creó un procedimiento llamado "*ProtocoloContractNet*". Cuando el agente recibe un mensaje *cfp*, verifica si la estructura del contenido del mensaje que recibió es correcta. Si es así, significa que el mensaje esta bien construido (sintáctica y semánticamente) y por lo tanto llama a la función "*buscarAccion*", la cual busca el nombre de la acción a realizar, si no encontró el acto envía un mensaje *NotUnderstood,* indicando que el acto no es soportado. De lo contrario, va a invocar la función "*Decidir_Accion*", con sus parámetros correspondientes, donde se implementan las especificaciones necesarias para que el programador establezca las reglas para decidir si se realiza o no la acción, dependiendo del conjunto de hechos que fueron recibidos como parámetro. Si la función se evalúa a verdadero, significa que la acción puede ser realizada, por lo tanto invoca la función "*elabora_propuesta*". En este método, el programador debe elaborar y enviar un mensaje *propose* con las condiciones para formar la propuesta respectiva (puede aplicar algoritmos diversos para representar la función de decisión). De lo contrario si la acción no puede ser realizada, entonces envía un mensaje *Refuse*, indicando su desinterés por realizar la acción solicitada.

Cuando el agente iniciador de la conversación recibe un mensaje *propose,* este verifica si el contenido de la propuesta recibida esta bien estructurado. Si está correcto si no ha transcurrido el tiempo de espera de las propuestas entonces almacena las propuestas. En caso de que se reciban *propose* fuera de tiempo llama al procedimiento "*ExecRejectProposal*" en el cual envía un mensaje *reject-proposal* indicando el motivo por el cual fue rechazada, entre otras cosas.

Si un agente participante de la conversación recibe un mensaje *accept-proposal,* significa que el agente iniciador aceptó su propuesta. Esto implica automáticamente que deberá realizar la acción solicitada por medio del mecanismo interno de los agen-

tes (su método InvocarAccion). El agente participante puede generar y enviar un mensaje *failure* cuando la acción no se realizó correctamente o se interrumpió por alguna circunstancia. De otra manera, un mensaje *success* se envía para informar que la acción se realizó de manera satisfactoria junto con los resultados obtenidos. Cuando se recibe un *failure*, *not-understood*, *refuse*, *reject-proposal* o *success* se termina la negociación entre los agentes implicados (iniciador -participante).

Existen unos procedimientos adicionales para lograr el funcionamiento del Protocolo *FIPA-ContractNet*, como es el caso del método "*IniciarProceso*" para empezar la secuencia de mensajes. Se invoca desde una aplicación, indicándole al agente iniciador, los nombres de los agentes a los que enviará el mensaje *cfp*, el tiempo que deberá estar esperando propuestas, la acción solicitada y las condiciones que se establecen para realización de dicha acción.

Una vez transcurrido el tiempo establecido para la interacción, se procede a analizar propuestas en donde se invoca a la función "*AceptaPropuesta*" para que el programador analice las condiciones para decidir si una propuesta es aceptada. Si está de acuerdo en la propuesta que recibió entonces invoca al método "*ExecAcceptProposal*", indicando que ha aceptado la propuesta, de lo contrario invoca al método "*ExecRejectProposal*", diciendo así que no desea que realice la acción, y debe indicar el motivo por el cual la propuesta no fue aceptada.

Para el caso de la implementación que aquí se presenta (para los tres protocolos), se sustituyó el acto comunicativo *inform-done* especificado en algunos protocolos por el acto *success* que, aunque no es parte de la biblioteca de actos comunicativos de FIPA, se propone su uso para expresar la necesidad de informar los resultados que se obtienen al ejecutar acciones por los agentes (esto es más consistente con el lenguaje de contenido FIPA-RDF0 utilizado para expresar los diferentes objetos de contenido de los mensajes).

4 Aplicación de demostración

Con la finalidad de mostrar la manera en que una aplicación puede utilizar agentes y los PI que aquí se han descrito se presenta esta sección. La idea es que se presenta una aplicación programada en el lenguaje Visual Basic 6.0 que contiene un grupo de siete agentes, cuya clase fue creada con la herramienta CAP-AgentTool, y que incorpora en su funcionalidad el uso del protocolo de interacción FIPA -ContractNet. Los nombres de los agentes utilizados en la aplicación son: AgenteCN1, AgenteCN2, así hasta AgenteCN7. El agente AgenteCN1 es el encargado de iniciar la conversación planteada por el protocolo ContractNet, por medio de una solicitud de propuestas al resto de los agentes.

Lo más importante que hay que resaltar es que los protocolos llegan a ciertos puntos o estados en los que se requiere tomar decisiones que son específicas del dominio de las aplicaciones que los utilizan. En este sentido, la implementación de estas partes específicas referentes al protocolo en cuestión requiere varias consideraciones.

Es importante tomar en cuenta que se considera que los agentes pueden decidir la realización de las acciones necesarias para lograr sus objetivos. Esto lleva a que se necesite implementar mecanismos que permitan al programador del protocolo tomar las decisiones apropiadas. En el caso particular de ContractNet, se han incluido varios métodos que ayudan a tomar tales decisiones. Por ejemplo, cuando a un agente parti-

cipante se le pide que envíe una propuesta para realizar una acción, este agente debe decidir si puede o quiere realizar la acción solicitada por medio de la implementación de su método respectivo, tal y como se explicó en la sección 3.5. De igual manera, específico de la aplicación determinar la forma de generar una propuesta, analizarlas, y aceptarlas si es que el agente así lo ha determinado. En la implementación de este ejemplo, estas funciones de decisión se han resuelto por medio de la asignación costos numéricos asignados arbitrariamente a las acciones solicitadas y los límites aceptación de propuestas para cada ejecución. En la figura 3 se muestra la interfaz la aplicación y el despliegue de los diferentes mensajes intercambiados entre los agentes.

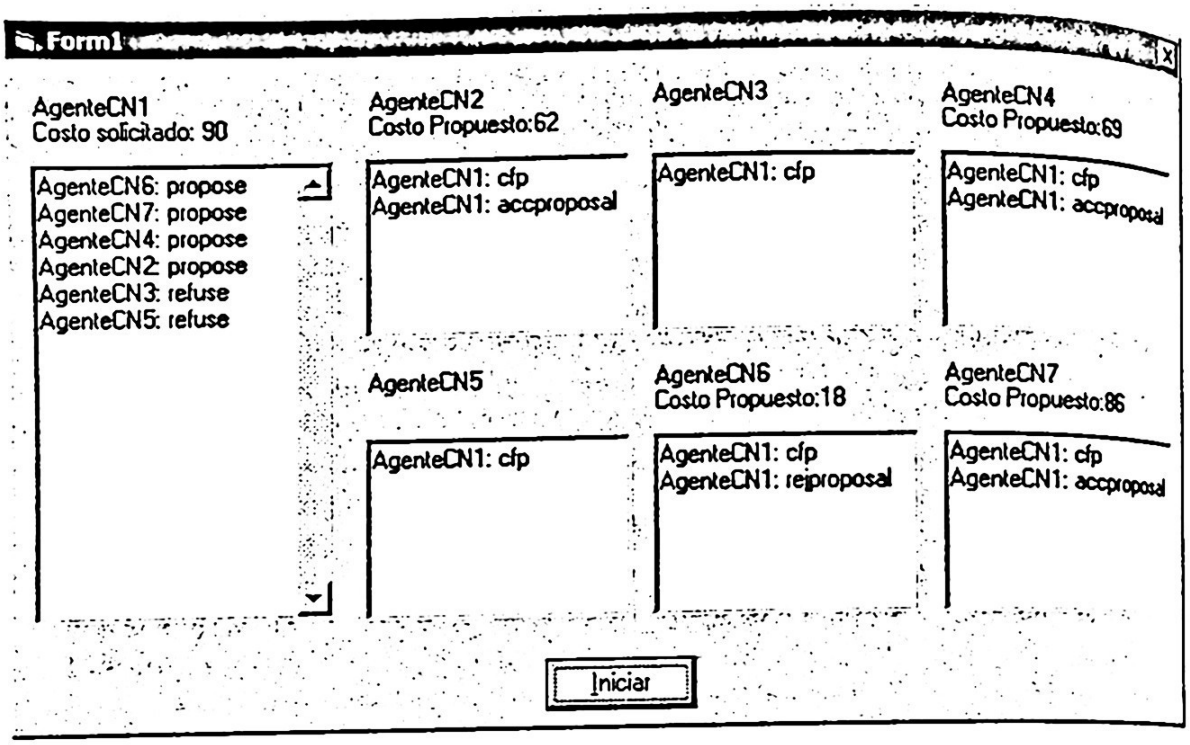


Fig. 3. Prueba del protocolo *FIPA-Contract-Net*

5 Conclusiones

Se estudió con detalle las características de la especificación de FIPA con respecto la implementación de tres protocolos de interacción que son parte de su biblioteca protocolos de interacción. Estos son los protocolos *FIPA-request*, *FIPA-query* y *FIPA-ContractNet*.

Se revisó la definición general de cada uno de los protocolos y a partir de esto logró desarrollar un mecanismo de implementación que se enfoca en las secuencias generales establecidas para dar seguimiento a los protocolos en sus distintas fases de interacción, dejando los detalles específicos de las aplicaciones abiertos en cada caso, por medio de funciones generales. La implementación presentada fue desarrollada con la herramienta de creación de agentes CAP-AgentTool, en donde se añadió la parte necesaria para permitir a los programadores que especifiquen si sus agentes van a lizar los protocolos implementados. Por medio de esta parte se pueden generar

agentes con la funcionalidad básica necesaria para implementar cada uno de los protocolos. Se explica el proceso que deben seguir los programadores de agentes para lograr una implementación específica de los protocolos seleccionados durante la creación del agente. Los agentes creados con esta herramienta tienen la característica de que son desplegados a través de una plataforma de agentes CAP.

Por último, se explican los detalles de la implementación de los protocolos a través de agentes de ejemplo y aplicaciones de prueba de la funcionalidad. En cada paso de la implementación se hace énfasis en el trabajo que debe realizar el programador y los detalles que debe tomar en cuenta para lograr un funcionamiento adecuado de los mismos.

Referencias

1. R. H. Gutman, A.G. Moukas, y P. Maes. Agent-mediated electronic comerse: A survey. The Knowledge Engineering Review, 13(2):147-159, 1998.
2. J. Yamamoto y K. Sycara. A stable and efficient buyer coalition formation scheme for e - marketplaces. In proceedings of the 5th International Conference on Autonomous Agents, 2001.
3. FIPA Interaction Protocol Library Specification http://www.fipa.org/specs/fipa00025/XC00025E.html
4. Contreras, M., 2001. Desarrollo de una infraestructura MultiAgentes para organizaciones virtuales. CIC-IPN, México D.F. 2001.
5. Sheremetov, L. & Contreras M., 2001. Component Agent Platform. CEEMAS-2001
6. Genesereth, M. R. & Ketchpel, S. P., 1994. Software Agents, *Communications of the ACM* 37 (7), 48-53.
7. Appleman Dan, Desarrollo de componentes COM/ActiveX con Visual Basic 6, Editorial Prentice Hall, Madrid España, año 2000
8. Germán, E. 2002. Desarrollo de una herramienta para la creación de agentes sobre la plataforma de agentes componentes. CIC-IPN, México D.F. 2002
9. *FIPA Communicative Act Library Specification*, http://www.fipa.org/fipa00037/
10. *FIPA ACL Message Structure Specification*, http://www.fipa.org/specs/fipa00061/
11. FIPA Request Interaction Protocol Specification http://fipa.org/specs/fipa00026/
12. *FIPA RDF Content Language Specification*. http://www.fipa.org/specs/fipa00011/
13. FIPA Query Interaction Protocol Specification. http://fipa.org/specs/fipa00027/
14. FIPA Contract Net Interaction Protocol Specification. http://fipa.org/specs/fipa00029/

Sistema multi-agente para control de manufactura flexible basado en la teoría de restricciones

Jorge Martínez Muñoz[1], J. Juan Guerra Vásquez[1]

[1] Centro de Investigación en Computación del Instituto Politécnico Nacional
{george, jguerrav}@correo.cic.ipn.mx

Resumen. Los Sistemas Multi-Agente, representan parte de los esfuerzos recientes en la solución de problemas que requieren planificación emergente y distribuida entre los participantes. En este trabajo se presenta el diseño e implementación de un sistema orientado al control flexible de manufactura y basado en la Teoría de Restricciones. Dicha Teoría plantea una nueva forma de concebir a la empresa y en especial, a la parte de manufactura. Se presentan además, resultados obtenidos de la experimentación en una plataforma abierta para desarrollo de agentes (JADE).

1 Introducción

Las tendencias actuales en el área de manufactura han dejado la producción masiva para dar lugar a la producción personalizada. Las empresas deben adaptarse a la demanda especializada de cada consumidor. Pare resolver esta situación, se requiere de sistemas con capacidad de adaptación dinámica y flexible. En el presente trabajo se describe la implementación de la Teoría de Restricciones (TOC) en un Sistema Multi-Agente (SMA) orientado al control flexible de manufactura. Con ello se pretende experimentar con la fusión de dos ramas del conocimiento, Administración de la Producción e Inteligencia Artificial, en la solución de un problema: calendarización dinámica de partes/componentes en la manufactura de un producto.

2 Definición del problema

El problema se toma de un caso de estudio general propuesto por Jennings en [1]. Se trata de un sistema de manufactura que comprende un conjunto de máquinas (M1-M9) para transformación o ensamble de componentes. El grupo de máquinas es alimentado por tres almacenes de materia prima. Con respecto al problema original, se aplicaron dos modificaciones: cuando una máquina falla, el componente que se encontraba procesando es desechado en lugar de mandarlo a reproceso y se agregó explícitamente un nuevo tipo de almacén para productos terminados. La configuración del problema se muestra en la Figura 1.

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 118–125, 2003. © Centro de Investigación en Computación, IPN, México.

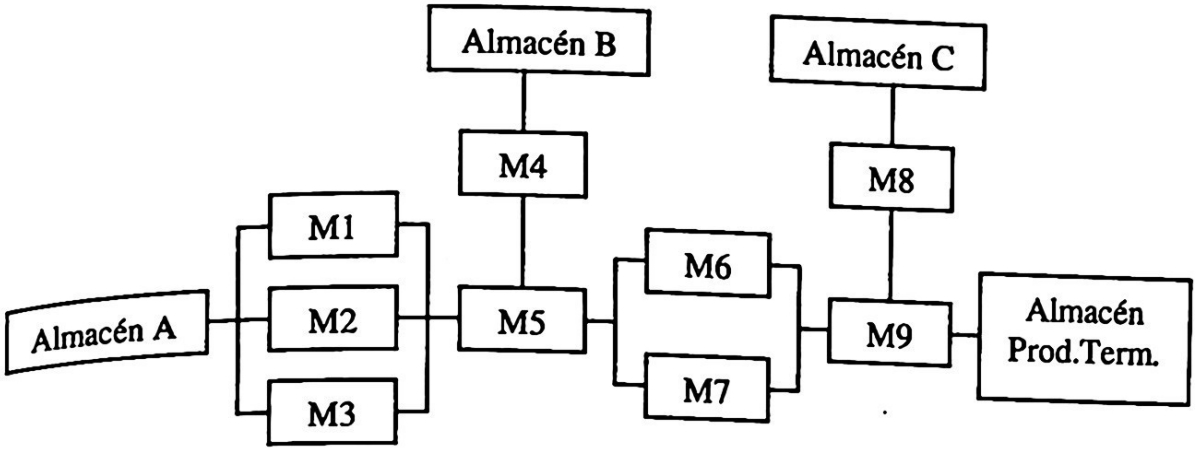


Fig. 1. Sistema Flexible de Manufactura.

3 Teoría de restricciones

Esta teoría se propuso como una alternativa cuya intención principal es la contemplación de una empresa, no como componentes aislados que colaboran entre sí, sino como un todo que debe ser optimizado. Una de las aplicaciones en un entorno similar al problema que aquí se presenta está en [3].

Para resolver el caso de estudio, se aplicaron únicamente aquellos conceptos de la Teoría de Restricciones con mayor relación al tipo de problema. En un sistema de manufactura se presentan fenómenos que afectan la operación de la misma. Por una parte se tienen los eventos dependientes y son aquellos procesos que por la arquitectura de la línea, no pueden operar a menos que reciban la salida de procesos anteriores a ellos; por otro lado, están las fluctuaciones estadísticas, como ejemplo se pueden mencionar las máquinas M3, M4 y M8 del caso de estudio. Ellas tres tienen capacidad de realizar dos tipos distintos de operaciones, de antemano es difícil pronosticar cuál operación ejecutaran, lo cual afecta el promedio de tiempo en línea que acumulen cada una de ellas. Cualquier retraso en un proceso anterior se acumula en los procesos subsecuentes.

Es importante aprender a distinguir entre dos tipos de recursos: Cuellos de Botella No Cuellos de Botella. Ambos están definidos con respecto a su capacidad, el primero de ellos es aquel cuya capacidad es igual o menor a la demanda que hay de él, mientras que el segundo tiene capacidad sobrada según la demanda. Los Cuellos de Botella indican cuándo alimentar al sistema con más inventario.

Típicamente, las empresas cometen el error de balancear la capacidad con la demanda. Lo importante es balancear el flujo de producto por la línea de producción con la demanda del mercado. El hecho de mantener a las máquinas trabajando todo el tiempo no es señal de eficiencia, finalmente, lo único que se logra es acumulación de inventario en los almacenes de productos en proceso. De aquí se obtiene la consideración más importante en el modelo desarrollado: detectar por medio de un análisis cuál es la máquina Cuello de Botella, que a su vez trabajará como el calendarizador de nuevos productos. No importa cuan rápido trabajen las otras

máquinas, eventualmente todas se verán obligadas a sincronizar sus tiempos operación con los números que maneja el Cuello de Botella.

4 Diseño del Sistema Multi-Agente

Cada elemento presente en el modelo de producción estudiado, está representado un agente. Para cada almacén de materia prima (A, B, C), existe un agente que corresponde y que alimenta de manera inicial al sistema. Por su parte, cada una materias primas y las partes resultantes de su respectivo ensamblaje, representadas por agentes Componentes; la meta de estos últimos es llegar hasta almacén de producto terminado.

Adicionalmente existe un agente en el sistema por cada máquina presente modelo, en este caso se identificaron nueve agentes máquina (M1-M9). La función estos agentes es la de prestar sus servicios a los agentes parte/componente que soliciten. Dichos servicios consisten de las operaciones que son capaces de realizar; en un modelo de producción algunas máquinas pueden tener operaciones en común. La máquina más lenta de acuerdo a los parámetros que definen sus operaciones, cuello de botella.

El almacén de producto terminado también tiene su representante. En el sistem este agente cumple la función de llevar el inventario de productos exitosamen procesados, así como de informar la demanda de los mismos de acuerdo a criteri preestablecidos (p.ej. prioridades de manufactura).

El proceso de producción inicia con la liberación de una materia prim (parte/componente) por parte de algún almacén. Esta es una tarea periódica en funci a la velocidad de operación del cuello de botella. Al ingresar la materia prima sistema, solicita al almacén de producto terminado, le informe cuál producto falta. A continuación, por medio de un servicio externo, se le proporciona secuencia de operaciones que integran el proceso tecnológico aplicable componente.

Cuando en alguna parte del proceso se requiere de un ensamble, el agente parte/componente solicita al almacén correspondiente la creación de una pareja componente. Con ello, al nacer la pareja, ya tiene conocimiento del producto formará e inmediatamente solicita información de las operaciones que requiere formar dicho producto. Ambos agentes componentes negocian con las máquinas respectivas para sincronizar su arribo simultáneo al punto de ensamble. Cuando pareja de agentes es ensamblada, muere y dar lugar alnacimiento de un nuevo agente compuesto con las características de ambos. Si un agente componente que integra pareja es desechado durante el procesamiento de una máquina, éste notifica la falla otro agente perteneciente a su pareja y muere. El sobreviviente solicita a su vez almacén respectivo, la creación de una refacción con las mismas características elemento fallido.

5 Implementación

El sistema se implementó en el entorno para desarrollo de agentes JADE [4]; dicha plataforma está basada en el lenguaje de programación Java. Se codificaron cuatro tipos de agentes: Almacén de Materia Prima (Warehouse), Almacén de Producto Terminado (FPWarehouse), Máquina (Machine) y Parte/Componente (Component). Adicionalmente, se desarrollaron dos agentes de utilería. El primero es una máquina de inferencias (WrapperProlog) que informa a los componentes el proceso tecnológico por el cual deben pasar. Se trata de un *wrapper* al intérprete SWI-Prolog. El segundo consiste de un servicio para presentación de resultados, análisis estadístico y gráfico en tiempo real (WrapperGraph). Éste se implementó por medio de un agente *wrapper* a la hoja electrónica de cálculo SCalc de OpenOffice [5].

Para comunicarse entre sí, los agentes usan protocolos definidos por los estándares FIPA[6], específicamente: FIPA-Request, FIPA-Query y FIPA-ContractNet; éste último es de uso intensivo en la negociación entre componentes que desean ser procesados y máquinas que ofrecen el tipo de operación requerida.

La implementación se enfocó fuertemente en el aspecto de que las interacciones entre agentes se basan en la solicitud y prestación de servicios, es decir, cada agente tiene la capacidad de solicitar y prestar servicios a los demás agentes del sistema.

La figura 2 muestra información adicional sobre el tipo de hardware y sistemas operativos que dieron soporte a la implementación y experimentación.

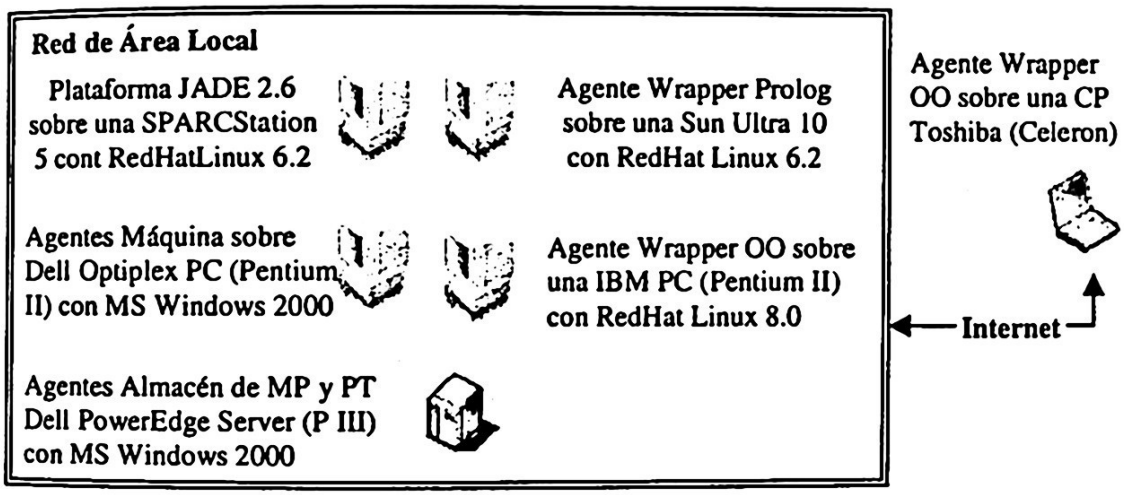


Fig.2. Diagrama de despliegue para la implementación y experimentos realizados.

El agente que inicia la simulación y se encarga de controlar la calendarización de la producción es el agente máquina Cuello de Botella. Este agente solicita al almacén correspondiente, la liberación de un agente parte/componente que desata el proceso para la generación de un producto: esta actividad es llevada a cabo periódicamente en función a la velocidad de operación del cuello de botella

Una vez que el agente *Component* (A,B,C) ha sido liberado por un agente *Warehouse*, procede a definir cuál es el producto final del que deberá formar parte. Esta decisión se toma considerando criterios tales como una producción equilibrada o

bien productos ordenados por prioridad. Para lo anterior, el agente Component solicita dicha información al agente almacén de productos terminados (*FPWarehouse*), que es responsable de suministrar esa información (artículo que necesita ser producido). Una vez que el agente *Component* tiene la información del producto a formar, necesita conocer la operación que requiere para poder transformándose en un producto terminado. Esto se logra solicitando el servicio inferencia al agente *wrapper* respectivo. La respuesta que obtiene es una lista de operaciones que deben ser aplicadas sobre el agente parte/componente. continuación se presentan una serie de negociaciones entre componentes y máquinas bajo el protocolo Fipa-Contract-Net; cada componente elige a la máquina cuya propuesta represente el menor tiempo de espera para ingresar a procesamiento.

Otra operación que el agente componente realiza, es preguntar al agente *wrapper* de software de inferencia, con cuál tipo de agente parte/componente debe ser ensamblado para generar un componente compuesto. Con esta información se solicita al almacén correspondiente la liberación del componente requerido y se estima el tiempo que tardarán los integrantes de la pareja en llegar al punto de ensamblaje. Cuando uno de los dos agentes pueda llegar antes que su pareja al punto de ensamblaje, procede a renegociar su ingreso a procesamiento con la(s) máquina(s) intermedia(s) a fin de que ambos arriben al mismo tiempo a dicho punto.

Después de que dos agentes parte/componente han sido ensamblados, mueren, lo que permite crear un nuevo tipo de agente parte/componente con las características de los anteriores (ej. mueren A y B para formar un componente AB). El nuevo agente conserva como objetivo formar el mismo producto que sus integrantes originales; al igual que ellos, requiere de informarse cuáles son las operaciones que deben ser ejecutadas sobre él y en consecuencia, debe negociar con las máquinas respectivas.

Las máquinas pueden fallar, cuando eso sucede el agente parte/componente que está siendo procesado en el momento de la falla, se desecha y muere. En el sistema cuando un agente parte/componente es notificado de la falla en su procesamiento, notifica la situación a su agente pareja. Éste último solicita la creación de un nuevo compañero con las características del anterior y procede a esperarlo en el búffer de salida de la máquina mas cercana al punto de ensamblaje. Aquí se presentan dos situaciones, el primero es que la creación del nuevo agente pareja (refacción) no repercuta en el rendimiento del sistema y pueda crearse y procesarse sin problemas hasta ensamblarse con su respectiva pareja. El segundo caso es que la creación de dicho agente de refacción sí repercuta en la planificación de las operaciones de la máquina cuello de botella, lo cual implicará que el agente parte/componente que esta en algún buffer espere por más tiempo ser ensamblado. Éste segundo caso se presenta generalmente cuando el agente parte/componente desechado se encontraba próximo a formar un producto terminado.

6 Experimentos y resultados

Los experimentos se enfocaron sobre dos objetivos principales: estudiar el comportamiento del sistema bajo el modelo de administración de producción propuesto y, evaluar el desempeño de la implementación sobre la plataforma seleccionada (JADE).

Con respecto al primer punto, los experimentos comprendieron los agentes para nueve máquinas (M1-M9), tres almacenes de materia prima (A, B, C) y un almacén de producto terminado (FPWH). Se consideraron un total de nueve operaciones en el proceso de producción (O1-O9). Las máquinas M1 y M2 realizaron las operaciones O1 y O2 respectivamente, la máquina M3 las mismas operaciones que M1 y M2; a M4 le correspondieron las operaciones O3 y O4. Las máquinas M6 y M7 desempeñaron la misma operación (O6), mientras que M8 tuvo la capacidad de desempeñar las operaciones O7 y O8. Las máquinas que llevaron a cabo las operaciones de ensamblaje O5 y O9 fueron M5 y M9 respectivamente.

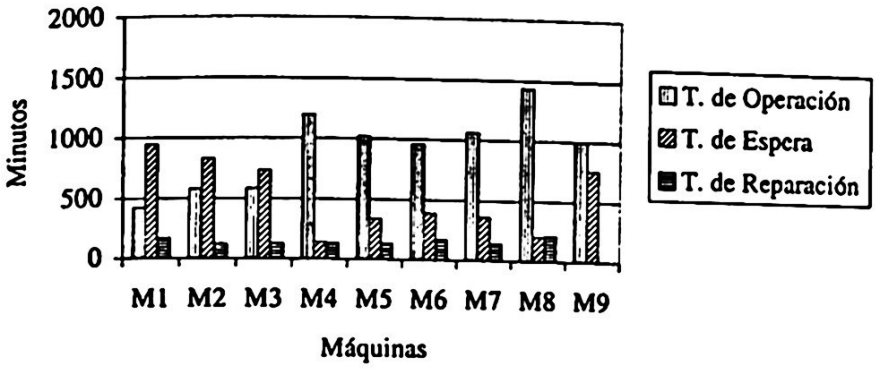


Fig. 3. Operación de las máquinas.

La figura 3 muestra los resultados correspondientes a los tiempos de operación de las máquinas. M1, M2, M3 presentan un tiempo de espera mayor al tiempo de operación debido a que tienen operaciones comunes y el trabajo de las tres en conjunto satisfacen los requerimientos del tiempo simulado. Algo semejante se presenta con las máquinas M6 y M7. M8 presenta un considerable tiempo de espera, lo cual es inadmisible según la Teoría de Restricciones. M8 debería ser la máquina que nunca se detiene, a menos que se deba a una falla. Se detectó que fallas en la ejecución en algunas máquinas, inciden sensiblemente en los tiempos del recurso cuello de botella, sobre todo cuando el componente desechado involucra componentes B o AB. El tiempo de espera en M8 es también consecuencia de los procesos tecnológicos seleccionados para el caso de estudio.

El tiempo de espera de una máquina está constituido por retrasos acumulados en fallas de máquinas que le anteceden y el tiempo que pierde al ajustarse a la velocidad del cuello de botella.

Se observó que la calendarización del proceso de producción en función a la velocidad del cuello de botella, permite contar con un sistema fluido y con un mínimo de inventario en proceso. Éste se da como una consecuencia de fallas en las máquinas, esto es, un componente fallido implica que su compañero exitoso se quedará en espera del arribo de una nueva refacción.

El segundo tipo de experimentos se dividió en tres configuraciones: todos los agentes ejecutándose en una misma computadora (el servidor Dell PowerEdge), los agentes distribuidos en una red local (Figura 2) y el agente wrapper OpenOffice ejecutándose en una computadora portátil desde otro país. Para poder presentar los resultados al usuario, este agente se comunica de manera intensiva con el resto del

sistema. Por ello se decidió realizar dos experimentos para cada configuración, con sin el agente wrapper.

En total se simularon seis, doce y veinticuatro horas de operación que se pueden apreciar en la Figura 4. Cuando el agente wrapper está ausente, los tiempos que tarda la simulación en terminar son muy similares. Sin embargo, si el agente wrapper OpenOffice se encuentra presente, llama la atención que los experimentos en red local tardaron menos tiempo en ejecutarse que los experimentos centralizados. Aquí tiene una consecuencia natural de la distribución de tareas. Finalmente, se tiene que experimento con la computadora portátil en otro país, requirió de mayor tiempo las otras dos modalidades para completarse.

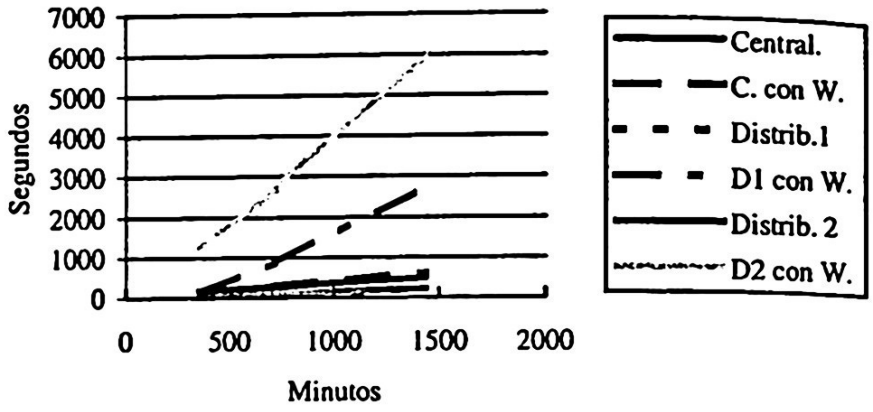


Fig. 4. Relación Tiempo Simulado - Tiempo Real de los experimentos.

7 Conclusiones

En apariencia, la Teoría de Restricciones parece una idea contrapuesta a la filosofía de los Sistemas Multi-Agente. En esta última, se insiste en la minimización dependencias entre las entidades participantes en el sistema con el fin de proteger autonomía. Sin embargo, no hay garantía de que el comportamiento global, resultado de las interacciones, sea el mejor que se pudiera esperar. En el desarrollo del presente trabajo, se mantiene la necesidad de un planteamiento general bajo el cual los Agentes colaboren (o compitan), sin obligarlos a establecer dependencias que impidan funcionamiento autónomo.

El motor de las interacciones entre los agentes es la prestación y solicitud servicios que en un momento dado necesiten, también resalta el uso de distintos protocolos de comunicación en función al tipo de agente con que pretenden interactuar.

Finalmente, dos de las mejoras que se proponen para extender la funcionalidad modelo son: minimizar el tiempo de espera para el cuello de botella de acuerdo estrategias de control y detectar de manera automática a dicho recurso. Actualmente se le especifica de manera explícita al sistema, cuál es la máquina más lenta de todo grupo.

Referencias

Jennings, NR. On agent-based software engineering. Artificial Intelligence, 117: 277-296, (2000).

Goldratt, Eliyahu M. La Meta. Ediciones Castillo. (2002).

Baker A.D., A Survey of Factory Control Algorithms which Can be Implemented in a Multi-Agent Heterarchy: Dispatching, Scheduling, and Pull, J. of Manufacturing Systems, 17(4): 297-320, (1998).

Java Agent DEvelopment Framework <http://jade.cselt.it>

OpenOffice Suite <http://www.openoffice.org>

Foundation for Intelligent Physical Agents <http://www.fipa.org>

Sistema multiagentes para el comando oral de un robot móvil

Oscar Pérez Márquez [1], Luis Villaseñor Pineda [2]

[1] Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 14 Sur
Puebla, México.
osmarp79@hotmail.com

[2] Laboratorio de Tecnologías del Lenguaje
Instituto Nacional de Astrofísica, Óptica y Electrónica
Calle Luis Enrique Erro #1.
Sta. María Tonanzintla, Pue., México
villasen@inaoep.mx

Resumen. El presente trabajo presenta el diseño y los resultados de la implementación de una arquitectura multiagentes para el comando oral de un robot móvil. La idea central fue la búsqueda de una plataforma flexible capaz de integrar todos los elementos posibles para poder brindar una solución satisfactoria al comando oral. Por supuesto esto no es una tarea fácil, la máquina debe escuchar la orden, comprenderla y ejecutarla. A pesar de que estas tareas son extremadamente complejas, en situaciones con contextos cerrados y bien definidos es posible llegar a tener éxito. Este es el caso del comando de un robot móvil. Actualmente la plataforma está completa y funciona correctamente. En esta primera etapa se han integrado diversos procesos hasta obtener una plataforma básica y funcional. Sin embargo, la plataforma brinda el medio para incluir agentes especializados o intercambiar los existentes.

1 Introducción

La capacidad de entender y usar el lenguaje natural es un tema que la ciencia ficción da por hecho. Por ejemplo, en la película "2001 Odisea del Espacio" la computadora HAL 9000 era capaz de interactuar con los seres humanos por medio del lenguaj natural sin ningún problema. Por supuesto, la distancia entre la ficción y la realidad este tema es aún enorme. Sin embargo, la idea de interactuar con una máquina usand el lenguaje, en particular el lenguaje oral, nos brindaría una enorme ventaja. El simpl hecho de no necesitar aprender un medio artificial de comunicación sería suficiente. Por supuesto esto no es una tarea fácil, la máquina debe escuchar la orden, comprenderla y ejecutarla. A pesar de que estas tareas son extremadament complejas, en situaciones con contextos cerrados y bien definidos es posible llegar tener éxito. Este es el caso del comando de un robot móvil.

El presente trabajo presenta el diseño y los resultados de la implementación de arquitectura multiagentes para el comando oral de un robot móvil. La idea central

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 126–134, 2003. © Centro de Investigación en Computación, IPN, México.

la búsqueda de una plataforma flexible capaz de integrar todos los elementos posibles para poder brindar una solución satisfactoria. Actualmente la plataforma está lista y funciona correctamente. En esta primera etapa se han integrado diversos procesos hasta obtener una plataforma básica y funcional. Sin embargo, la plataforma brinda el medio para incluir agentes especializados o intercambiar los existentes. Actualmente la plataforma integra aplicaciones completas como son el reconocedor de voz –en este caso se utiliza un sistema comercial– y el subsistema de control de movimientos de bajo nivel del robot –un sistema propio del robot sobre el cual trabajamos–. Por otro lado, se desarrolló un sistema para el procesamiento sintáctico-semántico de la orden enunciada por el usuario para poder adecuarla al contexto del robot y así conseguir su realización.

La siguiente sección explica brevemente qué es un agente y qué es un sistema multiagentes. Ella termina con la presentación del *toolkit* usado en nuestro sistema. La tercera sección presenta la arquitectura de nuestro sistema y describe los agentes principales y sus funciones específicas. Finalmente, se describen los resultados obtenidos hasta ahora y el trabajo futuro a realizar.

2 ¿Qué es un agente?

Actualmente no hay una definicion aceptada globalmente de un agente de software [1,2]. En vez de esto, los agentes se definen con base en la línea de investigación y/o aplicación donde se desarrollan. Sin embargo, la siguiente definición es suficiente para discutir algunos aspectos de los agentes.

> "Un agente de software es una entidad computacional que desempeña de forma automática tareas delegadas por el usuario" [1]

Esta definición implica una metáfora con un asistente personal, donde el agente realiza tareas en vez del usuario. Ejemplos de estos agentes son: agentes de filtrado de correo electrónico, agentes de recuperación de información y agentes de automatización de tareas de escritorio.

Una definición más formal dice:

> "un agente es todo aquello que puede considerarse que percibe su ambiente mediante sensores y que responde o actúa en tal ambiente por medio de efectores; por otro lado un agente racional deberá emprender todas aquellas acciones que favorezcan obtener el máximo de su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en todo el conocimiento incorporado en tal agente" [3]

Otro aspecto por definir es agente *inteligente*. Según [4] podríamos decir que es aquel agente capaz de realizar acciones autónomas de forma ***flexible***, es decir, cumpliendo con estas tres características:

- Reactividad: se refiere al hecho de que un agente debe poder sensar el estado del ambiente dentro del cual se encuentra inmerso y en función de esto, actuar,

respondiendo de manera adecuada a cambios producidos en el mismo. efectos producidos pueden modificar el estado de su entorno.

- pro-actividad: se podrá tomar la iniciativa en las decisiones, tener comportamiento basado en objetivos.
- Habilidad social: mostrar capacidad para interactuar con otros agentes incluso con humanos)

Lo ideal en un agente inteligente es lograr un compromiso entre comportamien basado en objetivos y conducta reactiva. La razón es que un agente deberá intent cumplir sus objetivos sistemáticamente, pero deberá también darse cuenta de cuán el procedimiento que actualmente sigue ya no es el apropiado, decidiendo en momento qué nueva acción llevar a cabo.

2.1 Arquitecturas concretas para agentes inteligentes

Consideraremos cuatro clases de agentes, dependiendo de cómo toman las decision [4]:

- Basados en lógica, a través de deducciones lógicas.
- Reactivos, que transforman directamente situaciones en acciones.
- Creencia – deseo – intención, que dependen de la manipulación de estructuras datos que representan las creencias, los deseos y las intenciones del agente.
- Arquitecturas en capas, en las que se realiza a través de varias capas de softwa cada una de las cuales representa un nivel de abstracción diferente sobre el entorno.

Frecuentemente los agentes no son desarrollados de forma aislada, sino con partes de un sistema complejo poblado por una diversidad de agentes, los sistem Multiagente (MAS) [9]. Los MAS son sistemas computacionales en los cuales vari agentes semi-autónomos interactúan o trabajan juntos para desempeñar un conjun de tareas, o para satisfacer un conjunto de metas.

2.2 Agentes inteligentes Jack

Los agentes inteligentes están siendo usados para modelar conductas simpl racionales en un amplio rango de aplicaciones distribuidas. En particular, arquitecturas multiagentes basadas en el modelo BDI han sido usadas exitosamente situaciones donde el modelo de razonamiento humano y la conducta en equipo necesarios.

La compañía Agent Oriented Software (AOS) que esta en Melbourne, Australia construido los agentes inteligentes Jack, que es un *toolkit* en Java para desarroll sistemas multiagentes. Jack utiliza agentes inteligentes basados en la conducta razonamiento BDI [6].

JACK es un entorno de desarrollo orientado a agentes. Esta completamente integrado con el lenguaje de programación Java. JACK consiste de los siguientes componentes claves:

- El lenguaje de agente de Jack. JAL(Jack Agent Language) es usado para desarrollar sistemas basados en agentes integrando nuevas clases, interfaces y métodos a Java.
- El compilador de JACK El compilador preprocesa los archivos JAL y los convierte a Java. Este código de Java puede ser entonces compilado en una máquina virtual de Java para correr en el sistema.
- El Kernel de JACK Provee un conjunto de clases que dan a los programas JAL su funcionalidad orientada a agentes.

Un agente de Jack es un componente de software que puede exhibir conducta de razonamiento bajo estimulación reactiva (manejando eventos) y proactiva.
Cada agente de Jack tiene:

- Creencias (*beliefs*) acerca del mundo (su conjunto de datos).
- Eventos a los cuales responder.
- Metas que desea alcanzar.
- Planes que describen como manejar las metas o eventos que podrían surgir

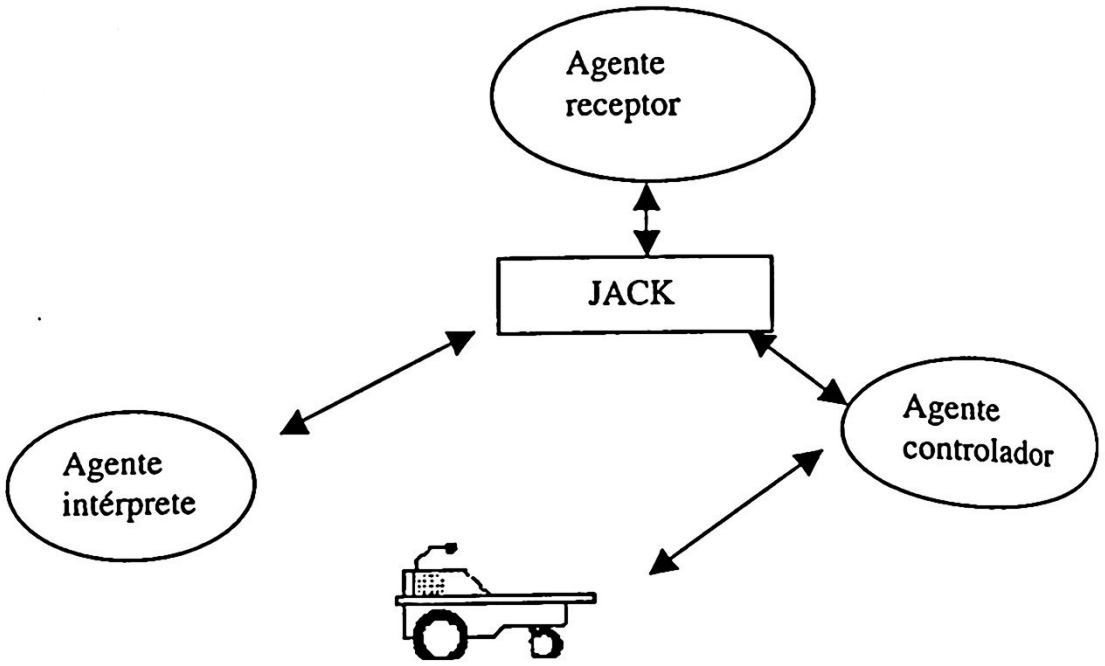
La forma en la que funciona un agente Jack es la siguiente: al momento en que un agente de JACK es instanciado en un sistema espera a que sea dada una meta o experimente un evento al cual deba responder, una vez que se recibe un evento o meta el agente inicia una actividad para manejarlo, esto lo hace buscando que plan es el más adecuado para manejar dicho evento y entonces ejecuta el plan. La ejecución del plan podría involucrar interacción con las relaciones de creencia de un agente u otra estructura de datos de Java. Los planes pueden llevarse a cabo o fallar, bajo ciertas circunstancias, si el plan falla el agente pudiese intentar otro plan.
Dentro de un proceso se puede tener múltiples agentes o también se puede tener agentes en diferentes procesos y sobre diferentes máquinas comunicándose uno con otro.

3 Arquitectura del sistema

La implementación de nuestro sistema se realizó utilizando la plataforma multiagentes Jack, la cual nos permite manejar un ambiente distribuido en donde se pueden tener un grupo de agentes trabajando en procesos que se encuentran en diferentes máquinas. Esto es de particular interés en nuestro caso, ya que existen procesos muy costosos y bastará con repartir adecuadamente la carga de cómputo para mejorar el rendimiento de nuestra aplicación.
El sistema esta dividido en tres procesos que son: el receptor de la orden, el intérprete de órdenes y el controlador de movimientos. Cada uno de estos procesos es llevado a cabo por un agente, todos ellos comunicados a través de la plataforma multiagentes. Cada agente registra sus capacidades de manera que pueda recibir peticiones de los otros. Por supuesto, los agentes pueden ejecutarse dentro de una misma máquina o en máquinas distintas, el máximo número de máquinas que se

pueden utilizar para la ejecución de nuestro sistema debe ser el mismo que el número de agentes utilizados.

El esquema del sistema es el siguiente:



Las siguientes secciones detallan cada uno de estos procesos.

3.1 Receptor de la orden

La función principal que realiza este agente es la captura de la orden pronunciada el usuario y su transferencia al agente intérprete. Para el reconocimiento de voz utilizó el sistema comercial Dragon Naturally Speaking. Este se encarga transformar la señal acústica en texto. Para mejorar el rendimiento del reconocedor un proceso de entrenamiento debe realizarse. El entrenamiento consiste en leer un texto durante aproximadamente 15 minutos.

El agente receptor presenta una interfaz al usuario en la cual el agente recibe peticiones y muestra información acerca de lo que está haciendo el robot. Será través de esta interfaz que el agente recibe el texto *reconocido* por Dragon y una obtenida la orden esta es transmitida al siguiente proceso.

La GUI contiene componentes que nos permiten visualizar información acerca Robot como la velocidad a la que se mueve, las coordenadas de donde se encuentra, estado de la comunicación, etc. La figura 1 muestra esta interfaz. Además nos permi elegir entre varias opciones para el manejo del sistema como son: modo de entrac (voz o teclado), elegir si se va a trabajar con el robot físico o con el simulador y permite establecer la conexión con los demás agentes.

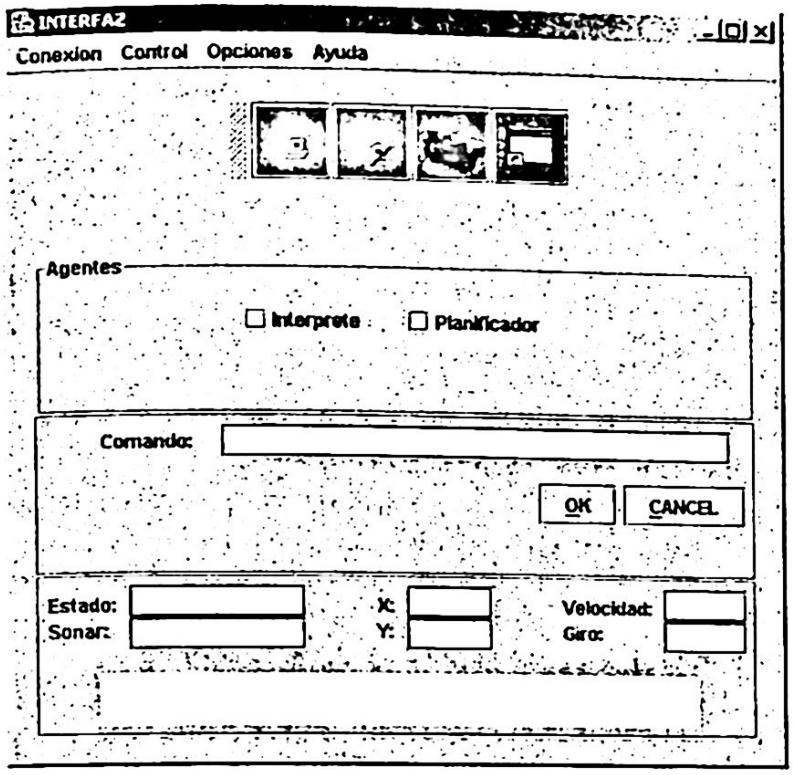


Fig. 1. Interfaz para la recepción de la orden.

El agente Receptor realiza otras tareas, como es la de comunicarse con el agente controlador para conectarse al robot, obtener datos del robot como la velocidad a la que avanza, el estado en el que se encuentra y las coordenadas que tiene robot actualmente para así poder desplegarlas en la interfaz de usuario.

3.2 Intérprete de órdenes

El intérprete se encarga de transformar el texto de la orden, expresado en lenguaje natural, a comandos expresados en el lenguaje de control del robot.El intérprete es un agente en Java que encapsula un motor de Prolog. Así el intérprete está propiamente programado en Prolog y usa el formalismo de DCGs (Definite Clause Grammars) para representar las estructuras gramaticales manejadas [10]. Al terminar la transformación de la orden, si se trata de una orden válida, se entregará el comando correspondiente al controlador de movimientos.

Las DCGs facilitan la programación de gramáticas libres de contexto. Una gramática en notación DCG es directamente ejecutada por Prolog. Además, la notación DCG permite incorporar la semántica del lenguaje en las reglas de producción de tal modo que el significado de un enunciado pueda ser manejado al mismo tiempo que la sintaxis.

Nuestra gramática depende en gran medida de las tareas que pueda hacer el robot, y por el momento se tienen frases que involucran movimientos simples del robot. Las frases que utilizamos en nuestra gramática prácticamente las podemos clasificar en tres tipos:

Órdenes simples: Estas órdenes están compuestas por un verbo en imperativo y algún adverbio. Por ejemplo: camina, avanza, retrocede, alto, camina más rápido, vete un poco más despacio.

Ordenes compuestas: Estas órdenes están formadas por una frase verbal, una frase nominal indicando la distancia o grados de giro y una frase preposicional indicando dirección. Por ejemplo, camina dos metros hacia adelante, camina rápido 4 metros hacia delante, camina hacia atrás, gira a la derecha 90 grados, gira 180 grados hacia derecha, camina hacia la izquierda. Es también posible construir frases donde la frase verbal se da por sobreentendida (elipsis), por ejemplo: hacia adelante un metro, metros hacia delante, a la derecha 90 grados, 90 grados a la izquierda.

Ordenes para manipulación de la cámara: Dado que el robot esta equipado con una cámara también es posible dirigirla a la una dirección específica. Por ejemplo, mira hacia la derecha, mira hacia la izquierda, mira hacia el frente.

3.3 Controlador de movimientos

La principal función del controlador es ejecutar el comando que recibe del intérprete. Además tiene otras funciones como envíar información al agente receptor de la orden sobre el estado del robot, velocidad, las coordenadas en las que se encuentra y distancia a la cual se encuentra el objeto más cercano. Los comandos que por ahora puede ejecutar el robot son simples como por ejemplo de giro, de traslación, modificación de velocidad.

Este agente está a cargo de salvaguardar al robot. Cuando el robot está movimiento este agente toma las lecturas de los sonares y en caso de detectar obstáculo cercano envía al robot la orden de detenerse para evitar así cualquier choque.

El agente controlador es un agente en java que encapsula el software de comando del robot Pioneer el cual está programado en C. Esto se hizo a través de métodos nativos. Un método nativo es un método de java cuya implementación está escrita otro lenguaje de programación. Los métodos nativos se integran a Java utilizando JNI (Java Native Interface). JNI es una interfaz de programación nativa y permite que código de Java que corre dentro de la máquina virtual, interopere con aplicaciones librerías escritas en otro lenguaje de programación, tales como C, C++ ensamblador. [8]

Los métodos nativos que se implementan en este proyecto están escritos en lenguaje de programación C, además de usar las instrucciones que controlan al robot, las cuales están contenidas en la librería PAI y la librería de Saphira. Saphira es entorno de desarrollo de aplicaciones robóticas desarrollado en el centro Inteligencia Artificial del Stanford Research Institute (SRI) bajo la dirección de Kurt Konolige [7]. Este programa nos permite crear nuestras propias funciones lenguaje C y hacer uso de los comandos de control directo que se encuentran definidos en su librería. Saphira incluye un software de simulación del robot, el cual utilizamos para probar nuestro sistema antes de hacerlo con el robot físico.

PAI es una librería de funciones y definiciones en C para controlar un robot móvil Pioneer. Tiene acceso a todas las funciones del robot. PAI hace uso de las funciones

de control de bajo nivel de Saphira. Esto provee la opción de usar algunas funciones de Saphira en un programa.

En el agente controlador se declaran los siguientes métodos nativos:

```
public native int Estado();
public native int ObtenerX();
public native int ObtenerY();
public native int SonarDistancia();
public native int SonarFrente();
public native int ConectarSimulador();
public native int ConectarRobot();
public native void Move(String n);
public native void Turn(String n);
public native void Stop();
public native void Desconectar();
public native void setVelocidad(String n);
public native void InicializaCamara();
public native void Pan(String n);
public native void Tilt(String n);
public native void PanTilt(String p,String t);
public native void Zoom(String n);
public native void Sonido();
```

Estas funciones se compilaron en un DLL para así ser accesibles desde el agente controlador.

4 Resultados obtenidos hasta ahora

Actualmente, el sistema se encuentra funcionando y ha sido probado en condiciones reales. Para ello, diferentes usuarios han sido invitados a usar el sistema y han comandado oralmente el robot. Diferentes configuraciones han sido probadas: los tres agentes en una sola máquina así como un agente por máquina. En todos los casos, los resultados han sido satisfactorios.

Por otro lado, nos encontramos en la fase de estudio de la cobertura de nuestra gramática. Para ello se han diseñado una serie de experimentos sencillos donde se pondrán a varios usuarios a comandar el robot. Cada sesión será grabada y se deberá establecer si aún hay formas no previstas de comandar oralmente al robot.

5 Trabajos futuros

Ya con la plataforma funcionando se tienen tres proyectos a futuro:

- Transmisión de la señal de video de la cámara del robot a la terminal de comando. Se desea poder ver desde el puesto de control la imagen percibida con la cámara del robot. Gracias al comando vocal de la cámara el robot podrá mostrarnos su entorno.
- Durante las sesiones de prueba con el robot, se descubrió que existe una diferencia entre el resultado final deseado y el resultado final obtenido. Esto se

presenta dadas las imperfecciones de la superficie, pequeñas diferencias entre el arranque de los motores, etc. Esto provoca que la posición final esperada no coincida con la real. Un trabajo a futuro es la consideración de mapas de localización para reposicionamiento.

- Comando multimodal del robot, voz y ademanes. En convenio con otros centros de investigación será posible coordinar esfuerzos e intentar la integración de un sistema que sea capaz de fusionar la voz y el reconocimiento de ademanes para comandar el robot.

Agradecimientos

Los autores agradecen al CONACYT por el apoyo financiero para la realización de este trabajo (No. Ref. 31128A). En particular, la beca de tesis de licenciatura recibida por el primer autor. Los autores también agradecen el apoyo del Laboratorio Tecnologías del Lenguaje y el Laboratorio de Robótica Móvil del INAOE.

Referencias

1. Alper Caglayan and Colin Harrison, Agent Sourcebook, Wiley Computer Publishing, 1997
2. Jörg P. Muller ,The Design of Intelligent Agents, A Layered Approach , Springer-Verlag Berling Heidelberg, 1996
3. Stuart Russell, Peter Norving, Inteligencia Artificial, Un enfoque moderno, Ed. Prentice Hall, 1996
4. Weiss, Gerard. Multiagent Systems: A modern Approach to Distributed Artificial Intelligence, Edited by Gerhard Weiss. MIT Press, 1999.
5. Manuel Pérez, Sistema Multiagentes para creación y mantenimiento de hipertexto a partir de documentos en español, Tesis de Maestría en Ciencias Computacionales, INAOE, Puebla México, Febrero 2003.
6. Nick Howden, Ralph Rönnquist, Andrew Hodgson, Andrew Lucas, JACK Intelligent Agents-Summary of an Agent Infrastructure, Agent Oriented Software Pty. Ltd.
7. Miguel Cazorla, Domingo Gallardo, Introducción a Saphira, Universidad de Alicante, Octubre 1998.
8. Beth Stearns, Java Native Interface, Sun Microsystems, 1995-2002.
9. Michael N. Huns and Munindar P. Singh, Readings in Agents, Morgan Kaufmann Inc, 1998.
10. Manual de referencia de SICStus Prolog 3.8.7, Octubre 2001.

Desarrollo de interfaces de voz para aplicaciones web

J. Quiané-Ruiz [1], J. Manjarrez-Sánchez [2]

Centro de Investigación en Computación
Instituto Politécnico Nacional
Unidad Profesional "Adolfo López Mateos" Zacatenco
Col. Linda Vista, C.P. 07738
México D.F.
ufo@correo.cic.ipn.mx [1], jorgerms@cic.ipn.mx [2]

Resumen. La gran popularidad de los teléfonos celulares digitales y de los asistentes personales digitales (PDA), han impactado en la forma de desarrollar aplicaciones sobre Internet, por lo cual se están demandando más procedimientos sobre Internet. Pero el problema de estos dispositivos es su pequeña capacidad de visualización. Las aplicaciones basadas en Web de hoy en día están diseñadas para equipos de escritorio, lo cual las hace difícil de visualizarse en los dispositivos ya mencionados. Este artículo presenta una solución a este problema al desarrollar aplicaciones web basadas en VoiceXML, y así poder acceder a nuestra información desde cualquier lugar utilizando cualquier tipo de teléfono.

1 Introducción

El VoiceXML es un lenguaje para aplicaciones de voz en Internet implementando los servicios de telefonía, y está basado en la norma XML (Extensible Markup Language). Este tipo de aplicaciones son ejecutadas sobre un servidor de voz (intérprete VoiceXML) mediante cualquier tipo de teléfono, y dicho servidor contiene los servicios de telefonía y de Speech, y la comunicación con el al servidor web o de servicios es por medio del protocolo http (la información puede estar almacenada en el mismo servidor de voz o en el servidor web).

En este artículo se explica el desarrollo de aplicaciones VoiceXML utilizando un emulador y enlazando el código de voz con una página jsp.

2 Componentes de Speech

Las aplicaciones VoiceXML interactúan con el usuario por medio de diálogos de voz, lo cual implica servicios de reconocimiento de voz (ASR), de Text-to-Speech (TTS), reproducción de archivos de audio, reconocimiento de tonos de marcado (DTMF), y de telefonía. Y los componentes de Speech son los que le permiten a las

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 135–148, 2003. © Centro de Investigación en Computación, IPN, México.

aplicaciones VoiceXML hacer todo lo anterior. Lo importante de esta tecnología, es de qué separa totalmente a estos componentes de la aplicación, lo cual hace que desarrollador no se preocupe por la programación de los mismos (programación de bajo nivel).

Hay dos tipos de Speech, que son:

Speech input: este se refiere a la entrada de información por parte del usuario, cual puede ser por medio de voz o por tonos de marcado. La información entrante procesada por un reconocedor de voz, para su uso posterior dentro del sistema.

Speech output: en éste se encuentran tres tipos de salida hacia el usuario, el pregrabado, el streaming y el sintetizado. Donde el pregrabado son archivos de audio familiares para los formatos de las computadoras de escritorio, tal como los WAV. El streaming audio es muy similar al pregrabado, a diferencia que en este el audio proporcionado en tiempo real. Y por último el sintetizado, que es por medio de un TTS Engine [1].

3 Intérprete VoiceXML

Éste es el componente más importante en esta tecnología, ya que es el que funciona como servidor de voz, contiene todos los componentes Speech, las aplicaciones VoiceXML y JSP ó ASP. Y es el encargado de ejecutar las aplicaciones VoiceXML, recibir las llamadas telefónicas, de llevar el dialogo con el cliente, de procesar peticiones del cliente, etc.

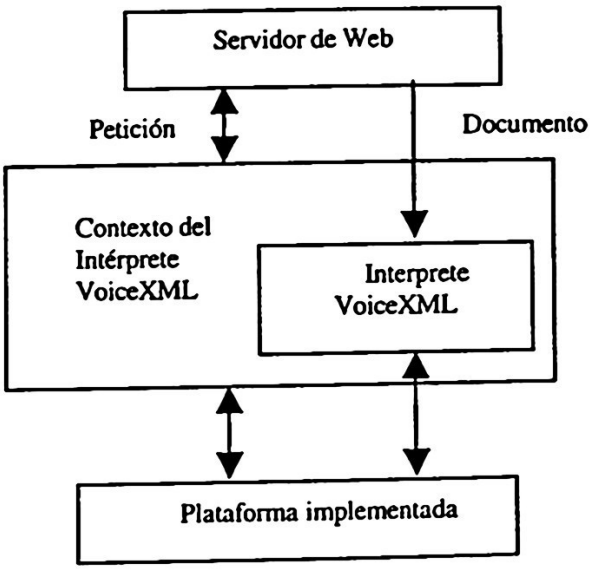


Fig. 1. Esta es la arquitectura presentada por el W3C (World Wide Web Consortium), donde se pueden ver los elementos principales de dicha arquitectura así como la interacción que hay entre ellos.

Donde un servidor del documento (ejemplo: un Web Server) procesa peticiones cliente, a través del intérprete VoiceXML, con el contexto del intérprete de VoiceXML. El servidor elabora los documentos de VoiceXML en la respuesta, que

procesados por el intérprete VoiceXML. El contexto del intérprete de VoiceXML puede supervisar entradas del usuario en paralelo al intérprete VoiceXML. Por ejemplo, un contexto del intérprete de VoiceXML puede esperar a escuchar siempre una frase especial de escape que lleve al usuario a un ayudante personal de alto nivel, y otra puede esperar a escuchar las frases del escape que alteran preferencias del usuario como: volumen ó características Text-to-Speech.

La plataforma implementada es controlada por el contexto del intérprete VoiceXML y por el intérprete VoiceXML. Por ejemplo: en un uso interactivo de la respuesta de la voz, el contexto del intérprete VoiceXML puede ser responsable de detectar una llamada entrante, de adquirir el documento inicial de VoiceXML, y de contestar a la llamada, mientras que el intérprete VoiceXML conduce el diálogo después de la respuesta.

La plataforma implementada genera acontecimientos en respuesta a acciones del usuario (ejemplo: la entrada hablada ó por carácter recibido) y a acontecimientos del sistema (ejemplo: expiración del contador de tiempo). Algunos de estos acontecimientos son actuados sobre y por el intérprete VoiceXML, según lo especificado por el documento de VoiceXML, mientras que otros son actuados sobre y por el contexto del intérprete VoiceXML. [10]

Actualmente, los intérpretes utilizan por lo menos un dispositivo de hardware sofisticado (costoso), lo cual los hace difíciles de adquirir por los desarrolladores individuales, pequeñas y medianas empresas, y por el sector académico. También ofrecen sus servicios sobre la Internet para la realización de pruebas de las aplicaciones VoiceXML, pero dichos servicios son ofrecidos en la unión americana ya que son productos americanos (BeVocal Café, Tellme Studio, VoiceGenie Developer Workshop, por no mencionar todos).

En este caso para poder ejecutar nuestros ejemplos se puede utilizar un emulador para este tipo de aplicaciones, tales como (solo algunos de ellos):

- Cambridge Voice Studio
- WebSphere Voice Server SDK
- Motorola Mobile ADK

4 Introducción a las aplicaciones VoiceXML

En esta sección nos enfocaremos a su desarrollo, viendo la estructura y los el ementos principales de una aplicación VoiceXML, en esta misma sección veremos una pequeña comparación de dos aplicaciones simples, como sería en HTML y como en VoiceXML.

Antes de pasar a ver VoiceXML debemos saber que XML es una norma estándar de marcado para darle una estructura a los documentos de texto en la Internet, en sí podemos decir, que es un metalenguaje para describir otros lenguajes. Una de las características principales de los documentos XML es que deben de ser documentos bien formados sin excepción alguna, para que la información pueda ser interpretada

y transformada a otro lenguaje. XML no sustituye al HTML mas bien es un formato complementario para este tipo de aplicaciones [7].

4.1 Estructura básica de las aplicaciones VoiceXML

Como VoiceXML es un lenguaje que surge a partir de la norma XML, por lo tanto son documentos bien formados, detalle que debemos tener siempre presente al desarrollar este tipo de aplicaciones.

La estructura general de un documento VoiceXML es la siguiente [2]:

```
<? xml version = "1.0" ?>
<vxml version = "2.0">
    [Body]
</vxml>
```

donde la primer etiqueta que va en nuestra aplicación "<? xml versión = "1.0" ?>" es la que nos indica que vamos a trabajar sobre un documento XML y sobre que versión de la misma trabajaremos y esta siempre debe de ir como encabezado de nuestro documento VoiceXML.

La segunda etiqueta "<vxml versión = "2.0">" es la que nos indica que el documento con el cual trabajamos es un documento VoiceXML, y en esta misma etiqueta de igual forma indicamos sobre que versión es la que vamos a trabajar.

Dentro de las etiquetas de "<vxml versión = "2.0">" y "</vxml>" se encontrará todo el cuerpo de nuestro programa.

Como podemos ver la etiqueta "<vxml versión = "2.0">" tiene su respectiva etiqueta de cierre "</vxml>", y podemos ver que la primer etiqueta de nuestra aplicación no tiene su respectiva etiqueta de cierre, y esta será la única etiqueta que no tendrá su etiqueta de cierre en nuestras aplicaciones VoiceXML.

4.2 Desarrollo de una aplicación VoiceXML de ejemplo

En este punto veremos lo principal para el desarrollo de aplicaciones VoiceXML, haciendo la comparación a como sería en HTML la misma aplicación.

Una aplicación en donde en HTML sería como la siguiente:

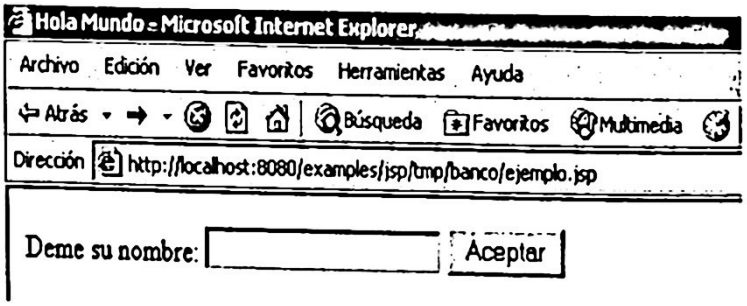


Fig. 2. Esta pantalla es generada al momento de ejecutar nuestra aplicación, y es una aplicación que contiene tres elementos, una etiqueta, un cuadro de texto, y un botón.

En este programa se le pide al usuario su nombre, donde el mismo lo escribe en el cuadro de texto y a terminar de escribirlo se lo indica al programa haciendo un click sobre el botón aceptar.

Y en caso de que usuario haya introducido un nombre nulo se le muestra una pantalla donde se le pide de nuevo su nombre, y al momento de que el usuario introduzca un nombre se le muestra la pantalla siguiente:

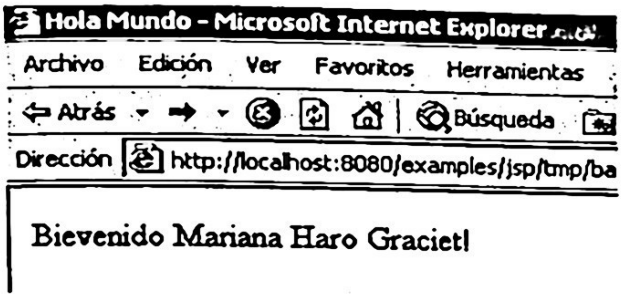


Fig. 3. Mensaje de bienvenida

Esta aplicación presentada, desarrollada en VoiceXML, él código quedaría de la siguiente forma:

```
<!-- Parte 1 -->
<?xml version="1.0"?>
<vxml version="2.0" mode="TTS">
    <form>
    <!-- Parte 2 -->
    <noinput>
          <prompt>
          El nombre no puede ser nulo
          </prompt>
    </noinput>
    <!-- Parte 3 -->
    <field name="nombre" type="spelling">
          <prompt>
          Deme su nombre
          </prompt>
          <!-- Parte 4 -->
          <filled>
                <prompt>
                ¡Bienvenido <value expr="nombre"/>!
                </prompt>
                <exit/>
          </filled>
    </field>
    </form>
</vxml>
```

En la primera parte del código tenemos 3 etiquetas, la primera es la definición de un documento "xml" y definimos la versión de xml a trabajar con el atributo "version"de dicha etiqueta. La segunda etiqueta es la que se utiliza para indicar que tra-

bajaremos con un documento "VoiceXML" (como la etiqueta "html" en HTML), y de igual forma se indica la versión del documento con el atributo "version", y con el atributo "mode" definimos la manera en que el intérprete trabajará con el documento, en nuestro caso será en modo "TTS" que es modo Text-to-Speech. Y la última etiqueta de la primer parte del código es para crear una forma de trabajo en el documento, esta etiqueta es muy parecida a la etiqueta "form" de HTML, y mayormente los documentos VoiceXML son divididos en una o mas formas.

En la segunda parte del código vemos dos nuevas etiquetas, que son "noinput" y "prompt". Donde la primer etiqueta es un evento que se ejecuta únicamente cuando se introduce una cadena vacía como nombre, y la segunda etiqueta es utilizada para indicarle al Parser que el texto que se encuentre entre la etiqueta de abertura y cierra de la misma debe ser reproducido por el TTS Engine, esta etiqueta sería como en HTML escribir en el browser (<h1>¡Hola!</h1>).

La tercera parte contiene una etiqueta nueva, es la "field". Esta nos permite esperar por alguna respuesta por parte del usuario. Dicha respuesta se almacen en una variable que sería la del nombre del campo (field), donde el nombre se le asigna al campo por medio de su atributo "name". Además, podemos definir el tipo del campo, eso se hace por medio el atributo "type". Esta etiqueta viene siendo como la etiqueta "field" de HTML, y esta parte del código sería como el respectivo código en HTML que genera la pantalla mostrada en la Fig. 2.

En la última parte del código se ve la etiqueta "filled", esta etiqueta va ligada a la etiqueta "field" y se ejecuta únicamente cuando se almacena algún valor en la etiqueta "field", viene siendo como el botón aceptar de la aplicación HTML.

Las interfaces de interacción con el usuario de esta aplicación no son visuales como las mostradas por la aplicación HTML, si no, son interfaces auditivas, al ejecutar la aplicación se escucha un dialogo inicial como el que se muestra en la Fig. 4, y dicho diálogo inicial es generado por el código etiquetado como tercera parte. En la figura vemos la transcripción del diálogo, donde *system* es la aplicación y *user* es el usuario.

```
System: Déme su nombre
User: [espera respuesta]
```

Fig. 4. Este diálogo es generado al momento de ejecutar la aplicación VoiceXML.

En caso de que la respuesta sea nula (cadena vacía), se lanza el evento "noinput" de la aplicación, y el diálogo escuchado por dicho evento es el ilustrado en la Fig. 5.

```
System: El nombre no puede ser nulo
System: Déme su nombre
User: [espera respuesta]
```

Fig. 5. Este diálogo se genera al momento de introducir un nombre nulo.

Y finalmente si el usuario realmente introduce su nombre (Mariana Haro Graciet), la etiqueta "filled" entra en acción.

```
System: ¡Bienvenido Mariana Haro Graciet!
```

Fig. 6. Este diálogo se genera al momento de introducir un nombre correcto.

5 Transformación de la aplicación web

En esta sección veremos como se puede adaptar una aplicación web existente en una aplicación VoiceXML (con interfaces de voz).

5.1 Model–View–Controller (MVC)

Toda aplicación contiene código de presentación, de procesamiento de datos y de almacenamiento de datos, y la arquitectura de las aplicaciones difiere según como se distribuye este código.

La arquitectura que la mayoría de las aplicaciones cumple es la MVC, que es una arquitectura de tres capas basadas en componentes, donde la meta es unificar las aplicaciones cliente servidor y las aplicaciones basadas en la Web [8].

Donde el modelo (Model) es el que encapsula el estado de una aplicación, expone la funcionalidad de una aplicación, y responde a las consultas del estado de la aplicación. Estos vienen siendo componentes bean, jsp, servlets, etc.

La vista es la presentación de los datos al usuario, esta es la que suministra al modelo, permite al controlador seleccionar una vista, y este manda los datos del usuario al controlador. Estas son las páginas HTML tradicionales.

Y por último el controlador, define el comportamiento de la aplicación, selecciona una vista para responder cada una de las funcionalidades, y es el que se encarga de actualizar al modelo.

Si la aplicación a modificar está estructurada de esta manera, la parte a sustituir es la correspondiente a la vista, en lugar de código HTML y JSP tendremos diálogos de voz mediante VoiceXML.

5.2 Implementación de interfaces de voz para una aplicación legada

El ejemplo de aplicación legada que utilizaremos, tiene las funcionalidades típicas de un banco, donde se puede realizar lo siguiente (ver Fig. 7):

- Consultas
- Retiros
- Depósitos
- Altas a clientes
- Modificaciones a clientes
- Bajas a clientes

- Creación de cuentas
- Registro de clientes online
- Entre otras...

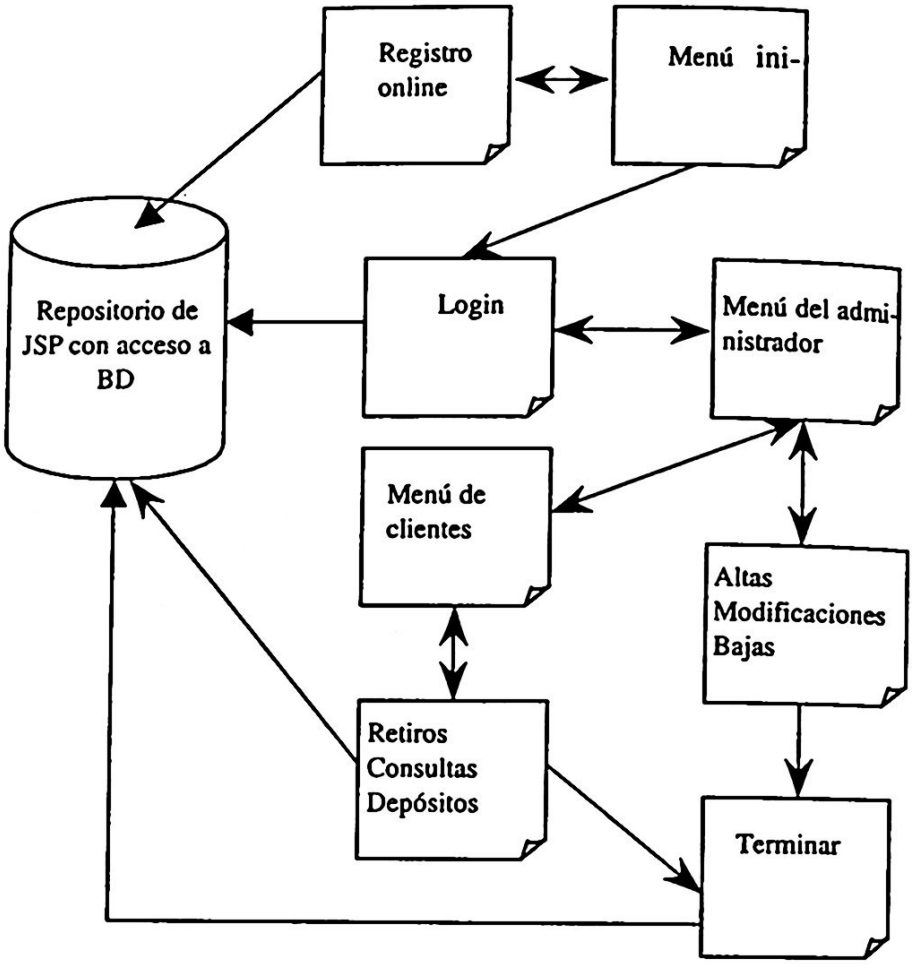


Fig. 7. Bosquejo general de la aplicación básica de bancos

Esta aplicación funciona normalmente desde cualquier navegador, pero ahora requerimos proporcionar acceso mediante voz. Como mencionamos, la parte que cambia es la que correspondía a todas las páginas que veía el usuario, y de hecho, virtualmente siguen existiendo, sólo que ahora además de verse, se escuchan. De modo que la aplicación proporciona acceso dual: el tradicional vía Web y ahora el telefónico.

Para poder ingresar a dicho sistema el usuario debe de proporcionar su user y su password del registro online, el diálogo encargado de pedirle la información es generado por la forma (form) "login" (name="login") del documento principal.vxml, cuyo código es presentado a continuación:

```
<?xml version="1.0"?>
<vxml version="2.0" mode="TTS">
    <!-- Parte 1 -->
    <form>
        <block>
            <prompt>
            Welcome to the CIC bank! Whenever you need
            help for navigate this application, just
            say help.
            </prompt>
        </block>
    </form>
    <!-- Parte 2 -->
    <menu>
        <noinput>
            <prompt>
            I didn't hear you
            </prompt>
        </noinput>
        <nomatch>
            <prompt>
            I didn't understand you
            </prompt>
        </nomatch>
        <help>
            Just follow the intructions in this menu
        </help>
        <prompt>
            <break msecs="5000"/>
            Select one of this options<enumerate/>
        </prompt>
        <choice next="#login">
            customers
        </choice>
        <choice next="#register">
            not customers
        </choice>
        <choice next="#end">
            exit
        </choice>
    </menu>
    <!-- Parte 3 -->
    <form name="login">
        <noinput>
            <prompt>I didn't hear you</prompt>
        </noinput>
        <nomatch>
            <prompt>
                I didn't understand you
            </prompt>
        </nomatch>
        <field name="user" type="spelling">
            <help>Just say your user</help>
            <prompt>Tell me your user</prompt>
        </field>
        <field name="pwd" type="spelling">
```

```
                    <help>Just say your password</help>
                    <prompt>Tell me your password</prompt>
                    <filled>
                          <!-- Parte 4 -->
                          <if cond="user=='administrator'">
                            <if cond="pwd=='system2003'">
                                <goto next="menu3.vxml"/>
                            <else/>
                                <submit
                                next="http://localhost:8080/ex
                                am-
                                ples/jsp/tmp/banco/login.jsp"/
                                >
                            </if>
                          <else/>
                            <submit
                            next="http://localhost:8080/exampl
                            es/jsp/tmp/banco/login.jsp"/>
                          </if>
                    </filled>
              </field>
              <!-- Parte 5 -->
              <block name="register">
                    <goto next="menu2.vxml"/>
              </block>
              <block name="end">
                    <prompt>
                          Thank you for visit us!, good bye!
                    </prompt>
                    <exit/>
              </block>
        </form>
  </vxml>
```

La primera parte del código anterior, es la que da la bienvenida al usuario y aquí podemos ver una etiqueta nueva par nosotros, que es "block", esta se utiliza para establecer un grupo de directivas para ser ejecutadas en orden. En este caso únicamente se encuentra la etiqueta "prompt" que es la encargada de decir el texto de bienvenida siguiente:

System: Welcome to the CIC bank! Whenever you need help for navigate this application, just say help.

Fig. 8. Diálogo de bienvenida de la aplicación

En la segunda, se genera el menú de opciones las que el usuario puede acceder (ver Fig. 9), y en esta misma parte podemos ver varias etiquetas no conocidas hasta este momento, que son: "menu", "nomatch", "help", "break", "enumerate", y "choice". La primer etiqueta es la que nos permitirá crear una lista de opciones para elección del usuario. La segunda etiqueta es muy parecida a la etiqueta "noinput",

diferencia de que esta se ejecuta al momento de que el usuario introduce una cadena que no coincida con el tipo esperado por el sistema. La tercer etiqueta, es de gran utilidad ya que esta nos permitirá que el usuario sepa siempre que hacer dentro del sistema, en el momento que el usuario diga la palabra clave 'help', el texto que se encuentre dentro de estas etiqueta se dictará en ayuda al usuario. La etiqueta "break", es la etiqueta que nos permitirá realizar una pausa en cualquier momento del sistema, el tiempo de la pausa es definido por el atributo "msecs" y el tiempo se encuentra en milisegundos. La etiqueta "enumarate", es la encargada de numerar ascendentemente y automáticamente las opciones del menú, entonces como podemos ver, esta siempre se encontrará dentro de la etiquetas "menu" y antes de las opciones de dicho menú. Y la última etiqueta es la encargada de definir las opciones del menú, y siempre tendrían definido el atributo "next", que es la acción que se ejecutará al seleccionar dicha opción, en el caso de esta aplicación se hace referencia a elementos del mismo documento, por lo tanto deben de ir precedidos por el carácter #, si la referencia es a otro documento no va precedido por dicho carácter.

La tercer parte del código, es la que realiza el primer diálogo con el usuario y es donde se pide el user y el password del usuario (ver Fig. 10), en caso de que la entrada sea nula o incorrecta se vuelven a pedir los datos anteriores (user y password).

```
System: Select one of these options
        For customers, say customers or press one.
        For not customers, say not customers or
        press two.
        For exit, say exit or press three
User: [espera de respuesta]
```

Fig. 9. Menú generado por el documento principal.vxml, donde podemos ver palabras agregadas por el intérprete en el menú, como "For customers, say or press one" donde la única palabra que fue definida en el documento fue "customers".

```
System: Tell me your user
User: ufo
System: Tell me your password
User: xxxxxx
```

Fig. 10. Diálogo donde se pide el user y el password

En caso de que sean ingresados correctamente el user y el password se ejecuta la parte 4 del código. En esta parte podemos identificar cuatro etiquetas nuevas, dos corresponden a la sentencia de condición 'if' y son "if" y "else", las otras dos etiquetas son "goto" y "submit" que corresponden a las etiquetas de referencias a toros elementos o documentos. Las etiquetas de condición ("if" y "else") son del mismo funcionamiento que en cualquier otro lenguaje de programación (if (var==cond) en C) y la condición a evaluar se define por medio del atributo "cond" donde se evalua a

la variabale, esto es lo mismo tanto para la etiqueta "if" y "else", a diferencia que el "else" se ejecuta cuando no entra al "if" correspondiente.

El llamado a la aplicación JSP se hace por medio de la siguiente etiqueta:

"<submit next = " http://localhost:8080/examples/jsp/banco/login.jsp "/>"

La página JSP será la encargada de realizar el login a la base de datos y devolver el resultado por medio de otra página VoiceXML. La etiqueta "goto" es utilizada para hacer referencia aun elemento del mismo documento o a otro documento y el "submit" es para hacer referencias a otros documentos, pero la diferencia entre los dos radica en que el "submit" manda todas las variables de la forma al que corresponde como referencia al documento referenciado y el "goto" no, al referencia lo hacen ambos por medio del atributo "next".

La última parte del código, es la que define los bloques ("block") para las otras dos opciones del menú (Register y Exit), donde podemos ver una nueva etiqueta, que es la etiqueta "exit" y esta es la que cierra por definitivo la aplicación en curso, y el diálogo escuchado como despedida al usuario es el mostrado enseguida.

```
System: Thank you for visit us! good bye!
```

Fig. 11. Este es último diálogo escuchado por el usuario y es el que despide al mismo.

6 Acceso a base de datos con VoiceXML

Todo el acceso a base de datos, no lo hace VoiceXML en sí, si no lo hace por medio de aplicaciones externas, que pueden ser JSP o ASP. En la aplicación básica de bancos utilizamos Java Server Pages para el acceso a bases de datos.

6.1 Java Server Pages (JSP)

Esta es la tecnología presentada por SUN para el desarrollo de aplicaciones dinámicas basadas en Web, y esta nos permite separar la parte dinámica de la parte estática (HTML: Hyper Text Markup Language). Este tipo de aplicaciones son desarrolladas con código 100% Java, gran ventaja para los desarrolladores de aplicaciones Java ya que no tienen que introducirse en un nuevo lenguaje para desarrollar este tipo de aplicaciones. Y una de las grandes ventajas de este tipo de aplicaciones es que reutilizan todas las librerías (API's) de Java, lo cual les da una gran robustez y funcionalidad [5].

Este tipo de aplicaciones es totalmente compatible con las aplicaciones VoiceXML así como también por parte de Microsoft lo son las Active Server Pages (ASP), y por medio de estas es que las aplicaciones VoiceXML hacen la mayoría de sus operaciones, logrando con eso una gran robustez como aplicaciones web. En nuestro caso utilizaremos las JSP para el acceso a las base de datos en nuestras aplicaciones VoiceXML.

6.2 Integración de un documento JSP

VoiceXML hace referencia a las JSP, por la etiqueta ya vistas ("goto" y "submit"), para el caso del login del usuario se empleo la etiqueta "submit", ya que se envían parámetros al documento JSP [4].

El fragmento de código que realiza el acceso a la base de datos para realizar el login es el siguiente (login.jsp):

```
// Declaración de variables de la conexión a la BD
Connection con;
Statement stmt;
ResultSet rs;
// Se guardan los valores de los parámetros enviados
// por el documento VoiceXML
String usuario = request.getParameter("user");
String contraseña = request.getParameter("pwd");
String query   = "";
// Se carga el controlador de JDBC y se hacen los
// objetos de conexión.
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
con = DriverManager.getConnection("jdbc:odbc:Banco", "","");
stmt = con.createStatement();
rs   = stmt.executeQuery("SELECT * FROM logs");
// Se hace la búsqueda del usuario en la base de datos
while(rs.next())
{
     if(usuario.equals(rs.getString("User")))
     if(contraseña.equals(rs.getString("Pwd")))
     {
      // Si se encuentra al usuario se guarda su id y
      // su user
      int tmp = rs.getInt("idCliente");
      Integer id = new Integer(tmp);
      session.setAttribute("idCliente", id.toString());
      session.setAttribute("Nombre", usuario);
      Statement stmt2;
      stmt2 = con.createStatement();
      // De igual forma se cargan sus cuentas (Tipo y
      // numero)
      ResultSet rs2;
      rs2 = stmt2.executeQuery("SELECT * FROM cuentas WHERE
      idCliente = " + tmp );
      while(rs2.next()){
       String tipo = rs2.getString("Tipo");
       int numero = rs2.getInt("Numero");
       Integer num = new Integer(numero);
       session.setAttribute("Tipo", tipo);
       session.setAttribute("Numero", num.toString());
      }
     // Finalmente se carga el documento principal de
     // la aplicación VoiceXML
     %><goto next="http://server/banco/menu.vxml" />
   <%}
}
```

En la última parte del código anterior, podemos observar una etiqueta de VoiceXML ("goto"), esto se puede hacer siempre y cuando se estructure el documento JSP como un documento VoiceXML (ver subtema 4.1), y así podemos mezclar código JSP con código VoiceXML y tener una mayor robustez en nuestras aplicaciones.

7 Conclusiones

En este artículo, se dieron a conocer las principales características de las aplicaciones VoiceXML, así como las principales etiquetas que nos permiten desarrollar documentos VoiceXML, y además de cómo tener acceso a bases de datos. Los ejemplos fueron ejecutados sobre el emulador "Voice Studio de Cambridge", este no tiene la capacidad de probar nuestras aplicaciones por vía telefónica (ningún emulador la tiene), los únicos que la tienen son los servicios ofrecidos sobre Internet ya mencionados, pero todos estos en la unión americana, por lo tanto no son óptimos para los que no pertenecemos a la unión americana. Debido a esto, en el CIC estamos trabajando en la implementación de un intérprete VoiceXML y un ambiente de desarrollo, los cuales no requieren de dispositivos de hardware costosos y superan a la mayoría de los existentes en que además se apegan a la última versión de VoiceXML (2.0), y permiten interactuar con un teléfono desde el emulador, es decir, probar las aplicaciones en un ambiente real.

Referencias

1. Bob Edgar: The VoiceXML handbook, CMP Books, New York, U.S., 2001
2. Sharma Chetan, Kunins Jeff: VoiceXML, WILEY, Indianapolis, Indiana, U.S., 2002
3. Beasley Rick, Farley Mike, O'Reilly John, Squire Leon: Voice Application Development with VoiceXML, SAMS, New York, U.S., 2002
4. Nordin A. Brandon: VoiceXML 2.0 Developer's Guide, McGraw-Hill/Osborne, Berkeley, California, U.S., 2002
5. Tom Myers, Alexander Nakhimovsky: Professional Java XML programming with servlets and JSP, Wrox Press, Birmingham, UK, 1999
6. Simon Brown: Professional JSP, Wrox Press, Birmingham, UK, 2001
7. David Megginson: Structuring XML documents, Prentice Hall PTR, Upper Saddle River, NJ, 1998
8. Erich Gamma: Design Patterns, Adison – Wesley, U.S., Abril 1996
9. Beat Signer, Moira C. Norrie, Peter Geissbuehler and Daniel Heiniger
10. http://www.w3.org/Voice
11. http://www.voicexml.org

Estudio sobre la navegación en la Web usando voz

Areli Torres González [1], Luis Villaseñor Pineda [2]

[1] Facultad de Ciencias Computacionales, BUAP
Av. San Claudio y 14 Sur
Puebla Pue.
are67@hotmail.com.mx

[2] Laboratorio de Tecnologías del Lenguaje
Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Tonantzintla, Puebla.
villasen@inaoep.mx

Resumen. El presente trabajo muestra un estudio empírico de la interacción hombre-máquina al usar la voz para la navegación en la Web. Para ello se grabaron una serie de experimentos en condiciones controladas, y se realizó un análisis de las transcripciones correspondientes. El análisis realizado hasta ahora se ha enfocado principalmente al rendimiento del reconocedor de voz y posteriormente se extenderá a las implicaciones del uso de la voz en la interacción hombre-máquina. Entre los primeros resultados obtenidos tenemos una tasa de reconocimiento del 76.82% y una tasa de ejecución del 72.74%. También se han identificado un primer grupo de puntos críticos en la interacción vocal, tanto a nivel del proceso de reconocimiento de voz como de la utilidad de la voz al navegar en la Web.

Introducción

Actualmente Internet es una de las herramientas más importantes con las que cuenta el hombre moderno. Proporcionando múltiples productos y servicios de información, tales como el Web (Word Wide Web), en donde es posible la navegación hipertextual y permite el acceso a información en forma de texto, sonido, e imágenes animadas [1]. Gracias a los recientes avances en el reconocimiento automático de voz se ha iniciado el uso comercial de esta tecnología en la navegación en la Web [2]. El presente trabajo muestra un estudio empírico de interacción hombre-máquina al usar la voz para la navegación en Internet. El objetivo general de este estudio fue la evaluación de la aplicación de esta técnica de interacción. Para ello se grabaron una serie de experimentos en condiciones controladas, y se realizó un análisis de las transcripciones correspondientes. Las siguientes secciones presentan la plataforma usada para llevar a cabo los experimentos, la descripción de los mismos, y el análisis de sus transcripciones. Por

Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 149–158, 2003. © Centro de Investigación en Computación, IPN, México.

último, se presentan las ventajas y desventajas que podemos concluir de este estudio, tanto a nivel del proceso de reconocimiento de voz como de la utilidad de voz al navegar en la Web.

2 Descripción de la plataforma de navegación

Para llevar a cabo este estudio se preparó una plataforma de navegación. En esta sección se mencionan algunas de sus principales características. Esto con el fin conocer las condiciones en las cuales se realizaron los experimentos.

Tabla 1 . Comandos vocales para la navegación en Web

NAVEGAR EN LA WEB	
Ir atrás	Detener
Ir adelante	Ir a la página de inicio
DESPLAZARSE POR UNA PÁGINA WEB	
Página abajo	Desplazar hacia abajo
Página arriba	Desplazar hacia arriba
Línea abajo	Detener desplazamiento
Ir al comienzo	Más rápido
Ir al final	
MANEJO DE IMÁGENES (HIPERVINCULO)	
Hacer clic en Imagen	Elegir n (n = 1...20)
Imagen	Ir n abajo (n = 1...20)
COMANDOS GLOBALES	
Pulsar Entrar	Marco siguiente
Cambiar a ventana siguiente	Edición
Cambiar a ventana anterior	Seleccionar Todo
A Dormir	Ir abajo
A Trabajar	Ir al inicio
Archivo	Ir arriba
Guardar como	Marco anterior
Cancelar	

2.1 Materiales

Para efectos de este estudio se limitó la navegación a un sólo sitio web. Se buscó sitio con una estructura fija pero cuyo contenido cambiará día con día. En nuestro caso decidimos utilizar el sitio de La Jornada, un periódico de circulación nacional (La Jornada Virtual http://www.jornada.unam.mx/). Este fue elegido ya que presenta una estructura fija pero cuya información cambia diariamente.

También se hizo uso del sistema de reconocimiento de voz *Dragon Naturally Speaking*. Este paquete comercial incluye un conjunto de comandos orales que nos

permiten controlar Internet Explorer y seguir los vínculos con la voz. Además nos ofrece un conjunto de herramientas para la creación de nuevos comandos *orales* [3][4], lo que nos permite crear nuestros propios comandos para así poder automatizar una serie de tareas adicionales sobre el Explorador. En la tabla 1 se enlistan los comandos que un usuario puede usar para navegar en la Web y en la Tabla 2 se muestran los comandos adicionales que se crearon. En particular, se crearon comandos orales para controlar los estados de la ventana del Explorador y poder seleccionar marcos.

La recopilación de los datos para este estudio se hizo usando dos herramientas de apoyo: *Techsmith Camtasia Recorder*, la cual nos permitió capturar el video de las sesiones experimentales; y *Teleport Pro*, una herramienta que nos permitió bajar todo un sitio web y así trabajarlo de manera local, evitando largos tiempos de espera.

Tabla 2. Comandos Adicionales

MANIPULACIÓN DE VENTANAS	
Maximizar	Restaurar
Minimizar	Cerrar
MANEJO DE MARCOS	
Seleccionar Marco	

3 Descripción de los Experimentos

Para lograr el objetivo del estudio de la navegación en la Web mediante la voz se diseñaron una serie de experimentos cuyo propósito principal era obtener un conjunto de resultados que nos permitieran concluir las ventajas y desventajas de este tipo de navegación, así como las características que presenta este tipo de interacción [6].

Los experimentos fueron realizados por 10 personas. A cada una de ellas se le asignó una tarea –la misma tarea para todas– a realizar navegando en el sitio de la Jornada. A continuación se detallan cada uno de los pasos del experimento.

3.1 Etapas de los Experimentos

1. Explicación breve sobre la prueba y sus objetivos. Se explicó brevemente al usuario el objetivo y los alcances de la prueba. Este punto era importante pues se quería dejar claro que él no sería el evaluado sino que con su ayuda se deseaba evaluar al sistema.
2. Entrega del 1er cuestionario. Previamente a la realización de la tarea, a cada usuario le fue entregado un breve cuestionario. Se solicitaron sus datos

personales e información de control de carácter general, para obtener el de cada uno de ellos.

3. Adecuación del sistema de reconocimiento al usuario. En esta etapa, siguieron los pasos para adaptar Dragon a la voz del usuario, esto con el fin obtener la mejor taza de reconocimiento posible. A través del *entrenamiento sistema* se adaptan los modelos acústicos a las características específicas cada hablante; como timbre, tono, volumen y acento. Dicho entrenamier consistió en leer en voz alta y clara parte de un libro o una historia dura aproximadamente cinco minutos.
4. Explicación y uso de comandos vocales por medio de un ejemplo. A usuario se le explicaron cada uno de los comandos que podía utilizar navegar sobre *La Jornada Virtual.* Se les proporcionó una lista con todos comandos y sus especificaciones (ver la Tabla 1 y 2). Después se llevó a un ejemplo sencillo con el fin de mostrarle a cada usuario como usar comandos vocales. El ejemplo consistió en:
   - Buscar en la sección de Espectáculos, dentro del sitio de *La Jorna Virtual,* un reportaje sobre Fernando Sariñana y guardarlo en disco.
   - Ir a *La Jornada de Oriente* y visitar la sección de Puebla, al termi regresar a *La Jornada Virtual.*
5. Elaboración de los ejercicios. Para ello se seleccionó previamente un conju de ejercicios que el usuario realizaría haciendo uso de la mayoría de comandos de navegación. Cada ejercicio incluyó una serie de pregun asociadas, las cuales deberían ser contestadas conforme se llevará a cabo ejercicio. A continuación se enlistan los ejercicios solicitados:
   - Entra a la pagina de La Jornada, elige un encabezado de tu interés y léelo.

     ¿Cómo se titula el encabezado?
     ¿De cuantos párrafos consta dicho encabezado?
   - Busca la sección de cartones y contesta lo siguiente:

     ¿Cuál es el título del cartón que más te agrado?
     ¿Quién es el autor de dicho cartón?
   - Entra a la sección de CineGuía, de la cartelera mostrada:

     ¿Qué película te gustaría ver?
     ¿Qué clasificación es?
     ¿Quiénes son los protagonistas?
   - Elige una sección del periódico al azar. De dicha sección menciona la noticia artículo más impactante y guárdalo en disco.

     Título de la Noticia/Artículo
   - Visita brevemente el sitio de la Jornada de Oriente y regresa a la página Jornada al terminar.

     ¿Qué estados conforman la Jornada de Oriente?

6. Entrega del 2do cuestionario. Al terminar los ejercicios se solicita la opinión general del usuario a cerca de la navegación por medio de voz así como la descripción de los posibles problemas que enfrentó.

Tabla 3. Tiempos de Grabación

USUARIOS	TIEMPO DE GRABACIÓN
usuario 1	16.57 min.
usuario 2	11.10 min.
usuario 3	09.01 min.
usuario 4	04.57 min.
usuario 5	18.48 min.
usuario 6	27.49 min.
usuario 7	17.47 min.
usuario 8	16.59 min.
usuario 9	13.16 min.
usuario 10	12.30 min.
Tiempo promedio de grabación	14.673 min.

4 Transcripción de las grabaciones

El primer paso antes de iniciar propiamente el análisis de los resultados fue la transcripción de las grabaciones. Durante cada experimento se grabó un video para mantener un registro completo de todas los pasos realizados por el usuario para llevar a cabo la tarea asignada. La Tabla 3 muestra los tiempos de grabación por usuario.

LO QUE SE DIJO	LO QUE SE RECONOCIO	LO QUE SE EJECUTO
Ir atrás	Ir atrás	Cargo la página anterior

Fig. 1. Transcripción de un comando

La transcripción consiste en llevar a texto cada una de las intervenciones del usuario junto con los resultados obtenidos. En este caso se recopilaron tres tipos de resultados por cada intervención:

- *Lo que se dijo.* Se transcribió ortográficamente la elocución pronunciada por usuario.
- *Lo que se reconoció.* Se transcribió el texto reportado por Dragon resultado del proceso de reconocimiento de voz. En caso de que el programa no reporta ningún texto reconocido colocamos un guión (-).
- *Lo que se ejecutó.* Se describió la acción que realizó el navegador. En caso de que el programa no llevaba a cabo ninguna acción colocamos un guión (-).

De esta manera, se creó una tabla por cada sesión. Las tablas estaban constituidas por tres columnas representando los tres tipos de resultado para cada intervención.

5 Análisis de las transcripciones

Tomando como referencia las transcripciones de cada usuario se realizó su análisis, mediante el estudio de dos relaciones posibles: *lo que se dijo-lo que se reconoció* *lo que se dijo – lo que se ejecutó.*

Es importante recordar que este estudio está en proceso y en este trabajo presentan los resultados iniciales que se enfocan principalmente en comportamiento del reconocedor de voz. Las dos relaciones anteriores tratan capturar este comportamiento.

5.1 Lo que se dijo-lo que se reconoció

Para esta relación se trabajó con las columnas uno y dos de las transcripciones. Las frases dichas por el usuario se clasificaron en tres tipos:

- *Comandos.* Las elocuciones que ordenaban a Dragon la ejecución de una acción.
- *Vínculos.* Las frases dichas por el usuario que correspondían a vínculos entre páginas.
- *Otros.* Son las frases que no corresponden a ninguno de los casos anteriores. trata de frases que el usuario intuitivamente creó para designar una acción; aquellos vínculos que aparecían sobre la página web como si se tratará de textos pero que en realidad eran imágenes.

Cada una de las intervenciones del usuario fue clasificada usando estos criterios. Posteriormente se obtuvieron el número de intervenciones totales del usuario, como el número de intervenciones reconocidas y las intervenciones reconocidas errores. Un error es cuando el proceso de reconocimiento de voz substituía la orden

pronunciada por el usuario con alguna otra orden válida. Es decir, Dragon reportó algo parecido a lo dicho por el usuario pero que no es lo inicialmente solicitado.

La tabla 4 muestra los resultados de la relación 1. En ella podemos observar que de las 686 frases dichas entre los diez usuarios Dragon reconoció 556 (81.05%) de las cuales sólo 527 (76.82%) fueron órdenes solicitadas expresamente por el usuario. Es decir, 29 frases fueron mal interpretadas.

Tabla 4. Relación 1: Lo que se dijo – lo que se reconoció

RELACIÓN 1:	LO QUE SE DIJO	LO QUE SE RECONOCIO	
USUARIOS	DIJO	SE RECONOCIO	SE RECONOCIO SIN ERRORES
usuario 1	104	65	57
usuario 2	58	53	51
usuario 3	45	34	34
usuario 4	33	30	30
usuario 5	82	60	58
usuario 6	94	89	83
usuario 7	62	47	46
usuario 8	84	65	58
usuario 9	60	53	51
usuario 10	64	60	59
Número de Frases totales:	686	556	527

Tabla 5. Relación 2. Lo que se dijo – lo que se ejecutó

RELACION 2:	LO QUE SE DIJO	LO QUE SE EJECUTO	
USUARIOS	DIJO	SE EJECUTO	SE EJECUTO SIN ERRORES
usuario 1	104	57	53
usuario 2	58	54	51
usuario 3	45	31	31
usuario 4	33	30	30
usuario 5	82	53	53
usuario 6	94	77	76
usuario 7	62	42	42
usuario 8	84	55	54
usuario 9	60	53	52
usuario 10	64	57	57
Número de Frases totales:	686	509	499

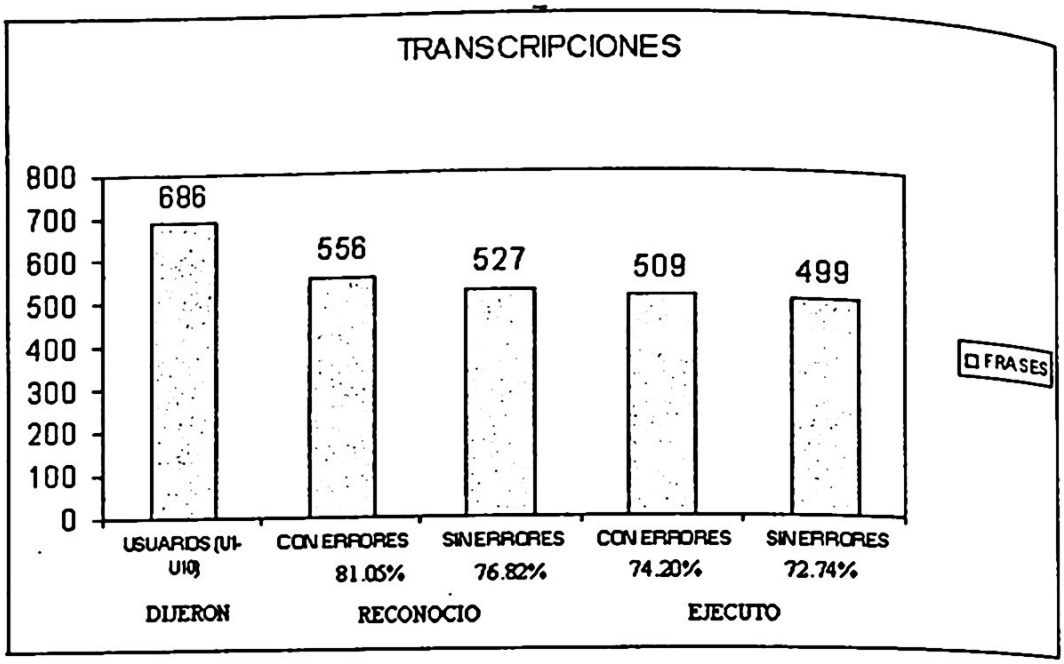
5.2 Lo que se dijo – lo que se ejecutó.

En esta relación se tomaron como referencia la primera y tercera columna de las transcripciones de cada usuario. De igual forma como en la primera relación, las dos columnas se clasificaron en tres tipos: comandos, vínculos y otros.

De igual manera, se calcularon para cada usuario el número de veces que lo ejecutó, no lo ejecutó o bien se ejecutó otra cosa. Los resultados obtenidos se muestran en la tabla 5. De esta tabla notamos que de las frases dichas por los usuarios se ejecutaron en total 74.20% y un 72.74% sin errores.

Así podemos concluir que la tasa de ejecución durante estos experimentos fue un 72.74%. Es interesante notar, que tanto en el proceso de reconocimiento como ejecución existen sustituciones realizadas por el sistema. En resumen los resultados totales se presentan en la tabla 6.

Tabla 6. Resultado del análisis de las transcripciones



5.3 Perfil de los usuarios

De los datos recabados a través de los cuestionarios es importante recalcar siguientes:

- de la muestra de diez usuarios, ocho eran hombres y dos mujeres.
- siete de los usuarios tenían estudios en el área de Ciencias Computacionales, uno en Lingüística, uno en Matemáticas y uno en Diseño Gráfico
- el rango de edad fue de 21 años a 27 años en general, excepto un usuario tenía 54 años.
- el 40% de los usuarios había tenido contacto previo con sistemas reconocimiento automático de voz (p.e. Sistemas de dictado)
- el 70% opinó, después de realizada la prueba, que el uso de la voz facilita navegación.

6 Resultados preliminares

Dos tipos de conclusiones se desprenden de este estudio hasta este momento. primeras son conclusiones con respecto al sistema de reconocimiento de voz (en caso en particular, Dragon Naturally Speaking), pero que pueden aplicarse cualquier sistema de reconocimiento automático de voz. Un segundo grupo

conclusiones presentan observaciones sobre el uso de la voz en este tipo de situación.

Sobre el desempeño del sistema de reconocimiento de voz tenemos los siguientes comentarios:

El medio ambiente de trabajo: El ruido ambiental es un serio problema, se debe buscar disminuirlo y si es que lo hubiera, que éste sea constante.

Los locutores: El uso de la voz, independientemente de la aplicación, depende del tiempo dedicado en el entrenamiento del sistema de reconocimiento. Mientras más se adapte el sistema a nuestra voz la tasa de reconocimiento será mejor. Esto implica cuidados e interés por parte del usuario que en ocasiones serán un obstáculo.

El sistema de cómputo utilizado: Es importante contar con un equipo potente, de acuerdo a los requerimientos de Dragon Naturally Speaking, para contar con un tiempo de respuesta adecuado.

El modelo de lenguaje: El sistema de reconocimiento inclina su reconocimiento por un modelo de lenguaje predeterminado. Esa es la principal explicación de los errores introducidos por el reconocedor al sustituir órdenes válidas correctamente pronunciadas por el usuario. El uso de frases cortas o la presencia de un modelo de lenguaje que prefiere cierto tipo de construcción gramatical sobre otro son las principales razones.

A pesar de no haber terminado el estudio respecto al uso de la voz en la navegación podemos presentar la siguiente lista de comentarios preliminares, contrastándolos en ventajas y desventajas.

Entre las ventajas de este modo de interacción podemos reportar las siguientes, tanto por ser observadas directamente, como remarcadas por los usuarios en los cuestionarios de control.

- *Realizar diferentes actividades*: El usuario tendrá la libertad de usar sus manos para efectuar otra tarea mientras navega, siempre y cuando tenga un conocimiento completo de la estructura de la página o sitio web a consultar.
- *Cómoda y de fácil manejo.*
- *Útil para personas con capacidades diferenciadas.* (p.e. con problemas motores en sus extremidades superiores)

Entre las desventajas deseamos señalar las siguientes:

- *Uso de un sub-lenguaje:* El usuario debe memorizar comandos específicos lo que restringe fuertemente el uso del lenguaje oral, e impone restricciones a la "naturalidad" esperada de este tipo de comunicación.
- *División entre la tarea de navegación y el contenido*: El esquema de navegación estudiado pretende imponer una clara división entre la tarea de navegación y el contenido de las páginas. Al navegar usando la voz esta división desaparece. Varios usuarios presentaron en más de una ocasión, y a

pesar de haber recibido una demostración de la navegación, preferencia por uso de comandos que se relacionaban directamente con el contenido.

7 Conclusiones

El presente estudio es parte de una serie de esfuerzos para poder delimitar claramente cuales son los fenómenos propios de la comunicación oral humana que deben presentarse en la comunicación oral hombre-máquina para que ésta sea realmente útil [5] [7].

Este trabajo es de particular importancia pues estudia la integración de la voz un ambiente diseñado y desarrollado para la presentación gráfica de información. Una situación donde la intervención humana podría hacerse de manera verbal pero la presentación de la respuesta sería fundamentalmente gráfica. Bajo estas condiciones la comunicación oral hombre-máquina no presenta ningún paralelismo con la comunicación oral humana.

Agradecimientos

Los autores agradecen al CONACYT por el apoyo financiero para la realización este estudio (No. Ref 31128A). En particular, la beca de tesis de licenciatura recibida por el primer autor para el desarrollo de este trabajo. Los autores también agradecen el apoyo del Laboratorio de Tecnologías del Lenguaje del INAOE.

Referencias

1. Hahn, Harley. *Internet, Manual de Referencia.* - McGraw-Hill, 1994.
2. Bernal Bermúdez Jesús, Bobadilla Sancho Jesús, Gómez Vilda Pedro: *Reconocimiento Voz y Fonética Acústica.* –Alfaomega Grupo Editor, 2000.
3. Dragon Naturally Speaking: www.dragonsys.com.
4. Dragon Naturally Speaking: *Creating Voice Commands.* –Learnout & Hauspie, September 2000.
5. David N. Chin & Martha E. Crosby. Introduction to the Special Issue on Empirical Evaluation of User Models and User Modeling Systems, *User Modeling and User-Adapted Interaction*, Vol. 12, No. 2-3, Kluwer, 2002
6. Diane J. Litman & Shimei Pan. Designing and Evaluating an Adaptive Spoken Dialogue System. *User Modeling and User-Adapted Interaction*, Vol. 12, No. 2-3, Kluwer, 2002.
7. Lynnette Hirshman & Henry S. Thompson. Overview of Evaluation in Speech Natural Language Processing. *Survey of the State of the Art in Human Language Technology.* Ronald Cole (ed.) National Science Foundation, 1995.

Procesamiento de textos electrónicos para la construcción de un corpus*

Sofía N. Galicia-Haro

Centro de Investigación en Computación, Instituto Politécnico Nacional
Av. Juan de Dios Batiz S/N, 07738 México, D. F.
(+52 55) 5729-6000 Ext. 56544, Fax 5586-2936
sofia@cic.ipn.mx

Resumen. Las colecciones de textos o corpus con etiquetado en diferente proporción, de cero a la totalidad, y en diferentes niveles (morfológico, sintáctico, semántico) son recursos muy útiles que se emplean para diferentes trabajos en investigación teórica y en diversas aplicaciones de lenguaje natural. Sin embargo este recurso no existe para todos los lenguajes, principalmente por la gran cantidad de esfuerzo requerido para compilarlos y para realizar las anotaciones. En este trabajo presentamos los pasos y resultados iniciales para crear un corpus de varios millones de palabras del español en su variante mexicana, de una forma automática.

1 Introducción

La obtención de información del uso del lenguaje a partir de colecciones de textos ha sido una práctica común en la rama de la lingüística aplicada que trata con el dis e-ño y la construcción de bases de datos léxicas, es decir, en la lexicografía. En el procesamiento lingüístico de textos mediante computadora también se ha vuelto una práctica común investigar los fenómenos que principalmente aparecen en materiales sin restricciones, empleando colecciones muy grandes de textos, es decir, corpus.

Los investigadores reconocen el potencial de explotación de corpus grandes para resolver problemas en los diferentes niveles de análisis que en la lingüística general se considera componen el procesamiento lingüístico: léxico, sintáctico y semántico. Sin embargo, los corpus más útiles para el procesamiento lingüístico de textos mediante computadora requieren anotaciones o etiquetas, en cada uno de esos niveles (categoría gramatical, género, número; estructura sintáctica; marcas de significado, concepto, etc.).

Entre los usos más importantes de los corpus etiquetados está el entrenamiento de métodos para resolver distintas tareas, por ejemplo para asignación de categorías gramaticales de palabras desconocidas [17], para vincular frases preposicionales a las palabras correctas [15], para desambiguación del sentido de palabras [14], para la corrección de errores semánticos en los textos [4], etc.

La principal forma de obtención de corpus etiquetados ha sido la compilación y anotación manual. Esta es la razón principal de que no existan corpus etiquetados para

Trabajo realizado con el apoyo parcial de CONACyT, SNI, CGPI y PIFI-IPN, México.
A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 159–170, 2003. © Centro de Investigación en Computación, IPN, México.

la mayoría de los lenguajes, entre ellos el español en su variante mexicana. Si además consideramos la construcción de corpus grandes, del orden de decenas de millones palabras, los esfuerzos manuales requeridos serían enormes. Por lo que actualmente investigan métodos automáticos tanto para la compilación como para la etiquetación de corpus grandes (en [1] se presenta una discusión sobre tiempo y costo en la construcción de corpus anotados muy grandes).

En este trabajo presentamos algunas observaciones y resultados referentes compilación automática de un corpus grande de textos para el español en su variante mexicana, a partir de textos electrónicos obtenidos de Internet. Primero presentamos las características de los corpus y enseguida los resultados hasta ahora obtenidos.

2 Corpus para el procesamiento de lenguaje natural

La compilación de corpus de textos, implica la solución de diversos problemas texto en sí mismo. Las fuentes que formarán el corpus deben analizarse respecto rios criterios: la adquisición de los textos con la información requerida, la cobertu del corpus respecto a los fenómenos lingüísticos requeridos, la confiabilidad en corpus, etc.

Para que un corpus se considere como representativo del lenguaje total debe cont ner en proporción a su uso todas las palabras y construcciones del lenguaje. Así para construir este corpus, idealmente sería deseable lograr una muestra muy grande representativa del lenguaje general. Mientras mayor es el corpus se espera un may número de palabras diferentes, lo que implicaría una mayor cobertura del diccionar del lenguaje, y principalmente supone mayor evidencia de los diversos fenóme lingüísticos (topicalización[1], oraciones subordinadas, pasiva refleja, etc.). Que sea presentativa supone diversos niveles culturales del lenguaje, diversos temas y gén ros. Sin embargo, estas cualidades no implican una a la otra, es más, en algunos se contraponen. Existe otro compromiso a considerar entre calidad y cantidad, hecho de tener un corpus grande no garantiza que posea la calidad esperada.

Idealmente el corpus debería estar balanceado entre esas cualidades. Sin embarg parece no ser posible balancear un corpus apropiadamente, al menos no sin un elev do esfuerzo. Los métodos de muestreo, por ejemplo para seleccionar calidad, son caros. Así que asumimos los problemas obvios de trabajar con datos no balanceadc ya que para construir un corpus balanceado requeriríamos mucho tiempo y un muy elevado.

Asumiendo la imposibilidad de tener un corpus con todas las cualidades deseabl limitamos las cualidades del corpus a las más importantes para los objetivos perseg dos en nuestro caso: la información requerida y el tamaño. El objetivo principal compilación de un corpus de textos mexicanos es el análisis sintáctico de textos restricciones, por lo que los textos de periódicos se consideran una buena fuente información requerida. Uno de los mayores problemas en el análisis sintáctico diante computadora es el enlace de grupos nominales y grupos preposicionales

[1] Tipo de desplazamiento por el cuál un sintagma puede aparecer en la posición oracional tópico. Por ejemplo: *Que llegues a hacerlo francamente no lo creo.*

palabras correctas[2]. Por lo que consideramos de suma importancia, que el corpus contenga uso extenso de frases preposicionales y de complementos.

Respecto a que el corpus tenga la información requerida, por ejemplo, [3] indica el diferente uso de frases preposicionales según el género de los textos. En [16] encontraron que hay diferencias significantes entre las frecuencias de los complementos que una palabra rectora puede tener y la categoría gramatical de ellos, en diferentes corpus. Los autores identificaron dos fuentes distintas para esas diferencias: la influencia del discurso y la influencia semántica. Los cambios en las formas de lenguaje que se usan en diferentes tipos de discurso originan la primera. La influencia semántica se basa en el contexto semántico del discurso. Así que la inclusión de diferentes géneros sería muy adecuada.

Respecto al tamaño del corpus, los corpus actuales están en el rango de un millón de palabras a cientos de millones, dependiendo del tipo, es decir, si son texto plano o con etiquetas de diversas clases. Por ejemplo, en [2] discuten que para obtener buenas aproximaciones de probabilidades, el corpus tiene que ser suficientemente grande para evitar los datos esparcidos y para reflejar el uso natural del lenguaje. Las autoras usaron el Wall Street Journal, un corpus de un millón de palabras. A diferencia de ellas, en otros trabajos, no emplean el corpus completo sino subcorpus con características específicas para su investigación [18], [15], [9], lo que implica un esfuerzo de selección de textos. Por lo que un corpus de varios millones de palabras sería la solución más rápida comparada con la obtención de material específico.

Para nuestro objetivo, análisis sintáctico de textos en español sin restricciones, la mayor importancia del corpus radica en la posibilidad de obtener los argumentos de verbos, adjetivos y sustantivos, para reducir el número de variantes de estructuras. Los argumentos corresponden a los complementos seleccionados semánticamente. En [16], los autores explican que conforme la cantidad de contexto circundante aumenta (yendo de una sola oración a un discurso conectado) decrece la necesidad de expresar manifiestamente todos los argumentos del verbo. Esta situación también se presenta en las oraciones muy largas. Por lo que en nuestro caso, son de gran utilidad aún las frases cortas.

2.1 La Web como fuente de textos electrónicos

Para construir rápidamente un corpus grande, de decenas de millones de palabras se requiere la disponibilidad de un vasto repositario de documentos electrónicos. Una fuente obvia de textos electrónicos es la Web que contiene millones de textos actuales aunque no soluciona por sí misma el problema de representatividad del lenguaje.

La Web comúnmente ofrece centenas de millones de páginas distintas, además de miles de nuevas páginas cada día. El problema de esta fuente es que las páginas son muy diversas en contenido y estilo. Muchas son inapropiadas para la construcción de corpus pero existen muchas otras útiles, de diversos temas y géneros.

Aunque es posible obtener una gran cantidad de textos al azar y seleccionar manualmente los más adecuados, no sería la mejor forma de construir rápidamente un

[2] Por ejemplo, en la frase, *compró contra su voluntad un traje nuevo en la tienda de la esquina*, se presentan tres grupos preposicionales: *contra su voluntad, en la tienda, de la esquina*. Mientras los dos primeros se enlazan al verbo, el tercero se enlaza a *tienda*. El grupo nominal *un traje nuevo* también se enlaza al verbo.

corpus. Para nuestro objetivo es posible obtener automáticamente miles de páginas periódicos mexicanos en sus diversas secciones.

2.2. Etiquetado de corpus

Existen diversos niveles para el etiquetado de los corpus, entre ellos: léxico, sintácti- co, semántico. En cada uno de estos niveles pueden existir a su vez otros niveles etiquetado. En el nivel léxico se considera usualmente el etiquetado del lema y las tegorías gramaticales. Por ejemplo, la palabra *textos* se etiquetaría: *texto* (lema), *tantivo masculino plural* (categoría gramatical). Estas asignaciones pueden tener versos grados de detalle. Por ejemplo, en el Penn Treebank [10] se utilizan etiquetas de categorías gramaticales, y 12 para puntuación y otros símbolos; mientras en el Brown Corpus [7] se distinguen 87 etiquetas simples y se permite a partir ellas la formación de etiquetas compuestas.

En el nivel sintáctico, las etiquetas muestran la estructura de la oración, es decir, describen cómo las palabras de la oración se relacionan y cuál es la función que cada palabra realiza en la oración. Usualmente se muestra agrupando las palabras mediante paréntesis, y adicionalmente etiquetando esos grupos. Existen diferentes grados realización de la estructura jerárquica de la oración, debido a que una estructura com- pleta requiere mayor tiempo de aprendizaje del esquema por parte de los anotadores más tiempo para etiquetar las oraciones. Por ejemplo, en el desarrollo del Penn Tree- bank se ignoró, en la primera etapa, la distinción de argumentos y adjuntos en la ción. Aunque el etiquetado de argumentos es crucial para la interpretación semántic del verbo. Los adjuntos corresponden a los complementos que expresan las cunstancias en las que se da la acción. Por ejemplo, en la frase, *compró contra su voluntad un traje nuevo*, el grupo preposicional *contra su voluntad* es un junto que expresa un modificador a la acción *comprar*, pero no es un partic pante de la acción del verbo.

En el nivel semántico, por ejemplo, se ha considerado el etiquetado de significac y de tipo (o concepto) en el desarrollo del *Italian Syntactic-Semantic Treebank* [5].

La mayor parte del etiquetado de corpus en los niveles sintáctico y semántico realizado de forma manual. Considerar esta tarea en un corpus muy grande requeri demasiados recursos humanos. Sin embargo, hay que considerar que se están expl rando numerosos métodos para aprovechar la situación donde se dispone de algu cantidad de datos etiquetados y una cantidad mucho más grande de datos sin etiqu tar, por ejemplo: clasificación de documentos por temas [13], [8].

3. Construcción del corpus

Seleccionamos cuatro periódicos mexicanos que diariamente sitúan en la red una siderable parte de su publicación y que por el tipo de organización permitía una tracción automática por períodos mensuales y anuales. Coleccionamos los textos los artículos de diferentes secciones (economía, política, cultura, deportes, etc.) dur- te los años de 1998 al 2000, todos en formato HTML.

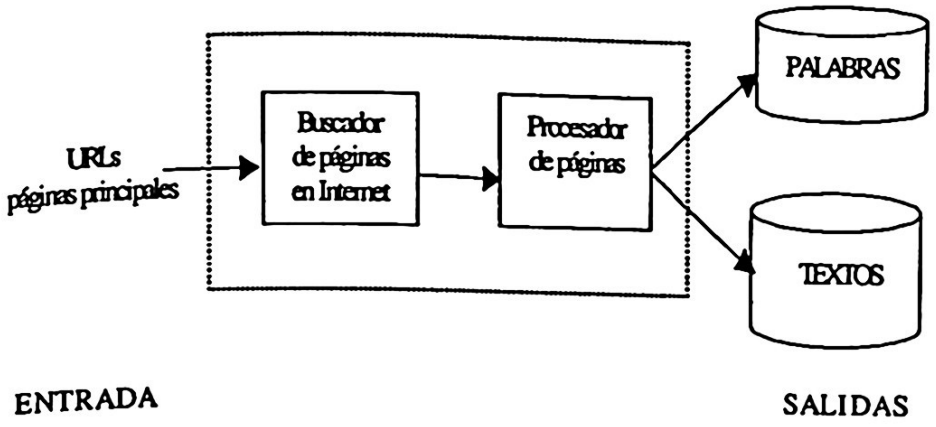


Fig. 1. Primera etapa de la construcción del corpus

En este trabajo presentamos la compilación de los textos, su formateo, su normalización y la etiquetación de grupos de palabras. La realización de estas tareas se llevó a cabo en dos etapas que se describen a continuación.

Primera etapa. En esta etapa (Fig. 1) mediante las direcciones de las páginas principales de cuatro periódicos nacionales coleccionamos 1540 MB en textos en formato original y mediante una serie de pasos que se listan a continuación obtuvimos 1092 MB de texto plano. Todos estos pasos fueron realizados mediante programas en lenguaje PERL.

1. **Buscador de páginas.** La entrada corresponde a las direcciones de las páginas iniciales de cuatro periódicos mexicanos. A partir de estas cuatro direcciones se buscan las direcciones correspondientes a cada día de los años considerados y en forma recursiva se obtienen las páginas incluidas.

2. **Filtro de textos.** Se eliminan los anuncios, los pies de fotos, porciones de código de lenguajes de programación, etc. Se normalizaron algunos signos de puntuación, por ejemplo: comillas (´¨'|"|¨`).

3. **Eliminación de marcas HTML.** La razón para eliminar estas marcas es que su uso no es consistente aún dentro del mismo periódico. Las marcas HTML tampoco aportan información relevante del texto para la construcción del corpus (como palabras extranjeras, marcas semánticas, etc.).

4. **Asignación de estructura.** Se marcan títulos, subtítulos, cuerpo del texto, párrafos y oraciones. Los párrafos se respetaron tal y como aparecen en la versión electrónica. Las oraciones se separaron mediante heurísticas del uso: de punto, signos de admiración e interrogación, números, y abreviaturas comunes.
   Un ejemplo de estructura es el siguiente:

```
<ARTICULO>
2001/dic01/011229/004n1pol
<TITULO>
```

```
PRD y PT votaron en contra, mientras que el Verde Ecologist
se abstuvo.
<SUBTITULO>
Priistas y panistas alcanzaron acuerdo sobre ISR
<AUTOR>
ROBERTO GARDUÑO Y CIRO PEREZ SILVA
<PARRAFO>
Diputados de PAN y PRI aprobaron esta madrugada, con el vot
en contra de PRD, PT y la abstención del PVEM, modificacion
a la Ley del Impuesto sobre la Renta (ISR), destrabando así
uno de los temas fundamentales de la miscelánea fiscal para
allegar mayores recursos al gobierno foxista. El coordinador
de los diputados perredistas, Martí Batres, declaró que el
articulado -con 286 páginas y 86 transitorios- llegó casi
cinco horas después de iniciado el debate, por lo que era
posible analizarlo, y en él se introdujeron elementos desco-
nocidos para los legisladores, que no fueron consensuados
```

Ejemplos de heurísticas para separar oraciones son las siguientes:

- Si se presenta el patrón H. K dónde H es número y K comienza mayúscula:
  Separar en dos oraciones con K iniciando la segunda oración si antes de encuentran varias palabras y la precedente no está en la lista ELEMENTOS *ciso, apartado, capítulo,* etc.)

- Si se presenta el patrón H. K dónde H es una palabra de la ABREV (*sra, Sra, sr, Sr, Dr,* etc.) mantener el grupo unido.

5. **Palabras totales.** Se obtuvo adicionalmente el total de las palabras diferentes contradas en el corpus.
   Las estadísticas de los dos últimos pasos se presentan en la siguiente tabla:

PERIÓDICO	# 1	#2	#3	#4
Número de oraciones	2,927,723	1,328,157	208,298	1,696,358
Número de palabras	87,597,168	38,387,767	5,652,358	45,702,200

Segunda etapa. En esta etapa (Fig. 2) se obtienen las palabras correctas de acuerdc los siguientes pasos.

1. **Separación de las palabras en correctas y erróneas.** Mediante el empleo de corrección ortográfica de un procesador de palabras y la búsqueda en la lista de labras de dos diccionarios para el español, obtuvimos automáticamente las palabr marcadas como "correctas", es decir, reconocidas en dichos recursos. Inicialmen de 751360 palabras diferentes, 60% estaban marcadas como incorrectas.

2. **La corrección ortográfica** usando el procesador de palabras Word se utilizó diante una macro en Visual Basic que marca automáticamente las palabras inc rrectas de la lista completa de palabras obtenidas en el último paso de la etapa ant rior. Los diccionarios se convirtieron de sus archivos electrónicos a un forma texto que considera la palabra y su categoría gramatical. Un programa en PER

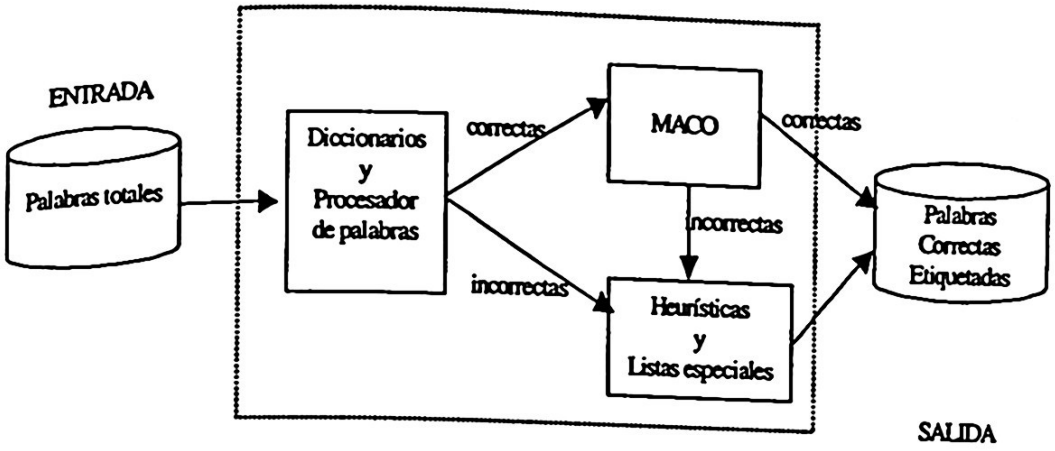


Fig. 2. Segunda etapa de la construcción del corpus

que utiliza esos archivos marca au las palabras correctas y erróneas conforme a esos recursos.

Entre las palabras inicialmente incorrectas encontramos los siguientes casos:

- *Algunas palabras mexicanas específicas*
  Una revisión manual no exhaustiva nos permitió identificar palabras como *ámpula* que no aparecen en DRAE[3], ni en el Diccionario de María Moliner, pero que si aparece en el DEUM[4].
- *Palabras de origen indígena*
  Existen palabras de origen náhuatl, maya, otomí, etc. Por ejemplo: *zotehuela, xochimilca, xochiteca, xoconostle*, etc. Para reconocer estas palabras obtuvimos de Internet y de manuales especializados una lista de palabras de origen indígena para comparar las palabras obtenidas de los textos electrónicos de periódicos. De esta forma se obtuvo un nuevo grupo de palabras correctas.
- *Palabras extranjeras*
  Los textos contienen palabras extranjeras, principalmente del inglés. Usamos la herramienta del procesador de palabras para detectar palabras "correctas" inglesas y francesas, también mediante una macro en Visual Basic.
- *Formas de palabras no contenidas en los recursos empleados*
  Mediante reglas lingüísticas se obtuvo un nuevo grupo de palabras correctas, reconociendo los sufijos de diminutivos, plurales y otras formas regulares.
  Un ejemplo de estas reglas es:
  Si una palabra terminada en "itos" (*abrigaditos*) se encuentra en su forma adjetivo o sustantivo masculino (*abrigado*) en el diccionario, se marca como correcta y diminutivo.
- *Palabras compuestas mediante guiones*
  Los grupos de palabras conectados mediante guiones son muy comunes en el inglés. En cambio, en los manuales de estilo del español se indica que su uso es

[3] Diccionario del español de la Real Academia Española. Espasa, Calpe, 21 ed. 1995
[4] Diccionario del Español Usual en México. Ed. Colegio de México. México, 1996.

muy restringido. Sin embargo, los textos de periódicos muestran el uso de guiones para diversos propósitos:

a) Para enfatizar un grupo de palabras. Ejemplo: *única-y-mejor-ruta*

b) Para evocar consignas. Ejemplo: *sí-se-puede, duro-duro-duro*

c) Para indicar la pronunciación lenta de una palabra. Ejemplo: *é-x-i-t-o, zoo-l gi-co*

d) Para expresar varios atributos diferentes. Ejemplo: *étnico-nacionalista-soci*

e) Para mostrar los vínculos entre corporaciones o nombres reales. Ejempl *ABB-Alsthom, ACDH-ONU, ADM-Dreyfus-Novartis-Maseca*

f) Para indicar una ruta. Ejemplo: *Durango-Mazatlán*

g) Para indicar competencias deportivas (fútbol, básquetbol, etc.). Ejempl *Atlas-Tecos, Alemania- Francia*

h) Para algunas palabras de origen inglés. Ejemplo: *e-mail, e-business, zig-z* etc.

También se encontraron algunos errores introducidos por la copia de los textos ginales en su forma impresa, que contienen el uso principal del guión en el españ para separar sílabas, por ejemplo *de-rechos*

Todos los casos anteriores se procesaron primero como compuesto de palabras, decir, separándolas y revisando que cada una fuera una palabra "correcta". Al esta suposición se procesaron como partes de una sola palabra, es decir, uniéndola revisando que fuera una sola palabra correcta. En general, una marca de sustantivo adjetivo puede asignarse. Esta clasificación se utilizará en el futuro para una asigr ción semántica a cada componente.

Para obtener una mayor cantidad de textos correctos se requerirá de alguna clase corrección de errores puesto que permanecen más del 50% de palabras erróneas: gráficos, ortográficos, etc. La cantidad de errores en las palabras de origen indíger también es alta ya que la mayoría de los hablantes del español desconocen su ortog fía correcta.

Hasta estos resultados no se ha llevado a cabo una evaluación del sistema. evaluación de diagnóstico, un perfil del funcionamiento del sistema respecto a un pacio de posibles entradas, implicaría la construcción de una prueba representati grande y confiable. Esta evaluación requiere a su vez de una inversión significante ra crear: conjuntos de pruebas, procedimientos bien documentados y programas; más de implementar y depurar esos procedimientos.

4 Etiquetado morfológico

Para el etiquetado morfológico consideramos las etiquetas del corpus LEXESP[5] nivel léxico, que ha sido empleado para investigaciones de nuestro grupo de traba

[5] El corpus LEXESP nos fue proporcionado amablemente por Horacio Rodríguez de la versidad Politécnica de Cataluña, en Barcelona, España.

Utilizamos las 275 etiquetas que se emplean en él. Esta cantidad de etiquetas se debe principalmente a la concordancia en género, número y persona que existe en el español.

El corpus LEXESP tiene las categorías PAROLE[6] [11]. La clasificación de categorías gramaticales en PAROLE la presentamos a continuación, donde sólo se detallan las claves completas para los determinantes (Tabla 1), indicando los rasgos considerados.

Adjetivo (A). Ejemplo: *frágiles* <AQ0CP00>
Adverbio (R). Ejemplo: no <RG000>
Artículo (T). Ejemplo: *la* <TDFS0>
Determinante (D), véase ilustración 1. Ejemplo: *tal* <DD0CS00>
Sustantivo (N). Ejemplo: *señora* <NCFS000>
Verbo (V). Ejemplo: *acabó* <VMIS3S0>
Pronombre (P). Ejemplo: *ella* <PP3FS000>
Conjunciones (C). Ejemplo: *y* <CC00>
Numerales (M). Ejemplo: *cinco* <MCCP00>
Preposiciones (SPS00). Ejemplo: *a* <SPS00>
Números (Z). Ejemplo: *5000* <Z>
Interjecciones (I). Ejemplo: *oh* <I>
Abreviaturas (Y). Ejemplo: *etc.* <Y>
Puntuación (F). Todos los signos de puntuación (.,:;-¡!'¿?"%). Ejemplo: "." <Fp>
Residuales (X). Las palabras que no encajan en las categorías previas. Ejemplo: *sine* <X>

Tabla 1. Descripción del marcado morfológico para los determinantes

Tipo		Persona	Género		Número		Caso	Poseedor
Valor	Clave		Valor	Clave	Valor	Clave		
Demostrativo	D	1	Femenino	F	singular	S	0	0
Posesivo	P	2	Masculino	M	Plural	P		
Interrogativo	T	3	Común	C	Invariable	N		
Exclamativo	E							
Indefinido	I							

Un ejemplo de etiquetas en el corpus, es el siguiente, para la palabra *bajo* que puede ser tanto una forma verbal, como preposición, adverbio, sustantivo o adjetivo:

bajar<VMIP1S0> bajo<SPS00> bajo<RG000> bajo<NCMS000> bajo<AQ0MS00>.

El valor común de género se emplea tanto para femenino como para masculino, por ejemplo: *alegre*. El valor invariable en número se emplea tanto en singular como en plural, por ejemplo, el pronombre *se*.

[6] www.ub.es/gilcub/castellano/proyectos/europeos/parole.html

El etiquetado se realizó automáticamente mediante el programa MACO [6] sarrollado por el grupo de Procesamiento de lenguaje natural de la Sección de Inte gencia artificial, Departamento de software de la Universidad Politécnica de Catal en colaboración con el Laboratorio de lingüística computacional de la Universidad Barcelona.

Esta última etapa considera el marcado de palabras especiales, entre las cuales tán las preposiciones compuestas y los nombres propios.

1. Preposiciones compuestas

Existen muchas preposiciones compuestas en el español, además de las preposicio simples. Grupos de palabras como *al cabo de*, *a fin de*, *con respecto a*, requieren manejo conjunto. Conforme a la lista de preposiciones compuestas de [12] se etiq taron automáticamente, mediante un programa en PERL, las preposiciones compu tas en el corpus de textos electrónicos de periódicos.

2. Nombres propios

Se encontraron 168,333 compuestos diferentes de palabras con mayúsculas (inicia zando las palabras o de forma generalizada). Estos compuestos se repiten tenien 1,804,959 ocurrencias en los textos. Los grupos que se encontraron corresponden

- *Acrónimos.* Por ejemplo: PRD, PT, ONU.
- *Frases de nombres.* Por ejemplo: *Convergencia por la Democracia, Ley del puesto sobre la Renta, Luz y Fuerza del Centro.*
- *Nombres.* Por ejemplo: *San Lázaro, Benito Juárez.*

En esta asignación se presentan los siguientes problemas para los cuales se desarrollando heurísticas:

- *Coordinación.* Por ejemplo: *Luz y Fuerza del Centro y Lotería Nacional* (Luz Fuerza del Centro, Lotería Nacional), *Margarita Diéguez y Armas y Carlos gilio Ferrer* (Margarita Diéguez y Armas, Carlos Virgilio Ferrer.)
- *Nombres compuestos donde solamente la primera palabra tiene mayúscula.* ejemplo: *cantando México lindo, Cucurrucucú, La malagueña y Amorcito coraz como La boa, El mudo, Luces de Nueva York, Perfume de mujer.*
- *Varios nombres compuestos ligados.* Por ejemplo: *Hospital de Traumatología Lomas Verdes* (Hospital de Traumatología, Lomas Verdes), *Comisión de Prot ción al Empleo y Previsión Social de la Asamblea Legislativa del Distrito Fede* (Comisión de Protección al Empleo y Previsión Social, Asamblea Legislativa Distrito Federal).

Las heurísticas consideran los casos de uso de preposición como "por la" que une sistemáticamente grupos de nombre de persona como en : *Juan Ramón Fuente por la Federación de Colegios de Personal Académico* (Juan Ramón Fuente, Federación de Colegios de Personal Académico) pero si une grupos de bras comunes como en *Alianza por la Ciudad de México.*

5 Conclusiones

Indicamos la utilidad de las colecciones de textos con etiquetado en diferentes niveles, es decir, su empleo para diferentes trabajos en investigación teórica y en diversas aplicaciones de lenguaje natural.

Presentamos el desarrollo de un recurso para el procesamiento lingüístico de textos en español: un corpus de textos de decenas de millones de palabras con etiquetado morfológico. Presentamos el proceso que requiere la colección de textos obtenida para la construcción del corpus. Detallamos el método empleado para la compilación del corpus y para el anotado léxico.

La principal ventaja de nuestro método de obtención es que la mayor parte del trabajo se ha realizado en forma automática, lo que reduce tiempos y costos en la compilación.

Referencias

[1] Banko, M. and Brill, E. Mitigating the Paucity of Data Problem. In Proceedings of the Conference on Human Language Technology, San Diego, Ca. 2001

[2] Berthouzoz, C. and Merlo, P. *Statistical ambiguity resolution for principle-based parsing.* In Proceedings of the Recent Advances in Natural Language Processing. Pag. 179-186, 1997

[3] Biber, D. Using Register. *Diversified Corpora for general Language Studies.* Computational Linguistics 19 (2) pp. 219—241, 1993.

[4] Bolshakov, I. A., A. Gelbukh. On Detection of Malapropisms by Multistage Collocation Testing. Proc. of NLDB-2003, 8th International Workshop on Applications of Natural Language to Information Systems. Lecture Notes in Computer Science, Springer-Verlag, 2003, to appear.

[5] Calzolari, N. Corazzari, O. & Zampolli, A. *Lexical-Semantic tagging of an Italian Corpus.* Second Conference on Intelligent text processing and Computational linguistics. CICLing-2001.

[6] Carmona, J., S. Cervell, L. Màrquez, M.A. Martí, L. Padró, R. Placer, H. Rodríguez, M. Taulé & J. Turmo. *An Environment for Morphosyntactic Processing of Unrestricted Spanish Text.* First International Conference on Language Resources and Evaluation (LREC'98). Granada, Spain, 1998.

[7] Francis, W. N. and Henry Ku•era. *Frequency Análisis of English Usage: Lexicon and Grammar.* Houghton Mifflin. 1982

[8] Gelbukh, A., G. Sidorov, A. Guzman-Arenas. *Use of a weighted topic hierarchy for document classification.* In Václav Matoušek *et al.* (Eds.). Text, Speech and Dialogue. Proc. TSD-99. Lecture Notes in Artificial Intelligence, N 1692, Springer-Verlag, 1999, pp. 130–135.

[9] Gelbukh, A., G. Sidorov, *et al. Compilation of a Spanish representative corpus.* In Intelligent Text Processing and Computational Linguistic. Proc. CICLing-2002. Lecture Notes in Computer Science, N 2276, Springer-Verlag, 2002, pp. 285–288.

[10] Marcus, M., Santorini, B. and Marcinkiewicz, M. *Building a large annotated corpus of English The Penn Treebank.* Computational Linguistics 19, 2, 1993.

[11] Martí M.A., Rodríguez H., Serrano J. *Declaración de categorías morfosintácticas.* Documento interno ITEM n-o2. Universidad Politécnica de Cataluña, España, UB, 1997.

[12] Nañez Fernández, E. *Diccionario de construcciones sintácticas del español*. Preposiciones. Ed. de la Universidad Autónoma de Madrid, España 1995.

[13] Nigam, K. et al. Using Maximum Entropy for Text Classication. In Proceedings of IJCAI-99 Workshop on Machine Learning for Information Filtering, pp. 61-67. 1999

[14] Pedersen, T. *An Ensemble Approach to Corpus-based Word Sense Disambiguation*. Conference on Intelligent text processing and Computational linguistics. CICLing-2000

[15] Ratnaparkhi, A. *Statistical Models for Unsupervised Prepositional Phrase Attachment*. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics. Montreal, Quebec, Canada, 1998 http://xxx.lanl.gov/ps/cmp-lg/9807011

[16] Roland, D. and D. Jurafsky. *How Verb Subcategorization Frequencies are Effected Corpus Choice*. In Proceedings International Conference COLING-ACL'98. August 10-14 Quebec, Canada, pp. 1122-1128, 1998.

[17] Weischedel, Ralph; Marie Meteer, Richard Schwartz, Lance Ramshaw, and Jeoe Palmucci. *Coping with ambiguity and unknown words through probabilistic models*. Computational Linguistics, 19(2): 359-382, 1993.

[18] Yeh, Alexander S., M. B. Vilain. *Some Properties of Preposition and Subordinate Conjunction Attachments*. In Proceedings International Conference COLING-ACL'98. August 10-14 Quebec, Canada, pp. 1436-1442, 1998. http://xxx.lanl.gov/ps/cmp-lg/9808007.

Caracterización de series de tiempo con escalamiento multidimensional

V. Rivera-Mancera [1], E. Bautista-Thompson [2], J. Figueroa-Nazuno [3]

Centro de Investigación en Computación, Instituto Politécnico Nacional
Unidad Profesional "Adolfo López Mateos" Zacatenco
Col. Linda Vista, C.P. 07738, México D.F.
rivera_mancera@hotmail.com [1] ebautista@correo.cic.ipn.mx [2]
jfn@cic.ipn.mx [3]

Resumen. Dentro de las técnicas de análisis multivariadas podemos citar el Escalamiento Multidimensional (Multidimensional Scaling MDS). El MDS es una técnica de interdependencia que trata de representar en un espacio geométrico de pocas dimensiones, las proximidades existentes entre un conjunto de objetos en base a métricas de similitud. El problema de identificar series de tiempo con propiedades similares que son indicativas de su dificultad de predicción, motivó en el presente trabajo, el aplicar la técnica MDS para identificar agrupaciones de series que cumplen lo anterior. Para ello se caracterizó un conjunto de 30 series de tiempo de diferentes orígenes (natural y artificial), a las cuales se les calculó un conjunto de propiedades representativas de origen computacional, topológico, estadístico, espacial, y temporal

Introducción

El escalamiento multidimensional (Multidimensional Scaling MDS), permite obtener información cuantitativa y cualitativa de las posibles relaciones de similitud entre objetos (en este caso de series de tiempo), mediante el escalamiento multidimensional métrico y el escalamiento multidimensional no-métrico respectivamente. El MDS es una técnica de representación espacial que trata de visualizar sobre un mapa, un conjunto de propiedades cuya posición relativa se desea analizar. El propósito del MDS es transformar las propiedades de las series de tiempo y llevarlas a una matriz de distancias susceptible de ser representada en un espacio multidimensional. El MDS está basado en la comparación de objetos, de forma tal que si un individuo juzga a los objetos A y B como los más similares, es debido a que la técnica de MDS coloca a los objetos A y B en el gráfico a una distancia entre ellos que sea la menor respecto a cualquier otra distancia entre cualquier otro par de objetos [1, 2].

Existen otras técnicas multivariadas, como son el análisis factorial y el análisis cluster, que persiguen objetivos muy similares al MDS pero difieren en una serie de aspectos, cómo por ejemplo: en el MDS no es necesario especificar cuáles son las variables a emplear en la comparación de objetos, algo que es fundamental en el análisis factorial y en el análisis cluster. Sin embargo, la utilización de alguna de estas técnicas no supone que no se pueda utilizar el escalamiento multidimensional, sino

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 171–180, 2003. © Centro de Investigación en Computación, IPN, México.

que esta última técnica puede servir como complemento a las otras técnicas multivariadas.

Los objetivos de éste trabajo son:

1) Encontrar propiedades que indiquen la predictibilidad de las diferentes series tiempo.
2) Aplicar la técnica MDS para la caracterización de las mismas.

2 El modelo general de escalamiento multidimensional

De modo general, podemos decir que el MDS toma como entrada una matriz proximidades, $\Delta \in M_{nxn}$, donde n es el número de series de tiempo. Cada elemento δ_{ij} de Δ representa la proximidad entre las propiedades para las series de tiempo ST_i y ST_j donde $\delta_{ij} = (P_i - P_j)^2$, P_k es una propiedad de la serie de tiempo k-ésima.

$$\Delta = \begin{bmatrix} \delta_{11} & \delta_{12} & \cdots & \delta_{1n} \\ \delta_{21} & \delta_{22} & \cdots & \delta_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{n1} & \delta_{n2} & \cdots & \delta_{nn} \end{bmatrix}$$

El MDS inicia con una matriz aleatoria $X \in M_{nxm}$, donde n, al igual que antes, número de series de tiempo, y m es el número de dimensiones. En nuestro tomaremos $m = 2$. Cada valor x_{ij} representa la coordenada de la ST_i en dimensión j.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix}$$

A partir de esta matriz X se puede calcular la distancia existente entre dos series tiempo cualesquiera i y j, simplemente aplicando la fórmula general de la distancia Minkowski:

$$d_{ij} = \left[\sum_{t=1}^{m} \left(x_{it} - x_{jt} \right)^p \right]^{1/p}$$

donde p puede ser un valor entre 1 e infinito, en nuestro caso tomaremos $p =$ partir de estas distancias podemos obtener una matriz de distancias que denominan $D \in M_{nxn}$.

$$D=\begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix} \tag{4}$$

La solución proporcionada por el MDS debe ser de tal modo que haya la máxima correspondencia entre matriz de proximidades inicial Δ y la matriz de distancias obtenidas D, la cuál se logra modificando la matriz X de forma iterativa.

$$X=\frac{BX}{2n} \tag{5}$$

En donde B tiene como elementos:

$$b_{ij}=\frac{-2\delta_{ij}}{d_{ij}} \quad si \quad i\neq j \tag{6}$$

$$b_{ii}=\sum\sum\frac{2\delta_{ik}}{d_{ik}} \quad si \quad i=j \tag{7}$$

$$b_{ij}=0 \quad si \quad d_{ij}=0 \tag{8}$$

3 Modelos de escalamiento multidimensional

Existen dos modelos básicos de MDS que son: el modelo de escalamiento métrico y el modelo de escalamiento no métrico. En el primero de ellos consideramos que los datos están medidos en escala de razón o en escala de intervalo y en el segundo consideramos que los datos están mediados en escala ordinal. No se ha desarrollado todavía ningún modelo para datos en escala nominal. Para el estudio de las series de tiempo sólo usaremos el modelo de escalamiento no métrico, ya que es éste el que permite formar clases o agrupaciones, a diferencia del escalamiento métrico que arroja información cuantitativa.

Modelo de escalamiento no métrico

A diferencia del escalamiento métrico, el modelo de escalamiento no métrico no presupone una relación lineal entre las proximidades y las distancias, sino que establece una relación monótona creciente entre ambas, es decir, si $\delta_{ij}<\delta_{kl}\Rightarrow d_{ij}\leq d_{kl}$. Su desarrollo se debe a Shepard (1962) quién demostró que es posible obtener soluciones métricas asumiendo únicamente una relación ordinal entre proximidades y distancias. Posteriormente Kruskal (1964) mejoró el modelo [1, 2].

El procedimiento se basa en los siguientes apartados:

1) Obtención de una matriz $X\in M_{nxm}$ de coordenadas aleatorias, que nos da la distancia entre las series de tiempo.

2) Comparación de las proximidades con las distancias contenidas en obteniéndose las disparidades.
3) Cálculo del S-Stress (explicado más adelante en está sección).
4) Modificación de la matriz X mediante la ecuación 5 con el fin de minimizar el Stress.
5) Ir al paso 2 hasta que el S-Stress alcance el valor deseado

Tanto para el modelo métrico como para el modelo no métrico es necesario obtener un coeficiente que nos informe sobre la precisión del modelo. Sabemos que las distancias son una función de las similitudes, es decir:

$$f : \delta_{ij}(x) \rightarrow d_{ij}(x) \tag{9}$$

de esta forma se tiene que $d_{ij} = f(\delta_{ij})$. Esto no deja ningún margen de error. Sin embargo, en las proximidades empíricas es difícil que se dé la igualdad, por lo que generalmente ocurre que $d_{ij} \approx f(\delta_{ij})$, $f(\delta_{ij}) = a + b\delta_{ij}$ donde a y b son constantes que se deben de determinar. A las transformaciones de las proximidades por f se denomina *disparidades.* A partir de aquí podemos definir el error cuadrático como:

$$e_{ij}^2 = (f(\delta_{ij}) - d_{ij})^2$$

Como medida de la precisión del modelo utilizaremos el S-*Stress* definido como:

$$S-Stress = \sqrt{\frac{\sum_{i,j}\left(f(\delta_{ij})^2 - d_{ij}^2\right)^2}{\sum_{i,j}\left(d_{ij}^2\right)^2}} \tag{10}$$

4 Análisis experimental de las series de tiempo

Se tomaron 30 series de tiempo que han sido reportadas en la literatura especializad en la evaluación de técnicas de predicción y modelado [3, 8] y que además corresponden a diversos orígenes (experimental o generadas por modelc matemáticos). La selección de las propiedades calculadas que caracterizan a las serie de tiempo se realizo considerando que cuantifican características de tipo estadístico topológico, computacional, espacial y temporal [3, 4, 7].

A continuación se da una descripción de las propiedades calculadas:

Exponente de Lyapunov. El exponente principal de Lyapunov, mide la evolución trayectorias vecinas en el espacio fase. Mide la inestabilidad de la dinámica sistema debido a cambios en sus condiciones iniciales [3, 6].

Exponente de Hurst. El exponente de Hurst permite determinar si el fenómeno representado por la serie de tiempo presenta correlaciones de largo alcance (memoria y persistencia de largo alcance) [3, 6].

Dimensión de Capacidad. La dimensión de capacidad es similar a la dimensión de Hausdorff y mide el grado de auto-similitud del sistema (comportamiento invariante ante cambios de escala espacial), permite cuantificar el grado de heterogeneidad de la señal a diferentes escalas [3, 6].

Dimensión de Correlación. La dimensión de correlación mide la cantidad de veces que la trayectoria del sistema pasa por una vecindad dada en el espacio fase, cuantifica la correlación espacial local entre puntos de la trayectoria en el espacio fase, sin tomar en cuenta el grado de correlación temporal [3, 5].

Frecuencia Dominante. Permite determinar si existe alguna frecuencia característica de la señal. Las series de carácter aleatorio poseen espectros amplios sin ninguna frecuencia dominante, lo mismo se aplica en cierto grado para las series caóticas. Las series periódicas y cuasi-periódicas poseen picos bien definidos [3, 5].

Entropía Espacio-Temporal. La entropía-temporal, cuantifica de forma global la no correlación de los datos mediante el análisis de recurrencia [3, 5].

Entropía de Shannon. La entropía de Shannon, es una medida de la cantidad de información que se obtiene al tomar una medida para especificar el estado del sistema [5, 6].

Porcentaje de Determinismo. Permite medir el grado de determinismo en el sistema, por medio del análisis de mapas de recurrencia [3, 5].

Porcentaje de Recurrencia. Permite medir el grado de recurrencia (periodicidad y estructura) entre los datos de la serie, que es indicativo de patrones repetitivos en la serie de tiempo, por medio del análisis de mapas de recurrencia [3, 5].

Reglas de Producción. La generación de gramáticas a partir de una serie de tiempo permite dar una medida de complejidad (computacional) en la cual a mayor número de reglas de producción necesarias para generar una gramática, mayor es la dificultad para la predicción o modelado de la serie de tiempo [3, 8].

La tabla 1 nos muestra las 30 series de tiempo y los valores calculados.

5 Método experimental del escalamiento multidimensional

Una vez obtenidas las propiedades de las series de tiempo se aplicó el procedimiento de MDS, para cada una de las propiedades para el conjunto de 30 series de tiempo.

El diagrama 1 muestra el procedimiento de MDS para las series de tiempo, usando el algoritmo ALSCAL (Alternating Least Squares SCALing), en el cuál se usaron las propiedades de las series de tiempo, para construir una matriz a través de la ecuación $\delta_{ij} = (P_i - P_j)^2$. Con esta matriz se cálculo la matriz de distancias. Este proceso se repitió hasta que el S-Stress presento una mejoría mínima de 0.001.

Tabla 1. Propiedades calculadas experimentalmente de las treinta series de tiempo

series de tiempo	exp. Hurst	dim. capacidad	dim. correlación	exp. Lyapunov	frecuencia Dominante	reglas de producción	entropía (Shannon)
Seno	0.948	0.246	0.228	0.517	-0.001	27	3.712
Qp2	0.002	0.679	0.923	0.925	0.13	66	2.446
Mkg	0.078	0.983	1.025	1.481	0.068	75	0.159
Log	-0.059	0.941	0.93	0.76	0.38	61	0.035
Lorentz	0.756	0.965	1.025	0.601	0.001	63	4.789
Rossler	0.562	0.994	1.026	1.049	0.017	72	2.459
Ikeda	-0.02	0.983	1.019	1.452	0.324	71	
Henon	-0.33	0.997	0.991	2.301	0.46	69	0.02
D1	0.312	0.986	2.053	1.288	0	92	0.852
Laser	0.088	0.961	2.096	0.949	0.13	66	2.935
Dowj	0.563	0.887	1.035	0.144	0	65	6.435
Kobe	0.016	0.904	1.053	1.102	0.22	79	3.744
Ecg	0.748	0.934	1.051	0.242	0	53	3.51
Eeg	0.204	0.890	1.876	1.0199	0.041	82	
Brownian	0.537	0.971	1.103	2.043	0	85	2.74
Whiten	0.001	0.983	2.086	1.606	0	73	
Ascii	-0.147	0.485	2.011	2.709	0.288	80	2.107
Cantor	0.001	0.661	0.658	5.728	0.422	72	
El niño	0.447	0.971	1.637	2.322	0	83	3.526
Hiv DNA	0.0014	0.953	5.017	0	0	9	
Human DNA	0.562	0.983	1.037	0.322	0	51	7
Loviana	0.509	0.958	1.027	1.069	0	77	5.574
Plasma	0.062	0.821	0.967	3.383	0.08	91	2.866
Primos	-0.01	0.044	3.55	0.594	0.5	78	1.295
Qp3	-0.008	0.605	0.96	-1.383	0.447	75	4.232
Sp500	0.015	0.859	1.026	3.569	0.035	82	2.585
Start	0.164	0.963	1.144	3.289	0.018	90	3.102
Tent	-0.243	0.971	1.029	0.477	0.5	13	3.401
Vanderpol	0.497	0.989	0.989	1.864	0.112	28	4.981

6 Resultados

A continuación se muestran los análisis correspondientes a tres de las propiedades más características para la cuantificación de la dificultad de predicción de las series tiempo. Para hacer la comparación se tomaron las clasificaciones de las series propuestas por Figueroa [3] (periódicas, cuasi-periódicas, caóticas, complejas estocásticas) la cual se basa en el comportamiento observado de la dinámica de señal.

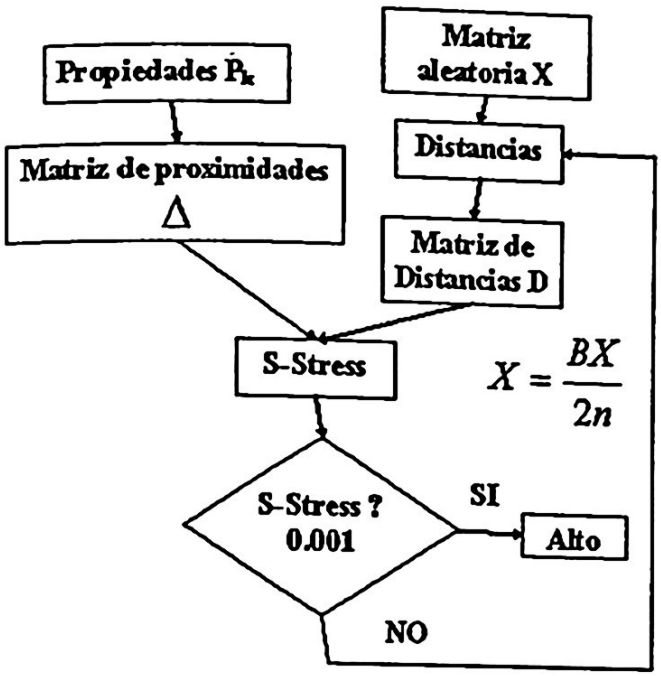


Diagrama 1. Algoritmo ASCAL

Al hacer el análisis MDS del exponente de Lyapunov (ver Fig. 1), se encontraron tres conjuntos que se diferencian por el nivel de inestabilidad que presenta la serie, cuando se le cambian las condiciones iniciales: las series del conjunto I tiene alta estabilidad a los cambios iniciales, las series del conjunto II son medianamente estables a cambios iniciales, y las series del conjunto III son poco estables a cambios iniciales.

Al hacer el análisis MDS del número de reglas de producción (ver Fig. 2), se encontraron tres conjuntos que se diferencian por el nivel de la complejidad computacional: las series del conjunto I tiene alta complejidad computacional, las series del conjunto II presentan complejidad computacional media, y las series del conjunto III presentan baja complejidad computacional.

El análisis MDS de entropía de Shannon (ver Fig. 3), nos muestra tres conjuntos que se diferencian por sus niveles de contenido de información: las series del conjunto I tiene alto contenido de información, las series del conjunto II presenta contenido de información medio, y las series del conjunto III tienen poco contenido de información.

Por último queremos hacer notar, que las relaciones de agrupamiento entre series de tiempo cambian dependiendo de la propiedad analizada con MDS, lo que nos indica que la clasificación (periódica, cuasi-periódica, caótica, compleja y estocástica) no permite caracterizar de forma completa su comportamiento. Por ejemplo, human DNA y HIV DNA aparecen en el mismo grupo en la Fig. 1 y en distintos grupos en las Fig. 2 y la Fig. 3. Ya que las dos series se toman como complejas se hubiera pensado que aparecería siempre en el mismo grupo con las diferentes propiedades.

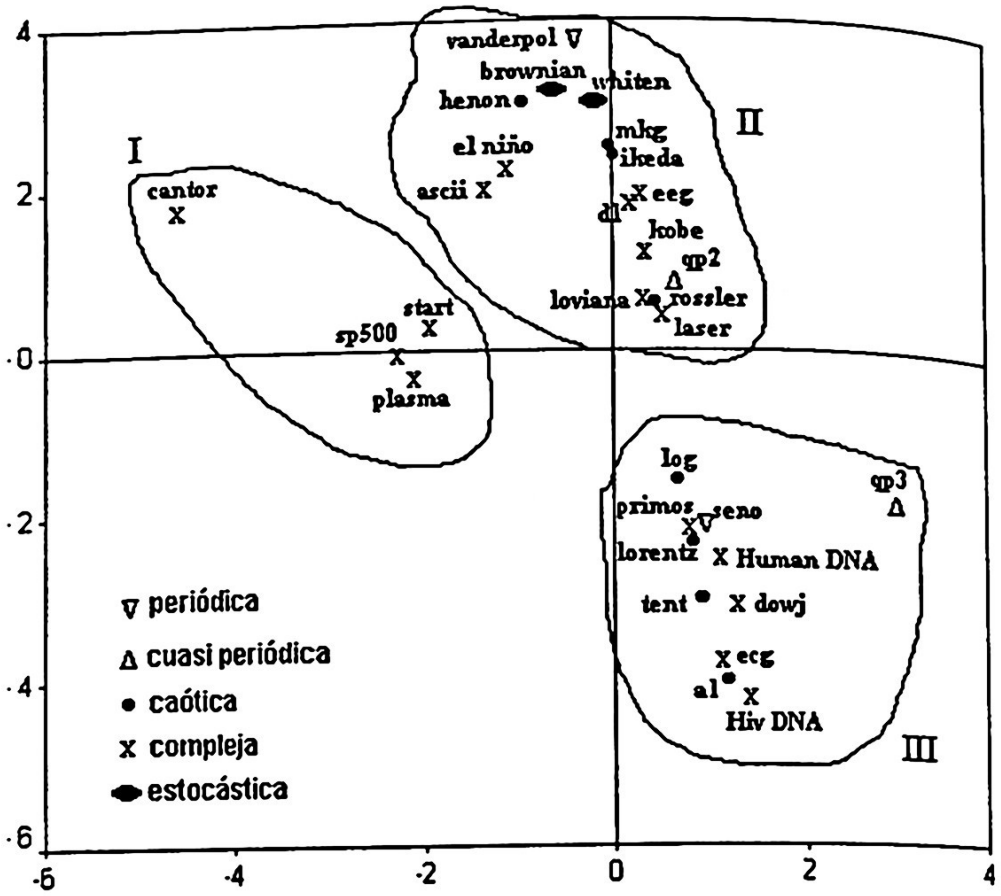


Fig. 1 MDS del exponente de Lyapunov de las treinta series

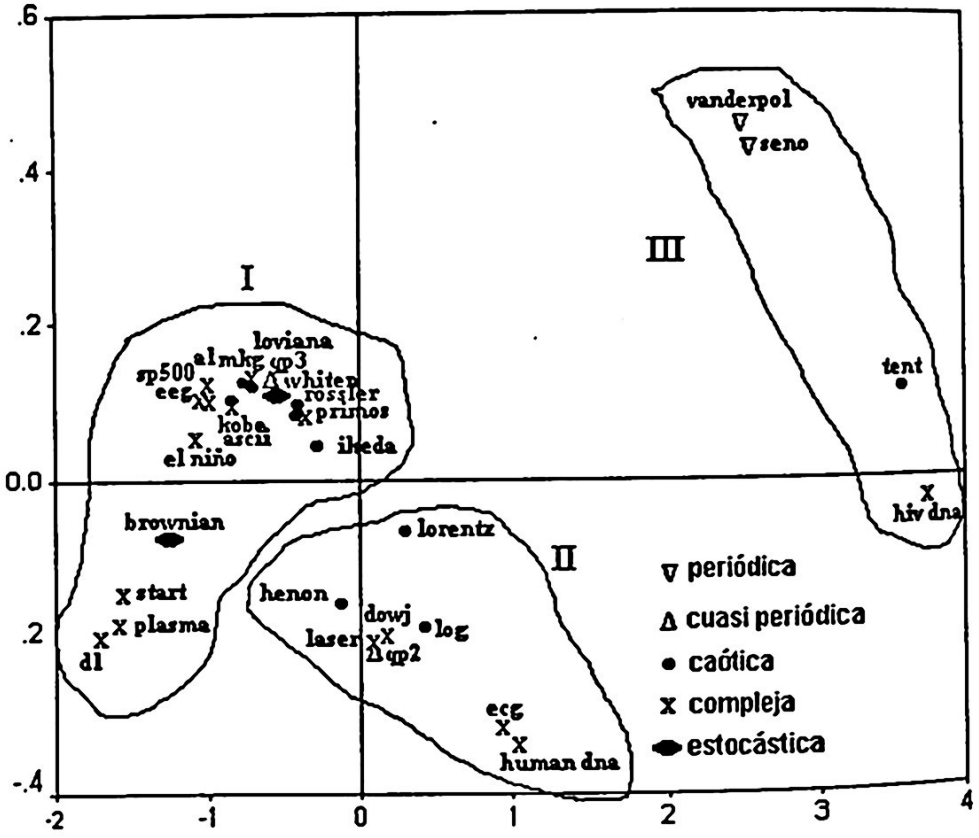


Fig. 2 MDS de reglas de producción de las treinta series

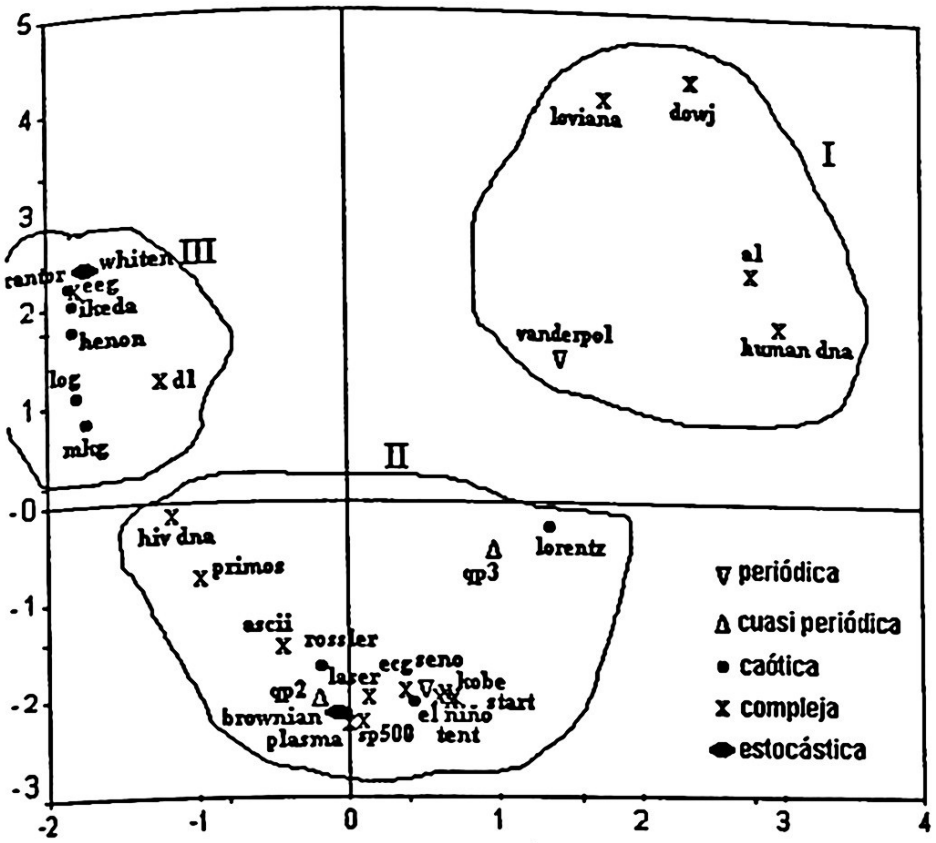


Fig. 3 MDS de entropía (Shannon) de las treinta series

7 Conclusiones

La relación entre la clasificación de referencia: (periódicas, cuasi-periódicas, caóticas, complejas y estocásticas) y las propiedades seleccionadas (exponente de Lyapunov, reglas de producción, entropía de Shannon) que permiten caracterizar la dificultad de predicción de las series de tiempo, es de carácter no lineal. Esto nos indica que clasificar una serie de tiempo únicamente por su comportamiento dinámico no es suficiente para caracterizar de forma completa a la misma. El análisis MDS ayuda a identificar clases o agrupaciones de series de tiempo con propiedades similares. La información del conjunto de propiedades puede ser utilizada por ejemplo, en la identificación de las relaciones entre las agrupaciones de series y así seleccionar el modelo de predicción más adecuado para las mismas.

Referencias

1. William R. Dillon & Matthew Goldstein (1984). "Multivariate Analysis: Methods and Applications" Editorial John-Wiley
2. Dallas E. Johnson (2000). "Métodos Multivariados Aplicados al Análisis de Datos" Thomson Editores
3. E. Bautista-Thompson, J. Figueroa-Nazuno. "Matriz de Conocimiento sobre la Complejidad de Predicción en Series de Tiempo". VII Congreso Iberoamericano

en Reconocimiento de Patrones. México, D. F. del 19 al 22 de Noviembre del 2002.

4. M.E Acevedo-Mosqueda, C.G. León-Vega & J. Figueroa-Nazuno "Medición de la Complejidad de Series de Tiempo". XLIV Congreso Nacional de Física, Morelia, Michoacán 2002
5. Espinosa-Contreras & J. Figueroa- Nazuno "Análisis del Comportamiento de la Perdida de Paquetes en la Red Internet con técnicas de la Dinámica No lineal". XLIV Congreso Nacional de Física, Morelia, Michoacán 2002
6. Robert Hilborn "Chaos and Nonlinear Dynamics An Introduction for Scientists and Engineers". Editorial Oxford University Press 2000
7. E. Bautista-Thompson, J. Figueroa-Nazuno. "Análisis de Propiedades que Caracterizan la Dificultad de Predicción en Series de Tiempo". XLV Congreso Nacional de Física. León, Guanajuato del 28 de Octubre al 1 de Noviembre del 2002.
8. R. Menchaca-Mendez, C. Sanchez-Rodrígez, J. Figueroa-Nazuno. "Predicción de Series de Tiempo Mediante Análisis Gramatical". XLIII Congreso Nacional de Física. Puebla, Puebla.

Evaluación de red neuronal artificial polinomial (RNAP) para el modelado de series de tiempo

E. Guzmán-Ramírez, E. Bautista-Thompson, J. Figueroa-Nazuno

Centro de Investigación en Computación, Instituto Politécnico Nacional.
Unidad Profesional "Adolfo López Mateos". Zacatenco.
Col. Lindavista, CP 07738. México D.F.
erick_igr@yahoo.com, ebautista@correo.cic.ipn.mx, jfn@cic.ipn.mx

Resumen. Se presenta la evaluación de RNAP (Red Neuronal Artificial Polinomial) para el modelado de un conjunto de treinta series de tiempo de diferente origen. En la literatura existen pocos trabajos reportados sobre el desempeño de técnicas de modelado para un conjunto amplio de series de tiempo, así como la comparación de diferentes técnicas de modelado. El objetivo de este trabajo es cuantificar el desempeño de RNAP para series con diferente Índice de Dificultad de Modelado construido a partir de tres propiedades características: Reglas de Producción, Exponente de Lyapunov y Entropía de Shannon. Se comparan los resultados obtenidos con aquellos reportados en la literatura para otras técnicas de modelado de series de tiempo, mostrándose que la capacidad de modelado de los algoritmos híbridos tales como RNAP es comparable con la de las mejores técnicas reportadas.

1 Introducción

En la literatura especializada se han reportado gran cantidad de técnicas para el modelado de series de tiempo, sin embargo, se encuentran pocos trabajos que estudien el desempeño de técnicas de modelado para un conjunto variado de series de tiempo y que además, comparen las diferentes técnicas entre sí [1].

En este trabajo se reporta un estudio sobre el desempeño o capacidad de modelado de series de tiempo, utilizando redes neuronales polinomiales de arquitectura adaptable, específicamente la Red Neuronal Artificial Polinomial (RNAP) [2].

El estudio de capacidad de modelado se llevo a cabo tomando como base el número de experimentos de modelado exitosos además de comparar dicha capacidad con respecto a un índice de dificultad de modelado que fue calculado para cada serie de tiempo. RNAP fue evaluada utilizando series de dos clases: series experimentales (económicas, físicas, fisiológicas) y series sintéticas en las cuales se tiene control sobre los parámetros de la función generadora (Seno, Lorentz, Henon, Ikeda).

El artículo consta de las siguientes secciones: en la sección 2 se presenta la descripción de los principios de funcionamiento de la RNAP. En la sección 3 se explican las tres propiedades seleccionadas para evaluar la dificultad de modelado de las series de tiempo: Reglas de Producción, Exponente de Lyapunov y Entropía de Shannon además de presentarse la caracterización de las series y definirse el Índice de Dificultad de Modelado. En la sección 4 se describe la metodología seguida para

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 181–192, 2003. © Centro de Investigación en Computación, IPN, México.

realizar los experimentos. En la sección 5 se presentan los resultados obtenidos y la comparación con otras técnicas, por último, en la sección 6, se presentan las conclusiones de este artículo.

2 Modelo RNAP

RNAP es un modelo de red neuronal que define su función de transición en términos de un sistema no lineal de ecuaciones polinomiales el cual es optimizado en la fase de entrenamiento por medio de una implementación de Algoritmo Genético con el fin obtener la mejor arquitectura de la red.

El modelo RNAP se define como:

$$\hat{y}_k = \left[\phi(x_{1,k}, x_{2,k}...x_{n_i,k}, x_{1,k-1}x_{2,k-1},..x_{1,k-1}...,x_{n_i,k-n_1}, y_{k-1}, y_{k-2},...,y_{k-n_2})\right]_{\phi\min}^{\phi\max}. \quad (1)$$

donde $\hat{y}_k \in \Re$, es la salida de la red, $\phi(x,y) \in \Re$ es una función no lineal, $x_i \in X$, son las entradas, $i=1,...,n$,; n_i es el número de entradas, $y_{k-j} \in Y$ son los valores anteriores de la salida $j=1,...,n_2$; n_1 es el número de retardos de la entrada, n_2 es número de retardos de la salida y X, Y, son conjuntos compactos de $\Re$.

La función no lineal $\phi(z)$ está dada por:

$$\left[\phi(z)\right]_{\phi\min}^{\phi\max} = \begin{cases} \phi_{\max}, & \phi(z) \geq \phi_{\max} \\ \phi(z), & \phi_{\min} < \phi(z) < \phi_{\max} \\ \phi_{\min}, & \phi(z) \leq \phi_{\min} \end{cases}. \quad (2)$$

donde $\phi_{\max}$ y $\phi_{\min}$ son los limites máximos y mínimos respectivamente.

Para simplificar la notación se rescribe (2) haciendo el siguiente cambio variable:

$$z = \{x_{1,k}, x_{2,k}...x_{n,k}, x_{1,k-1}x_{1,k-1},...x_{1,k-1}...,x_{1,k-n}, y_{k-1}, y_{k-2},..., y_{k-n_2}\}. \quad (3)$$
$$z = \{z_1, z_2...z_{nv}\}$$

Donde n_v es el número total de elementos de la descripción z, es decir, el numero de entradas, valores anteriores y valores anteriores de salida.

Dado el cambio de variable, la función $\phi(z_k) \in \Phi_p$ conforma una familia polinomios que pueden ser representados como:

$$\Phi_p(z_1, z_2,..., z_n) = \left\{ \begin{matrix} \phi(z) : \phi(z) = a_0(z_1, z_2,..., z_{nv}) + a_1(z_1, z_2,..., z_{nv}) + ... \\ + a_p(z_1, z_2,..., z_{nv}) \end{matrix} \right\}.$$

donde los términos $a_i(z_1,z_2,...,z_i)$ son polinomios homogéneos de grado total i, $i=0,...,p$.

2.1 Aprendizaje de RNAP

Primeramente se definen:

- Error de aproximación de RNAP

$$err_n(y^n,\phi(z))=\frac{1}{n}\sum_{k=1}^{n}(y_k-\phi(z_k))^2=\frac{1}{n}\sum_{k=1}^{n}\left(y_k-\hat{y}_k\right)^2, y^n=(y_1,y_2,...,y_n)\ . \qquad (5)$$

Donde $\phi(z_k)\in\Phi_p$ y n es el numero de puntos.

- Error óptimo:

$$opterr_n(y^n,\phi(z))=\min_{\phi\in\Phi_p}\{err_n(y^n,\phi(z))\}$$
$$=err_n(y^n,\phi_n^*(z)) \qquad (6)$$

donde $\phi_n^*(z)\in\Phi_p$ es la estimación optima de y^n.

La RNAP $\phi(z)\in\Phi_p$ aprende uniformemente la salida deseada con la precisión ε si

$$err_n(y^n,\phi(z))-err_n(y^n,\phi_n^*(z))>=\varepsilon$$
$$\varepsilon>=0 \qquad (7)$$

Una vez definidos estos conceptos el problema de aprendizaje de RNAP consiste en encontrar la estructura de $\phi\in\Phi_p(z)$ que verifica la desigualdad descrita en (7).

2.2 Optimización mediante algoritmo genético

El problema de aprendizaje de RNAP para una arquitectura específica puede ser representado por un problema de optimización con los siguientes dos pasos:

$$\min_{W_b}\left\{\min_{W\in\Re}\left\{err_n(y^n,\phi(z))\Big|_{W_b}\right\}\right\} \qquad (8)$$

donde $err_n\left(y^n,\phi(z)\right)\Big|_{W_b}$ es el error definido en (1) para un valor determinado del vector binario W_b.

El objetivo del algoritmo genético es encontrar el valor de W_b para el cual el vector de pesos W hace que se cumpla (7). Este arreglo puede ser obtenido utilizando el método de mínimos cuadrados, y queda expresado como:

$$W_{W_b}=\text{argmin}\left\{err_n\left(y^n,\phi(z)\right)\Big|_{W_b}\right\}$$
$$W_{W_b}=\Gamma^N\sum_{k=1}^{n}y_k\left(M_b(z_k)\right)\Big|_{W_b} \qquad (9)$$

donde:

$$M_b(z_k) = M .* W_B^T$$

$$\Gamma^N = \left(\sum_{k=1}^{n} M_b(z_k)^T \right)^{-1} . \tag{10}$$

Los pasos que sigue el Algoritmo Genético para obtener la óptima estructura RNAP se pueden resumir de la siguiente manera:

1. Se obtiene población inicial de forma aleatoria.
2. Se selecciona al mejor conjunto de individuos de la población inicial utilizando como función objetivo el error de aproximación de RNAP definido en (5).
3. Se obtiene una nueva población A_g utilizando los operadores de recombinación mutación propios de AG [2].
4. Se reevalúa la función objetivo utilizando el error de aproximación de RNAP
5. Se regresa al paso 3 hasta que se alcance el máximo número de generaciones hasta que alguno de los individuos extraídos del proceso de selección satisfaga desigualdad descrita en (7).

Dado que RNAP asigna dinámicamente el número de neuronas y de generaciones necesarias en el entrenamiento, los únicos parámetros manipulables exteriormente son:

- *Pot.* Representa la máxima potencia a alcanzar en la aproximación polinomial.
- *Eta.* Representa la dispersión del ruido presente en los datos de entrada.
- *Phiest.* Define la estructura de la matriz de aprendizaje.

3 Caracterización de las series e índice de dificultad de modelado

Se seleccionaron tres parámetros característicos de las series de tiempo determinar su dificultad de modelado: Reglas de Producción, Entropía de Shannon Exponente de Lyapunov, cuyo origen es, respectivamente: Teoría de la Computación, Teoría de la Información y Teoría de los Sistemas Dinámicos No Lineales.

A continuación se describen las características de dichos parámetros:

Entropía de Shannon. La Entropía representa la exactitud con la cual es posit predecir el comportamiento de un sistema dinámico, específicamente, la Entropía Shannon determina la cantidad de información que se puede obtener del estado de sistema a partir de cualquier medición que se realice [7].

Exponente de Lyapunov. El Exponente de Lyapunov es una medida de la divergenc de las trayectorias cercanas. Es equivalente a la tasa a la cual un siste unidimensional produce información, es decir, el Exponente de Lyapunov cuantif de forma precisa la exactitud del modelado para un punto futuro dado [7].

Reglas de Producción. A mayor número de Reglas de Producción se tiene una ma; complejidad computacional. Además, la tasa de crecimiento de las Reglas Producción obtenidas mediante análisis gramatical permite identificar regiones den

de las series de tiempo en donde se puede realizar un modelado más eficiente de las mismas [8].

Se seleccionaron un total de treinta series de tiempo las cuales se detallan en la Tabla 1 junto con las propiedades que las caracterizan [1].

Tabla 1. Clasificación y propiedades de dificultad de modelado de series de tiempo

Serie de Tiempo	Exponente de Lyapunov	Número de Reglas de Producción	Entropía de Shannon
Sine	0.517	27	3.7118
Vanderpol	1.864	28	4.981
Qperiodic2	0.925	66	2.445885
Qperiodic3	1.383	75	4.232
Mackey-Glass	1.481	75	0.15914973
Logistic	0.760	61	0.03522222
Lorenz	0.601	63	4.78931065
Rossler	1.049	72	2.45897942
Ikeda	1.452	71	0
Henon	2.301	69	0.0204499
Cantor	5.728	72	0
Tent	0.477	13	3.401
Primos	0.594	78	1.295
D1	1.288	92	0.8521749
A1	0.2234006	79	7.02
Ascii	2.709	80	2.107
El niño	2.322	83	3.526
HIV DNA	0.029	9	1
Human DNA	0.322	51	7.28
Lovaina	1.0693	77	5.574
Plasma	3.383	91	2.866
Laser	0.949	66	2.93549693
S&p500	3.569	82	2.585
Star	3.289	90	3.102
Down Jones	0.144	65	6.43519134
Kobe	1.102	79	3.74401017
ECG	0.242	53	3.51860944
EEG	1.199	82	0
Brownian Motion	2.043	85	2.74038235
White Noise	1.606	73	0

Para calcular la dificultad de modelado de una serie de tiempo es necesario, primero, introducir algunos conceptos utilizados en la definición de la misma.

Definición 1
El Índice de Exponente de Lyapunov de una serie de tiempo dentro de un conjunto de series es:

$$IEL = \frac{EL_i}{EL_{max}} . \quad (1$$

donde:
- EL_i es el Exponente de Lyapunov propio de la serie analizada.
- EL_{max} es el máximo Exponente de Lyapunov registrado dentro del conjunto de series de tiempo analizado.

Definición 2
El Índice de Reglas de Producción de una serie de tiempo dentro de un conjunto de series es:

$$IRP = \frac{RP_i}{RP_{max}} . \quad (12)$$

donde:
- RP_i es el número de Reglas de Producción propio de la serie analizada.
- RP_{max} es el máximo número de Reglas de Producción registrado dentro del conjunto de series de tiempo analizado.

Definición 3.
El Índice de Entropía de Shannon de una serie de tiempo dentro de un conjunto de series es:

$$IES = \frac{ES_{min} + 1}{ES_i + 1} . \quad (13)$$

donde:
- RP_i es el número de Reglas de Producción propio de la serie analizada.
- RP_{max} es el máximo número de Reglas de Producción registrado dentro del conjunto de series de tiempo analizado.

Después de definir estos conceptos el Índice de Dificultad de Modelado de una serie de tiempo dentro de un conjunto de series esta dado por la expresión:

$$IDM = IDL + IRP + IES . \quad (14)$$

4 Metodología de evaluación de la técnica

La metodología de evaluación de la técnica consistió en la realización de entre uno y diez experimentos por serie de tiempo buscando el resultado más acertado mediante variación de los parámetros de la red neuronal Pot, Eta y Phiest. En cada experimento se realizo el entrenamiento de la red con entre 950 y 1200 valores obteniendo un

modelado de los siguientes cincuenta valores de la serie. Los valores de entrada de cada serie de tiempo fueron normalizados antes de ser ingresados a la red neuronal con el objetivo de poder comparar el error obtenido para las diferentes series de tiempo utilizadas.

Para la evaluación del éxito del modelado de la serie de tiempo por parte de la técnica se utilizaron dos métodos. El primero consistió en la comparación de los datos originales con los obtenidos en el modelado mediante la graficación de ambos. El segundo método solo se aplicó a los resultados exitosos y consistió en el cálculo del error medio raíz cuadrada (RMSE) con los resultados del modelado y los valores originales de la serie. El RMSE esta definido por la expresión:

$$RMSE(x) = \left[\frac{1}{N} \sum_{i=1}^{N} \left(x_i^m - x_i^o \right)^2 \right]^{\frac{1}{2}} . \tag{15}$$

donde x_i^o representa el i-ésimo valor de la serie de tiempo evaluada y x_i^m representa el i-ésimo valor del modelado efectuado por la técnica.

Se eligió la utilización de esta medida de error debido a su propiedad de ser particularmente sensible a las diferencias grandes entre el valor original y el valor modelado [9].

5 Resultados

A continuación se presentan tres ejemplos que ilustran el funcionamiento de RNAP.

La figura 1 muestra el resultado del modelado de la serie Lovaina. Obsérvese que las diferencias entre los valores originales y los del modelado son muy pequeñas.

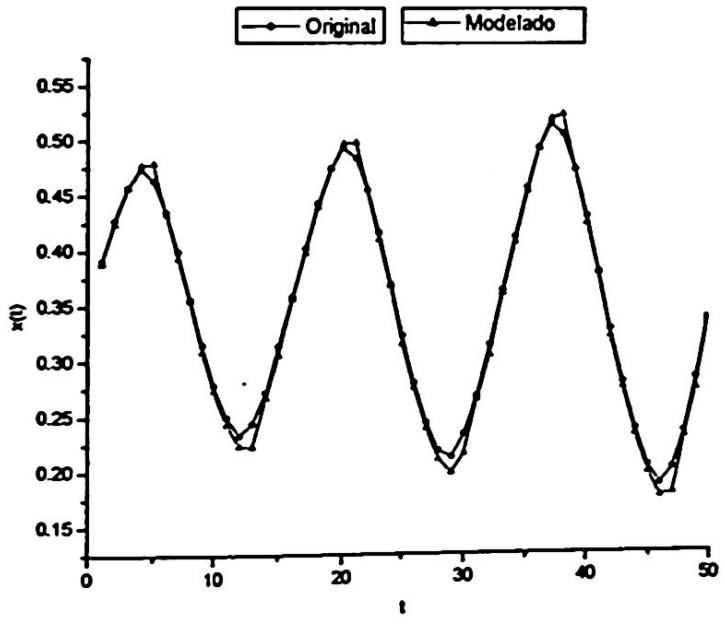


Fig. 1. Modelado y valores originales de la serie Lovaina

La figura 2 muestra el resultado del modelado de la serie Tent. Obsérvese que el modelado sigue la tendencia de la serie original presentando pequeñas discrepancias en las magnitudes de las crestas de la serie.

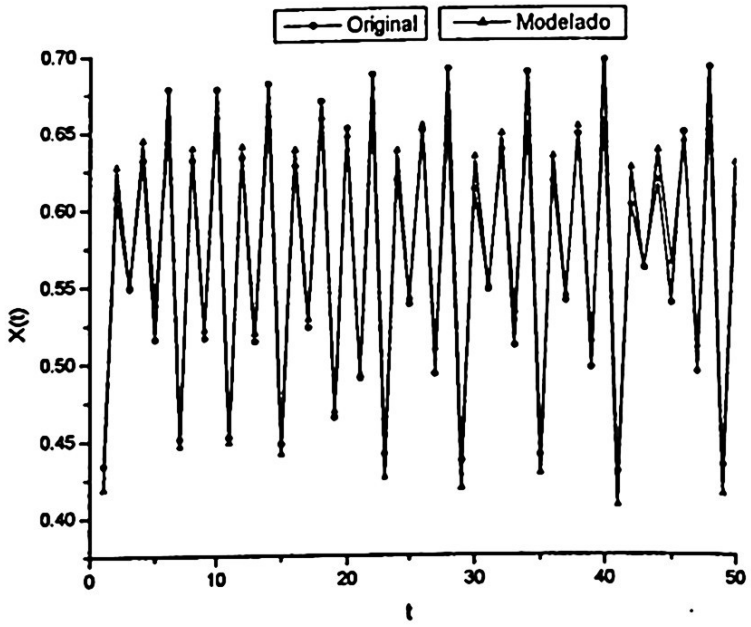


Fig. 2. Modelado y valores originales de la serie Tent

La figura 3 muestra el resultado del modelado de la serie Ikeda. Obsérvese como modelado sigue la tendencia de la serie original.

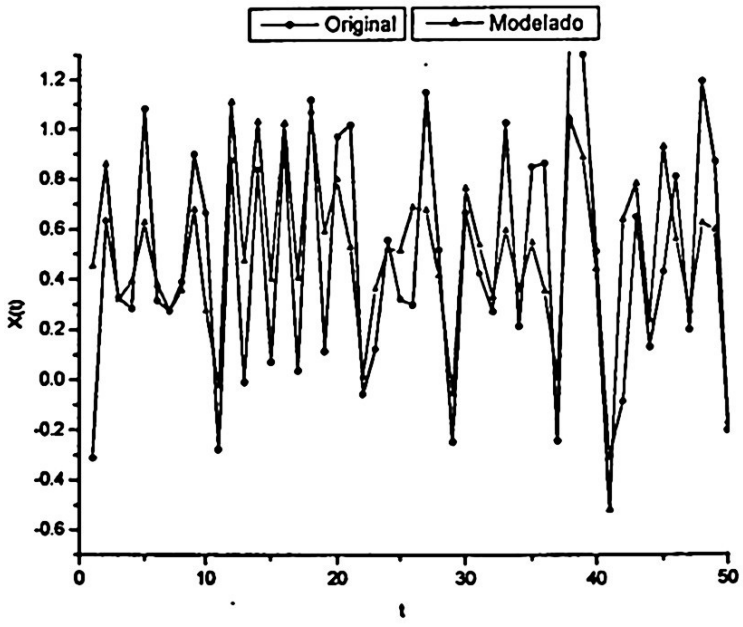
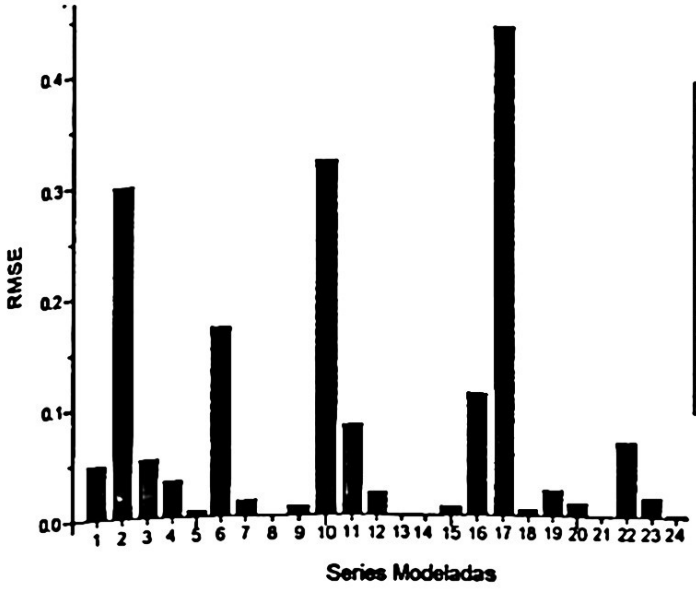


Fig. 3. Modelado y valores originales de la serie Ikeda

En la figura 4 se presentan los resultados de los 24 experimentos considerados exitosos después de ser evaluados con las dos metodologías descritas anteriormente. Se presentan los resultados en base al error medio raíz cuadrada obtenido.



Número	Serie	Número	Serie
1	A1 Santa Fe	13	Logistic
2	Brownian Motion	14	Lorenz
3	D1	15	Lovaina
4	Down Jones	16	Mackey
5	ECG	17	Plasma
6	EEG	18	QPeriodic 2
7	El niño	19	QPeriodic 3
8	Henon	20	Rossler
9	Human DNA	21	Sine
10	Ikeda	22	Star
11	Kobe	23	Tent
12	Laser	24	Vanderpol

Fig. 4. Resultados de modelado obtenidos en relación con el RMSE

En las figuras anteriores (1 a 3) se puede observar que los resultados de modelado de RNAP son muy acertados en la mayoría de los casos: 24 de 30 series fueron modeladas obteniendo un RMSE comparable y en algunos casos más bajo que el obtenido en experimentos similares reportados en la literatura utilizando otras técnicas [1].

Además, se puede notar que el modelado hecho por RNAP sigue de forma bastante fiel la tendencia de la serie, aún en los casos donde el modelado no es del todo acertado debido a que no se alcanzan las amplitudes de la serie original (tal es el caso de la serie Ikeda, ver Figura 3). Este dato es importante ya que existen múltiples aplicaciones en las cuales el conocer la tendencia del comportamiento del sistema es suficiente para poder obtener información relevante.

En la tabla 2 se comparan las treinta series de tiempo y el resultado obtenido en el proceso de modelado. Las series están ordenadas en forma descendente en base a su Índice de Dificultad de Modelado.

Analizando la Tabla 2 se puede observar que series de tiempo con una alta dificultad de modelado de acuerdo con sus propiedades características, tales como Henon, ECG e Ikeda pudieron ser modeladas exitosamente por RNAP.

Finalmente, se presenta una tabla comparativa de los resultados de modelado de RNAP y otros métodos con una capacidad de modelado alta de los cuales existen reportes en la literatura [1].

En la Tabla 4 se describe la naturaleza de cada una de las técnicas de modelado que se han reportado en la literatura [1].

Se observa de la tabla 3 que solo Anfis y Sugimay presentan resultados comparables con aquellos obtenidos por RNAP y que no se da el caso en el que alguna de las técnicas estudiadas presentara resultados exitosos con series que RNAP no logra modelar.

Tabla 2. Resultados obtenidos en base a la dificultad de modelado. M = Modelada, NM = No Modelada

Serie	Resultado obtenido	IDM
Cantor	NM	2.791
Henon	M	2.139
EEG	M	2.110
White Noise	NM	2.082
Ikeda	M	2.033
Mackey-Glass	M	1.945
Plasma	M	1.849
Star	M	1.806
S&p500	NM	1.803
D1	M	1.775
Logistic	M	1.768
Ascii	NM	1.673
Brownian Motion	M	1.558
El niño	M	1.538
Primos	NM	1.396
Kobe	M	1.271
Rossler	M	1.263
QPeriodic3	M	1.256
Lovaina	M	1.184
QPeriodic2	M	1.176
Laser	M	1.145
A1 Santa Fe	M	1.031
Lorenz	M	0.969
Down Jones	M	0.873
ECG	M	0.845
Vanderpol	M	0.800
Human DNA	M	0.737
HIV DNA	NM	0.603
Sine	M	0.599
Tent	M	0.453

Tabla 3. Comparación de varias técnicas con RNAP. M = Modelada, NM = No Modelada

Técnica de modelado / Serie de tiempo	RNAP	Pittnet-FFNBP	Pittnet-RBFNBP	Matlab-Anfis	Tisean-rbf	Sugi-may	Npredict-ARMA
Sine	M	M	M	M	M	M	M
Qperiodic 2	M	M	M	M	M	M	M
Mackey – Glass	M	M	M	M	M	M	M
Logistic	M	M	M	M	*NM*	M	*NM*
Lorenz	M	M	M	M	*NM*	M	M
Rossler	M	M	M	M	*NM*	M	M
Ikeda	M	*NM*	*NM*	M	*NM*	*NM*	*NM*
Henon	M	M	M	*NM*	*NM*	M	*NM*
D1	M	M	M	M	*NM*	M	*NM*
Laser	M	*NM*	*NM*	M	M	M	M
Dow Jones	M	M	*NM*	M	*NM*	M	M
Kobe	M	*NM*	*NM*	M	*NM*	M	M
ECG	M	M	M	M	*NM*	M	M
EEG	M	M	*NM*	*NM*	*NM*	M	M
Brownian motion	M	M	M	M	*NM*	M	M
White Noise	*NM*	*NM*	*NM*	*NM*	*NM*	*NM*	*NM*

Tabla 4. Técnicas de modelado analizadas

Técnica de Modelado	Tipo de Técnica
RNAP	Híbrido Red Neuronal con Algoritmo Genético.
Pittnet-FFNBP	Red Neuronal
Pittnet-RBFNP	Red Neuronal
Matlab-Anfis	Híbrido Red Neuronal con Lógica Difusa
Tisean-rbf	Teoría de los Sistemas Dinámicos no Lineales.
Sugimay	Teoría de los Sistemas Dinámicos no Lineales.
Npredict-ARMA	ARMA optimizada con Red Neuronal.

6 Conclusiones

Del análisis de los experimentos aquí reportados se puede concluir que RNAP presenta una capacidad de modelado comparable e incluso superior a la de técnicas como Anfis y Sugimay. Lo anterior tiene como sustento el hecho de que RNAP presenta resultados exitosos con series de tiempo con una dificultad de modelado alta,

algunos de los cuales no son alcanzados por las técnicas reportadas en otros trabajos. El empleo de algoritmos híbridos en la implementación de técnicas de modelado series de tiempo, como es el caso de RNAP y Anfis, han demostrado dar muy buenc resultados, por lo que se vislumbra que con éste tipo de implementación se pueda desarrollar técnicas eficientes para la solución de problemas de modelado, control entre otros. Por último, este trabajo muestra la importancia de contar metodologías estandarizadas para la comparación de diferentes técnicas de modelado

Referencias

1. E. Bautista-Thompson & J. Figueroa–Nazuno (2002). "Matriz de Conocimiento Sobre Complejidad de Predicción de Series de Tiempo". VII Congreso Iberoamericano Reconocimiento de Patrones. Ciudad de México.
2. E. Gómez-Ramírez (1999). "Adaptación de la Arquitectura de Redes Neuronal Artificiales y su Aplicación al Control Adaptable". Tesis doctoral. Ciudad de México.
3. A. González-Yunes, M. Ávila-Álvarez, E. Gómez–Ramírez, O. Mullet, F. Mazzanti, Vilasis–Cardona (2000). "Redes Neuronales para Identificación y Predicción de Series Tiempo". Revista Centro de Investigación, Universidad la Salle. Ciudad de México.
4. B. Martín del Brio & A. Sanz-Molina (2002). "Redes Neuronales y Sistemas Difusos. Alfaomega–RA-MA.
5. E. Gómez-Ramírez, A. S. Posnyak & R. Lozano (2000). "Control Adaptable Utilizan Redes Neuronales Artificiales Polinomiales". Revista Centro de Investigació Universidad la Salle. Ciudad de México.
6. R. Urbach. (2000). "Footprints of Chaos in the Markets". Prentince–Hall.
7. R. Menchaca-Mendez, C. Sanchez-Rodríguez & J. Figueroa-Nazuno. "Predicción Series de Tiempo Mediante Análisis Gramatical". XLIII Congreso Nacional de Fisi Puebla, Puebla.
8. Diebold, F. X. (2001). "Elements of Forecasting". South-Western Thomson Learning.

Simulación computacional de la actividad eléctrica cerebral

Lizette P. Castro, L. A. Torres

Universidad Iberoamericana León
Blvd. Jorge Vértiz, 37000 León,Gto., México
lizettepaulina@yahoo.com

Resumen. El cerebro humano es un órgano donde se puede hacer un modelado de la actividad eléctrica cortical registrada en la superficie del cráneo (EEG); esto puede contribuir a caracterizar el funcionamiento normal y distinguirlo de la anormalidad. Asimismo, permite una mejor comprensión de los fenómenos eléctricos que subyacen a procesos tan importantes como el pensamiento, la memoria y el aprendizaje. Se realizaron estudios comparativos entre el EEG registrado convencionalmente y un modelo físico basado en la ecuación de un potencial generado por un dipolo eléctrico, a través de una simulación computacional utilizando la técnica de Monte Carlo. La simulación permite generar registros de ondas de baja frecuencia, similares a la actividad alfa fisiológica del EEG y estimar la localización de posibles fuentes de actividad anormal.

1 Introducción

La modelación es una herramienta que tiene gran utilidad para comprender el funcionamiento de los sistemas vivos. El cerebro humano puede considerarse como un sistema biológico complejo que puede ser modelado a través de técnicas computacionales. En este trabajo se presenta la simulación de registros electroencefalográficos (EEG), generados por un sistema de Monte Carlo acoplado para la actividad eléctrica de una región del cerebro.

El modelo utilizado está basado en un potencial (V) generado por un dipolo subcortical, registrado en el cuero cabelludo. En donde la cabeza humana es modelada por tres esferas concéntricas [1,2]. Esta simplificación preserva elementos importantes de la cabeza, con el propósito de que el modelo se asemeje en forma y con determinados parámetros a la cabeza real [1,3]. En combinación con la simulación del potencial de un dipolo profundo, se considera también a la cabeza como un circuito simple resistor-capacitor (RC), que produzca un EEG similar a los obtenidos en registros de pacientes normales. Para esto se propone el circuito RC simulado en donde la entrada de voltaje sea el potencial generado por el dipolo [4].

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 193–197, 2003. © Centro de Investigación en Computación, IPN, México.

El primer requerimiento para un modelo simple como el que se presenta, es que la salida de la actividad eléctrica de la simulación Monte Carlo se aproxime a la actividad eléctrica de un registro normal de EEG.

2 Antecedentes

La trascendencia de estudiar al cerebro humano mediante la interpretación de los registros de EEG nos obliga a considerar la simulación como una parte potencialmente importante para el diagnóstico, en el trabajo diario del especialista ó investigador en el área de neurociencias y una oportunidad para la interacción entre ingenieros en sistemas computacionales. El explicarse cuantitativamente el origen de la actividad eléctrica, así como la posible correlación entre la forma de onda de los registros EEG y algún trastorno en el funcionamiento cerebral, ha sido uno de los motivos para indagar en el estudio del cerebro mediante modelos físicos y matemáticos en conjunto con las herramientas computacionales.

En la actualidad los distintos métodos para localizar las diferentes fuentes de potencial en el cerebro difieren entre sí por los modelos que se usan para representar a la cabeza, las formulas usadas en el cálculo del potencial en el cuero cabelludo, la forma en que la referencia del electrodo es tratada y, sobre todo, los métodos computacionales aplicados a las neurociencias [5].

Los resultados obtenidos del Monte Carlo en este trabajo están basados en un modelo de tres capas esféricas [1,2] y un circuito RC simulado [4]. El potencial sobre cualquier punto del cuero cabelludo esta basado en una ecuación que proporciona el potencial [1]. En otros trabajos se han propuesto modelos de circuito RC [4] para obtener las señales, sin embargo la relevancia de éste, consiste en mostrar resultados en donde el voltaje de entrada al circuito RC sea considerado el mismo potencial simulado.

El primer requerimiento para la construcción del modelo es que la salida del circuito RC produzca una salida similar a la del EEG que se produce en un paciente normal. El planteamiento de general puede ser formulado de la siguiente forma: dada una caja negra con dos salidas, ¿qué circuito podría dar una salida que produzca un voltaje con formas de onda similares a las del EEG, de tal manera que se involucren parámetros de la cabeza real de una persona, como son los radios del cerebro, cráneo, cuero cabelludo y sus resistividades respectivas.

3 Metodología

La cabeza es modelada por tres esferas concéntricas [2]. La esfera interna representa al cerebro, la intermedia al cráneo y la externa al cuero cabelludo (fig. 1). Para grarlo se generó el voltaje mediante el modelo de las 3 esferas concéntricas. Una vez generado, se le hizo pasar como voltaje de entrada a través del circuito RC mulado.

En este modelo el radio del cerebro y del cráneo fue de 8 y 8.5 cm respectivamente, mientras que el de la cabeza fue de 9.2 cm [2]. Para el cerebro y el cuero cabelludo la resistividad se consideró de 2.22 Ohms y la del cráneo de 177 Ohms.

Estos valores numéricos indican un valor típico fisiológico promedio. Los radios fueron normalizados al radio de la cabeza, los resultados finales fueron a = 0.87, b = 0.92 y c = 1.0. La conductividad ya normalizada a la del cráneo fue C1 = C3 = 1.0 y C2= 0.0125

El modelo de tres capas esféricas, que se aplicó en el Monte Carlo, proporciona el potencial (V) generado por el dipolo en el cuero cabelludo. Está dado por la siguiente ecuación [1].

$$V = \frac{1}{4\prod SR^2} \sum_{n=1}^{\infty} \frac{X(2n+1)^3}{d_n (n+1)n} b^{n-1} \left[nm_r p_n(\cos\alpha) + m_t P_n'(\cos\alpha)\cos\beta \right]$$

donde:

R es el radio externo de la cabeza (cuero cabelludo).
S es la conductividad del cerebro.
X es la conductividad del cráneo / la conductividad del cerebro.
b es la distancia del dipolo al centro /R.
• es el ángulo polar sobre un punto en la superficie.
• es el ángulo azimutal.
r1 y r2 son el radio del cerebro y del cráneo respectivamente
P_n son los polinomios asociados de Legendre.

Posteriormente a la generación de los potenciales, éstos fueron introducidos en las ecuaciones básicas del circuito RC simulado. Los voltajes de salida del circuito fueron registrados de tal manera que los rangos de frecuencia pudieran ser modificados.

4 Resultados

Se obtuvieron registros EEG_t al hacer distintas corridas de 1000 eventos cada una, inicialmente sin hacer pasar al potencial por el Monte Carlo que simula el circuito RC. Después se muestran resultados obtenidos al introducir la señal de potencial en el circuito simulado (fig. 2).

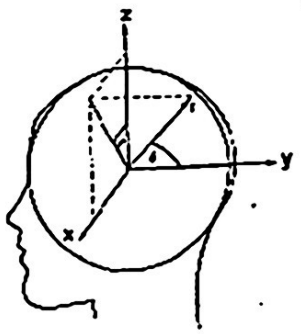


Fig 1. Modelo del cerebro basado en 3 esferas concéntricas

Una vez que se obtienen tanto los registros del EEG real (fig. 3) como los simulados del EEG_s se hace el análisis estadístico que consiste en la comparación entre las señales reales y las simuladas (EEG y EEG_s). Esto se hizo mediante un ajuste por mínimos cuadrados; el cual permitió seleccionar la mejor curva simulada [6,7]. La señal equivalente o la más aproximada fue aquella en donde la suma es mínima. La siguiente ecuación se utilizó para efectuar el cálculo de la comparación:

$$\text{Suma} = \sum_{i=1}^{n} (v(i)_{cal} - v(i)_{obs})^2$$

donde:
$V(i)_{cal}$->son los potenciales generados por el Monte Carlo (EEG_s)
$V(i)_{obs}$-> son los potenciales proporcionados por el registro real de un paciente normal.

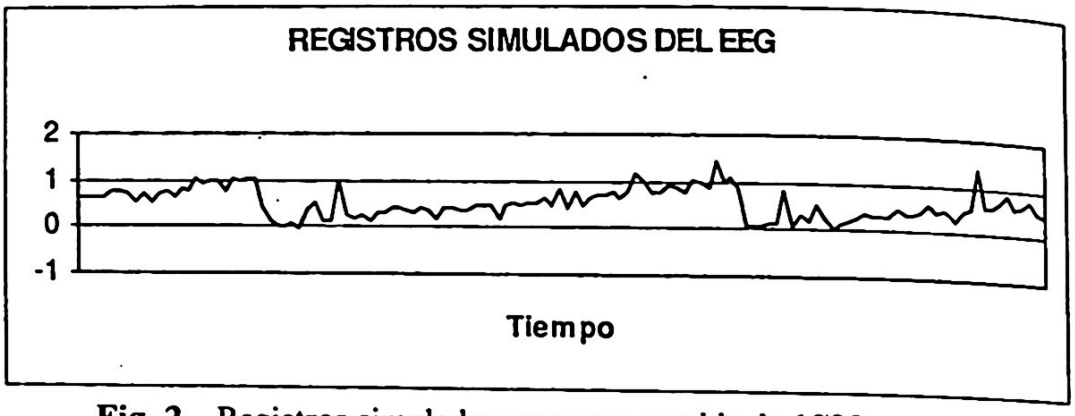


Fig. 2. Registros simulados para una corrida de 1000 eventos

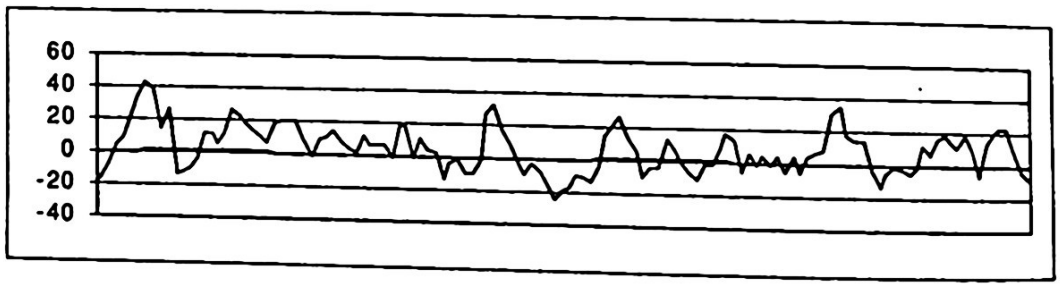


Fig. 3. Registros reales obtenidos de la actividad eléctrica de un paciente normal

Los gráficos de los registros tienen una duración de dos segundos y sus voltajes que se proporcionan van desde –29.9 hasta 35.9 microvolts.

5 Conclusiones

Las señales que se obtuvieron mediante el modelo de acoplamiento permiten generar actividad eléctrica, la cual se asemeja en algunos intervalos a las frecuencias que se registran en un EEG real. Se pueden detectar tanto ondas lentas como rápidas cuyas frecuencias varían entre 4 y 13 Hz.

El simulador permitirá manipular distintos parámetros, así cómo complicar la configuración del circuito para intentar la simulación de registros EEG de algunos trastornos que causan paroxismos como la epilepsia.

Referencias

[1] *Salu* Y, Cohen L., Rose D., Kuffa C., Hallet M. *An improved method for localizing electric Brain Dipoles.*, IEEE Transactions on Biomedical Engineering, Vol 37, No. 7, pp. 699-705, July 1990.

[2] Yao Dezhong. *High-resolution EEG mapping: a spherical harmonic spectra theory and simulation results.* Cli-nical Neurophisiology 111, pp. 81-92, Agosto 2000.

[3] Niderhauser Joel. *Simple electric model for epilepsy.* IEEE potentials, pp. 35-39, February 2002.

[4] Eshel Y.,Witman S., Rosenfeld M., Abboud S. *Correlation between skull thickness asymetry and scalp potential estimated by a numerical model of the Head.* IEEE Transactions of Biomedical Engineering, Vol 42, No. 3, pp. 242-249, March 1995.

[5] Lyle Borge-Graham. *Facilitating the science in computational neuroscience.* Nature neuroscience, Sup p 1191, Vol 3, November 2000.

[6] Rosenfeld M., Tanami R., Abboud S. *Numerical solution of the potential due to dipole sources in volume conductors with arbitrary geometry and conductivity.* IEEE tranasactions on biomedical engineering, pp. 679, vol. 43, No. 7, July 1996.

[7] Pardey J., Roberts S. *A review of parametric modelling techniques for EEG analysis.* Med. Eng. Physics, Vol 18, 2-11, January 1996.

Arquitectura hardware para el seguimiento de objetos utilizando procesamiento piramidal

Marco Aurelio Nuño Maganda, Miguel Arias Estrada

Instituto Nacional de Astrofísica, Óptica y Electrónica,
Luis Enrique Erro # 1, Santa María Tonanzintla, Puebla, México.
nmaganda@cseg.inaoep.mx , ariasm@inaoep.mx

Resumen. En este trabajo se presenta una arquitectura hardware para el seguimiento de objetos. La ventaja de esta arquitectura con respecto a otras es el uso de pirámides para hacer el seguimiento. La arquitectura se divide en dos grandes bloques: la arquitectura que obtiene la pirámide de la imagen y la arquitectura que efectúa el proceso de correlación de la imagen con un patrón dado, utilizando operaciones basadas en pirámides. Se presentan resultados para ambos bloques de la arquitectura y se bosqueja el funcionamiento de ambos bloques interactuando uno con el otro.

1 Introducción

La visión por computadora es una de las principales ramas de la Inteligencia Artificial que se explora actualmente[1]. Las aplicaciones de visión obtienen un conjunto de imágenes de una determinada fuente, las procesan y arrojan resultados en forma de imágenes o de datos que después son analizados dependiendo de la aplicación.

A lo largo del tiempo se han utilizado varios dispositivos para el procesamiento digital de imágenes. Uno de ellos es la PC, la cual comenzó a ser utilizada como elemento de procesamiento cuando se comenzaron a producir en serie y su costo se redujo. Una de las desventajas de la PC es la forma secuencial de realizar el procesamiento. Se ha contemplado el uso de supercomputadoras o estaciones de trabajo dedicadas, pero éstas tienen un costo muy elevado con respecto a las PCs. Otra de las alternativas es el uso de plataformas de desarrollo con FPGAs (Field Progammable Gate Arrays) como elemento central de procesamiento. Los FPGAs son dispositivos digitales programables que a diferencia de los microprocesadores, no se programa en ellos un conjunto de instrucciones, sino la descripción de una arquitectura en función de componentes básicos [2]. Posteriormente, a través de las herramientas proporcionadas por los fabricantes, se implementa esa arquitectura mapeando la descripción de los componentes básicos en los recursos disponibles en el FPGA. La potencia de cómputo de los FPGAs se mide en el número de bloques lógicos o en el número de compuertas equivalentes contenidas en él. El desarrollo de FPGAs con mayor cantidad de recursos cada vez nos permite la implementación de sistemas complejos de visión en tiempo real. En la actualidad existen dispositivos FPGA con capacidad equivalente a millones de compuertas digitales.

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 198–208, 2003. © Centro de Investigación en Computación, IPN, México.

Por otro lado, el seguimiento de objetos tiene múltiples aplicaciones en diversos sectores, como son la industria, en robótica y en aplicaciones militares. Se han explorado diversas técnicas de seguimiento de objetos, algunas muy robustas y otras dedicadas a determinadas características. Se han intentado varios tipos de seguimiento, como pueden ser por extracción de esquinas, por extracción de bordes, por modelado 3D de objetos, por segmentación, por umbralización, por fusión de sensores y por correlación. Todos los tipos tienen sus ventajas y desventajas y también son dependientes de la aplicación del seguimiento en sí.

Una de las representaciones de imágenes utilizada para el procesamiento de video es la pirámide multiresolución, que separa la imagen original de una resolución dada en un conjunto de imágenes de diferentes resoluciones espaciales [3].

2 Principios

La potencia de la pirámide consiste en que las operaciones basadas en ella son mucho más rápidas que las mismas operaciones sobre la imagen original [3]. Esta simplificación se debe a que se trabaja con resoluciones más burdas, en donde hay menos píxeles que deben ser procesados. Cada nivel es ¼ más pequeño que el nivel precedente, resultando en una reducción en el procesamiento de un factor de 16, 64, 256 y así sucesivamente.

Muchos algoritmos que trabajan a nivel piramidal son llamados algoritmos burdos-a-finos (coarse-to-fine algorithms). Estos algoritmos procesan la imagen en una resolución muy burda (muy baja) y obtienen un primer resultado del procesamiento, el cual es impreciso, pero da una noción del resultado que se desea obtener. Los resultados se refinan repitiendo el procesamiento a resoluciones mayores, utilizando como referencia los resultados obtenidos en resoluciones más bajas.

2.1 Pirámide Gaussiana

Para generar una pirámide gaussiana, se parte de la suposición de que la imagen está representada por el arreglo G_0 de F Filas por C columnas. Cada valor representa la intensidad de luz del punto correspondiente de la imagen. Esta imagen es el nivel 0 de la pirámide gaussiana. El nivel 1 de la pirámide contiene una imagen G_1, la cual es reducida a partir de una versión de la imagen G_0 a la que se le aplicó un filtro pasa-bajas [4]. Cada valor dentro del nivel 1 es calculado como un promedio ponderado de los valores en el nivel 0 dentro de una ventana de 5x5. Cada valor del nivel 2, representado por G_2, es obtenido a partir de los valores del nivel 1 aplicando la misma ventana.

El proceso de promediación ponderada de nivel a nivel está implementado por la función *Reduce:*

$$G_k = \text{Reduce}\,(G_k\text{-}1) \tag{1}$$

Lo cual significa que para los niveles $0 \le l \le N$ y nodos i, j, $0 \le i \le F$, $0 \le j \le C$

$$G_l(i,j) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} w(m,n) G_{l-1}(2i+m, 2j+n). \tag{2}$$

donde N se refiere al número de niveles de la pirámide mientras que F_i y C_i son dimensiones del i-ésimo nivel. La densidad de los nodos es reducida por la mitad una dimensión o por cuatro en dos dimensiones de nivel a nivel El patrón de pesos debe cumplir con las siguientes propiedades:

a) Por lo general, se utilizan máscaras de tamaño 5 x 5, aunque no hay restricción con respecto al tamaño de la máscara.

b) Debe ser separable

$$w(m,n) = \hat{w}(m)\hat{w}(n)$$

c) Los componentes del patrón deben estar normalizados

$$\sum_{m=-2}^{2} \hat{w}(m) = 1$$

d) Además, estos componentes deben ser simétricos

$$\hat{w}(i) = \hat{w}(-1) \quad para\ i = 0,1,2$$

En la Fig. 1 se muestra el resultado de obtener la pirámide gausiana de una imagen. En la tabla 1 se muestra el número de filas y columnas que tendrá cada una de imágenes de la pirámide, partiendo de una imagen con 480 filas por 640 columnas.

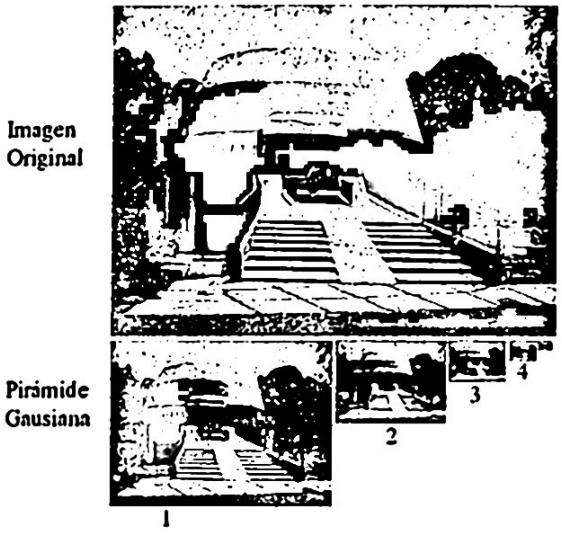


Fig. 1. Pirámide Gaussiana de la figura "INAOE"

Tabla 1. Número de filas y columnas de la pirámide de la imagen "INAOE"

Nivel	Filas	Columnas
0	480	640
1	240	320
2	120	160
3	60	80

2.2 Medidas de correlación

Se han explorado diversas medidas de similitud basadas en correlación. Las diferentes medidas exploradas durante el desarrollo de este trabajo se mencionan a continuación:

1) Suma de Diferencias Absolutas (SDA) [5]

$$SDA=\sum_{i,j}\left|I_1(x+i,y+j)-I_2(x+i+d,y+j)\right| \tag{6}$$

2) Suma de Diferencias al Cuadrado (SDC) [5]

$$SDC=\sum_{ij}\left(I_1(x+i,y+j)-I_2(x+i+d,y+j)\right)^2 \tag{7}$$

3) Correlación Cruzada Normalizada (CCN) [6]

$$CCN=\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\frac{fr(i,j)fp(i,j)}{\sqrt{fr(i,j)^2}\sqrt{fp(i,j)^2}} \tag{8}$$

3 Sistema propuesto

Se propone una arquitectura hardware basada en un FPGA de seguimiento de objetos. Esta arquitectura se divide en dos grandes bloques, la arquitectura que obtiene la pirámide gaussiana de una imagen y la arquitectura para la correlación de objetos basándose en los resultados arrojados por la primera.

3.1 Arquitectura para la obtención de pirámides

Se diseñaron dos variantes de la arquitectura. Estas variantes dependen del número de píxeles almacenados por dirección de memoria. Se supone el uso de imágenes en niveles de gris, donde un píxel ocupa 8 bits (1 byte). Entonces, si tenemos una memoria de 16 bits, se pueden codificar 2 píxeles por localidad. En el caso de memorias de 32 bits, se pueden codificar 4 píxeles por localidad.

Los módulos básicos que integran la arquitectura son:

1. Memoria RAM. En esta memoria se almacena la imagen a procesar, así como las imágenes que son arrojadas como resultado del procesamiento. La arquitectura fue diseñada para ser flexible con cualquiera de estas dos restricciones:

   a) Se dispone sólo de una memoria principal, en donde además de almacenarse la imagen a procesar, también se almacenan los resultados del procesamiento.

   b) Se dispone de más de una memoria. En una de éstas se almacena la imagen a procesar y en otras se almacenarán los resultados del procesamiento.

Para esta arquitectura se tomó en cuenta la primera restricción. Para pasar a la segunda restricción, debe hacerse un cambio mínimo en el código sin necesidad alterar el funcionamiento de los otros módulos que integran la arquitectura.

2. Generador de Direcciones. Este módulo se encarga de proporcionar a la arquitectura la dirección RAM en donde se leerá el dato o la dirección RAM en donde se almacenará el resultado. El Generador de Direcciones recibe datos de los Generadores de Control, los cuales se encargan de determinar cuándo se tiene listo en banco de registros el contenido de una localidad de memoria de resultados. En caso de que no se necesite hacer escritura a memoria, el Generador de Direcciones genera la dirección de lectura del siguiente dato a procesar.

3. Procesadores de Convolución. Estos procesadores se agrupan en niveles de procesadores. Cada nivel puede tener 2 ó 3 procesadores, dependiendo del tamaño de palabra de las memorias (2 procesadores para memorias de 32 bits y 3 procesadores para memorias de 16 bits). La principal característica de los Procesadores de Convolución consiste en que entre más niveles de procesadores se declaren, arquitectura tendrá un mejor desempeño. El número mínimo de niveles que deben declararse es uno, mientras que el máximo dependerá de los recursos disponibles en el FPGA en donde se va a sintetizar el diseño.

4. Banco de Registros. En este Banco de Registros se almacenan los resultados parciales de cada uno de los Módulos de Convolución. El Generador de Direcciones se encarga de leer el contenido de los registros del Banco cuando alguno de módulos Generadores de Control le indica que es tiempo de escribir en la memoria de datos el resultado de las operaciones realizadas por los Módulos de Convolución. En otro caso, los resultados de los Módulos de Convolución seguirán siendo almacenados en el Banco de Registros.

En la Fig. 2 se muestran los componentes de la arquitectura piramidal, así como señales de control que entran a los bloques. Estas señales de control son generadas el generador de direcciones, el cual recibe señales de control de los generadores control y decide en qué momento hacer una operación de escritura y en qué momento hacer una operación de lectura a la memoria de datos.

3.2 Arquitectura de correlación

Los componentes principales de la arquitectura de correlación son:

1. Memoria de Imagen. Como se mencionó en la arquitectura piramidal, la imagen original se almacena en una memoria y el resultado de la pirámide puede estar almacenado en la misma memoria o en una memoria diferente. La Memoria Imagen es la región de memoria en donde se encuentra el nivel de la pirámide gausiana que será utilizado para hacer la correlación con el patrón a buscar.

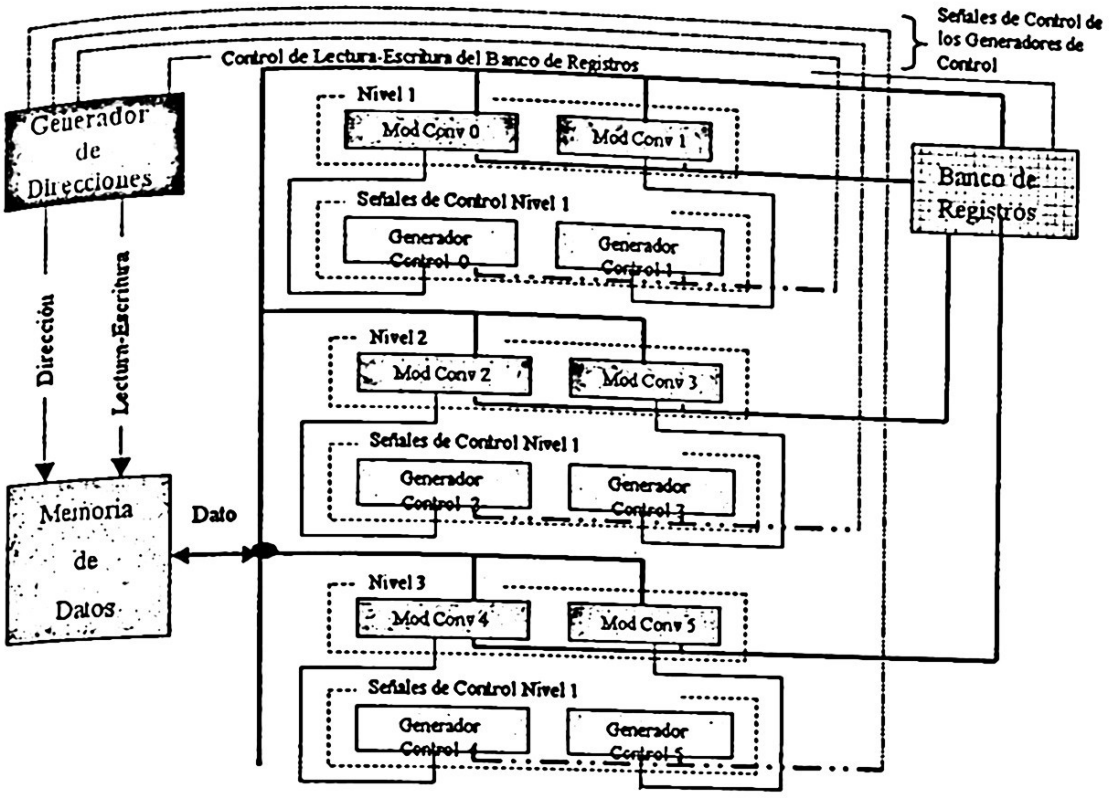


Fig. 2. Bloques Principales de la Arquitectura Piramidal

2. Memoria de Patrón. En el proceso de seguimiento de objetos, se debe buscar la similitud entre un patrón y una imagen. Entonces, la memoria patrón contendrá este patrón que se desea correlacionar. Si se cuenta con una sola memoria para realizar todo el procesamiento, esta memoria será una región diferente de la memoria principal en donde está almacenado el patrón.
3. Control de Correlación. Este se encarga de enviar las señales de control a las memorias de Datos y de Patrón, a los registros de Imagen y de Patrón y a los registros de Comparación Local y Global.
4. Registros de Imagen. Hay dos registros que se encargan de almacenar temporalmente dos palabras de memoria con la finalidad de obtener la correlación de 4 ventanas al mismo tiempo, ya que se desea realizar el menor número de lecturas repetidas como sea posible.
5. Registro de Patrón. Hay un registro que se encarga de leer de la imagen de patrón una palabra que se correlacionará con las 4 posibles ventanas de correlación.
6. Bloques de correlación (fcorr). Estos bloques se encargan de hacer el proceso de correlación de los valores de los registros de imagen y de patrón. Con esto, además de hacer un sistema altamente modular, podemos programar cualquier función de correlación deseada.
7. Comparador Local (ComparadorL). Una vez que se han terminado de procesar las cuatro ventanas de correlación, el control de correlación envía una señal a este comparador para obtener el mejor coeficiente de correlación de los cuatro que se

están procesando actualmente. Con esto, se tiene una idea de cuál de las ventanas es la que tiene mayor similitud con el patrón buscado.

8. Comparador Global (ComparadorG). Se encarga de comparar el contenido del Registro de Mejor Correlación con la salida del Comparador Local. En caso que el Mejor Local sea "mejor" (dependiendo de la medida de correlación) se actualizará el Registro de Mejor Correlación.
9. Registro de Mejor Correlación (MejorGlobal). En este registro se va almacenando el valor de mejor correlación de toda la imagen. Este registro servirá para saber cuánto cambió el grado de correlación con respecto al cuadro anterior.
10. Registros de Coordenadas. En estos registros se almacenan los valores donde va encontrando la mejor correlación. Al final del barrido de la imagen, estos registros contienen la fila y la columna de la mejor correlación, la cual sirve para desplegar el recuadro donde se encuentra tentativamente el objeto o para que en la siguiente iteración del proceso, se tomen esos resultados para acotar la búsqueda.

En la Fig. 3 se muestra la conexión entre todos los bloques que integran arquitectura de correlación, así como las principales señales de control y de datos.

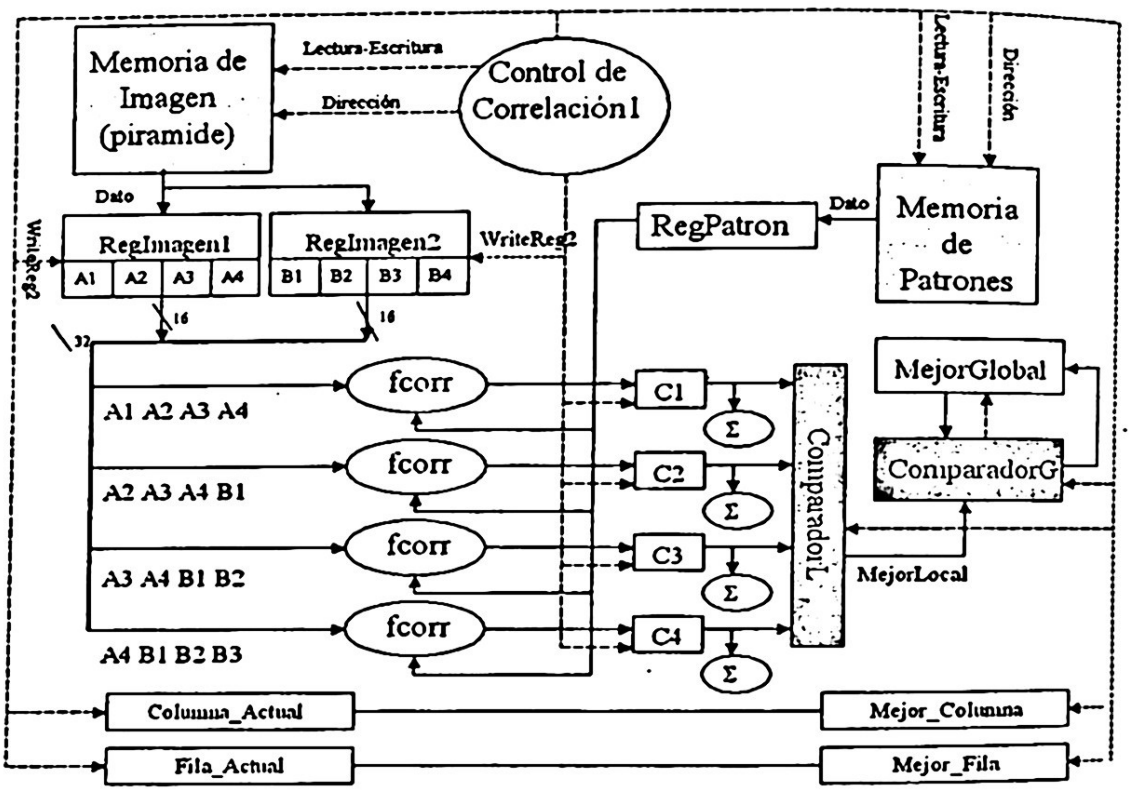


Fig. 3. Arquitectura de Correlación

3.3 Integración de las arquitecturas anteriores

En esta sección se describe brevemente la integración de las arquitecturas de pirámides y de correlación para dar paso a la arquitectura de seguimiento.

La arquitectura piramidal se encarga de obtener la pirámide de la imagen actual, y posteriormente pasa las direcciones donde se localizan las imágenes a la arquitectura

de correlación. La arquitectura de correlación se encarga de decidir si es el primer cuadro a procesar. En caso de ser el primer cuadro, toma las coordenadas de inicio de patrón (fila y columna) y hace una copia de la región correspondiente al patrón en una región de memoria diferente de donde se encuentra almacenada la pirámide. Con este proceso se obtiene la pirámide del patrón, que es una copia de los respectivos píxeles de las imágenes que componen la pirámide correspondientes al patrón a seguir. En caso de no ser el primer cuadro, entonces se hace la búsqueda del patrón almacenado en el cuadro anterior, iniciando por el nivel más alto de la pirámide. La búsqueda se divide en dos partes: la búsqueda hecha en el nivel más burdo (o de menor resolución), en donde la correlación del patrón buscado se lleva a cabo con toda la imagen. En las búsquelas hechas en niveles menos burdos, la búsqueda se realiza acotando el espacio de búsqueda de acuerdo con los resultados obtenidos en el nivel inferior. En la Fig. 4 se muestra de manera gráfica este proceso. Una vez que se ha llegado al nivel cero de la pirámide, se actualiza el patrón en todos los niveles, se envía una señal de término de proceso y se procesa la siguiente imagen.

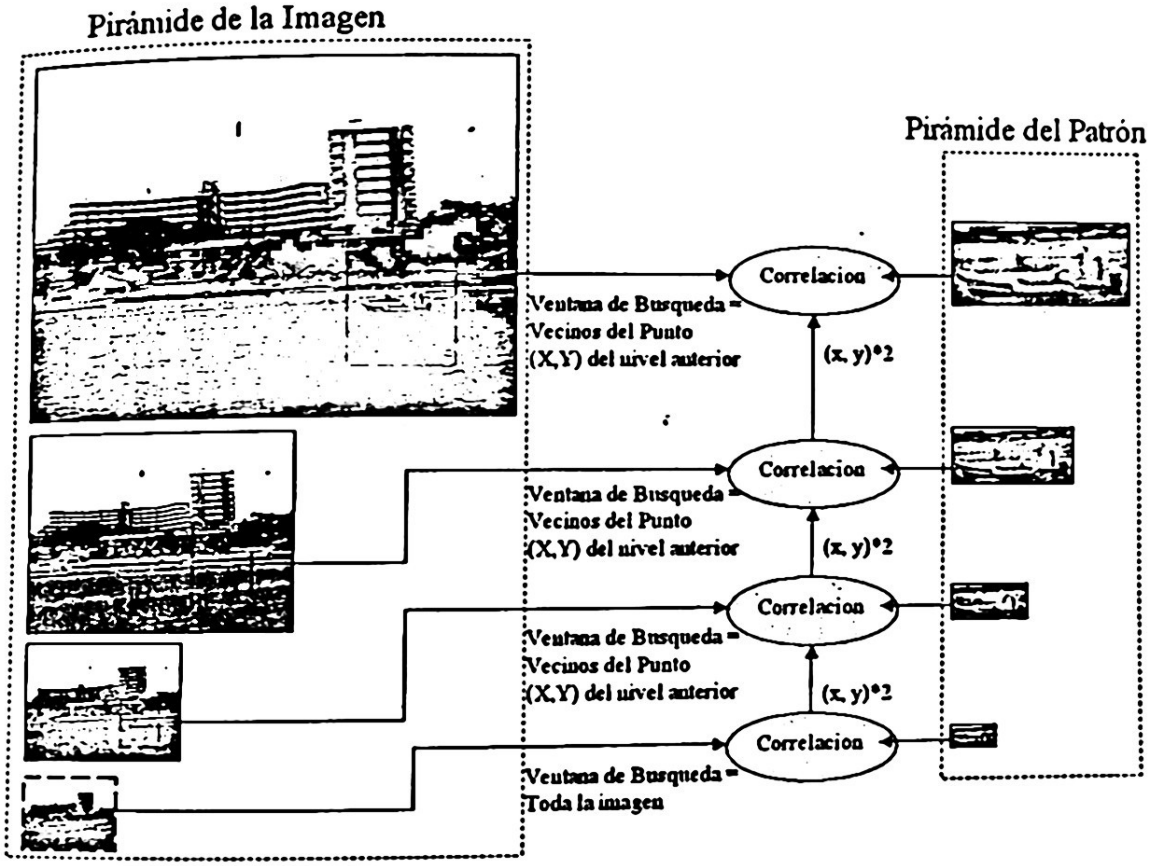


Fig. 4. Integración de las Arquitecturas Piramidal y de Correlación

4 Implementación y resultados

Los resultados presentados para la arquitectura de seguimiento fueron obtenidos por medio de simulaciones hardware. Handel-C es una herramienta innovadora que propone la reducción del tiempo de desarrollo de arquitecturas hardware, utilizando un lenguaje basado en el estándar ANSI-C. Las arquitecturas piramidal y de correlación

fueron codificadas en este lenguaje. Para la obtención de los resultados presentados, las imágenes utilizadas se convirtieron en archivos de texto, que se abrieron en Handel-C para pasarse a la arquitectura y posteriormente obtener archivos de texto donde se almacenaron los resultados del procesamiento. Una vez que se obtuvieron los resultados, se realizó una herramienta de desplegado de las secuencias (en MATLAB) que utilizará los datos arrojados por la arquitectura para desplegar la posición tentativa del objeto.

En las Figuras 5 y 6 se muestran los resultados para cuatro imágenes de dos diferentes secuencias. En la primera secuencia se observa que el seguimiento del objeto es bueno, en la segunda secuencia, se utiliza una secuencia un poco más difícil debido a que se introduce una serie de movimientos rotacionales que dificultan el seguimiento para algunos sistemas, pero que en el caso de esta arquitectura, los efectos de rotación sobre el seguimiento final son mínimos.

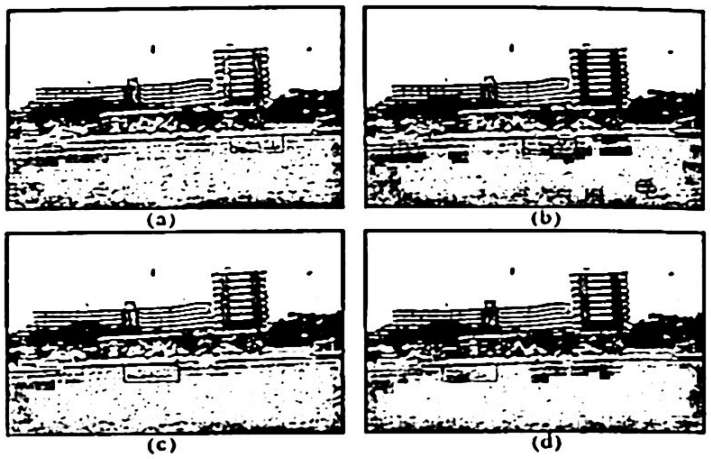


Fig. 5. Resultados del Seguimiento para la Secuencia "Marina"

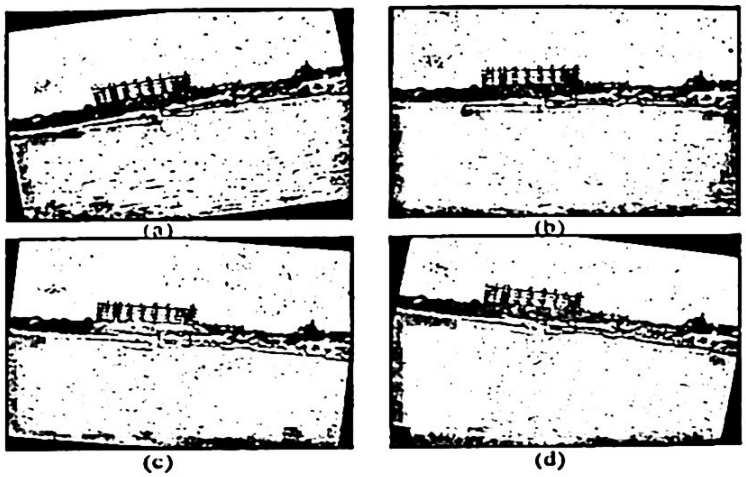


Fig. 6. Resultado del Seguimiento para la Secuencia "Casita Rotada"

En la Fig. 7 se muestra el resultado de la implementación de la arquitectura piramidal. Para su implementación se utilizó la tarjeta RC1000 de Celoxica, la cual se comunica con la PC por medio del bus PCI. La interfaz con la PC se programó en Visual C++.

Fig. 7. Arquitectura Piramidal en Funcionamiento

Se efectuaron varias pruebas a la arquitectura antes de su implementación. Una de ellas consistió en obtener el número de ciclos de reloj utilizados para procesar la imagen completa, y de esta forma obtener el tiempo de procesamiento de una imagen (suponiendo una frecuencia de reloj de 20 Mhz), que en este caso fue de 480 líneas por 512 píxeles. Los resultados obtenidos se resumen en la tabla 2.

Tabla 2. Rendimiento de la arquitectura en ciclos de reloj y en tiempo de procesamiento.

Número de Niveles de Procesadores en Paralelo	Rendimiento en Ciclos	Tiempo de Procesamiento (ms)	Imágenes por Segundo
1	262, 080	13 ms	76
2	181, 440	9.1 ms	109
3	154, 560	7.7 ms	129
4	141, 010	7.1 ms	141

Posteriormente se hizo la implementación, la cual fue realizada en la tarjeta RC1000 de Celoxica, que tiene como elemento central de procesamiento un FPGA Virtex (fabricado por Xilinx) con capacidad de 2 millones de compuertas equivalentes. Los recursos utilizados por la arquitectura para el procesamiento de pirámides se muestran a continuación:

Tabla 3. Recursos utilizados por la arquitectura del FPGA Virtex

Recurso del FPGA	Total Utilizado	Porcentaje Utilizado
Slices	6, 583 de 19, 200	34 %
4 Input LUT	8, 795 de 38, 400	22 %

Conclusiones

En este trabajo se presentó una arquitectura para el seguimiento de objetos que tiene como características principales la flexibilidad con respecto al grado de paralelismo que se desea implementar. El grado de paralelismo implementado dependerá de los

recursos disponibles en el FPGA utilizado para su implementación, de la velocidad que se desee alcanzar y de la aplicación que utilizará el seguimiento de objetos.

También se puede concluir que Handel-C es un lenguaje que acorta el tiempo de diseño e implementación de arquitecturas hardware complejas. Esto permite hacer un mayor número de pruebas y verificaciones antes de llegar a un diseño final que puede ser utilizado en aplicaciones donde se requiera menor espacio o menor consumo de potencia. Además, se pueden diseñar aplicaciones que tengan un alto rendimiento con la finalidad de alcanzar el desempeño en tiempo real.

6 Trabajo Futuro

Se ha validado la arquitectura de seguimiento a nivel simulación. Como trabajo futuro se hará la implementación definitiva en la Tarjeta RC1000 de Celoxica, además de hacer experimentos con otras aplicaciones de la pirámide gaussiana, como son la generación de mosaicos de imágenes, la estabilización de imágenes, entre otras. También queda como trabajo futuro optimizar el código para obtener un rendimiento superior, con la finalidad de implementar otros algoritmos que utilicen el seguimiento de objetos.

Referencias

1. Pajares, G., De La Cruz, J.: Visión por Computador. Editorial. Ra-ma., México, (1995)
2. Brown, S., Rose, Jonathan.: Architecture of FPGAs and CPLDs: A Tutorial., IEEE Design and Test of Computer, Vol. 13, No. 2, pp. 42-57, (1996)
3. Sarnoff Corporation.: Pyramid-Based Video Processing., http://www.sarnoff.com/government_professional/vision_technology/core_technology/pyramid_video_processing.asp, (2002)
4. Burt, P.J., Adelson E .H.: The Laplacian pyramid as a compact image code. IEEE Transactions on Communications. COM-31, pp 532-540, (1983)
5. Xicoténcatl P, Juan Manuel.: Arquitectura Hardware de Visión Estéreo en Tiempo Real. INAOE, México. (2000)
6. Peregrina A, Cesar Darío.: Seguimiento de Objetos por medio de Visión Activa. INAOE, Mexico. (2002)

La fusión de imágenes a nivel características para la detección y seguimiento de objetos

Janeth Cruz Enriquez, Leopoldo Altamirano Robles

[1] INAOE, Departamento de Ciencias Computacionales, Luis Enrique Erro No. 1, Sta. Ma. Tonantzintla, Puebla, 72840, México
jcruze@ccc.inaoep.mx, robles@inaoep.mx

Resumen. En este trabajo se presenta un algoritmo para la detección y seguimiento de un objeto mediante la fusión de datos proporcionados por dos sensores pasivos, un sensor infrarrojo y un sensor visible. La fusión se realiza a nivel características, las cuales se extraen de las imágenes de cada sensor para formar una imagen fusionada con las regiones resultantes que representan a los posibles objetos de interés sobre la escena. Las tres características locales seleccionadas se basan en la diferencia de contraste existente entre los objetos de interés y el fondo. El seguimiento del objeto en movimiento se mantiene mediante la correlación de un patrón de la imagen de muestra sobre los posibles objetos usando una ventana del tamaño del objeto, la cual se obtiene mediante la extracción adaptiva del fondo.

1 Introducción

Actualmente existe un gran número de algoritmos de detección y seguimiento de objetos. La mayoría de estos realizan la detección con los datos que provienen de un solo sensor. El uso de un solo sensor ocasiona desventajas que reducen la eficiencia del sistema, entre ellas: blancos falsos, susceptibilidad a contramedidas, incertidumbre que puede existir en los datos, deficiencia en cambios de escenarios, etc.

Para solucionar esta problemática se presentaron resultados a principios de los 80's con la creación de un nuevo modelo conocido como fusión de datos multisensor, el cual se relaciona con la asociación, correlación y combinación de datos e información de múltiples fuentes de información para obtener posiciones exactas y estimaciones de identidad, así como una completa valoración de situaciones y riesgos [4].

La fusión surge con la necesidad de llevar a cabo entre otras cosas la detección, identificación y seguimiento de blancos aéreos en el campo militar [9]. Muchas son las ventajas con el uso de múltiples sensores, donde cada sensor puede ser usado para complementar los datos de otros sensores, proveer una cobertura mayor y eficiente en la estimación del estado de blancos y decisiones [1]. Así como en el monitoreo cooperativo de blancos que no pueden ser visualizados por un simple sensor [2], [7].

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 209–219, 2003. © Centro de Investigación en Computación, IPN, México.

La desventaja de usar un sensor para realizar el seguimiento de objetos con una operatividad diurna-nocturna y la eliminación de blancos falsos fueron parte de motivación de este trabajo, en el cual se plantea la fusión adaptiva de los datos obtenidos de diferentes sensores, un sensor visible y un sensor infrarrojo, en la detección y seguimiento de objetos. El presente trabajo forma parte de un proyecto mas amplio, cuyo objetivo es la fusión adaptiva de sensores bajo una arquitectura fusión distribuida en aplicaciones de seguimiento de objetos.

Los sensores usados en este trabajo operan en diferentes rangos del espectro electromagnético pero ambos son sensores pasivos, es decir, estos sensores sólo reciben la radiación emitida por los objetos, a diferencia de los sensores activos, como radar que envía una señal y la recibe después de ser reflejada por el objeto. Un sensor visible capta la cantidad de luz que emiten o reflejan los objetos en la escena dentro del espectro visible y un sensor infrarrojo representa en una imagen la cantidad de calor que emiten los objetos en la escena.

Diversos trabajos se han realizado para la detección y seguimiento de objetos con estos sensores de forma individual. Sin embargo, los algoritmos de detección y guimiento de objetos con sensores infrarrojos deben ser más robustos al ruido comparados con los sensores visibles, no obstante, los sensores infrarrojos pueden trabajar sobre escenarios donde no existe iluminación, lo cual no es posible con un sensor visible.

La fusión de estos sensores también se ha explorado a nivel características, donde estas son seleccionadas dependiendo del contexto de trabajo y los objetos interés, como las basadas en textura [6], contraste [7], color y forma. Sin embargo, presentan diversos inconvenientes, entre otros la cantidad de características a fusionar o en la selección de la función adaptiva de ponderación, la cual determina valor de cada característica en el proceso de fusión.

El propósito de este trabajo es fusionar la información que proviene de los dos sensores seleccionados, para generar una imagen fusionada que contenga únicamente regiones que satisfagan las características del objeto de interés. La detección basa en la fusión de tres características locales (diferencia de medias, gradiente promedio, varianza local) de las imágenes que provienen de ambos sensores, usando una ventana que corresponde al tamaño de los objetos de interés, el tamaño de esta ventana se determina usando un algoritmo robusto de extracción adaptiva del fondo. Después del proceso de detección donde se obtienen los posibles blancos en la imagen, es necesario efectuar la correlación existente entre los posibles blancos detectados y un patrón de entrada para mantener el seguimiento del objeto en movimiento.

2 Fusión multisensor

Existen tres niveles de fusión de datos de imágenes [4]: *Nivel pixel, nivel características* y *nivel decisión.* En la Figura 1 se observa como los niveles dependen de forma en la que se encuentran los datos del sensor en el momento de realizar fusión, es decir, el nivel es usado para describir el punto durante el procesamiento

de imágenes en el cual ocurre la combinación de los datos que provienen de diferentes sensores.

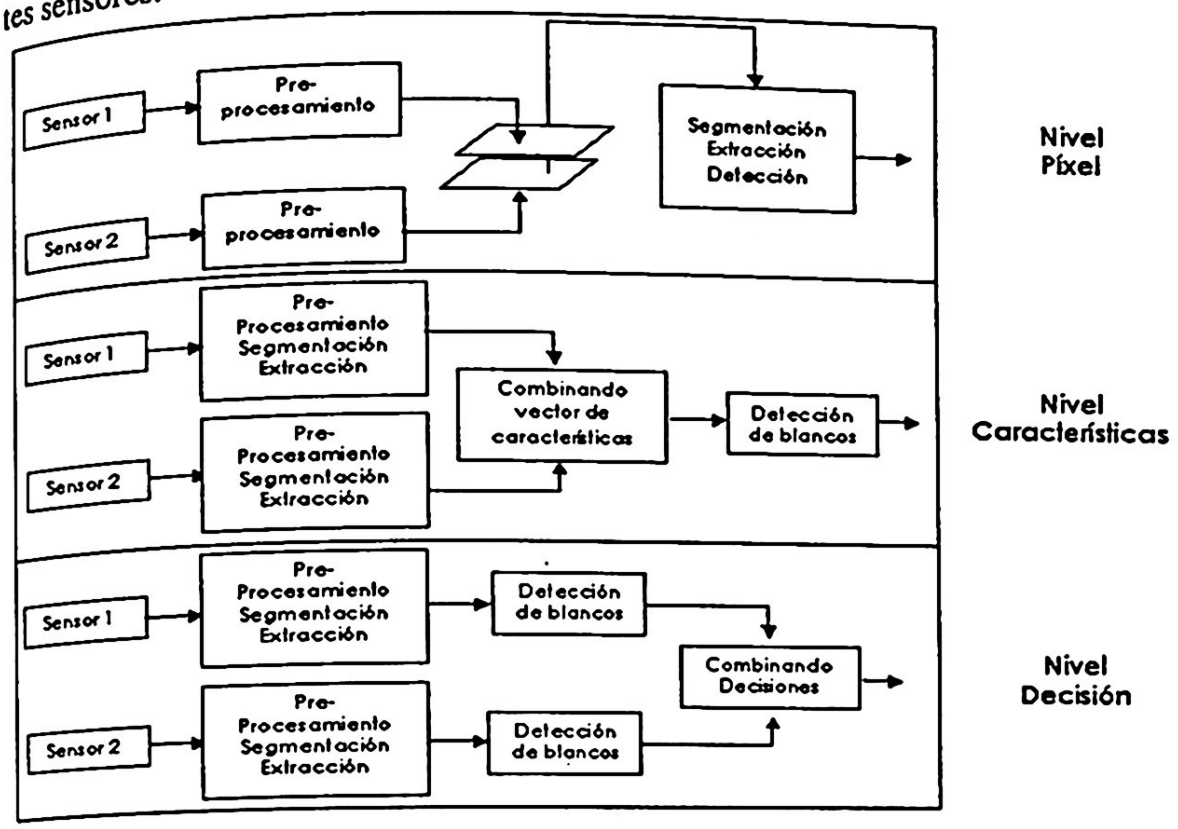


Fig. 1 Niveles de fusión, en la figura se observan los módulos del procesamiento de imágenes y sobre cuales se aplican los tres diferentes niveles.

El nivel mas bajo de fusión es a *nivel pixel* que consiste en la combinación de los pixeles registrados de todo el conjunto de sensores para después ejecutar diferentes funciones de detección y discriminación de objetos. El segundo nivel, *fusión a nivel características* combina las características de los objetos (dimensiones, área, textura, contraste, etc.) que se extraen de los datos que provienen del sensor antes de ser combinados.

El nivel mas alto, *fusión a nivel decisión* consiste en fusionar las decisiones propuestas en cada dato de entrada obteniendo como resultado una decisión. En este trabajo se muestran los resultados obtenidos de la fusión a nivel características para la detección de los objetos en movimiento.

3 Fusión adaptiva distribuida

El propósito de fusionar la información de diferentes sensores es complementar la información que proporciona un solo sensor, sin embargo, una problemática a resolver cuando se integran varios sensores es evaluar la información relevante que proporciona cada sensor. Aunque actualmente ya se tienen trabajos sobre la fusión de sensores, pocos son los que se basan en fusión adaptiva, lo cual permite tener un campo de investigación a explorar y proponer técnicas adaptivas de fusión usando

los sensores seleccionados. Además, parte importante para la comunicación de sensores es la arquitectura seleccionada, donde la mayoría de los sistemas de fusión sensorial usan una arquitectura centralizada que ocasiona la dependencia de información que generan los sensores, por lo que la tendencia en el diseño de estos sistemas es crear arquitecturas descentralizadas que permitan la funcionalidad sistema aun cuando un sensor presente problemas en su funcionamiento.

El modelo de los elementos básicos de un sensor en la detección y seguimiento de múltiples objetos requiere un proceso de asociación de datos el cual se define como asociación local y se muestra en la Figura 2, pero además cuando se incrementa el número de sensores se requiere de una asociación global para determinar que información proporcionada por diferentes sensores pertenece al mismo objeto físico.

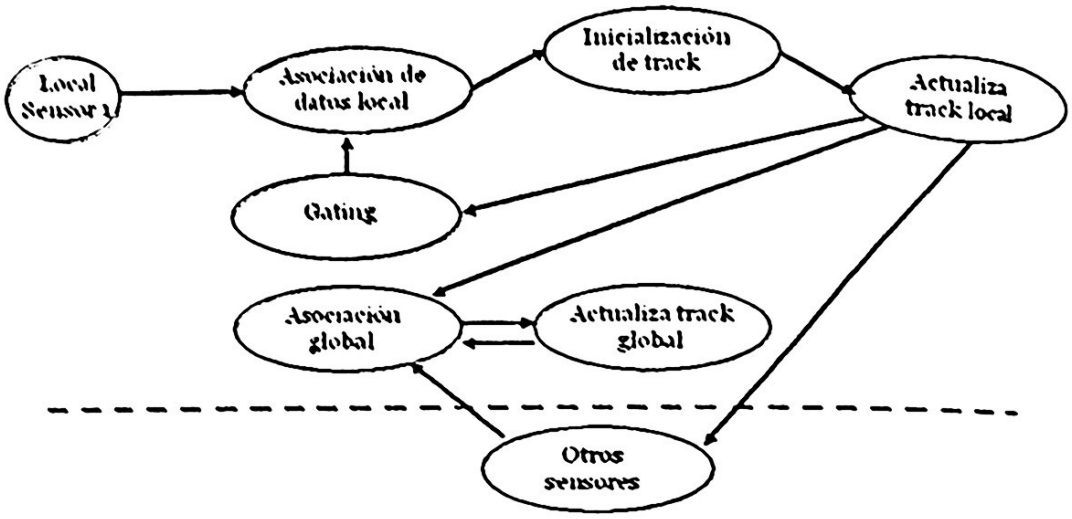


Fig. 2 Fusión distribuida, esta Figura muestra el modelo básico de un sistema de asociación global junto con la etapa de fusión de diferentes sensores.

En base al estudio del estado del arte se propone un algoritmo de fusión adaptiva que permita tener un seguimiento robusto a los cambios que sufra el objeto en forma y tamaño a través del tiempo, así mismo la importancia de realizar una fusión distribuida ha sido poco explorada por su complejidad pero proporciona ventajas el manejo de recursos del sistema. El objetivo es tener un algoritmo de seguimiento de múltiples objetos para cada sensor y después combinar de manera distribuida información de los objetos detectados.

4 Detección y seguimiento de objetos

La primer pregunta que se debe contestar en la selección de las características para la fusión es "¿Qué es importante en la imagen y como puede ser aislado?" [6], esto significa que es necesario determinar el tipo de objetos a detectar y como se representan en los sensores seleccionados. Por ejemplo, en aplicaciones militares, gene-

primero los objetos que son de interés deberán ser mas calientes o fríos que el fondo, cuando se usan sensores infrarrojos.

El sensor infrarrojo usado en este trabajo representa el nivel de calor de un objeto con niveles de gris, a medida que aumenta el nivel de gris significa que el cuerpo esta emitiendo mayor cantidad de calor, es decir, un objeto caliente se representará con un nivel de gris cercano al 255, sin embargo, un objeto frío tendrá un nivel de gris cercano al 0.

En este trabajo como ya se mencionó, se han digitalizado secuencias con dos sensores (infrarrojo y visible), donde cada una de las características se obtienen localmente usando dos ventanas, una interna y una externa. El tamaño de la ventana más pequeña es igual al objeto de interés.

4.1 Selección de características y tamaño de la ventana

Existen diferentes características que nos permiten identificar a un objeto. En [5] usan cuatro características locales: máximo local, diferencia de medias, gradiente promedio y variación local; por medio de estas características detectan automóviles fácilmente con secuencias obtenidas por dos sensores, infrarrojo y visible.

Estas características basadas en la diferencia de contraste también serán usadas para la detección de objetos en este trabajo. Sin embargo, solo se usarán 3 características: diferencia de media, gradiente promedio y variación local. Esto se debe a que el máximo local no proporciona información relevante que discrimine al objeto del fondo, en las secuencias obtenidas.

El tamaño de la ventana debe contener al objeto de interés ya que es de suma importancia obtener resultados confiables en la fusión de sensores y seguimiento [8]. Una técnica de segmentación es usada para determinar la ventana asociada al objeto de interés; la cual realiza un procesamiento temporal, basado en la representación multi-normal a nivel del pixel. Este método usado en [7] identifica a los pixeles del fondo en cada cuadro nuevo mientras actualiza el modelo de cada pixel.

Los pixeles etiquetados como fondo pueden integrarse como parte de los objetos usando un algoritmo de componentes conectados. Este algoritmo proporciona el tamaño de la ventana interior. El tamaño de la ventana exterior se determina de acuerdo al tamaño aproximado del objeto de interés [5].

4.1.1 Diferencia de Medias

La diferencia entre medias es una medida del nivel de gris promedio de una región del tamaño del objeto de interés (ventana interior), comparado con el nivel de gris de los píxeles vecinos que forman parte del fondo (ventana exterior).

Esta característica es ampliamente usada para detectar píxeles que son más calientes que el fondo y que no están contenidos en la ventana interior. La diferencia de medias detecta las regiones locales en donde los cambios de intensidad son abruptos, ésta se calcula por medio de las ecuaciones (1), (2) y (3) para cada pixel de la imagen:

$$C_{i,j}^{1} = \mu_1(i,j) - \mu_2(i,j)$$

$$\mu_1(i,j) = \frac{1}{n_1} \sum_{(k,l)\in N_1(i,j)} f(k,l)$$

$$\mu_2(i,j) = \frac{1}{n_2} \sum_{(k,l)\in N_2(i,j)} f(k,l)$$

Donde n_2 es el número de píxeles en $N_2(i,j)$, n_1 es el número de píxeles en la ventana interior que contiene a los objetos de interés. $N_2(i,j)$ contiene a todos los píxeles vecinos de la ventana centrada en el píxel (i,j), excepto los píxeles que son parte la ventana interior, como se muestra en la Figura 3.

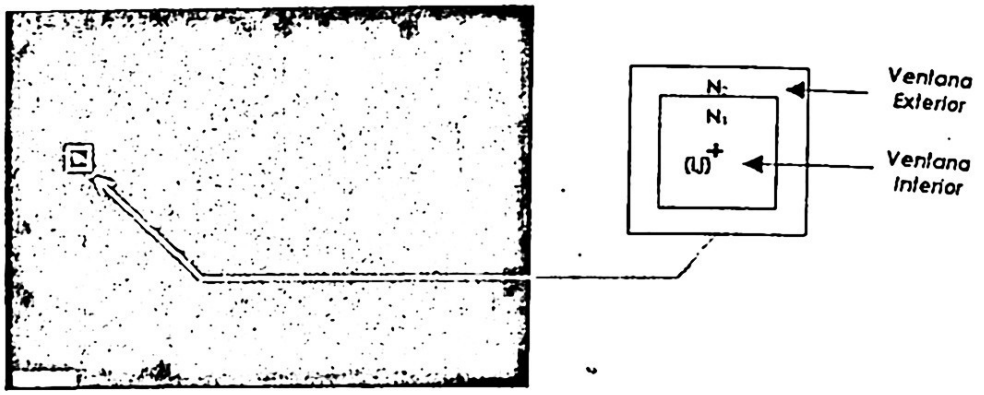


Fig. 3 Ventana interior y exterior, sobre las cuales se obtienen las tres características a fusionar.

4.1.2 Gradiente promedio

El gradiente promedio es una característica que detecta las variaciones de intensidad internas del objeto. El gradiente promedio se calcula con la siguiente ecuación:

$$C_{i,j}^{2} = \frac{1}{n_1} \sum_{(k,l)\in N_1(i,j)} G_1(k,l) - \frac{1}{n_2} \sum_{(k,l)\in N_2(i,j)} G_2(k,l) \tag{4}$$

La ecuación 4 es la diferencia entre el gradiente de la ventana interior y el de la ventana exterior. Los gradientes para cada ventana G_i se obtienen mediante la ecuación 5 que representa la diferencia entre los gradientes horizontales G_i^h y gradientes verticales G_i^v de la ventana.

$$G_i(k,l) = G_i^h(k,l) - G_i^v(k,l) \qquad i=1,2 \tag{5}$$

$$G_i^h(k,l) = \left| f(k,l) - f(k,l+1) \right| \tag{6}$$

$$G_i^v(k,l) = \left| f(k,l) - f(k+1,l) \right| \tag{7}$$

donde i=1 corresponde a la ventana interior y i=2, corresponde a la ventana exterior.

4.1.3 Variación local

La variación local es usada para detectar pequeños cambios en la variación de intensidad sobre las regiones locales. La variación local se obtiene de la siguiente manera:

$$C_{i,j}^3 = \frac{1}{n_1} \sum_{(k,l)\in N_1(i,j)} V_1(k,l) - \frac{1}{n_2} \sum_{(k,l)\in N_2(i,j)} V_2(k,l) \tag{8}$$

donde

$$V_1 = \left| f(k,l) - \mu_1(i,j) \right| \tag{9}$$

$$V_2 = \left| f(k,l) - \mu_2(i,j) \right| \tag{10}$$

4.2 Normalización de características

Después de obtener las tres características para cada pixel se deben normalizar de acuerdo a la imagen correspondiente, generando tres imágenes características. De esta manera el valor del pixel representa el número de desviaciones estándar que el pixel tiene de acuerdo a la imagen.

$$C_{i,j}^{m,N} = \frac{C_{i,j}^m - \mu_m}{\sigma_m} \quad \text{m=1,2,3} \tag{11}$$

Asimismo, se realiza la normalización (N) para cada característica (m), la media y la desviación estándar de cada característica se calculan sobre todos los píxeles de la imagen.

4.3 Fusión multisensor

Una vez normalizadas las características, el siguiente paso es combinarlas. Las características se fusionan de tal forma que cada una debe ser valorada o pesada de acuerdo a la información que proporciona para la detección del objeto, es decir, la característica de mayor peso es la más relevante para el tipo de objetos a detectar.

El peso de la característica se mide como el promedio de los N_t valores más altos de la característica, donde N_t representa el número de pixeles en la región del tamaño del objeto de interés. Este proceso enfatiza las regiones de interés mientras que resto se suprimen. La ecuación 12 obtiene la imagen multisensor combinando las características para ambos sensores.

$$F_{i,j} = \sum_{m=1}^{6} P_m C_{i,j}^{m,N} \tag{12}$$

donde

$$P_m = \frac{V_m}{\sum_{i=1}^{3} V_i} \qquad \text{m=1,2,3} \tag{13}$$

$$V_m = \frac{\sum_{S=1}^{N_t} C_S^{m,N}}{N_t} \tag{14}$$

4.4 Seguimiento de objetos

Después de la fusión de las características, se segmenta la imagen fusionada para detectar los posibles objetos de interés en la escena. Un patrón de búsqueda debe definirse para llevar a cabo el seguimiento. La correlación es procesada únicamente sobre los posibles objetos de interés y no sobre toda la imagen usando el patrón dado.

5 Resultados

Las secuencias de imágenes para este trabajo se adquirieron con un sensor infrarrojo PUMA con resolución de 640 x 480 y con un lente de 100 mm. Un sensor visible color DFK4003 con la misma resolución con un zoom de 8-108 mm. Ambos sensores se calibraron manualmente y se mantienen estáticos sobre la escena de interés.

El algoritmo fue evaluado con tres secuencias de imágenes adquiridas con ambos sensores sobre la misma escena con variaciones de iluminación. La Figura 4 muestra las imágenes adquiridas en diferentes instantes de tiempo t=0, t=20; las imágenes la izquierda fueron obtenidas con un cámara infrarroja y las imágenes de la derecha son en el rango visible, en las imágenes se aprecia un objeto en movimiento a largo del campo de visión de ambas cámaras. Este objeto es una podadora desplazándose sobre un campo. En ambos casos, existe diferencia de contraste del objeto de interés con respecto al fondo.

En la Figura 5 se muestran las imágenes resultantes de la fusión de las tres características para cada uno de los sensores, en estas imágenes se observa como la fusión realiza un filtrado de los objetos que satisfacen las características seleccionadas con respecto al fondo, aunque en ambos casos, aún se tienen varias regiones que serán eliminadas en el siguiente proceso de fusión multisensor.

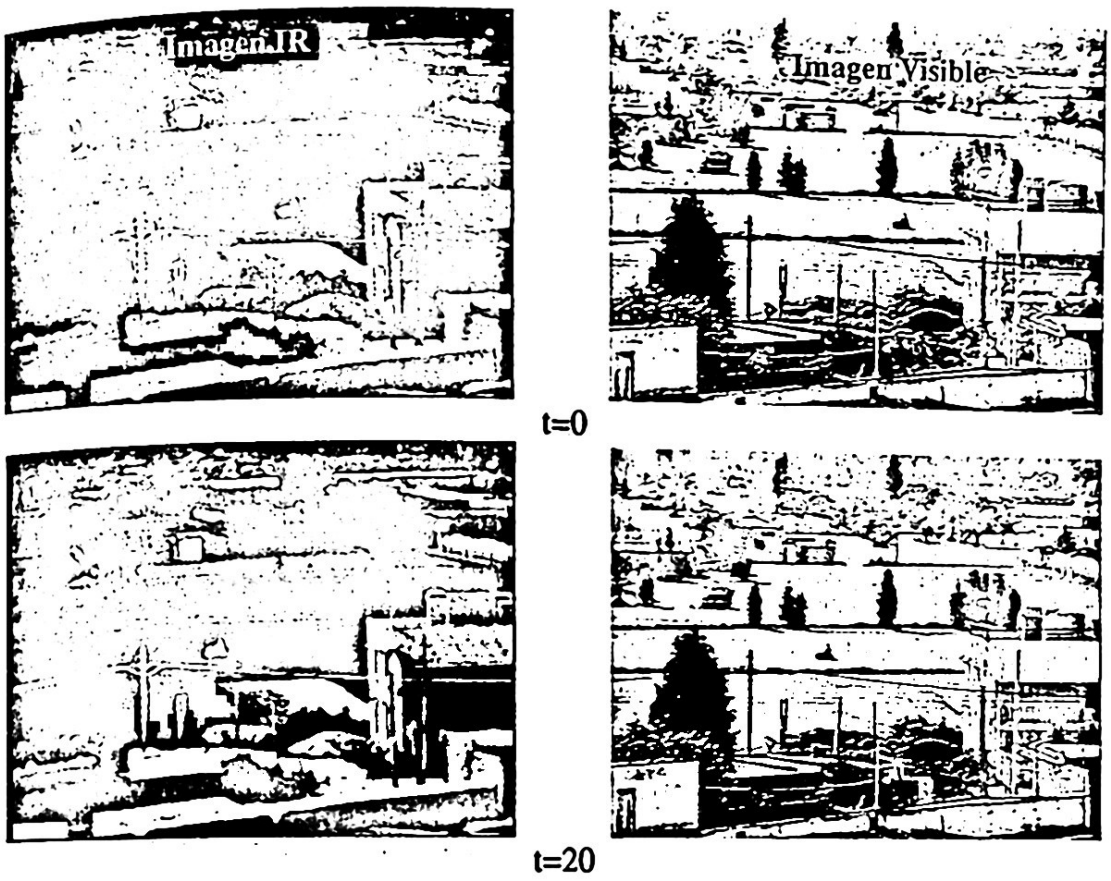


Fig. 4 Imágenes adquiridas usando los sensores infrarrojo y visible a diferentes tiempos. Nótese el objeto de interés (podadora).

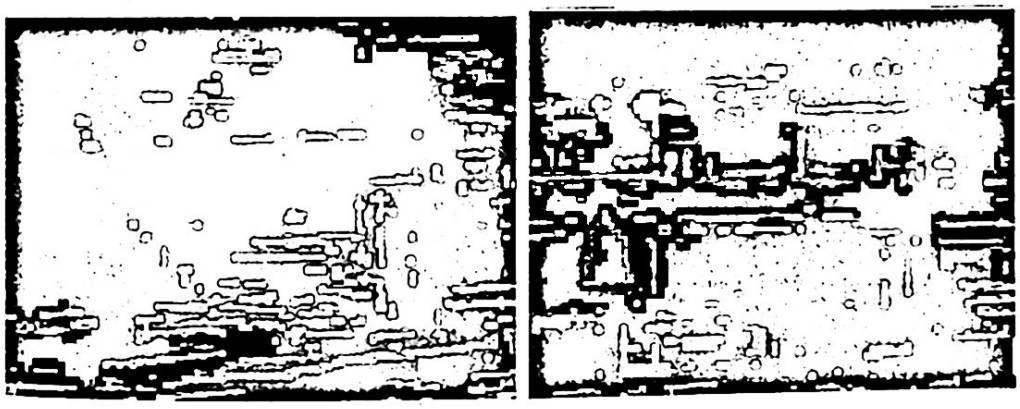


Fig. 5 Imágenes fusionadas

La figura 6 presenta la imagen multisensor resultante de combinar las seis características definidas en la sección cuatro, así como las regiones resultantes después de

realizar el proceso de segmentación. Finalmente la Figura 7 muestra el resultado del proceso de seguimiento. Los resultados en la detección y seguimiento del objeto para tres secuencias digitalizadas son satisfactorios ya que se discriminan en lo posible los objetos que no son de interés en la imagen con la fusión de las características seleccionadas para ambos sensores lo cual permite mantener el seguimiento del objeto en la escena.

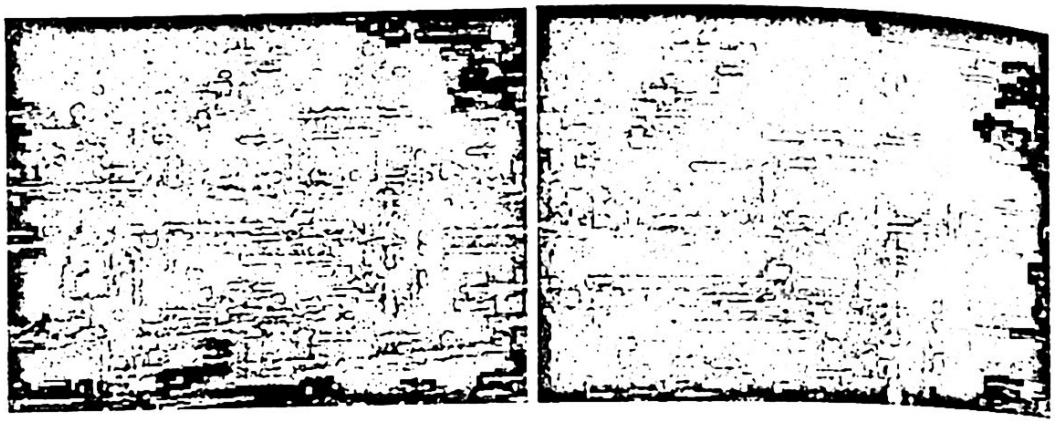


Fig. 6 Resultados de la fusión multisensor

Fig. 7 Resultados de la detección y seguimiento del objeto.

Al igual que en la figura 4 se muestran imágenes capturadas en diferentes instantes de tiempo para observar el seguimiento.

6 Conclusiones

En este trabajo se ha presentado un algoritmo de detección y seguimiento de objetos en secuencias de imágenes multisensor, el cual consiste de tres módulos principales: el primero, es determinar el tamaño de la ventana de los objetos de interés mediante un algoritmo de segmentación basado en la representación Multi-Normal a nivel pixel; el segundo es la fusión de características para la detección de objetos y último módulo es el seguimiento de objetos mediante correlación sobre los posibles objetos de interés detectados. La detección de la ventana de los objetos de interés

hace automáticamente. Su tamaño determina la eficiencia en el funcionamiento de los siguientes módulos, mostrando con nuestra propuesta resultados satisfactorios. Con la fusión de las características seleccionadas se logra eliminar la mayor cantidad de objetos que no son relevantes. El sistema completo se probó con 3 secuencias de aproximadamente 100 imágenes cada una.

Referencias

1. Beauvais M., Lakshmanan S., CLARK: A heterogeneous sensor fusion method for finding lanes and obstacles, Image and Vision Computing, Vol. 18, No. 1, 2000, 397-413.
2. Collins T. R., Lipton J. A., Fujiyoshi H., Kanade T., Algorithms for Cooperative Multisensor Surveillance, Proceedings of the IEEE, Vol. 89, No. 10, 2001, 1456-1477.
3. Fujiyoshi H., Lipton J. A., Real-time human motion analysis by image skeletonization, Proc. of the Workshop on Application on Computer Vision, October 1998.
4. Hall L. D., Llinas J., Handbook of Multisensor Data Fusion, CRC Press, Florida, 2001.
5. Kwon H., Der Z. S., Nasrabadi M. N., Adaptive multisensor target detection using feature-based fusion, Optical Engineering, Vol. 41, No. 1, January 2002, 69-80.
6. Lallier E., Farooq M., A Real Time Pixel Level Based Image Fusion Via Adaptive Weight Averaging, 3rd. International Conference on Information Fusion, FUSION 2000, Vol. 2, July 2000, WeC3-3 to WeC3-10.
7. Pavlidis I., Morellas V., Tsiamyrtzis P., Harp S., Urban surveillance systems: from the laboratory to the commercial world, Proceedings of the IEEE, Vol. 89, No.10, October 2001, 1478-1497.
8. Son G. J., Lim W. C., Choi I. Kim C. N., Adaptive Sizing of Tracking Window for Correlation-Based Video Tracking, IEICE Trans. Inf. & Syst., Vol. E85-D, No. 6, June 2002, 1015-1021.
9. Valet L., Mauris G., Bolon, A Statistical Overview of Recent Literature in Information Fusion, IEEE AESS Systems Magazine, March 2001, 7-14

Recuperación de formas 3D usando luz estructurada

Venustiano Soancatl[1] y Leopoldo Altamirano[1]

INAOE Departamento de Ciencias Computacionales Luis Enrique Erro No. 1
Sta. Ma. Tonantzintla, Puebla, 72840, México
venus@ccc.inaoep.mx, robles@inaoep.mx

Resumen La recuperación 3D usando luz estructurada es una técnica que permite recuperar modelos 3D de objetos con alta exactitud. En este artículo presentamos una breve revisión de la técnica de recuperación de profundidad proyectando una sola franja, explicamos el método para su extensión a múltiples franjas, hacemos una calibración de la cámara y proponemos una técnica sencilla de calibración del proyector. Así mismo proponemos usar un método de codificación binaria para resolver el problema de las discontinuidades de las franjas, presentamos también resultados obtenidos, conclusiones y trabajo futuro.

1. Introducción

En la industria en general hay una gran necesidad de medir con exactitud formas de objetos 3D y asegurar la calidad en la producción. En muchos casos la medición de estos objetos se hace mediante contacto manual en donde técnicos miden la superficie de los objetos para encontrar defectos. También posible usar robots para la medición de los objetos. En ambos casos tanto medición mediante contacto manual como en la utilización de robot el tiempo de medición es alto. Estas son algunas de las principales razones por las que es importante contar con un sistema de recuperación 3D basado en luz estructurada. Con un sistema de este tipo, puede realizarse inspección automática línea valiéndose de la medición de formas 3D del objeto bajo inspección, por ejemplo, se pueden detectar los defectos en el pintado del cuerpo de un panel inspeccionar abolladuras, efectuándose estos con alta rapidez, exactitud y sobre todo sin contacto.

Un sistema de luz estructurada normalmente consta de una cámara, un objeto o superficie a recuperar, una fuente de iluminación que puede ser un láser, proyector de diapositivas, un cañón o incluso un foco común y corriente que pueda proyectar una franja o patrón de luz sobre el objeto. Otras de las ventajas de usar esta técnica es que se puede hacer la medición de objetos sin la utilización de dispositivos de contacto y el hecho de que la cámara y la fuente de iluminación no son costosos.

Esta técnica consiste en proyectar patrones de luz sobre la superficie que queremos recuperar, estos patrones pueden ser puntos de luz, una sola franja,

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 220–231, 2003. © Centro de Investigación en Computación, IPN, México.

múltiples franjas, en general patrones de luz de los que conozcamos su geometría. La información 3D se manifiesta por sí misma en las deformaciones aparentes del modelo proyectado. Analizando estas deformaciones podemos obtener información acerca de la posición, orientación y textura de la superficie. El procedimiento para la obtención de la forma y dimensiones de los objetos se determina mediante la distancia de los objetos en relación con algún sistema de referencia. Un sistema de luz estructurada usa una técnica denominada triangulación en la que una fuente de luz emite un haz a un cierto ángulo sobre una superficie, el haz reflejado es captado por una cámara. Al estar la cámara y la fuente de iluminación separados una cierta distancia se obtiene un triangulo que al resolverlo se puede conocer la distancia a la que se encuentra el objeto.

1.1. Trabajo relacionado

Klette, Schlüns and Koshan [1] proponen una técnica de proyección de plano de luz en donde el centro óptico de la fuente de iluminación es considerado como el origen. El objetivo es calcular las distancias entre un plano de referencia y los puntos iluminados en el objeto, dicho plano de referencia es definido por el eje vertical Y del sistema de coordenadas del plano de luz y el eje X en el que se encuentra la cámara. El centro óptico de la cámara y el centro óptico de la fuente de iluminación se encuentran en el mismo eje. Los autores calculan un ángulo α_k y a partir del conocimiento de la distancia entre la cámara y el proyector y un espacio calibrado pueden calcular las distancias iluminadas en el objeto. Este método se verá con mayor profundidad en la siguiente sección ya que una parte de nuestro trabajo es una extensión de este método.

Boughet [2] implementó un sistema de luz estructurada de bajo costo que usa una lámpara de escritorio y un objeto con el que pueda proyectar una sombra que se asemeje a una franja. Se mueve esta franja manualmente de tal manera que barra toda la superficie del objeto. Calculan la posición del centro óptico de la fuente de iluminación colocando un objeto en posición vertical. Del conocimiento de la altura del objeto y de la longitud de la sombra que proyecta es posible encontrar el centro óptico de la fuente de iluminación.

Sotoca [3] presenta una aplicación de la técnica de luz estructurada para reconstrucción de espaldas humanas. El usa principalmente dos enfoques, el primero es reconstrucción con el eje óptico de la cámara perpendicular al plano de referencia y el segundo es reconstrucción con el eje óptico de la cámara con dirección arbitraria. La técnica es similar a la de Klette y la diferencia radica en que Klette pone el plano perpendicular al plano de referencia.

2. Proyección de una sola franja

El principio consiste en proyectar una franja de luz sobre el objeto como se muestra en la Fig. 1

Como ya se mencionó anteriormente, esta técnica proyecta un plano de luz sobre la escena. La idea es interceptar los rayos de proyección de la imagen con el plano de luz. Dicha intersección es visible como una franja de luz en la imagen.

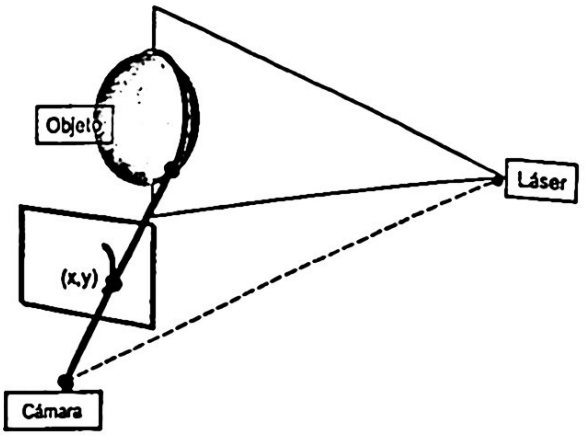


Figura 1. Proyección de una franja de luz

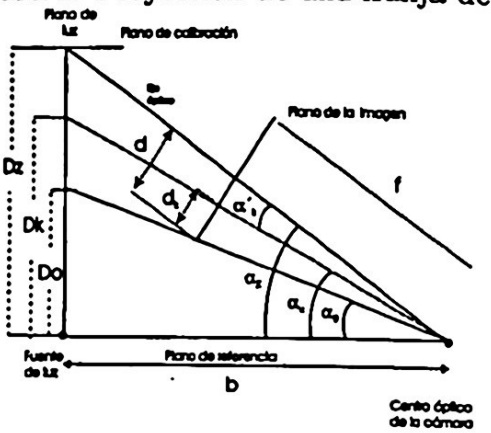


Figura 2. Representación esquemática de la técnica de proyección de un plano de

La tarea consiste en determinar las distancias entre los puntos iluminad del objeto y un plano de referencia (ver Fig. 2). Asumimos que la imagen es tamaño MxN y que los índices columna de esta matriz son $x = 0, 1, ..., M -$

Como un paso inicial, se lleva a cabo la calibración de la franja de luz para poder calcular los valores de profundidad. Primero se mide la longitud b es la distancia entre el centro óptico de la cámara y el proyector del plano luz. Se asume que se tiene un objeto de calibración plano, paralelo al plano referencia y puede moverse a alo largo del eje óptico del proyector del plano luz. El objeto de calibración se mueve de tal forma que la franja de luz proyectada coincida con la columna del centro de la imagen $x = M/2$. Posteriormente mide la distancia D_z entre el objeto de calibración y el plano de referencia. Para determinar la distancia D_0 se mueve el objeto de calibración de tal forma que franja de luz proyectada coincida con la primera columna de la imagen $x =$ entonces D_0 es la distancia entre el objeto de calibración y el plano de referencia.

Sean α_z el ángulo formado por el plano de referencia y el eje óptico de cámara y α_0 el ángulo formado por el plano de referencia y el plano que definido por la primera columna de la imagen y el centro óptico de la cámara (ver Fig. 2). Así:

$$\alpha_z = \arctan(\frac{D_z}{b}) \tag{1}$$

$$\alpha_0 = \arctan(\frac{D_0}{b}) \tag{2}$$

El objetivo es poder calcular cualquier distancia D_k entre D_0 y D_z. De la Fig. 2 podemos ver que

$$D_k = b \cdot \tan \alpha_k \tag{3}$$

donde

$$\alpha_k = \alpha_z - \alpha'_k \tag{4}$$

De esta última ecuación podemos ver que la solución al problema de la distancia D_k se reduce a encontrar α'_k. De la Fig. 2 también

$$d = f \cdot \tan(\alpha_z - \alpha_0) \tag{5}$$

al aplicar la fórmula de la adición de la tangente tenemos

$$d = f \cdot \frac{(D_z - D_0) \cdot b}{b^2 + D_z \cdot D_0} \tag{6}$$

Puesto que la imagen tiene M columnas la distancia d_k entre la columna 0 la columna k puede ser calculada por la siguente ecuación

$$d_k = \frac{2kd}{M} \tag{7}$$

$$\tan(\alpha'_k) = \frac{d - d_k}{f} \tag{8}$$

de (7) y (8) tenemos:

$$\tan(\alpha'_k) = \frac{d \cdot (M - 2k)}{M \cdot f} \tag{9}$$

$$= \frac{(D_Z - D_0) \cdot b \cdot (M - 2k)}{M \cdot (b^2 + D_z \cdot D_0)} \tag{10}$$

de (9) podemos resolver (3) y (4) que son las ecuaciones que nos dan la profundidad de la escena.

2.1. Proyección de múltiples franjas

Está técnica se basa en la técnica de proyección de una sola franja. Se asume al igual que para el caso de una sola franja, que las franjas se proyectan sobre un plano de calibración paralelo al plano de referencia y el objetivo es calcular las distancias D_ki de los puntos iluminados por las franjas en la escena al plano de referencia (ver Fig. 3).

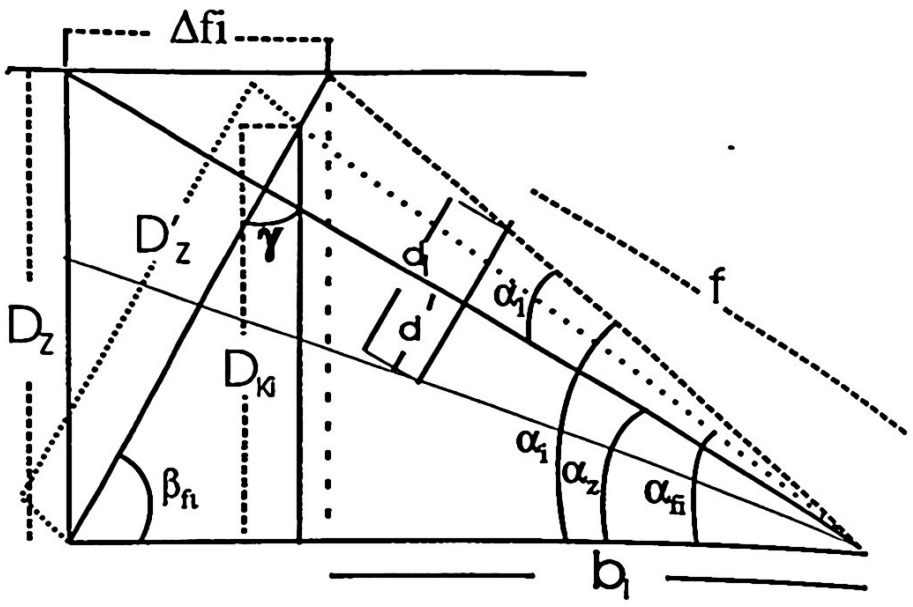


Figura 3. Representación esquemática para múltiples franjas

De la ecuación (9) tenemos

$$\tan(\alpha_l) = \frac{d_l - d}{f} \tag{11}$$

Podemos obtener d_l de la ecuación (7) sustituyendo k por l donde l es l-ésima columna en la imagen, así.

$$\tan(\alpha_l) = \frac{d \cdot (2l - M)}{M \cdot f} \tag{12}$$

$$= \frac{(D_Z - D_0) \cdot b \cdot (2l - M)}{M \cdot (b^2 + D_z \cdot D_0)} \tag{13}$$

Por lo tanto tenemos que $\alpha_i = \alpha_z + \alpha_l$ entonces

$$b_i = \frac{D_z}{\tan(\alpha_i)} \tag{1}$$

Ahora se calcula $\alpha_f i$ de la misma forma que α_i y $\Delta fi = |b - b_i|$ además $\beta_f i = \arctan(D_z/\Delta fi)$ entonces usando la ley de los cosenos

$$D'_z = \frac{b \cdot \sin(\alpha_f i)}{\sin(\beta_f i + \alpha_f i)} \tag{15}$$

y finalmente obtenemos

$$D_k i = D'_z \cdot \sin(\beta_f i)$$

Con esta última expresión podemos calcular la profundidad de cualquier punto iluminado en el objeto.

3. Calibración

Los métodos expuestos en las secciones anteriores dependen de la exactitud con la que se encuentren los valores D_z, D_0 y b ver Fig. 2. Por esta razón es necesario usar un modelo exacto para la calibración tanto de la cámara como del proyector.

3.1. Calibración de la cámara

La calibración de cámara consiste en encontrar los parámetros intrínsecos y extrínsecos de la cámara. A saber longitud focal f, el punto principal (C_x, C_y), los factores de escala S_x y S_y y los parámetros de rotación α, β, γ y de traslación T_x, T_y y T_z. Para tal efecto usamos un plano de calibración de 10 cm x 10 cm como el que se muestra en la Fig. 4.

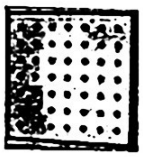


Figura 4. Plano usado para calibrar la cámara

Hay tres posiciones fundamentales del plano de calibración, la primera es la posición en plano trasero como se muestra en la Fig. 5, paralelo al plano xy. La segunda posición es perpendicular a la primera posición de forma tal que coincida con el plano yz y finalmente la tercera posición es perpendicular al segundo plano y de nuevo paralelo al plano xy como se muestra en la Fig. 5.

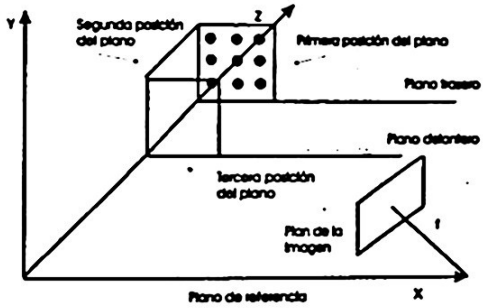


Figura 5. Plano usado para calibrar la cámara

Se pueden tomar varias imágenes del plano en diferentes posiciones, en particular nosotros tomamos 10 para llevar a cabo la calibración de la cámara.

3.2. Calibración del proyector

La calibración del proyector se hace después de haber calibrado la cámara. Primero se proyectan las franjas sobre el plano trasero, ver Fig. 6. Llamémosle

Δfi a la distancia entre la franja fp y alguna franja fi sobre el plano trasero. Así mos calcular las distancias entre la franja fp y cualquier franja fi. El siguiente paso es mover nuestro plano al plano delantero. Podemos saber qué tanto movi- mos nuestro plano dado que usamos un plano de calibración de la cámara que coincide con el plano yz. Sea esta distancia dm. El siguiente paso es medir de nue- vo las distancias entre el plano fp y los planos fi puesto que el plano delantero también esta calibrado.

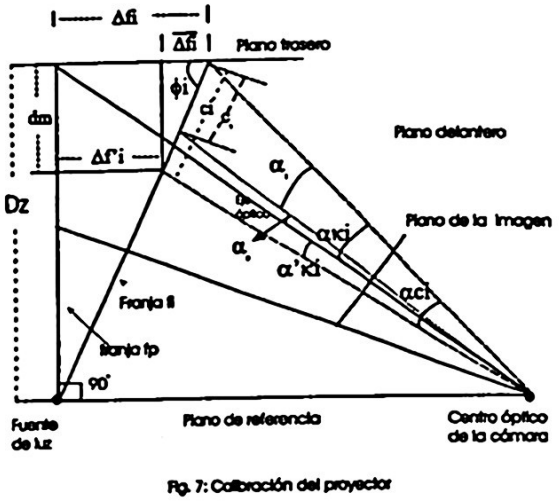


Fig. 7: Calibración del proyector

Figura 6. Calibración del proyector

Así conocemos las $\Delta fi, \Delta f'i$ y dm, de estos datos podemos calcular

$$\overline{\Delta fi} = \Delta fi - \Delta f'i \tag{1}$$

de aquí podemos obtener

$$\phi i = arctan(\frac{dm}{\overline{\Delta fi}}) \tag{18}$$

que es el ángulo comprendido entre el plano trasero y la franja fi. También podemos calcular la distancia D_z del plano trasero al centro óptico de la fuente de iluminación.

$$D_z = \Delta fi \cdot \sin(\phi i)$$

Ahora bien, podemos calcular la distancia $c1$ sobre la franja fi (ver Fig. mediante la ecuación

$$c1 = \sqrt{dm^2 + (\overline{\Delta fi})^2}$$

puesto que la cámara ya ha sido calibrada

$$d = \frac{M}{2} \cdot Sx$$

donde Sx es el factor de escala para la dirección x obtenida en la fase de calibración de la cámara y podemos obtener dk de la ecuación (7) donde k, es la columna k en el plano de la imagen, por lo tanto podemos calcular $\alpha_2, \alpha_k i, \alpha'_k i$ y por supuesto

$$\alpha c_i = \alpha'_k i - \alpha_k i \tag{22}$$

$$\alpha_1 = \alpha_k i - \alpha_2 \tag{23}$$

puesto que hay una correspondencia entre $\alpha_c i$ y ci tenemos

$$c1 = \frac{\alpha_1 \cdot ci}{\alpha_c i} \tag{24}$$

Una vez hecho esto, lo único que resta es calcular las distancias Dk_i al plano de referencia (ver Fig. 7).

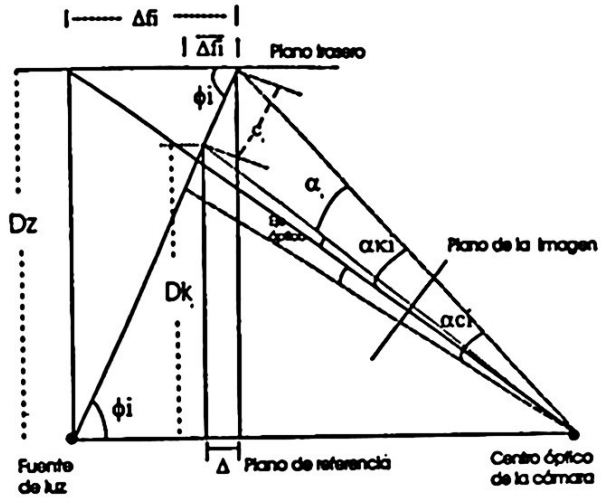


Figura 7. Distancias $D_k i$ al plano de referencia

$$\Delta = c1 \cdot \cos(\phi i) \tag{25}$$

y finalmente

$$Dk_i = (\Delta fi - \Delta) \cdot \tan(\phi i) \tag{26}$$

De está manera la exactitud de las mediciones ahora dependen de la calibración de la cámara y de la colocación del plano trasero y delantero. A diferencia de la sección 2 y 2.1 ya no es necesario medir D_z, D_0 y b puesto que para el método de calibración presentado no necesita de esos tres parámetros.

3.3. Codificación binaria

Hasta ahora se ha resuelto el problema de la profundidad para una y múltiples franjas, sin embargo aun queda el problema de las discontinuidades cuando

se proyectan múltiples franjas. El problema de saber que líneas en la imag pertenecen a una determinada franja es resuelto usando el método de codif cación binaria.

La idea para codificar las franjas de luz es proyectar secuencialmente señal binaria correspondiente a los digitos binarios de un código. En la Fig. señal binaria es traducida en franjas blancas y negras. Al proyectar una secuenc de n patrones produce 2^n franjas codificadas.

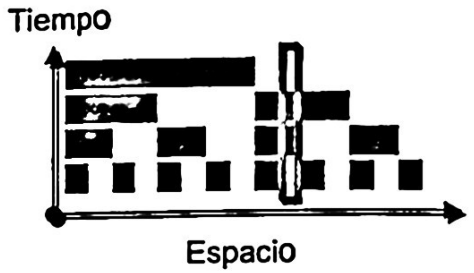


Figura 8. Codificación binaria

Para decodificar la posición de un pixel, la tarea principal es recuperar la intensidad binaria proyectada en la secuencia de patrones. Así por ejemplo, código para la franja del rectángulo vertical en la Fig. 8 es 1001.

4. Experimentos

Se han llevado a cabo varios experimentos para comprobar el funcionamient del sistema. Uno de ellos es la recuperación de un ratón de computadora com el que se muestra en la siguiente figura.

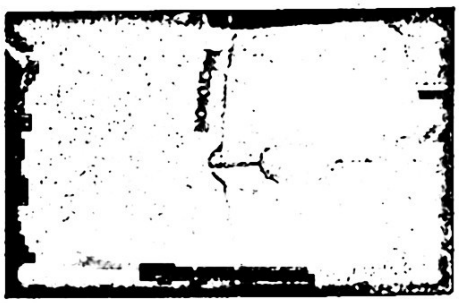


Figura 9. Objeto que queremos recuperar

Para recuperar el objeto de la Fig. 9 se proyecto una sola franja de luz objeto fue desplazado 55 veces de modo que se barriera la totalidad del objeto, para cada desplazamiento se tomó una imagen, el resultado de procesar cada una de las imágenes se muestra en la Fig. 10

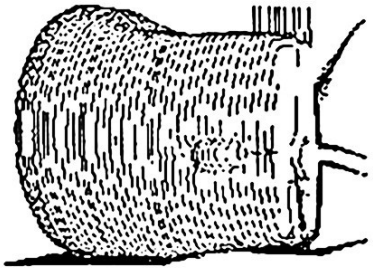


Figura 10. Resultado de la recuperación 3D del objeto de la Fig. 9

Para el caso de múltiples franjas se llevo a cabo el siguiente experimento. Se uso una superficie metálica (ver Fig. 11) a la que se le proyectaron múltiples franjas usando un proyector de diapositivas.

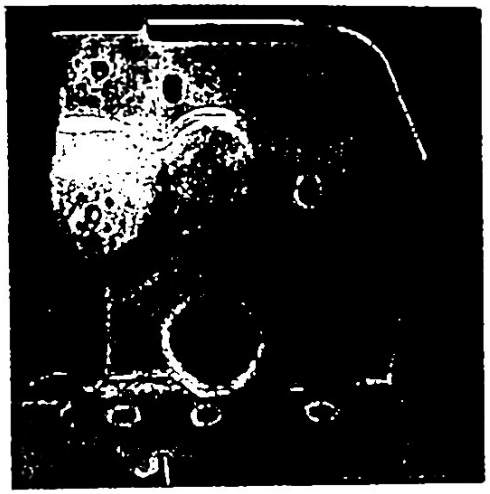


Figura 11. Objeto que queremos recuperar

El resultado de recuperar información 3D de la objeto de la Fig. 11 se muestra en la Fig. 12. El error obtenido de h1 es 0.01 mm, el de h2 es de 0.02 mm y el de h3 es de 0.06 mm (ver Fig. 12). Cabe mencionar que los puntos en las figuras 10 y 12 representan profundidades.

5. Conclusiones

Se ha hecho una breve revisión de la técnica de proyección de una sola franja y se ha propuesto un método para su extensión a múltiples franjas así como también se ha propuesto un método de calibración de la fuente de iluminación. También se presentaron resultados satisfactorios aplicado estas técnicas. Para el problema de discontinuidades en las franjas se ha usado una técnica de codificación binara. La proyección de una sola franja se implementa con mayor facilidad, un inconveniente es que se requieren muchas más imágenes y por lo tanto mayor tiempo

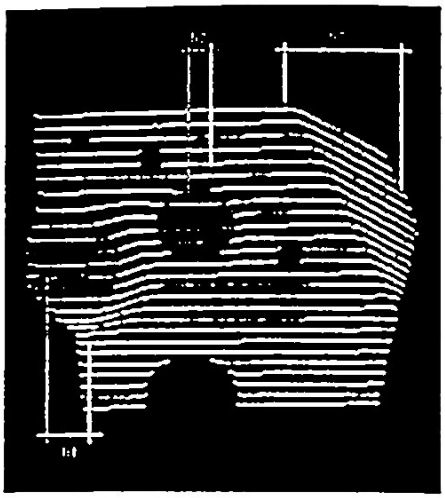


Figura 12. Objeto que queremos recuperar

de procesamiento que con la técnica de proyección de múltiples franjas. Los resultados en ambos casos son bastante buenos puesto que los errores reportados son pequeños. En cuanto a la proyección de múltiples franjas un inconveniente como ya se ha mencionado es la discontinuidad de las franjas proyectadas que se resuelve con la técnica de codificación binaria.

6. Trabajo futuro

Como trabajo futuro adicional al ya realizado se proponen los siguientes puntos.

- Implementar otras técnicas de codificación de franjas o codificación de patrones.
- Implementar la técnica de luz estructurada para objetos de grandes dimensiones (1–2 metros).
- Reconstrucción completa de los objetos usando alguna técnica de creación de mallas.

Referencias

1. Klette, Schlüns and Koschan Computer Vision Three-Dimensional Datafrom Images, Springer-Verlag, Singapore 1998.
2. Boughet Visual Methods for Thre-Dimensional Modeling, Phd thesis, California Institute of Technolgy Pasaden, California, 1999.
3. Sotoca Reconstrucción de Superficies en Entornos de Luz Estructurada: Aplicación a Espaldas Humanas, Tesis doctoral, Universidad de Valencia, Facultad de Ciencias Físicas, Departamento de Informática. Julio, 2001.
4. Huynh, Calibration of a Structured Light System: a Proyective Approach, Proc. IEEE Conf. Comput. Vision Patt. Recog, June 1997, Puerto Rico, pp. 225–230.
5. Rocchini, Cignoni, Montani, Pingi and Scopigno "Low Cost 3D ScannerBased on Structured Light, EUROGRAPHICS 2001 , Volume 20 (2001), Number 3.

6. Monks Measuring the Shape of Time-Varing Objects, Phd thesis. Faculty of Engineering and Applied Science Department of Electroncs and Computer Science, august 1994.
7. Chen, Brown and Song Overview ot three-dimensional shapemeasurement using optical methods, Optical Engineering Enero 2000.
8. Batlle, Mouaddib y Salvi Recent Progress in Coded Structured Light as a Technique to Solve the Correspondence Problem: a survey, Pattern Recognition, Diciembre de 1996.
9. Jeffrey A., Richard C. Kim and Steven K. Case Three dimensional inspection using multistripe structured light, Optical Engineering, Vol. 24, No 6, 966-974 p.1995.
10. Kowarschik Richard, Kühmstedt Peter Adaptive Optical three dimensional measurement with Strucutred light, Optical Engineering, Vol. 39, No.1, 150-158 p, 2001.
11. Yang Zaiming, Wang Error Analysis of 3D Shape Construction form Structured Lighting, Department of Computer Science, University of California, USA, Vol 29, No, 2 189-206.
12. Yi-Chih Hsieh Decoding structured light patterns for three-dimensional imaging system, Elsevier Science. 1999.

SymWeb: Modelado de una aplicación de simulación web mediante OOHDM

Omar Sosa-Tzec [1], Víctor Menéndez-Domínguez [1], Luis Vargas-Villamil [2]

[1] Facultad de Matemáticas – UADY.
Calle 8 x 21 Col. Ma. Luisa. CP 97177. Mérida, Yucatán.
tzec@cimat.mx, mdoming@tunku.uady.mx
www.uady.mx/sitios/matemati

[2] Facultad de Veterinaria y Zootecnia – UADY.
Carretera Mérida-Xmatkuil Km. 15.5. CP 97100. Mérida, Yucatán.
luis@avanzavet.com
www.uady.mx/sitios/veterina

Resumen. Una problemática que existe actualmente es como diseñar aplicaciones web. El desarrollo de estas aplicaciones debería ajustarse a un modelo ingenieril como de cualquier software. Entre las opciones que se proponen actualmente, se encuentra el *Modelo de Diseño Orientado a Objetos de Hipermedia* (OOHDM). En este trabajo se presenta el modelado con OOHDM de una aplicación de simulación web relacionada con nutrición animal, denominada SymWeb. Se describe brevemente cada una de las fases del modelo. Luego se presenta la metodología empleada en el desarrollo; se muestra una síntesis del análisis de requerimientos y cómo se conformaron los diagramas conceptuales y de navegación, así como la implementación. Se discuten los resultados obtenidos de la aplicación del modelo en el caso de SymWeb. Se concluyó que la principal aportación del OOHDM es facilitar la comunicación en un grupo interdisciplinario enfocados al desarrollo de una aplicación web.

1 Introducción

En la actualidad surge el interés en desarrollar aplicaciones que corran en la Internet, en particular en la Web. Estas aplicaciones se conocen con el nombre de *aplicaciones web* o de *hipermedia* [19]; entre ellas se encuentran las que realizan simulaciones.

La historia de la simulación web comienza como un área de trabajo académico el Congreso de Simulación de Invierno (WSC por sus siglas en inglés) [11]. *simulación web* es un término que introdujo Paul A. Fishwick [3] para representar conexión entre la Web y el campo de la simulación. Otros autores añaden a la definición que el propósito de la simulación web es ejecutar más simulaciones eficientes [21].

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 232–243, 2003. © Centro de Investigación en Computación, IPN, México.

En el caso de la simulación web, la Web puede servir como sistema operativo como el canal de distribución para sus aplicaciones, tomando las características de la Web a su favor, [8]. Sin embargo, este nuevo medio requiere explorar nuevas técnicas para el desarrollo de aplicaciones web, como por ejemplo el *Modelo de Diseño Orientado a Objetos de Hipermedia* (OOHDM).

2 Motivación

El Ciclo de Vida de Desarrollo del Sistema (CVDS) para una aplicación web es similar al de cualquier aplicación de software. Para desarrollarlo, se selecciona un *modelo* a seguir. El modelo se refiere al conjunto de productos, formalismos y mecanismos que se usan durante todo el CVDS. Los nombres de las etapas entre los diferentes modelos pueden cambiar, pero prácticamente son una variante del esquema básico del CVDS y que se arraigan a un tipo de conceptualización en particular (o metodología previa) dando su propia versión de este [19]. Al proceso de desarrollar una aplicación web bajo un enfoque ingenieril siguiendo un modelo establecido se le denomina como Ingeniería de Hipermedia.

En el caso de *SymWeb*, se consideró desarrollar, bajo un enfoque de Ingeniería de Hipermedia, una aplicación web para realizar la simulación de modelos dinámicos de nutrición animal mediante métodos numéricos. Estos modelos son producidos a través de un proceso de preprocesamiento de archivos generados por un software comercial de simulación.

El modelado en la producción animal permite reproducir los diferentes procesos digestivos y metabólicos que se llevan a cabo dentro del animal para así estudiarlos con una base cuantitativa y dinámica a más bajo costo que en la investigación pecuaria convencional" [7]. De esta idea surgió el proyecto de desarrollar un simulador de modelos matemáticos veterinarios abierto al público interesado dentro del proyecto 33722-B de CONACyT intitulado "Desarrollo de un Modelo Dinámico para la Simulación de la Producción Bovina de Carne en el Trópico". El proyecto recibió el nombre de *SymWeb*, por Simulación y Modelado vía Web.

La problemática se enfocó en la manera de cómo se llevaría a cabo el desarrollo. Después de un análisis comparativo de la literatura, derivó en el uso del OOHDM desarrollado por Daniel Schwabe y Gustavo Rossi [18].

3 Descripción del OOHDM

A grandes rasgos OOHDM es un modelo de construcción de grandes aplicaciones hipermedia; enfocado en dos aspectos críticos en el desarrollo de aplicaciones web, las cuales son la interfaz del sistema y el esquema de navegación, para lo cual se utiliza una colección de objetos.

Este modelo está constituido de las siguientes fases: Análisis de requerimientos, modelo conceptual, diseño navegacional, diseño abstracto de interfaz e implementa-

ción. Los procesos de desarrollo se modelan usando notación del Lenguaje Modelado Unificado (UML) [20].

3.1 Análisis de requerimientos

En esta primera fase se identifican los actores y las tareas que realizan, definiendo así un escenario. Los escenarios son agrupados para formar un caso de uso el cual representado utilizando diagramas de interacción de usuarios (UID)[15]. Estos diagramas proporcionan una representación gráfica de la interacción del usuario con sistema durante la ejecución de una tarea [18].

Los escenarios son descripciones narrativas de cómo la aplicación puede ser usada. A las descripciones narrativas o textuales de las acciones de los actores según sus roles se les conoce como Especificaciones de Escenario (EE) [15]. Siempre recomendable que los usuarios proporcionen tales descripciones. Si lo anterior no posible, los diseñadores deberán encargarse de redactarlas. Para especificaciones los escenarios muy generalizadas, se aclara el "cómo" elaborando Casos de Uso (CU). En sí, éstos son agrupaciones de escenarios que describen la misma tarea [15].

Como las representaciones gráficas pueden ser más fáciles de seguir, entonces hace uso de los UID. Éstos explican de manera gráfica los casos de uso. Un UID está conformado por elipses y flechas que representan la información a intercambiar y el procesamiento de la información, respectivamente.

3.2 Diseño conceptual

En esta fase se concibe la aplicación en términos de objetos y las relaciones existentes entre ellos [18], utilizando la Tecnología de Modelado de Objetos (OMT) [12]. Esto es, definir diagramas de clases para representar abstractamente la aplicación.

El diseño conceptual tiene como producto el esquema conceptual, el cual explica la manera en la que está conformada la aplicación: Las entidades que contienen información y como se relacionan. En este punto no se requiere qué información verá el usuario o cómo va ha acceder a la información, mucho menos la forma en que se implementará la aplicación.

3.3 Diseño navegacional

En este punto, se describe la estructura navegacional de una aplicación hipermedia en términos de contextos navegacionales que son generados de clases navegacionales [13]. El diseño navegacional es expresado en dos esquemas: Esquema de Clases Navegacionales y Esquema de Contexto Navegacional. En el primero se realiza especificación de nodos, los cuales son proyecciones de una clase y representan elementos que tienen la información. El esquema de contexto navegacional presenta las diferentes maneras en la que un nodo puede ser accedido según el esquema clases navegacionales. Estas maneras son los contextos [13].

Un contexto navegacional es un conjunto de nodos, enlaces y las denominadas clases contextuales. Puede ser definido en intención cuando presentan una propiedad particular presente en todos los nodos o enlaces; o bien, ser definidos en extensión, que es cuando se enumeran sus miembros [13].

Un diseño navegacional puede utilizarse para más de una aplicación. Los contextos tienen como función ayudar al usuario a realizar su tarea durante la navegación de la aplicación. Cuando se establece una plantilla de diseño navegacional que puede adaptarse a una aplicación según el contexto, se le llama esquema de contexto navegacional genérico, el cual está compuesto por representaciones genéricas de los contextos mencionados anteriormente. Existen diferentes notaciones gráficas para los contextos genéricos [13][14][16], pero básicamente son cinco: Contexto genérico, creación de instancia/modificación, estructura de acceso genérica o índice genérico, estructura de acceso jerárquica genérica y búsqueda genérica. Cada una de las notaciones para los contextos genéricos pueden estar acompañados de una tarjeta de especificación donde se definen las características como la relación definida por el enlace que los une (e.g. uno a muchos), tipos posibles de información que contienen (estáticos, dinámicos, jerárquicos), restricciones, etc.

3.4 Diseño abstracto de interfaz

El diseño de interfaz de usuario se auxilia del Diseño Abstracto de Interfaz (AID). Un AID se enfoca en saber cuales de los diferentes objetos navegacionales aparecerán, cuales activarán la navegación y cuales transformaciones de la interfaz se presentarán [15].

La conformación de un AID está dada por elementos llamados Vistas Abstractas de Datos (ADV). Estas ADV son propiamente la interfaz representada por los diferentes elementos con los que interactuará el usuario. Las ADV contienen los elementos claves que visualizará el usuario al acceder el nodo. Pueden ser cadenas de texto particulares, imágenes o información generada por el contexto. En el ADV se define primeramente los eventos relacionados con el acceso. Los demás elementos en la ADV que no están relacionados directamente con el nodo (i.e. elementos no proyectados en el nodo), son de auxilio en la navegación o poseen simplemente caracterísitcas decorativas. Se definen a éstos como decoradores de nodo.

3.5 Implementación

Por su naturaleza, OOHDM no expresa explícitamente como ha de ser la implementación de la aplicación. Sin embargo, presenta la noción de los objetos de navegación como vistas de las clases conceptuales, organiza el espacio de navegación con la introducción de contextos de navegación, separa los aspectos relacionados con la interfaz de usuario de aquellos navegacionales y explícitamente señala que existen decisiones de diseño que necesitan ser hechas solo al momento de la implementación [15]. Éstas son las características que pueden ayudar a establecer una arquitectura adecuada para implementación.

4 Resultados

Como *SymWeb* está enfocado a la simulación de modelos matemáticos veterinarios por parte de personas interesadas en el tema, se definieron los roles dependiendo las características particulares que puede poseer un usuario. El listado de roles (omitiendo tareas por cuestión de espacio) se presenta en el Cuadro 1.

Rol	Descripción
Usuario No Registrado	Usuario común; es decir, aquel que entra a navegar la aplicación con meros propósitos informativos, por curiosidad o por accidente.
Usuario Registrado	Usuario que ha pasado por el proceso de registro en *SymWeb*
Usuario Publicador	Usuario registrado que tiene la capacidad de publicar modelos matemáticos para otros usuarios registrados
Administrador	Usuario registrado único que tiene la capacidad de publicar modelos matemáticos en *SymWeb* pero también tiene la capadicad de administrar a los usuarios registrados

Cuadro 1. Nominación y descripción de roles de *SymWeb*.

El rol básico es el *Usuario No Registrado*. Este tipo de actor puede cambiar rol dependiendo del evento denominado *Registro de Usuario*, en donde este actor pasa ahora a ser parte de otro grupo (*Usuario Registrado*). Por otra parte, algunos estos usuarios registrados tienen la capacidad de poder ejecutar otro rol de manera simultanea: *Usuario Publicador*. Existe el actor de la aplicación que le indica a *Usuario Registrado* cuando puede desempeñar el rol alterno, éste es el *Administrador*, quien al fin y al cabo es un "actor multifacético", ya que puede desempeñar todos los roles dentro de la aplicación.

Según el análisis realizado en los requerimientos, se obtuvieron 10 especificaciones de escenarios que se consideraron básicas para el modelado de la aplicación, cuales van desde *navegar SymWeb* hasta *salirse de SymWeb*, entre otros.

En consecuencia, se realizaron 12 CU, los cuales se enlistan a continuación: Registro de Usuario, Inicio de Sesión, Actualización de Datos del Usuario, Simulación de Modelos Matemáticos, Graficación de Resultados, Publicación de Modelos Matemáticos, Búsqueda Interna General, Búsqueda Interna Avanzada, Dar de baja a Usuario, Dar de baja a un Modelo, Darse de baja como Usuario y Salirse de *SymWeb*.

Para el desarrollo de *SymWeb* se enriqueció el significado de las elipses que componen los UID para señalar el lugar (físico o abstracto) que se ve afectado por procesamiento de información. También se adaptaron éstos con etiquetas añadidas las elipses para facilitar su lectura. En la Figura 1 se presenta el UID para el Registro de Usuario. Los UID en *SymWeb* se realizaron en el orden determinado por CU.

Para abstraer las clases involucradas con *SymWeb* hay que tener en cuenta que ta es una aplicación que consiste en simular modelos matemáticos que generen sultados gráficos por usuarios que probablemente tendrán la capacidad de publicar tales modelos bajo el control de un administrador, quién por defecto publica modelos. El enunciado anterior encierra los puntos señalados en los escenarios reali-

zados previamente. De esta forma se obtuvieron las clases preliminares: Modelos Matemáticos, Gráficas, Usuarios y Administrador.

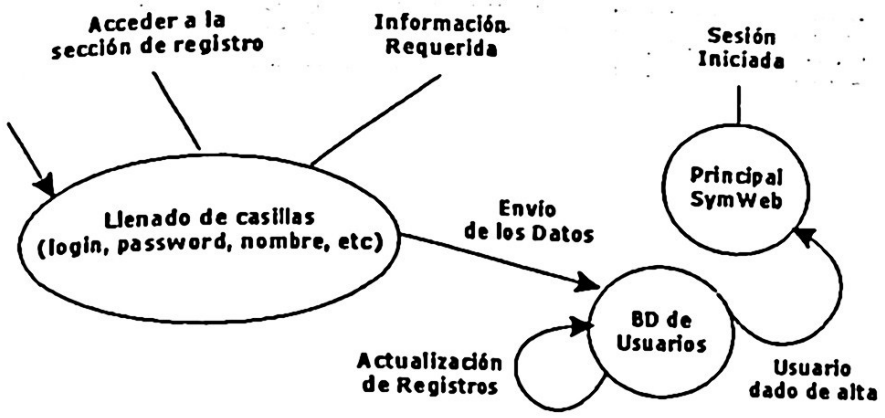


Fig. 1. Diagrama de Interacción de Usuario para Registro de Usuario

Como resultado de un diseño detallado, se abstrajeron 17 clases (presentadas sintéticamente en la Figura 2) que definen 11 enunciados que describen *SymWeb* de manera conceptual.

La definición de los nodos en *SymWeb* se basó en realizar preguntas concretas sobre los eventos y acciones asociados a la clase conceptual. Como resultado se obtuvo 8 tipos de nodos diferentes. Como puede observarse en la Figura 3 correspondiente al nodo *Administrador*, el nodo *admin* es una instancia de la clase *Admin* que se reconoce dentro de la aplicación con la variable *admin*. Este nodo hereda los atributos (e.g. login) y métodos (e.g. Darse_de_baja()) de la clase *Usuario*. Diseñando de manera similar los demás nodos, se establecen 4 esquemas de clases navegacionales para *SymWeb*, donde cada una está relaciona con el usuario actual de la aplicación.

En el caso de *SymWeb* se concluyó el diseño navegacional con tan solo un esquema de contexto navegacional (Figura 4). Esto se debe a que modelamos la aplicación *SymWeb* de manera particular, por lo que no llegamos a un esquema conceptual adaptable a todas las aplicaciones de hipermedia que hagan simulación. Para poder realizar este esquema fue necesario instanciar los contextos genéricos en contextos particulares dentro de los nodos.

En el esquema de contexto navegacional se pueden observar ciertos detalles. Primeramente, se tiene un índice general denominado "Opciones de *SymWeb*" que procede según el tipo de usuario que navegue (i.e. registrado o no registrado). Los usuarios no registrados solo acceden a los nodos generales del modelo y de gráfica. Si un usuario es registrado, entonces accede al nodo de "login" donde dispara el método *inicio_de_sesion()* para después acceder al "Menú General" que está representado por un índice (i.e. una lista de enlaces a las opciones). Adjunto a este índice están el de "Menú Publicación" y "Menú Administración", los cuales están etiquetados con un ovalo negro a su izquierda; este ovalo indica que son índices de acceso restringido y que en nuestro caso solo están disponibles para un usuario con permiso de publicar y para un usuario que tenga además permiso de administrador, respecti-

vamente. Se observa también que todos los usuarios pueden acceder a un índice de "Búsqueda", el cual devuelve información tanto de modelos encontrados, así como de archivos relacionados a éstos, almacenados en el servidor.

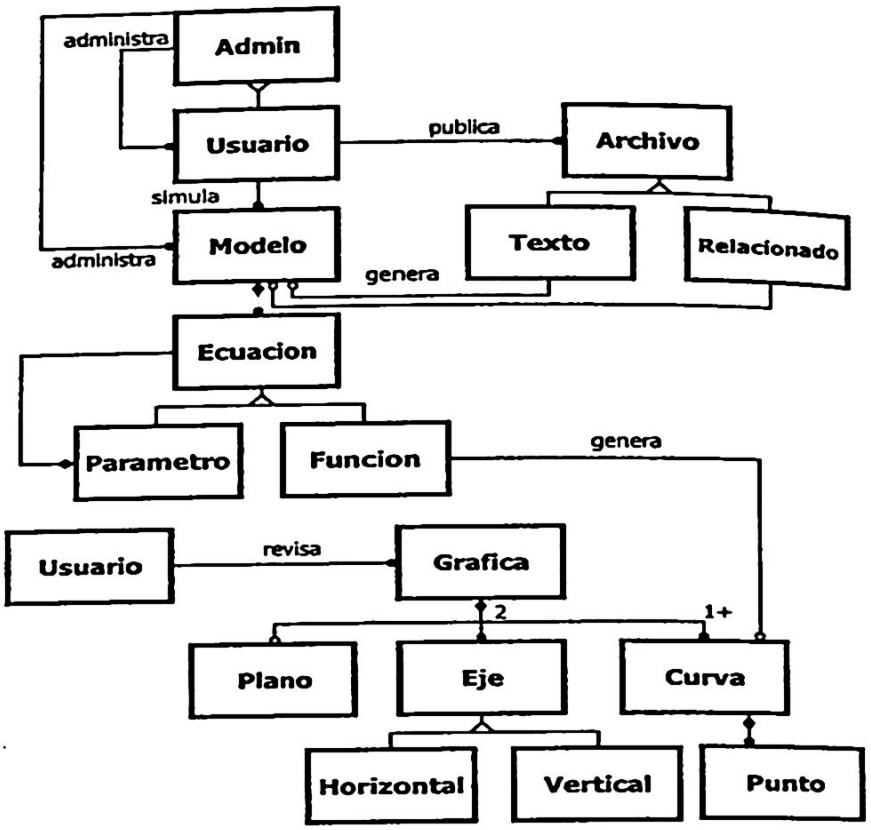


Fig. 2. Esquema conceptual de SymWeb (representación sintética)

NODE admin FROM Admin: admin INHERITS FROM Usuario
EsAdmin: booleano ToUsuario: Anchor(administra) ToModelo: Anchor(administra)
Admin_usr() Admin_mod()

Fig. 3. Nodo Administrador

Durante el proceso de AID, se definieron varios ADV como el mostrado en la Figura 5, los cuales indicaban que componentes de información serían utilizados secciones particulares de la navegación.

Con respecto a la implementación, se definió que *SymWeb* fuera una aplicación basada en páginas web dinámicas que interactúan entre ellas. Estas páginas fueron desarrolladas con la tecnología de Páginas de Servidor Activas (ASP) de Microsoft, incluyéndose también javascript para analizar rutinas de validación y otras funciones en el lado del cliente. La decisión de utilizar ASP se basó en la tecnología disponible al momento de iniciar el proyecto. Sin embargo, se tiene contemplado reali-

zar futuros trabajos que permitan comparar la tecnología empleada con otras existentes.

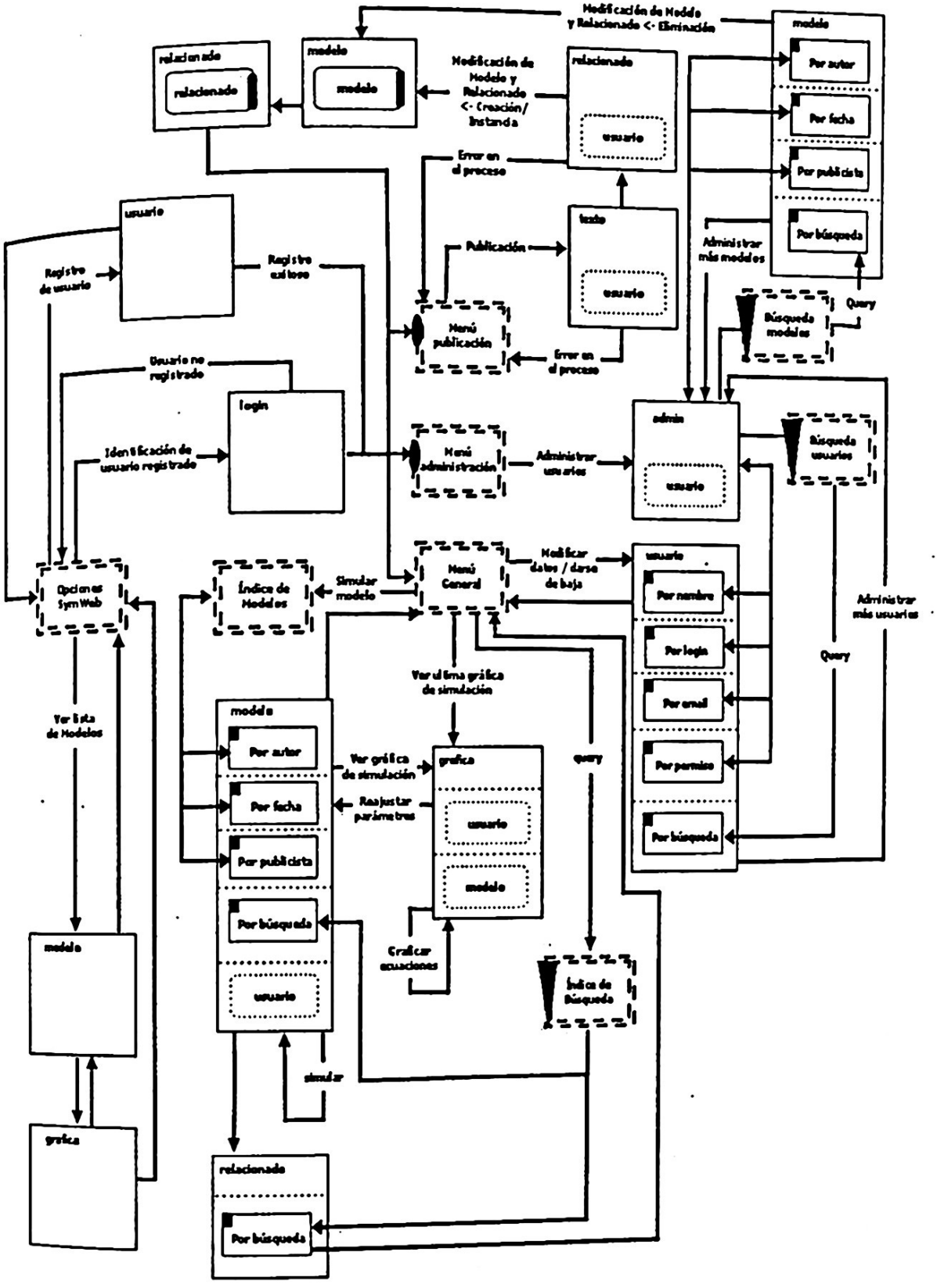


Fig. 4. Esquema de Contexto Navegacional para SymWeb

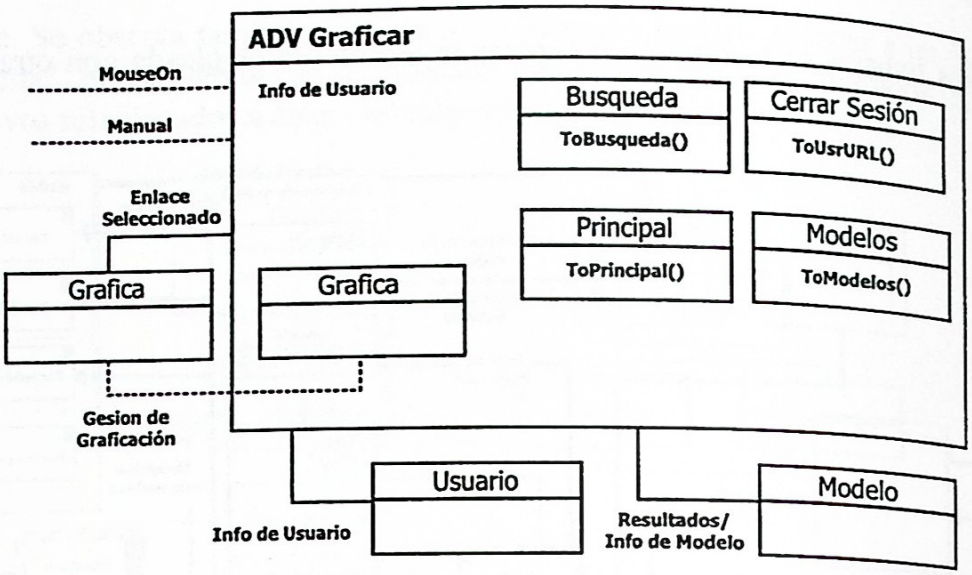


Fig. 5. Vista (sintetizada) de Datos Abstracto correspondiente a la graficación de resultados de un usuario registrado

El acceso por parte de los usuarios se realiza a través de sesiones. La entrada toda la información se hace a través de elementos de formulario, la cual es proces da siempre del lado del servidor. El proceso de publicación de un modelo se reali a través del preprocesamiento de un archivo de texto que contiene ecuaciones dif renciales generadas a partir de un software de simulación para PC [5]. Este proce se auxilia con un componente para cargar archivos en el servidor [1]. Cuando realiza la simulación de un modelo, la tabulación de resultados se establece en archivo XML que es pasado como parámetro a un componente de graficación [Se cuida que ninguna gráfica se presentada hasta que algun modelo se simule. manejo de los datos del lado del servidor se realiza utilizando tres tablas: Una pa los usuarios, otra para los modelos y una tercera para archivos que contengan inf mación sobre los modelos.

El sitio está compuesto por una raíz denominada *SymWeb* y que será el elemer final de la cadena correspondiente al URL. En esta carpeta se encuentran todas páginas a las que accede el usuario. En adición, se encuentran otras carpetas propósito específico (*e.g.* carpeta de modelos).

Se conceptualiza la GUI (figura 6) tomando en cuenta los cinco atributos de de nición de la usabilidad según Nielsen [9][10]. Se utilizó un símil de carpetas. Cu do fue necesario, se presentó al usuario retroalimentación o mensajes de error proc rando ser siempre claros. Se consideró una GUI en tonalidades verdosas debido perfil del usuario potencial y se empleó tipografía fácil de leer en monitor con el de incrementar la satisfacción subjetiva que marca la usabilidad web [9]. Incluso empleó en el diseño de interfaz los principios de la psicología Gestalt [2].

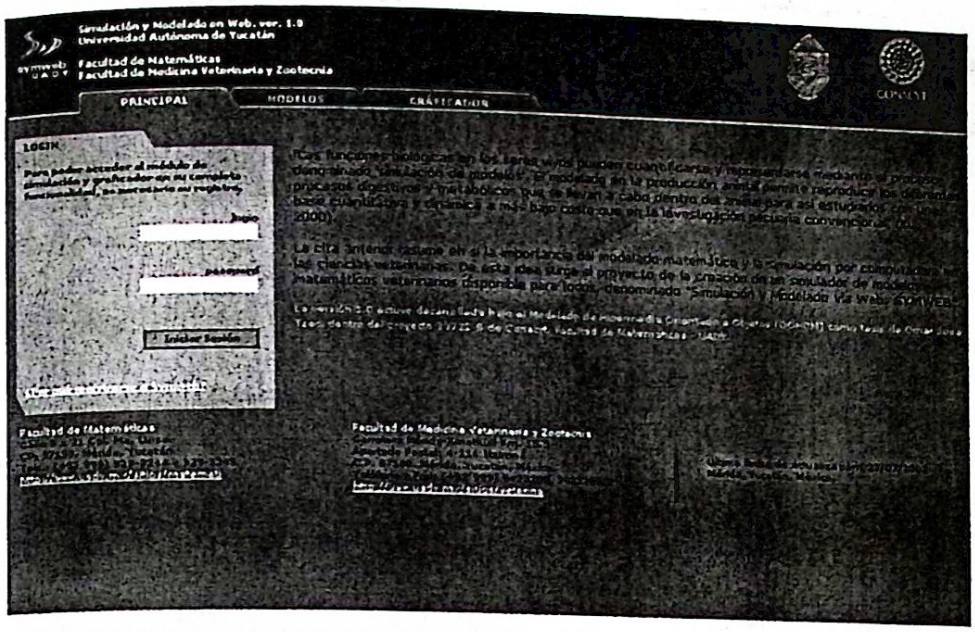


Fig. 6. Portal de acceso – Interfaz Gráfica de Usuario de *SymWeb*

5 Discusión

Para cualquier aplicación web, seleccionar una metodología de desarrollo con base en un modelo es una importante decisión. Hay que conocer las características del modelo a usar suponiendo, no garantizando, que modelará adecuadamente la aplicación. El OOHDM al ser un derivado de UML y OMT, tiene ventaja sobre otros modelos debido a que está en un contexto de desarrollo de software actual. Así, se modeló una aplicación simulación web para modelos veterinarios que es independiente de implementación; es decir, se podrá utilizar otra plataforma en un futuro utilizando este modelado. La ventaja de seguir este enfoque es que cualquier cambio a efectuarse en la aplicación derivará del modelado actual. En tal caso, se espera poder hacer un mapeo modelo a modelo de la aplicación.

Como desarrollador de software dentro de un grupo de simulación web, se tiene la ventaja de intervenir en el diseño para realizar modificaciones no consideradas por los investigadores de otras disciplinas dentro del proyecto. Para poder lograr una coperación adecuada entre la gente de desarrollo y los demás integrantes del proyecto, se acudió al Diseño Centrado en el Usuario (UCD), el cual fue desarrollado a través de constantes sesiones. El UCD casi no es considerado cuando se desarrollan aplicaciones; muchas veces el enfoque está tan solo en la técnica. En el caso de aplicaciones web como en *SymWeb*, el UCD es vital en el desarrollo, ya que el simple hecho de que el programa correrá sobre la Internet, implica que el éxito radicará en el máximo cumplimiento de los requerimientos de la comunidad global de usuarios. De la combinación de seguir el proceso ingenieril con el UCD, se observó que beneficia al proceso de desarrollo, ya que la gente encargada de esto debe estar en constante contacto con los demás integrantes y adentrarse lo suficiente en el contexto del campo de la ciencia que está relacionada con el proyecto.

En sí, lo más recomendable al enfrentar el desarrollo de una aplicación para simulación web es seguir un proceso de ingeniería de hipermedia. Además, se recomienda realizar prototipos una vez comenzada la ingeniería. En particular, aplicando el OOHDM en mezcla con el UCD, se observó que la participación del usuario brinda muy buenos resultados durante las etapas de diseño conceptual. Con respecto al diseño navegacional, éste debe ser realizado por el grupo de desarrollo y luego efectuar pruebas con el usuario sobre la navegación ya sea con prototipos o interfaces en papel. Esto implica que hay que tener presente también un buen desarrollo la interfaz desde el principio, sobre todo si los usuarios finales no están tan familiarizados con la Web, ya que además de aumentar el éxito de la aplicación, hace presentes nuevas especificaciones o se detectan limitantes relacionadas con el desarrollo, lo cual deriva en una redefinición de requerimientos o consideraciones para futuras versiones de la misma aplicación.

En la actualidad SymWeb pretender ser una herramienta que fomente el uso modelos matemáticos entre la comunidad académica veterinaria, permitiendo optimizar los recursos de la investigación tradicional, debido a que permite la utilización inmediata de los modelos publicados en SynWeb: sin requerir el uso de una aplicación comercial para realizar la simulación.

6 Conclusiones

El desarrollo de aplicaciones de hipermedia puede ser considerado bajo un enfoque ingenieril, tal y como sucede en el desarrollo de software a la medida; así, se observó que la aplicación del modelo ayudó a establecer puntos de control y realizar retroalimentación en el desarrollo. Con base en los resultados obtenidos, se puede determinar que la simulación web mediante páginas dinámicas permite una adecuada implementación dadas las condiciones de la Internet actual; se observó que este enfoque presenta grandes ventajas: 1) Administrar a los usuarios mediante sesiones, 2) Permite la personalización de la interfaz, 3) Se ajusta a los criterios de usabilidad web, 4) Es posible integrar componentes, lo cual fomenta el reuso de software.

El UCD facilitó el desarrollo de la aplicación al establecer desde el principio buen canal de comunicación con el usuario final y mantuvo en foco un diseño *hoc* a la naturaleza del proyecto (*i.e.* correr en la Web).

En la actualidad, a pesar de que la simulación web está en sus inicios, se observa que estar en un grupo de simulación web, requiere que las personas encargadas desarrollo de la aplicación posean la habilidad de sumergirse en el contexto proyecto, ya que este tipo de proyectos requieren de un enfoque multidisciplinario. También se afirma, con base en lo observado, que el grupo de desarrollo debe centrarse en el usuario y saber identificar sus necesidades.

Referencias

1. aspSmart. aspSmartUpload. www.aspsmart.com/aspSmartUpload/ (2002)
2. Dondis, D. A.: La sintaxis de la imagen. Introducción al alfabeto visual. GG Diseño. México. (1973)
3. Fishwick, P. A.: Web-Based Simulation: Some Personal Observations. Proceedings of the Winter Simulation Conference. (1996) 772-779
4. Fraternali, P.: Tools and Aproaches for Developing Data-Intensive Web Applications: A Survey. ACM Computing Survey, Vol. 31, No. 3 (1999)
5. HPS, Inc. Stella IV. URL: www.hps-inc.com (2002)
6. Infosoft. Fxgraph 4.0. URL: www.infosoftglobal.com/fXgraph (2002)
7. Ku-Vera, J.: Desarrollo de un Modelo Dinámico para la Simulación de la Producción Bovina de Carne en el Trópico. Proyecto 33722-B. Conacyt. (2000)
8. Kuljis, J.: A Review of Web Based Simulation : Whither we wander? Proceedings of the Winter Simulation Conference. (2000) 1872-1881.
9. Nielsen, J.: Usability Engineering. Morgan Kaufmann. USA. (1993)
10. Nielsen, J.: Usabilidad. Diseño de sitios Web. Prentince Hall. España. (2000).
11. Page, E.: The Rise of Web-Based Simulation: Implications for the High Level Arquitecture. Proceedings of the 1998 Winter Simulation Conference. (1998) 1663-1668.
12. Rumbaugh, M.: Object Oriented Modelling and Design. Pretince Hall Inc. (1991)
13. Schwabe D.: An Object Oriented Aproach to Web-Based Application Design. Theory and Practice of Objects Systems 4(4). Willey and Sons, New York. (1998) 1074-3224.
14. Schwabe, D.: 'Just Add Water' Applications: Hypermedia Application Frameworks. Extended abstract for the Hypertext '99 Workshop on "Hypermedia Development: Design Patterns in Hypermedia". In Press. (1999)
15. Schwabe, D.: Modeling interactions and Navigation in Web Applications. Proceeding of the World Wild Web and Conceptual Modeling '00 Workshop, ER '00 Conference. Springer. USA (2000)
16. Schwabe, D.: Web Design Frameworks: An Approach to Improve Reuse in Web Applications. Proceedings of the Second International Workshop on Web Engineering, WWW9 Conference. Springer. USA (2000)
17. Schwabe, D.: Engineering Web Applications for Reuse. IEEE Multimedia, Spring. (2001) 2-12.
18. Schwabe, D.: The Object-Oriented Hypermedia Design Model. www.telemidia.puc-rio.br/oohdm/oohdm.html (2002)
19. Sosa-Tzec, O.: Desarrollo de un Programa Web para la Simulación de Modelos Dinámicos Ligados a Base de Datos. Tesis. Universidad Autónoma de Yucatán. México. (2002)
20. Unified Modeling Language. UML Home Page. www.uml.org
21. Young, H.: Web-Based Simulation with the Remote Method Invocation. Hanyang University, South Korea. In Press. (1998)

Aplicando análisis de proximidad en Web-Mapping

Miguel F. Mata, Miguel Torres

Laboratorio de Geoprocesamiento
Centro de Investigación en Computación
Instituto Politécnico Nacional
Av. Juan de Dios Bátiz s/n Unidad Profesional "Adolfo López Mateos"
C.P. 07738, México, D.F., MÉXICO
migfel@sagitario.cic.ipn.mx, mtorres@cic.ipn.mx

Resumen. En el presente trabajo se propone un modelo para realizar análisis de proximidad a través de Web-Mapping. El modelo es una herramienta de extensión a un servidor de mapas, el cual está basado en una arquitectura de tres capas; en donde los siguientes módulos están incluidos: el Administrador de Análisis Espacial (AAE), el Módulo de Operaciones Espaciales (MOE), el Módulo de Recuperación Espacial (MRE) y una página JSP con un *applet* insertado. Estos componentes permiten la visualización de los datos y el despliegue del mapa en formato vectorial. Todos los módulos interactúan en una Interfaz Gráfica de Usuario (IGU), la cual es representada en un navegador de internet (cliente).

1 Introducción

Actualmente, los Sistemas de Información Geográfica (SIG) de Escritorio comerciales proporcionan diversas funciones de análisis espacial. Sin embargo, ante el impacto y desarrollo de la Internet en el área de la Geocomputación, la necesidad de implantar estas funciones en un servidor de Web de Mapas ha crecido de manera considerable. Es por ello, que en el presente trabajo se ha desarrollado la función de análisis de proximidad a través de Web-Mapping. En diversos sistemas de información espacial, el concepto de proximidad o *buffer*, se refiere a un polígono que encierra un área con una distancia específica de un punto, línea, nodo o polígono (conocida como área de influencia). De acuerdo a las primitivas de representación de datos espaciales (puntos, líneas y polígonos), a esta técnica se le denomina: *buffer de punto*, *buffer de línea*, *buffer de nodo* o *buffer de polígono* [1].

El análisis de proximidad se efectúa a través de la longitud del área de influencia, y éste proporciona como resultado la creación de nuevos objetos poligonales que rodean a los objetos sobre los que se realiza el análisis. Dichos objetos, normalmente se fusionan y se genera una superficie continua en la que sin solapamientos se calcula la proximidad "*buffers*" o área de influencia de los objetos determinados.

El buffer se aplica en los formatos de representación de los datos más comunes tales como: en *raster* sobre superficies de fricción (resistencia al desplazamiento) y *vectorial* sobre distancias euclidianas (en línea recta).

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 244–254, 2003. © Centro de Investigación en Computación, IPN, México.

El análisis de proximidad depende de las primitivas de representación de los datos espaciales y de la naturaleza del fenómeno a estudiar. El análisis de proximidad o buffer es uno de los más utilizados en SIG, esta técnica ha sido incluida en análisis en 3D con Java y VRML geográfico [2]. En este desarrollo, la técnica de análisis de proximidad ha implantada en una herramienta SIG de escritorio para propósito específico [3]. Los SIG comerciales proporcionan opciones en la aplicación de análisis de proximidad, tanto para la definición del buffer, como para los atributos que estarán implicados para determinar el área de influencia [4].

Por otra parte, el *Web-Mapping* es un concepto nuevo en los SIG orientados al Web, básicamente consta de las siguientes operaciones: paneo, zoom, selección de elementos y recuperación de la información a través de una página web en un modelo cliente-servidor [5]. Existen diferentes tipos de estructuras y tipos de datos espaciales en diversas ambientes tales como: bases de datos orientados a objetos, bases de datos relacionales, archivos de formato propietario etc. [8], a través de todas estas especificaciones es posible compartir, intercambiar, y transportar objetos geográficos en un sitio web [9]. Las páginas web con formato HTML carecen y limitan la capacitad para aplicar análisis espacial [10]. Por esta razón, es necesario utilizar páginas Java Server Page (JSP) con un applet incrustado. Esto con el objetivo de proporcionar al Web-Mapping la capacidad de efectuar análisis espacial [11].

Nuestra propuesta está enfocada a trabajar con análisis espacial, y en particular con los diferentes casos que implica el análisis de proximidad. Por ello, los datos geométricos y topológicos se trabajan de forma separada, de acuerdo a nuestro caso de estudio, de tal forma que mediante este mecanismo los usuarios pueden simular el funcionamiento de una aplicación SIG de escritorio.

El articulo esta organizado de la siguiente manera. En la sección 2 se presenta la arquitectura de Web-Mapping y la interacción con la función de análisis de proximidad. En la sección 3 se describe y detalla el funcionamiento de la aplicación; en la sección 4 se describe el proceso de análisis de proximidad y los diferentes tipos de análisis que existen. Algunos resultados parciales se muestran en la sección 5 y finalmente en la sección 6 se presentan conclusiones parciales del trabajo.

2 Arquitectura de Web-Mapping y análisis de proximidad

La arquitectura está basada en 3 capas, donde la capa 1 es la capa de presentación, la capa 2 es la capa de procesamiento de datos espaciales y la capa 3 es la capa del repositorio de datos espaciales. Dichas capas constan de un Módulo de Administración de Análisis Espacial (MAAE). El MAAE opera con un Módulo de Operaciones Espaciales (MOE), el cual de acuerdo al tipo de análisis a efectuar, seleccionará la función correspondiente a dicha operación. El Módulo de Recuperación Espacial (MRE) es una clase Java que se encarga de comunicar a la base de datos geográfica (vía JDBC) con el MOE.

El MRE recupera los objetos geográficos implicados en el análisis, esto para poder enviar una respuesta al Administrador de Análisis Espacial (AAE); de acuerdo a la respuesta recibida, éste configura los parámetros necesarios para que la página JSP se construya de manera adecuada. De igual manera, para el applet que se incrustará en la pagina web final. Este modelo se muestra en la Fig. 1.

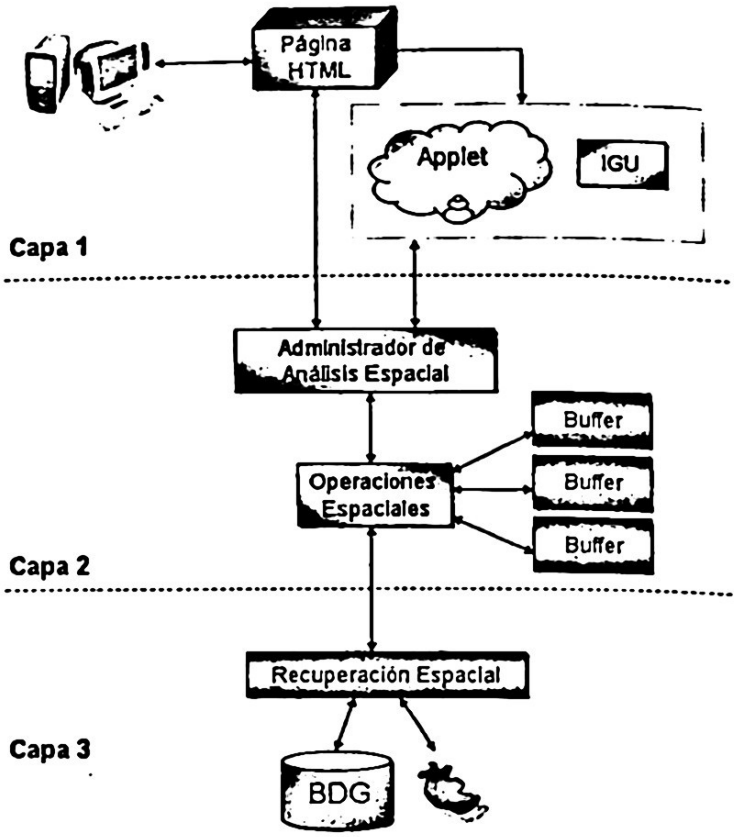


Fig. 1. Arquitectura de Web-Mapping para aplicar análisis de proximidad

La Fig. 1 muestra el Módulo de Operaciones Espaciales (MOE), el cual es el componente fundamental del sistema. Este se encuentra ubicado entre el Administrador de Análisis Espacial y el Módulo de Recuperación Espacial, cuya función principal es seleccionar el procedimiento o mecanismos necesarios para aplicar el análisis espacial requerido por el AAE. Una vez que el mecanismo ha sido seleccionado, se envía al MRE la petición de los datos espaciales y se espera el resultado. El MRE construye la consulta espacial necesaria para acceder a la información espacial en la base de datos geográfica (geométrica o topológica).

La consulta es ejecutada y el MRE envía el resultado al MOE, donde los resultados recibidos se envían al MAAE, el cual procesa los resultados y de acuerdo al tipo de representación obtenido, configura y construye el applet para que la IGU despliegue esta información en forma adecuada. El modelo general del Módulo de Operaciones Espaciales es presentado en la Fig. 2. Este módulo (MOE) es utilizado para seleccionar el tipo de operación espacial a realizar y envía los parámetros necesarios a cada función para que el análisis pueda ser aplicado. Una vez que la función es ejecutada espera la respuesta para comunicarse con MAAE.

En la Fig. 2 se muestra el funcionamiento del MOE, de acuerdo a la operación espacial seleccionada. Para este caso es un *buffer*, en donde esta función se encarga de seleccionar el tipo de primitiva de representación espacial y aplica el algoritmo de proximidad correspondiente. El resultado de dicho algoritmo es enviado como parámetro de retorno al MOE para que continúe con el proceso y se pueda realizar el despliegue final de los datos en la página web.

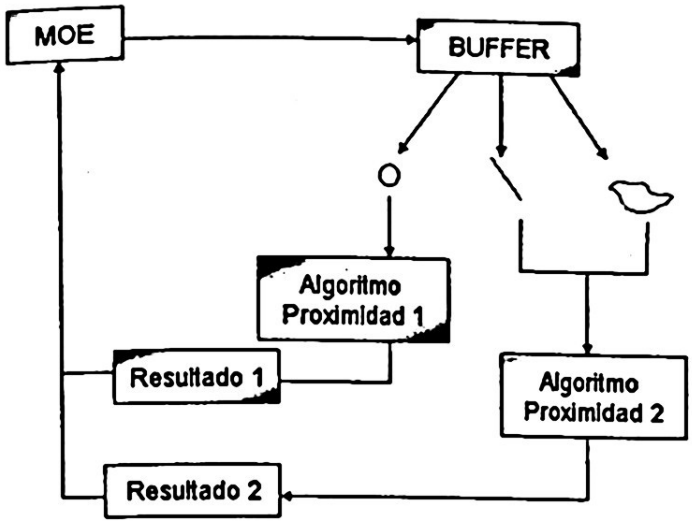


Fig. 2. Modelo de operación de MOE

3 Módulo de Análisis de Proximidad

El Módulo de Análisis de Proximidad es el encargado de realizar todos los procesos de análisis de acuerdo al tipo de primitiva que presente un objeto geográfico. Este módulo debe generar una cobertura espacial nueva en base al buffer estimado a la primitiva de representación y a los parámetros de influencia que sean considerados en el análisis. En la Fig. 3 se muestra el esquema funcional del Módulo de Análisis de Proximidad. Este mecanismo se encarga de generar la nueva capa de datos (capa buffer de datos), en base al siguiente proceso:

1. Se debe aplicar el buffer a cada capa de datos seleccionada.
2. Se debe generar una nueva capa de datos buffer.
3. Esta capa es enviada a un módulo de transformación, el cual es un Shell de Unix que hace la conversión de la capa de datos a formato de cobertura de Arcinfo.
4. Se realiza la conversión a un formato multi-capa para que pueda insertarse y presentarse como una capa de datos nueva en el sistema, el cual es posible procesar de manera transparente.

4 Proceso de Análisis de Proximidad

Algunas operaciones de SIG requieren de un punto, una línea o un polígono para poder aplicar *buffer* en los análisis. A continuación se describe los pasos del algoritmo de análisis de proximidad para el caso de primitivas espacial de tipo línea. Primero se asignan a cada segmento de línea el ancho de buffer apropiado, (fijo o por pesos). Llamaremos a este buffer distancia b; y cada segmento de línea es procesado individualmente, de tal forma que en base a la Fig. 4 , tendremos 3 segmentos de línea, donde cada segmento de línea tiene un nodo inicial (E_1, N1) y un nodo final (E_2, N2).

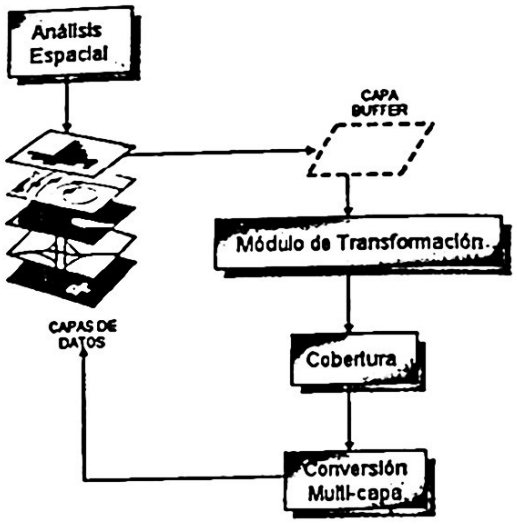


Fig. 3. Funcionamiento de la aplicación de análisis de proximidad

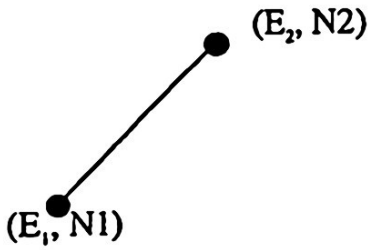


Fig. 4. Nodos iníciales de un segmento de línea

El segundo paso es utilizar estas coordenadas para calcular •x y •y entre los dos puntos a ser computarizados, esto se muestra en la Fig. 5.

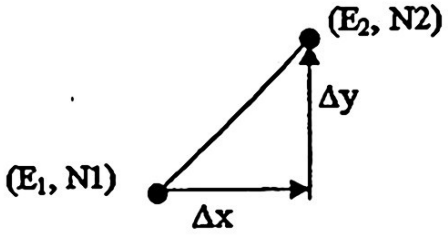


Fig. 5. Cálculo de •x y •y

El tercer paso consiste en la determinación de los puntos finales de las líneas buffer, las cuales se posicionan en cualquier lado del segmento de línea, a una distancia perpendicular b. En la Fig. 6 se ilustra este proceso las ecuaciones 1, 2, 3 y 4 determinan los valores correspondientes.

$$\begin{matrix} E_1 \pm b * Sen\left[Tan^{-1}\left(\dfrac{\Delta x}{\Delta y}\right)\right] \\ E_2 \pm b * Sen\left[Tan^{-1}\left(\dfrac{\Delta x}{\Delta y}\right)\right] \end{matrix} \qquad \begin{matrix} (1) \\ (2) \end{matrix}$$

$$N_1 \pm b * Cos\left[Tan^{-1}\left(\frac{\Delta x}{\Delta y}\right)\right]$$

(3)

$$N_2 \pm b * Cos\left[Tan^{-1}\left(\frac{\Delta x}{\Delta y}\right)\right] \quad (4)$$

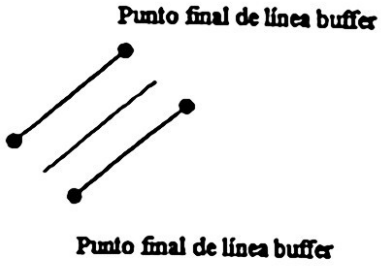


Fig. 6. Puntos finales de las líneas buffer

Es importante notar que si la línea es horizontal o vertical con respecto al eje de coordenadas, entonces las formulas deberán ser más simples. En este punto se efectúa cuarto paso, que consiste en examinar el valor de la tangente al segmento de línea, para determinar si los incrementos al este o al norte son sumados o restados dependiendo de la dirección de la línea. Una vez que las dos nuevas líneas paralelas buffer, han sido identificadas, el siguiente segmento de línea se procesa de la misma forma. Cuando las líneas de buffer paralelas para el siguiente segmento han sido identificadas, se procede al quinto paso donde las intersecciones de las líneas paralelas de buffer de cada segmento son calculadas y las nuevas coordenadas son asignadas a los vértices comunes. La prueba más sencilla para intersección de líneas puede ser utilizada en este punto. Estos pasos son repetidos hasta que último segmento de línea ha sido procesado. El sexto paso implica definir la terminación del buffer (en los puntos de inicio y los puntos finales de la línea). La Fig. 7 muestra estos puntos.

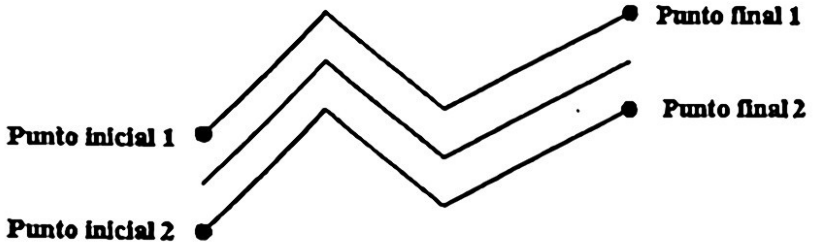


Fig. 7. Puntos finales de las líneas de buffer

Para finalizar el buffer, existen diferentes métodos, y se listan a continuación los más importantes:

- Truncado simple de las líneas paralelas de buffer.
- Adelgazamiento en la finalización del buffer paralelo para encontrar de esta forma los puntos finales.
- Encerrar el punto de inicio y el punto final de la línea con un semicírculo o un buffer de radio b (el cual se muestra en la Fig. 8).

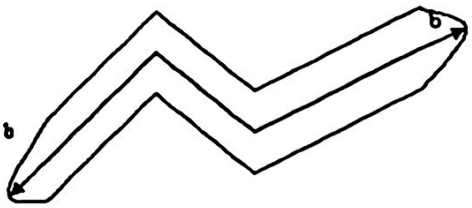


Fig. 8. Buffer de radio b

4.1 Buffer para datos tipo punto

Dentro de las 3 primitivas espaciales, hacer buffer en la primitiva espacial de tipo punto es la más sencilla, ya que el proceso solamente implica la creación de un polígono circular alrededor de cada punto con un radio igual al ancho del buffer. La Fig. 9 muestra el proceso de aplicación de buffer.

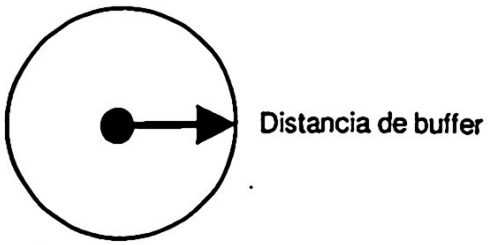


Fig. 9. Aplicación de buffer a primitiva espacial de tipo punto

Si hay múltiples puntos en la capa a la que se le aplicará el buffer, entonces se deben verificar las sobreposiciones que se produzcan. Las secciones que se sobrepongan deberán ser eliminadas, ya que el resultado de la operación será una capa de polígonos, la cual representa el área cubierta por los buffer sobrepuestos. Este proceso implica dos operaciones adicionales: *Intersección* y *Disolución.*

En la Fig. 10 se puede apreciar el resultado de la operación de intersección, los puntos de intersección son marcados gráficamente en color amarillo, para poder observar la siguiente operación de eliminación de líneas que no son necesarias en el buffer. La operación de intersección ayudará a eliminar las líneas que no serán necesarias debido a que se cruzan entre sí. La Fig. 11 muestra la operación de eliminación.

Una vez que se han eliminado las líneas de intersección, se aplica la operación de disolución, la cual genera un cuerpo sólido con las líneas que forman el buffer. La Fig. 12 ilustra el resultado de esta operación.

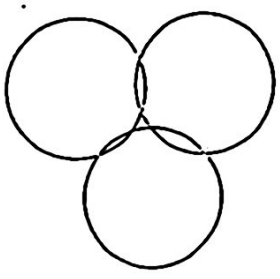


Fig. 10. Operación

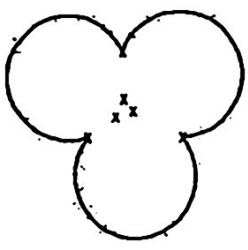


Fig. 11. Operación de

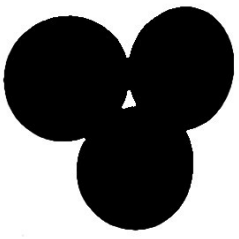
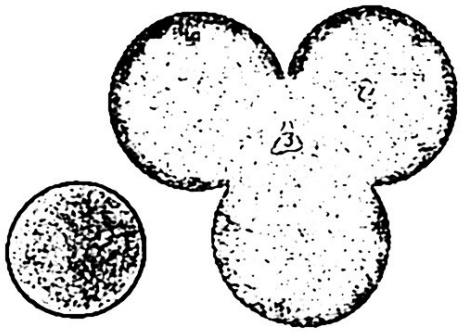


Fig. 12. Operación de

intersección | eliminación a partir de la intersección | disolución

El proceso de buffer da como resultado una nueva capa de datos de tipo polígono, cuales representan las zonas de buffer (por distancia fija o por pesos). La tabla de polígonos resultante tendrá identificadores para cada uno de los polígonos generados el proceso de buffer, y un atributo adicional que indica si el polígono en particular está dentro o fuera de la zona de influencia. En la Fig. 13 se muestra este proceso, donde 1 indica que el polígono está dentro de la zona de buffer y 0 indica lo contrario.



Punto	Distancia
1	1
2	1
3	0

Fig. 13. Resultado del buffer final

4.2 Buffer para datos de tipo polígono

algoritmo para polígonos utiliza el mismo proceso que el de buffer para datos de tipo línea. La diferencia consiste en que el buffer de polígono es generado sobre un solo lado de la línea que define al polígono. El método consiste en generar un buffer, cual rodea los límites del polígono (algunos SIG comerciales también dan la opción para generar un buffer dentro de los límites del polígono). La Fig. 14a y Fig. 14b muestran el proceso de buffer aplicado en datos de tipo polígono, como se puede apreciar el buffer se puede generar de manera interna o externa al polígono.

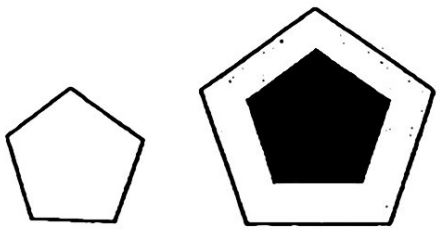


Fig. 14a. Buffer interno a un polígono

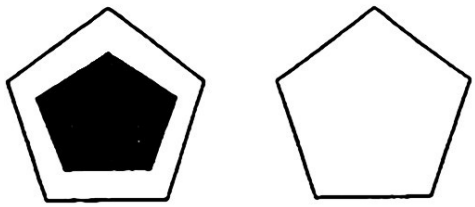


Fig. 14b. Buffer externo a un polígono

Resultados preliminares

Utilizando la aplicación SIG podemos recuperar datos espaciales de acuerdo a los parámetros requeridos. A su vez se puede aplicar análisis espacial de buffer a las apas de datos existentes para manipular el resultado con otras capas que ya han sido generadas como resultado de otros análisis espaciales. Los datos espaciales de la

Ciudad de México fueron utilizados para realizar las pruebas correspondientes al sistema desarrollado. A continuación se muestran algunos resultados de los análisis efectuados. La Fig. 15 muestra una capa espacial de México, en donde se aplica un análisis de buffer para determinar las áreas de afectación por crecimiento de ríos.

Fig. 15. Áreas de afectación por crecimiento de ríos

En la Fig. 16 se muestra el Distrito Federal. Aquí se aplicó el análisis espacial de buffer para conocer los edificios afectados por el ruido, ante la cercanía de una avenida y así poder tomar las medidas necesarias para solucionar el problema.

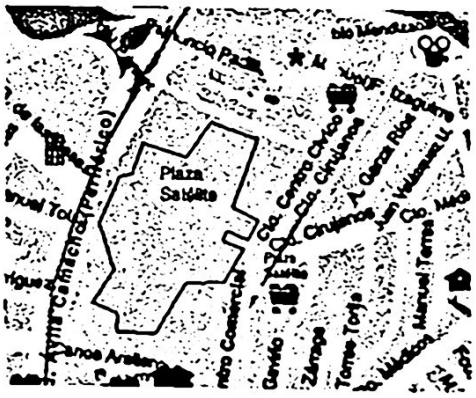


Fig. 16. Mapa del Distrito Federal donde se muestran las avenidas (color amarillo) en una cierta área, aquí se desea determinar el área de afectación por el ruido ante la cercanía de dicha calle

Otros resultados están enfocados a los segmentos de línea, los cuales representan avenidas. A estos segmentos se les aplican buffers de un peso inicial. Estos atributos han sido calculados por la presencia y ausencia de flujo de tráfico en las calles. (Por ejemplo el promedio del número de autos por minuto). Cada línea es identificada por un número o color en el SIG y la tabla indica los pesos mencionados (ver Fig. 17).

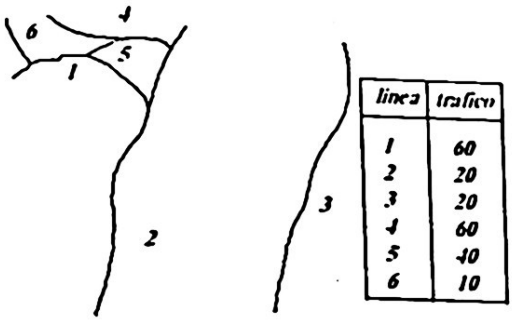


Fig. 17. Proceso de buffer para el mapa de la Fig. 16

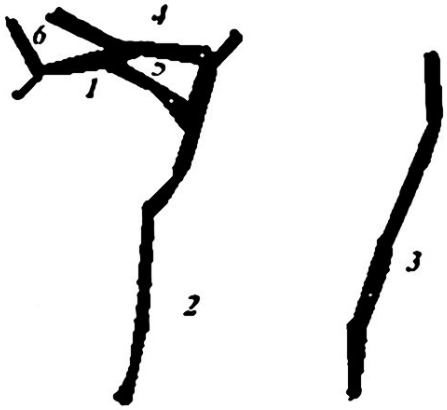


Fig. 18. Resultado del buffer donde se muestra el área de afectación por el ruido

6 Conclusiones

En el presente trabajo se ha presentado un modelo para proporcionar vía la web la capacidad de hacer análisis espacial, en particular el análisis de proximidad y sus casos de uso. Utilizando este modelo es posible analizar datos espaciales a través de un formato vectorial y los resultados de este análisis son generados en una nueva capa de datos, la cual realiza un proceso de transformación para que quede a disposición del usuario para nuevos análisis. De esta forma se ha dotado a Web-Mapping de la capacidad de aplicar análisis espacial en archivos de tipo *Shapefile* (archivos multi-capas). La especificación que definimos para aplicar el análisis de proximidad existe solo para aplicaciones de escritorio o SIG comerciales, no así para la Web, lo que implica que con este modelo es posible manejar los datos desde la Web. La posibilidad de manejar archivos en formato multi-capa proporciona la posibilidad de hacer análisis espacial sin que existan problemas de interoperabilidad, ya que la información que este formato proporciona puede ser convertida a otras especificaciones para su análisis.

El uso de JSP en Web-Mapping ofrece un mejor rendimiento en las consultas de recuperación espacial. Debido a que maneja los datos topológicos y geométricos de manera separada e individual. De esta manera las consultas que necesiten de la geometría de los datos se ejecutarán como se hace en las bases de datos tradicionales,

incrementando así el desempeño de la aplicación sin tener la necesidad de modificar la fuente de datos.

Agradecimientos

Los autores de este artículo agradecen al Instituto Politécnico Nacional y a la Coordinación General de Posgrado e Investigación por el apoyo brindado a este trabajo, así como a los revisores de este trabajo por sus pertinentes comentarios.

Referencias

[1] Moreno M., "La Generalización Automática de la Información Geográfica Multiescala", Centro de Investigación en Computación, Instituto Politécnico Nacional, México, D.F. (2001) 123-145

[2] Kyong-Ho K., Kwon L. & Jong-Hun L., "3D Geographical Analysis within JAVA VRML-based on GIS: Lantern Operation", Proceedings of International Conference on Geographical Information Systems and Spatial Theory, Orlando, Florida, USA (2000) 167-178

[3] Torres M., "Herramienta SIG de Escritorio para la Recuperación Manejo y Análisis de Datos Espaciales", Centro de Investigación en Computación, Instituto Politécnico Nacional, México, D.F., (2001) 45-61

[4] Hong Y., "Exploring Spatial Analysis in GIS", OnWordPress, Thompson Learning, USA (1997) 89-117

[5] Open-GIS Web Map Server Interface Specification: http://www.opengis.org/wms

[6] Open GIS Consortium: www.opengis.org

[7] Rigaux P., Scholl M. & Voisard A., "Spatial Databases with Application to GIS", Morgan Kaufmann Publishers, San Francisco, CA., USA (2001) 76-85

[8] Openshaw S. & Openshaw C., "Artificial Intelligence en Geography", John Wiley & Sons, New York, USA (1997) 91-123

[9] Torres M., Moreno M., Menchaca M. & Levachkine S., "Making Spatial Analysis with a Distributed Geographical Information System", Proceedings of IASTED International Conference on Databases and Applications, Innsbruck, Austria, February (2003) 1234-1240

[10] Garmash A., "A Geographic XML-based Format for the Mobile Environment", Proceedings of 34th International Conference on System Sciences, Maui, Hawaii (2001) 56-67

[11] Hardie A., "The Development and Present State of Web-GIS Cartography", International Journal of IMS, Vol. 27, No. 2 (1998) 11-26

Desarrollo de una autoridad de certificados digitales para una intranet universitaria

Dorian Omar González Cortés, Jesús Eduardo Torres Castañeda,
Petar Zaprianov Doychev y Arturo Pruneda Martínez

Escuela Superior de Computo del Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz S/N esquina Miguel Othón de Mendizával
Unidad Profesional Adolfo López Mateos
Col. Nueva Industrial Vallejo C. P. 07738, México, D. F
dorian_omar@hotmail.com, jetorresc@hotmail.com,
petar555@hotmail.com, apruneda@ipn.mx

Resumen. Un certificado digital proporciona la posibilidad de autentificar a una entidad para permitir el acceso a servicios proporcionados por diversos sistemas de información. La Autoridad de Certificados (AC), es la parte más importante de una infraestructura de llave pública (PKI), debido a que en ella descansa la confianza de toda la seguridad proporcionada por este sistema [1]. Los certificados digitales emitidos por la AC son el resultado de un proceso riguroso de verificación, que garantiza la legitimidad de la entidad certificada.

La AC requiere de los servicios de entidades auxiliares para su adecuado funcionamiento, por ejemplo una Autoridad de Registro (AR) y un repositorio de datos. En este trabajo se presenta el desarrollo de una AC, la cual provee certificados digitales que serán usados para autentificar a los usuarios de una Intranet universitaria. Se detalla cada parte que compone el sistema, además del proceso de certificación adecuado al entorno particular en el que se desarrolla el proyecto, así como políticas de seguridad necesarias para asegurar el uso adecuado de los certificados.

1 Introducción

Actualmente, el creciente uso del Internet, no solo como medio de consulta o comunicación, si no como un medio por el cual se llevan acabo transacciones, en las que se manejan datos sensibles, ha motivado un gran interés en la utilización de diversas herramientas para la protección de dichos datos.

El uso de una Infraestructura de Llave Pública (PKI) ofrece la plataforma necesaria para el uso de certificados digitales. Los certificados digitales proporcionan un tipo de identificación digital para una entidad, y por medio de éstos, una entidad A se puede identificar con otra entidad B, presentando su correspondiente certificado.

Un certificado digital es un documento emitido por una Autoridad de Certificados Digitales (AC), la cual firma este documento dándole validez para aquellas instancias que reconozcan a la Autoridad de Certificados, como una autoridad emisora de certificados de confianza.

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 255–265, 2003. © Centro de Investigación en Computación, IPN, México.

El certificado digital es una forma eficaz de distribuir la llave pública de la entidad certificada [8]; sin una adecuada distribución de la misma, el uso de criptografía asimétrica no serviría de mucho. De nada sirve que una entidad cuente con una llave pública si esta no es dada a conocer. La llave pública esta contenida en el certificado digital, y una forma de distribuir los certificados digitales es almacenándolos en un repositorio, él cual esta disponible a todas las personas para obtener el certificado de otra entidad.

Dentro de una Infraestructura de Llave Pública, la AC juega un papel central, ya sea que pertenezca a la propia PKI, o bien sea externa; por el simple motivo que AC se encarga de la gestión y emisión de los certificados digitales.

La confianza de los usuarios del sistema se deposita en la AC, y ésta debe proporcionar la garantía de que su llave privada está bien resguardada, ya que con esa llave se firman los certificados que emite. Cuando los usuarios dejan de confiar en la Autoridad de Certificados, toda la infraestructura de seguridad se viene abajo y los certificados dejan de tener validez [1].

Un certificado digital tiene un tiempo de validez finito, determinado por la AC de acuerdo a las especificaciones requeridas en el medio en el que se implementen los certificados.

La información que será incluida dentro del certificado digital, es verificada por Autoridad de Registro (AR), que son entidades autorizadas por una Autoridad Certificadora. La AR, es un puente de comunicación entre las entidades finales y la AC.

Este artículo trata sobre la implementación de certificados digitales en un entorno universitario, él cual tiene características particulares que serán tratadas en los siguientes párrafos; por lo que deben hacerse las adecuaciones necesarias para su funcionamiento. Estas características comprenden los componentes de la arquitectura propuesta, las tecnologías usadas, el proceso de certificación, así como las políticas adecuadas a este entorno y el ciclo de vida del certificado digital.

2 Características del entorno

Actualmente la Escuela Superior de Cómputo del IPN, no cuenta con sistemas de información propios para la administración de sus procesos internos. Es por eso que en la actualidad se está desarrollando un proyecto llamado "Intraescom", el cual tiene como objetivo el desarrollo de una Intranet que servirá para conectar las diversas áreas que comprende la escuela, y facilitar todo el proceso administrativo y académico dentro de ésta.

Dentro de un ambiente universitario, la implantación de una Intranet puede tener altos beneficios, cuando se toma en cuenta la reducción de tiempos en procesos que regularmente se realizan de manera manual.

Entre los sistemas que se están desarrollando, solo por mencionar algunos se encuentran: El Sistema de Control Escolar, el Sistema de Contabilidad y el Sistema de Administración de Activo Fijo.

En una Intranet, los usuarios tradicionalmente se identifican mediante un nombre de usuario (*username*) y una contraseña (*password*) para acceder a los recursos restringidos. Este método presenta la problemática del robo o falsificación de *password* cuando éste viaja por la red de manera insegura. Además, las contraseñas

suelen ser fácilmente atacadas, ya que teniendo tan solo algunos caracteres de ella, se pueden utilizar ataques conocidos como de prueba y error. Existen además otras razones por las cuales las contraseñas son un mecanismo de autentificación que no se considera seguro [2].

El proyecto de IntraEscom contempla la implementación de una Infraestructura de Llave Pública (PKI) para proveer los servicios de seguridad que requieren los usuarios y las aplicaciones. Una PKI se define como un conjunto de mecanismos criptográficos de llave pública basados en la existencia de dos llaves (una pública y otra privada) que se utilizan para garantizar la identidad del usuario, la confidencialidad y la integridad de la información transmitida [6].

Los certificados tienen la función de autentificar a las entidades que intentan entrar a un sistema, para permitir el acceso. Éstos estarán almacenados dentro de algún objeto físico conocidos como *tokens*, los cuales pueden ser un disco compacto, una tarjeta inteligente (*smartcard*), o una ikey [7]. El uso de un medio de almacenamiento portátil para el material criptográfico es necesario para que los usuarios puedan llevar su certificado correspondiente con ellos y no necesiten dejarlo en una máquina, que es donde normalmente se guarda el certificado así como la llave privada. Se pretende que el usuario desde cualquier computadora o terminal pueda usar su certificado digital, sin necesidad de copiarlos al disco duro, ya que esto representa un proceso molesto para el usuario, o bien, un posible robo.

Algunos *tokens* tienen características de seguridad que evitan el acceso a la información contenida dentro del mismo, si no se introduce una contraseña. Esto permite que a pesar de que el dispositivo de almacenamiento sea robado o perdido, no sea fácil acceder a la llave privada de la entidad certificada.

El uso de certificados digitales permite el uso del protocolo SSL (Secure Sockets Layer) para sesiones seguras, donde los datos van cifrados con una llave de sesión que acuerdan las dos entidades que se comunican. En los pasos del protocolo SSL, el servidor le presenta su certificado al cliente, este verifica la validez del certificado. El cliente puede o no presentar su certificado. Se crea una llave de sesión con un algoritmo que tanto cliente como servidor soportan., y como su nombre lo indica, solo se usa esa llave para esa sesión [5].

El protocolo SSL es soportado por los principales navegadores que existen actualmente, por ejemplo IExplorer, Netscape, Mozilla, etc. Para que la información aloja en un servidor pueda verse protegida mediante el protocolo SSL es necesario instalar un certificado digital, siendo este un requisito para establecer una conexión segura, dentro de los pasos del protocolo SSL [3].

3 Arquitectura propuesta

Los componentes del sistema son los siguientes:

- Autoridad Certificadora
- Autoridad de Registro
- Repositorio de Datos

En la Fig. 1 se muestra la interacción entre ellos.

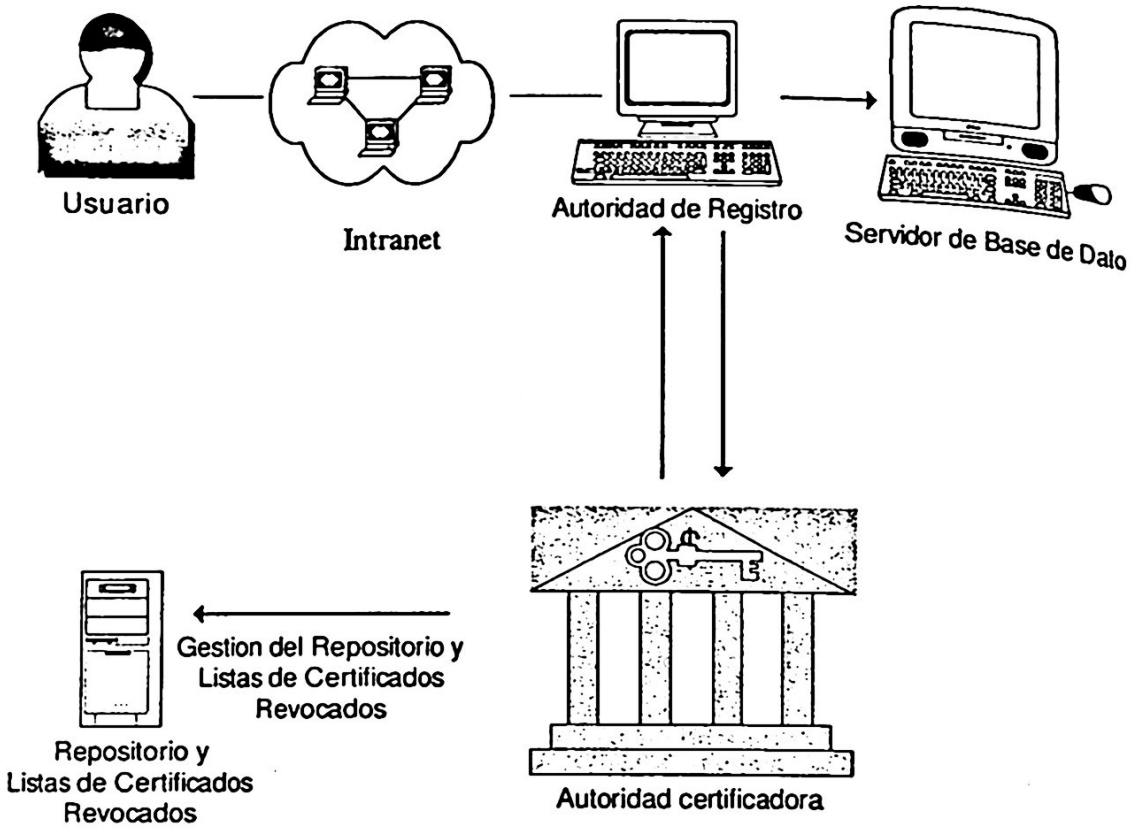


Fig. 1. Interacción entre componentes de la arquitectura

3.1 Autoridad Certificadora

La AC es la encargada de la gestión y emisión de los certificados, así como establecer las políticas adecuadas para asegurar su correcto uso. La AC esta aislada, (desconectada de la red) con el propósito de resguardar su llave privada y evitar accesos de entidades no autorizadas. Por el motivo anterior la comunicación con Autoridad de Registro es mediante dispositivos de almacenamiento físico, tales como CD-ROM, diskette, etc.

Las funciones de la Autoridad Certificadora son las siguientes:

- Emisión de los certificados de usuarios registrados y validados por la Autoridad de Registro (AR).
- Revocación de certificados (CRL - Lista de Certificados Revocados). Un certificado puede ser revocado por que los datos han dejado de ser válidos, llave privada ha sido comprometida o el certificado ha dejado de tener validez dentro del contexto para el que había sido emitido.
- Renovación de certificados.
- Distribución de certificados e información asociada a los mismos. Se debe ofrecer un mecanismo para el fácil acceso a los certificados. La forma mas empleada de publicar los certificados es un repositorio público.

La AC es operada por un administrador, el cual cuenta con una interfaz que permite emplear todas las funciones que la AC proporciona.

3.2 La Autoridad de Registro (AR)

La AR sirve de puente entre las entidades finales y la AC. Una AR tiene como principal función recibir los requerimientos de certificación, para el caso de esta Intranet, la información necesaria para hacer un requerimiento ya se encuentra capturada dentro de los sistemas que la componen, principalmente el de Control Escolar (para los alumnos), y el de Administración y Control de Personal (para los profesores y otros empleados no docentes). La AR obtendrá entonces de estos sistemas, los datos necesarios para hacer los requerimientos.

3.3 Repositorio de datos

Un repositorio de datos es una base de datos que tiene la característica de poder ser accedida fácilmente por el usuario que requiera la información.

En el repositorio de datos se almacena información que no necesita ser secreta, solo se necesita protegerla de posibles modificaciones no autorizadas. Dentro de esta información se encuentran los certificados digitales, a los cuáles los usuarios pueden acceder para verificar sus estado o bien para obtener la llave pública de otra entidad.

La información en este repositorio puede ser accedida mediante una dirección IP o un DNS, los cuáles deben darse a conocer a los usuarios al momento de la entrega del certificado.

3.4 Proceso de certificación

Dentro del entorno académico, todos los alumnos, así como personal académico y administrativo deben estar certificados por la AC. Este proceso inicia conjuntamente con el semestre escolar. En los siguientes puntos se explican los procesos de emisión de certificados, revocación y renovación

3.4.1 Creación de solicitudes de certificación

Las solicitudes de certificación son creadas por la AR, de la información obtenida de los sistemas de Control Escolar y de Administración y Control de Personal mencionados anteriormente. También se aceptan solicitudes de los usuarios mediante un formulario que se puede llenar en una pagina web y se envía a la AR Esto solo en casos en los que por alguna razón, los sistemas de Control de Escolar y el Sistemas de Administración y Control de Personal no contengan la información, o bien en el momento de la importación de datos, el registro de esa persona no era válido; se considera un registro válido aquel que contenga información de alguna persona con derecho a ser certificada, es caso de alumnos, estar inscrito, y en caso del personal, seguir laborando dentro del instituto.

La AR crea el par de llaves directamente dentro del medio físico donde serán almacenados, crea las solicitudes con la información de cada usuario. Una vez creados son exportados a la AC, para proceder a la emisión del certificado correspondiente.

3.4.2 Emisión de certificados

La AC recibe los requerimientos de certificación de parte de la AR, después Administrador selecciona aquellas solicitudes que serán firmados. Una vez firmad son exportados a la AR, para que sean almacenados dentro del *token* (ikey, tar inteligente) correspondiente, y ésta misma es la que hace entrega de los *tokens* usuarios finales. En la entrega se le pide al usuario que se identifique mediante credencial.

La Autoridad Certificadora, una vez que firmó el certificado y por lo tanto válido, lo agrega al repositorio de datos.

3.4.3 Casos de revocación

Los certificados durante el tiempo en el que son válidos, pueden ser revocados distintas causas, como perdida o robo del *token*, que la llave privada haya comprometida, o por un mal uso del certificado según lo estipulado en las políticas seguridad emitidas por la Autoridad Certificadora. También usuario puede pedir revocación a la AR, explicando los motivos por los cuales requiere revocar certificado.

Los certificados revocados se pueden consultar dentro de una lista de revocaci expedida por la AC. Esta lista de revocación esta firmada por dicha autoridad, emitida al final del día, solo en el caso de existir una revocación; en caso contrario, se emite dicha lista.

Una lista de certificados revocados solo contiene los datos relacionados con la que la creó, la fecha de creación, y la próxima actualización. Dentro de la lista revocados solo se coloca el número de serie del certificado, así como la fecha hora en la que fue revocado.

3.4.4 Casos de renovación

Los certificados se renuevan cada cambio de semestre, para esto el solicitante deb tener un registro válido en los sistemas correspondientes, ya sea alumno o personal la escuela. En el caso de los alumnos, este debe estar inscrito, por lo tanto debe alumno regular. El personal debe seguir perteneciendo al instituto.

El proceso de renovación comienza con la obtención de los registros por la AR, como se especifico en la emisión de certificados. Y los siguientes pasos son mismos que en la emisión de certificados.

También, tal como en la emisión de certificados, existe la posibilidad de solicitar renovación mediante el acceso a una pagina web, llenando todos los datos, enviándola.

3.5 Políticas de certificación

Las políticas de certificación juegan un papel muy importante dentro de un ambiente PKI donde se implementa una AC, ya que con ellas se puede asegurar funcionalidad y el uso adecuado de los certificados, así como prever posibles sanciones en el caso que no se cumpla con los reglamentos. Como ejemplo de estas políticas de certificación tenemos las siguientes:

I.- De los certificados digitales:

1. Se certificada a todo persona interna a la Escuela Superior de Computo que realice correctamente el proceso de solicitud de certificación o bien aquellas que tengan un registro válido en los sistemas de Control Escolar o Administración y Control de Personal, salvo los casos que en este mismo apartado se mencionan.
2. Se perderá el derecho de certificación por falsificación de datos durante el proceso de solicitud del certificado digital. El derecho de certificación será devuelto de acuerdo al criterio del Administrador de la AC.
3. La vigencia del certificado digital se llevará a cabo conjuntamente con el ciclo escolar (semestral). Después de este periodo el certificado es inválido, y no podrá usarse como medio de autentificación.
4. Se realizará la revocación automática del certificado en caso de:
   - Extravió o robo de la llave privada.
   - Uso indebido de información según los criterios establecidos en el Reglamento Académico.
   - El uso indebido de una llave privada ajena con o sin el conocimiento del poseedor legitimo de la llave privada.
5. La ruta de certificación para los certificados confiables dentro del sistema solo será directamente de la AC de la ESCOM.

II.- De la Autoridad de Certificados Digitales (AC):

1. La AC será operada única y exclusivamente por el administrador de la AC, que será responsable del uso correcto y protegido de la llave privada de ésta. Así como de cumplir las funciones de la AC.

III.- De la Autoridad de Registro (AR)

1. La AR será operada única y exclusivamente por el administrador de la AR. El administrador será el responsable de llevar acabo las tareas de verificación y autentificación de datos del solicitante.

IV.- Del repositorio de datos (RD)

1. El Repositorio de Datos será manejado exclusivamente por el administrador de la AC. La actualización del repositorio se realizará al final del día en caso de haber creado nuevos certificados digitales.

V.- De la lista de revocación de certificados digitales. (CRL).

1. La CRL será creada exclusivamente por el administrador de la AC y firmada por la llave privada de la AC. La CRL se actualizara al final del día en caso de haber nuevas revocaciones, o algún caso especial.

VI.- De la generación de llaves

1. El proceso de generación del par de llaves es llevado a cabo por la AR, después de haber recibido una información valida de certificación. El par de llaves es creado dentro del *token.*

VII. De la emisión de certificados.

1. Después de haber cumplido los requerimientos para una solicitud satisfactoriamente, la AC verifica la veracidad de esta solicitud, para proceder con la emisión del certificado correspondiente.

VIII. De la entrega del certificado.

1. El certificado se almacena en un medio físico (smart-card, i-key), la entidad certificada deberá acudir personalmente a recoger dicho medio físico, donde se encuentra almacenado su certificado digital y la llave privada.

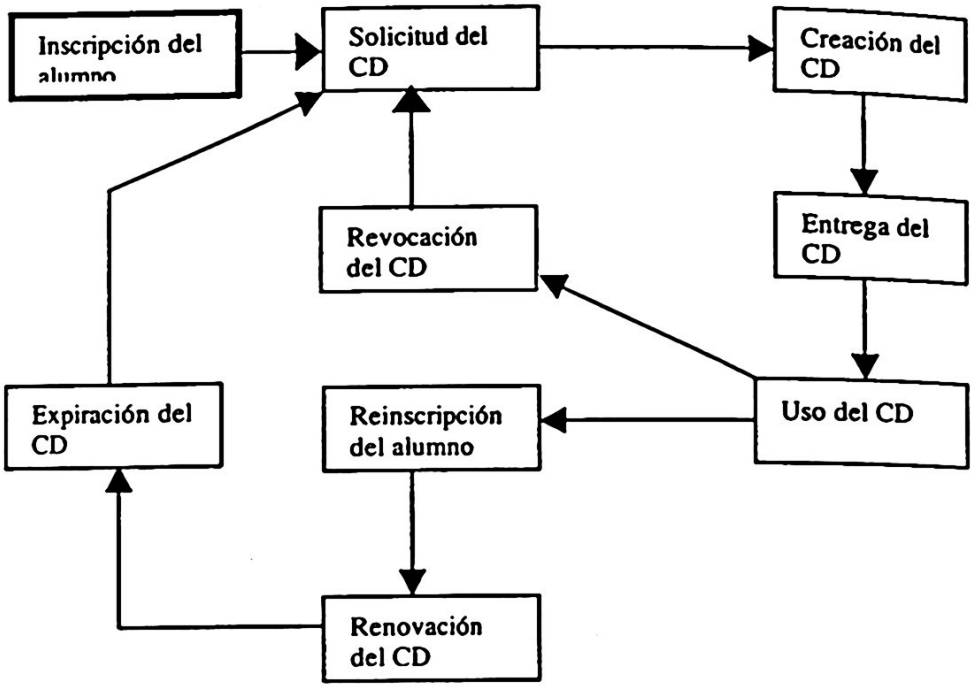


Fig. 2 Ciclo de vida del Certificado

4 Ciclo de vida del certificado digital

Los certificados digitales solo podrán pertenecer a alumnos que se encuentren inscritos en el semestre corriente, así como a personal docente y no docente internos a la escuela que se encuentren en situación regular.

1. El ciclo de vida de los certificados digitales inicia en la inscripción de los alumnos.
2. Se realiza la solicitud del mismo.
3. El Certificado Digital es creado.
4. Se entrega el Certificado Digital al dueño correspondiente.
5. A partir de este momento el certificado es valido para su uso, durante el cual podrá ser revocado.
   - En caso de revocación, se realizará nuevamente el proceso de solicitud

6. Después de la reinscripción de los alumnos, se podrá realizar el proceso de renovación de certificado digital, durante un breve tiempo antes de su expiración.

5 Desarrollo de la Autoridad de Certificados

Para el desarrollo de la AC debían de tomarse en cuenta 2 cosas: La utilización de funciones, librerías, o componentes que brindaran las funciones criptográficas que una AC lleva a cabo, y el desarrollo de las interfaces para administrar esta AC.

Para realizar las operaciones criptográficas se utilizaron las funciones que brinda OpenSSL, que es un popular proyecto de código abierto disponible en Internet. OpenSSL ofrece la posibilidad de crear el par de llaves (pública y privada), requerimientos de certificación, certificados digitales y lista de revocación.

OpenSSL maneja algoritmos como el RSA y DSA para la creación de llaves, la longitud de estas se le pueden indicar mediante las opciones que contiene en sus comandos. Usa el estándar x.509 para la creación de certificados, este estándar ha sido aceptado por la International Telecommunication Union-Telecomunication Standardization Sector (ITU-T) y por ISO/International Electrotechnical Comisión (IEC) [4].

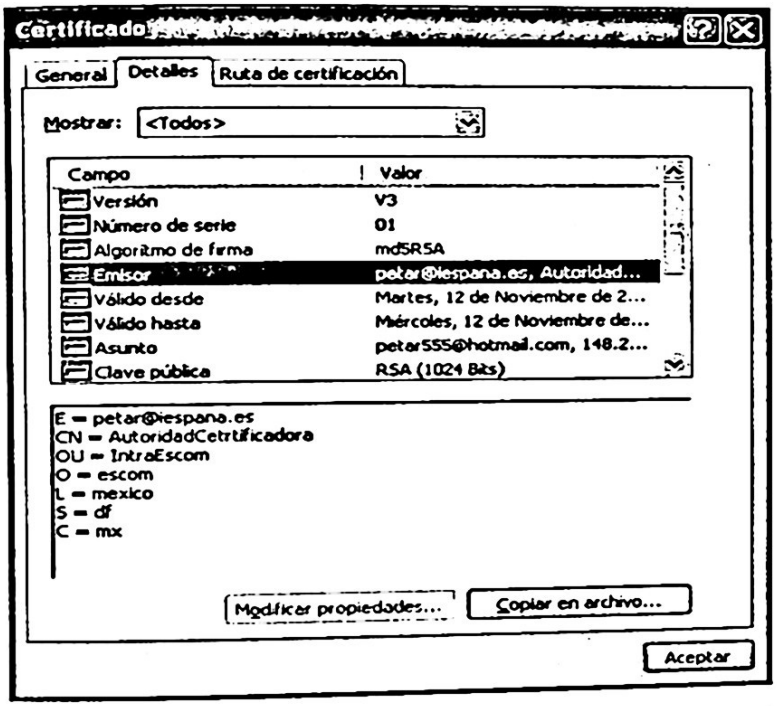


Fig. 3. Ejemplo de un certificado creado con la AC de ESCOM

La estructura del estándar x.509 contiene los campos necesarios para poder identificar a la entidad certificada. Un ejemplo de certificado digital puede verse en la siguiente figura 3.

La interfaz de la AC, se desarrollo como un software *standalone*, utilizando el lenguaje de programación Java sobre una plataforma Linux Mandrake 9.0. Las

interfaces implementan las funciones de OpenSSL, él cual habitualmente se usa mediante línea de comandos.

En la figura 4, se muestra una interfaz usada por el administrador de la AC, donde tiene la posibilidad de elegir en los requerimientos de certificación proporcionados por la AR. Después de elegir los requerimientos deseados, son firmados, la interfaz muestra los certificados que ya han sido firmados, así como el porcentaje de avance.

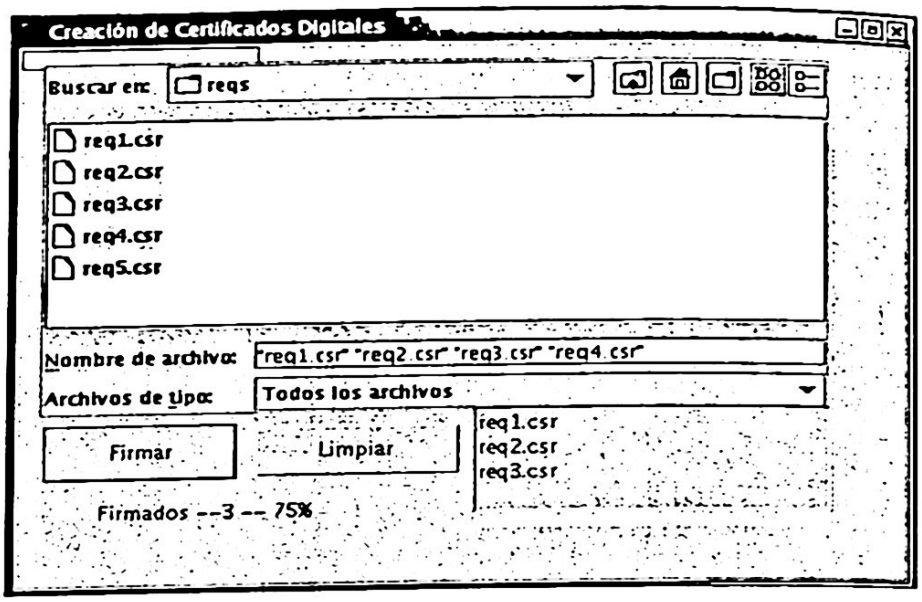


Fig 4. Interfaz de firmado de la AC

Las interfaces de la AR son mediante páginas web, que pueden ser visualizadas través de un navegador. Como se ha mencionado la AR no esta aislada, y mediante este tipo de interfaces se puede operar a distancia, aunque es preciso comentar que administrador de la AR debe contar previamente con un certificado digital para poder operar la misma. La AR utiliza un servidor web Apache sobre la plataforma Windows.

6 Conclusiones

El contar con una Autoridad de Certificados propia permite tener un mayor control os procesos de certificación, así como la creación de políticas de seguridad adecuadas l entorno, y de esta manera, implementar el nivel de seguridad deseado.

El uso de los certificados digitales genera un nivel alto de seguridad en cualquier mbiente, siendo además la base para el desarrollo de una PKI, que se vislumbra omo una de las soluciones de seguridad más completas y utilizadas en los próximos ños.

Los procesos que la AC lleva acabo para la certificación, deben ser planeados studiados con sumo cuidado, para que las entidades pueden confiar en los ificados que esta AC emita. Así también es importante la protección de la llave privada de la AC, donde recae toda la seguridad de una infraestructura de seguridad, donde una AC es la parte central.

Referencias

Pruneda A., "Desarrollo de una infraestructura de llave pública para aplicaciones de comercio electrónico seguro", CIC–IPN, México D.F. 2003, (M.S. Thesis).
Manual Pons Montorrel, "Control de accesos",
http://www.criptored.upm.es/guiateoria/gt_m013d.htm.
Certificados de seguridad SSL Server, Servidor Seguro Datacom,
www.datacommultimedia.com/certificados_seguridad_.htm
OpenSSL, "The Open Source toolkit for SSL/TSL", www.openssl.org/
HTMLWeb, Seguridad, Transacciones seguras en Internet.
http://www.htmlweb.net/seguridad/ssl/ssl_6.html.

[6] Menchaca, Pruneda., "Desarrollo de una Infraestructura de Llave Pública (PKI) para comercio electrónico", Congreso Internacional de Computación, CIC-IPN. México D.F., 2002.

[7] Rainbow Technologies: ikey 2000, http://latinamerica.rainbow.com/ikey/ikey2000.html

[8] Carlisle Adams, Steve Lloyd (1999), "Understanding Public –Key Infrastructure". Macmillan Technical Publishing.

Análisis y empleo de herramientas OLAP en el sistema de información corporativa de la CFE

Liliana Argotte, Norma Jácome, Manuel Mejía

Instituto de Investigaciones Eléctricas, Gerencia de Sistemas Informáticos,
Ap.Postal 1-475, 62001, Cuernavaca, Morelos, MÉXICO,
Tel / Fax. (01 777) 3 18 26 58
{largotte, nejacome, mlavalle}@iie.org.mx

Resumen. Se presentan, aplican y comparan las herramientas comerciales Oracle Express y Oracle Discoverer. Ambas fueron aplicadas en el contexto del Almacén de Datos que la CFE desarrolla denominado Sistema de Información Corporativa (SICORP), el cual involucra el manejo de grandes volúmenes de información histórica y actual. Dada la importancia y el auge que actualmente tiene el tema de Almacenes de Datos, las experiencias que se describen en el artículo son de gran utilidad para futuros desarrollos.

1 Introducción

1.1 Los sistemas de información corporativa

El estar bien informado a veces resulta una meta difícil de alcanzar, sobre todo cuando las empresas se desarrollan y crecen al ritmo de los tiempos actuales. Es partir de esta necesidad que surgen los Sistemas de Información Corporativa (SIC). Un SIC es el enfoque de la informática actual que busca apoyar al ejecutivo moderno en la toma de decisiones estratégicas de la empresa, proporcionándole información de calidad, resumida y oportuna a partir de los grandes volúmenes de datos actuales e históricos con que ya cuenta la organización; un SIC se integra con varias tecnologías y aplicaciones, como las redes locales y amplias, los manejadores de bases de datos, las herramientas de visualización de datos y otras, cuyo empleo dependerá de cada negocio, por lo que un SIC tendrá características muy particulares para cada organización. Lo que hace importante el concepto es que hoy día su implementación es perfectamente factible debido a los cambios tecnológicos en materia de hardware y software ocurridos en los últimos diez años. Entre otros, estos avances incluyen [1]:

- El surgimiento del Procesamiento Analítico en Línea (On Line Analytical Processing - OLAP) y su aplicación más allá de las áreas tradicionales de marketing y finanzas.
- El crecimiento de la computación Cliente/Servidor, que ha creado servidores de hardware y software más poderosos y sofisticados que nunca. Los servidores de

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 266–275, 2003. © Centro de Investigación en Computación, IPN, México.

hoy compiten con los mainframes de ayer y ofrecen arquitecturas de memoria tecnológicamente superiores, con procesadores de alta velocidad y capacidades de almacenamiento masivas.
- La aparición de modernos Sistemas de Gestión de Base de Datos (Data Base Management Systems – DBMS´s) que proporcionan mayor soporte para las estructuras de datos complejas.
- El uso generalizado del Internet e Intranet, acompañado de nuevas y más potentes aplicaciones.

1.2 El sistema de información corporativa de la CFE

La idea de crear un SIC para la Comisión Federal de Electricidad (CFE) surge en la Subdirección Técnica de la CFE. La Subdirección Técnica, a través de la Gerencia de Informática y Telecomunicaciones (GIT), es responsable de apoyar los objetivos estratégicos de la CFE, mediante la implementación de soluciones corporativas en materia de informática, que provean de información oportuna, integral y confiable a los ejecutivos de la institución, a fin de mejorar los procesos de planeación, operación y gestión de la misma. En virtud de lo anterior, la GIT elaboró un plan maestro de sistemas informáticos que integró acciones a mediano plazo y en el que se identifican proyectos que requieren de un gran esfuerzo y de la participación de todas las áreas de la CFE.

Tal es el caso del Sistema de Información Corporativa (SICORP), con el que se está creando un esquema de acceso a los datos de las diversas Subdirecciones de la CFE, concentrando información histórica y actual que apoye al proceso de toma de decisiones y que contribuya al incremento de la productividad en las áreas usuarias. Así, el SICORP será una herramienta indispensable para la planeación y gestión de la empresa en el nuevo milenio.

1.3 Almacenes de datos y herramientas de análisis de la información

Dentro de los distintos elementos o tecnologías que pueden conformar a un SIC destaca de manera importante el del Almacén de Datos (Data Warehouse), porque provee un ambiente para que las organizaciones hagan un mejor uso de la información que está siendo administrada por diversas aplicaciones operacionales ya existentes años atrás [2]. Un Almacén de Datos (AD) es una colección de datos en la cual se encuentra concentrada e integrada la información institucional y se usa como soporte para el proceso de toma de decisiones gerenciales. Esta integración de información no se logra mediante una simple copia de los datos operacionales, sino que se obtiene a través de un modelado especial de los datos y de procesos específicos de extracción, limpieza, transformación e inserción de datos [3].

Estrechamente ligadas al concepto de SIC y AD, se encuentran las modernas herramientas para interrogar, analizar, presentar, visualizar y descubrir conocimiento en los grandes volúmenes de información contenidas en estas tecnologías. En efecto, en tanto que el enfoque de SIC nos proporciona un marco general para la

conceptualización de sistemas de soporte en la toma de decisiones estratégicas y AD es el medio para la concentración de los datos de interés corporativo, las Herramientas para el Análisis de la Información (HAI) son el software que permite a los ejecutivos interrogar, buscar y visualizar, de manera sencilla, aquella información que apoye la toma de decisiones gerenciales.

Las HAI son una pieza fundamental en todo SIC, pues es la parte que el usuario final ve y con la que interactúa: la evaluación y elección de la (o las) HAI puede representar el éxito o el fracaso de un proyecto de SIC, pues el grado de aceptación del sistema depende en gran medida de que el usuario encuentre útiles y entendibles las HAI que se le proporcionen para trabajar. Por ser el componente con la cual ejecutivo accesa la información se le conoce como el front-end del sistema (en forma análoga, el back-end se refiere a los procesos que el usuario final no ve y que son los responsables de extraer, limpiar, transformar e insertar los datos en el AD).

Dentro de las HAI se encuentran distintas categorías de acceso: los reportes, aplicaciones, las de análisis y las herramientas para consultas-reportes a la medida.

En cuanto a los reportes estándar, son vistas oficiales (institucionales) de información, en formatos prediseñados y que son emitidos de manera periódica para distribución a un grupo generalmente extenso de usuarios. Esta es la forma más tradicional de proporcionar información en la empresa, sin embargo el contenido estos reportes es muy distinto a los reportes generados por los sistemas operacionales, pues están dirigidos a la alta gerencia, más que a los mandos medios encargados de la operación cotidiana de la organización.

Las aplicaciones están constituidas por software con interfaz del tipo apropiado para su uso con el ratón de la computadora (aplicaciones de acceso push-button) generan un conjunto relativamente pequeño de reportes clave orientados a una munidad usuaria muy específica de ejecutivos de alto nivel, que realizan consultas en la pantalla de su computador.

Las herramientas para consultas-reportes a la medida permiten a los usuarios crear sus propios reportes usando manejadores de consultas basadas en SQL como QMF, QBE, MS-Query o como las que contienen productos como Excel o Access.

La categoría de acceso denominada de análisis incluye aplicaciones sofisticadas para búsqueda de información y son las de más reciente integración a los SIC. minería de datos es una de estas aplicaciones, con la que se busca la información importante que se encuentra oculta e implícita en los datos, sin realizar interrogaciones específicas: se deja que el algoritmo minero encuentre qué relaciones, tendencias o patrones de comportamiento existen o existirán, y que permitirían mejorar los procedimientos que actualmente sigue la organización. Algunas de las técnicas computacionales usadas para realizar la minería son las redes neuronales artificiales, la inducción de árboles de decisión, las reglas de decisión, y los métodos basados casos similares (nearest neighborhood). También en esta categoría se encuentran facilidades de visualización, los paquetes de análisis estadístico avanzado, las facilidades de navegación por los datos (data-surfing), la inteligencia artificial, las simulaciones y las tecnologías para el procesamiento analítico en línea (OLAP). Esta última es típica en los AD, pues cuenta con la habilidad de examinar la información en varios niveles de resúmenes a lo largo de múltiples dimensiones del negocio

definidas en el AD) mediante una interfaz flexible e intuitiva; permite al usuario explorar distintos enfoques de búsqueda, dejándolo ir de lo muy general a lo muy particular (drilling down) o visceversa, lográndose realizar consultas no predefinidas y complejas, con gran facilidad, pues el usuario actúa como si rebanara y o uniera un cubo de información, capacidad que se conoce como pivoteo del dato o slice and dice.

La tecnología OLAP es el tema central del presente artículo. En las siguientes secciones se presentarán y compararán dos herramientas comerciales que fueron aplicadas en el SICORP, resaltándose las experiencias obtenidas, las cuales son de gran utilidad para desarrollos futuros.

2 La herramienta Oracle Express

2.1 Descripción

Oracle Express [4] es a la vez una base de datos multidimensional y un ambiente de desarrollo para construir aplicaciones OLAP que faciliten el análisis y la toma de decisiones. Almacena datos como objetos multidimensionales y puede acceder a una o más bases de datos Express para presentar información o bien a una base de datos relacional (RDBMS) cuando es necesario (Figura 1). Se manejan 2 tipos de conexiones a bases de datos relacionales: Direct Oracle Call Interface (OCI) y Open Database Connectivity Interface (ODBC).

El lenguaje utilizado en esta tecnología se denomina Express Language, el cual es un lenguaje de procedimientos almacenados (Stored Procedure Language, SPL), usado por las aplicaciones de desarrollo para almacenar, procesar, recuperar y navegar en los datos de las bases de datos Express.

La tecnología Express es implementada en 2 productos:

- Oracle Express Server 6: desarrollado para plataformas Microsoft Windows NT y Unix; servicio multithreaded; soporta múltiples instancias, usuarios y sesiones simultáneas; provee acceso a clientes remotos y locales.
- Personal Express 6: desarrollado para plataformas Windows 95 o Windows NT; aplicación multithreaded; atiende sólo un usuario; provee acceso sólo a clientes locales; soporta sesiones múltiples simultáneas.

2.2 Aplicación en el SICORP

Debido a que la CFE cuenta con licencias corporativas de Oracle, se optó por trabajar con Oracle Express Server 6.2.0 para el desarrollo de la aplicación OLAP, además de que Express Server y las aplicaciones Express, constituyen una inversión segura, que tendrá el soporte, mantenimiento y disponibilidad por muchos años de Oracle, con la seguridad de poder migrar la aplicación hacia los servicios OLAP que Oracle vaya abriendo en el mercado.

En SICORP se utilizó Oracle Express como Multidimensional OLAP (MOLAP), sin embargo es un producto híbrido que permite ambos modos: MOLAP y Relational OLAP (ROLAP) [5].

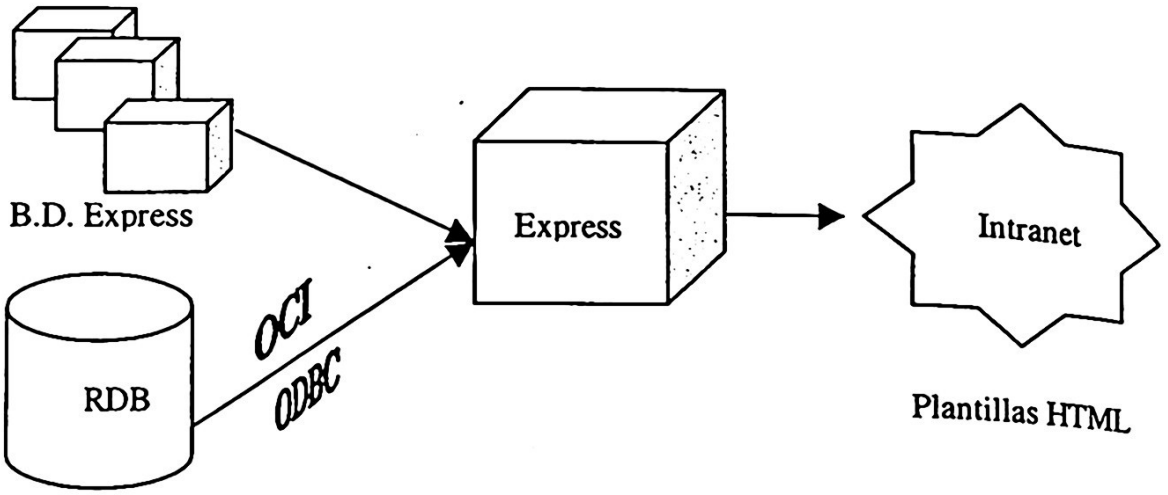


Fig. 1. Flujo de información en Oracle Express

El módulo que se utilizó para crear y cargar las dimensiones y hechos (variables en el lenguaje de Oracle) fue Express Administrator. Una vez creados los cubos de datos en Express, se hicieron plantillas html con código embebido del lenguaje Express para crear vistas en forma de tablas y gráficas que pudieran ser desplegadas en Web. Las facilidades que el usuario tiene en el manejo del cubo son las que Express proporciona, como el drilling down y slice and dice.

Empleando Express, durante el año 2001, se crearon alrededor de un centenar de cubos de áreas de la CFE como: Generación, Distribución, Comercial, Recursos Humanos, Capacitación y Finanzas.

Actualmente, la tecnología de bases de datos de Oracle y Express Server convergen para formar los Servicios OLAP Oracle 9i y llegar a ser la plataforma líder de aplicaciones analíticas en Internet [6].

Express Server 6.3 provee poderosas aplicaciones analíticas con capacidades para predecir, planear y dirigir. Por más de 20 años, Express Server y sus predecesores, Classic Express y Express MDB han sido líderes en la industria de servidores OLAP, por su desempeño y capacidades analíticas. Express Server ha mantenido su liderazgo, porque ha evolucionado con las necesidades analíticas a lo largo del tiempo y con las plataformas de desarrollo (primero en mainframe, después en Cliente/servidor y actualmente en Internet) [9].

2.3 Ventajas y desventajas

Ventajas:

- El módulo Express Administrator provee una interfaz sencilla para crear y mantener bases de datos de Express.

- Es fácil de usar y entender (se requiere de una a dos semanas de capacitación).
- Permite crear dimensiones a partir de archivos planos o de tablas de una base de datos.
- Permite crear jerarquías en las dimensiones.
- Puede tener múltiples variables dimensionadas.
- Permite aplicar fórmulas tanto en las agregaciones, ó sumarizados, como para cargar datos en una variable.
- Realiza automáticamente sumarizados (Rollup's).
- Permite aplicar criterios en el formato (color) de las cifras, para llamar la atención del ejecutivo para casos de especial interés.
- Se pueden realizar consultas a la base de datos Express mediante los siguientes módulos:
  - Express Analyser,
  - Express Web Publisher,
  - Express Objects,
  - Escribiendo un programa SPL.

Desventajas:

- No fue posible cargar datos mediante un programa OCI en una variable con más de 6 dimensiones, ya que no se podían ver los valores aún después de ejecutar los sumarizados (Rollup's), y aún cuando Oracle ofrece que la herramienta está diseñada para soportar más de 30 dimensiones.
- El tiempo de respuesta de Express al presentar los datos de los cubos, tanto en Express Administrator como al momento de crear las vistas con el lenguaje Express es lento, del orden de 3 minutos en una máquina conectada a la red local.
- A la fecha no se ha encontrado la forma para evitar la carga de applets en Web, al momento de desplegar un cubo en la máquina del cliente.
- Permite aplicar sólo una fórmula en los agregados, siendo que existen casos en la CFE que se requiere aplicar distintas fórmulas dependiendo del nivel de agregación..
- Actualmente no se ha logrado configurar un esquema de seguridad (privilegios de acceso a los datos), es decir, la herramienta no prevé esta necesidad.
- Tiene un costo elevado respecto a competidores como SQL Server.

3 La herramienta Oracle Discoverer

3.1 Descripción

Discoverer es una herramienta de consulta orientada al negocio. Permite al usuario, dentro de los diferentes niveles de la organización, obtener acceso inmediato a la información del negocio para la toma de decisiones. Soporta análisis multidimensional de consulta rápida y poderosas búsquedas de datos. Tiene una capa de metadatos que simplifica y mejora la visualización de los datos de los usuarios.

Está formado por tres componentes: un módulo para administradores, un módulo para usuarios finales y una capa de metadatos. El módulo de administradores sirve para crear y mantener los metadatos que tienen como finalidad facilitarle al usuario final el análisis y la construcción de vistas e informes. En el módulo para usuarios finales se puede llevar a cabo el análisis y desglose multidimensional para tener una información más detallada sin conocer la estructura de la base de datos fuente. Se puede tener un mapa visual de la base de datos y llevar a cabo consultas sofisticadas sin programación. Los datos se desglosan a través de diferentes niveles de resumen para descubrir información detallada del negocio. Los resultados de las consultas son desplegados en una variedad de vistas (tablas y gráficos) fáciles de comprender y sólo con la información necesaria para cada nivel directivo. Y por último la capa de metadatos sirve para ocultar la complejidad de la base de datos de tal manera que el usuario interactúa con sus datos pero en términos de su negocio, en su propio idioma. Dentro del módulo de administradores se crea esta capa.

Aunque algunos autores [5] consideran a Discoverer como una herramienta del tipo Desktop OLAP (DOLAP), nuestra experiencia en la explotación del SICORP nos hace incluirla dentro de las herramientas de tipo ROLAP de acuerdo a la definición que hace de ella Oracle [7,8].

3.2 Aplicación en el SICORP

Discoverer versión 3.1 Cliente/Servidor [7,8] fue la herramienta utilizada como ROLAP para la explotación del SICORP de la CFE. Se partió de una base de datos relacional diseñada en Oracle 8i que contiene diferentes estrellas (modelado multidimensional) que modelan cada una de las áreas que conforman a la CFE. Las áreas que se trabajaron con Discoverer fueron: Generación Hidroeléctrica, Gerencia Comercial, Recursos Humanos y Administración de la Distribución. Para cada una de ellas se implementaron una o más estrellas.

Cada área de la CFE fue creada en Discoverer como un área de negocio diferente a la que sólo los usuarios involucrados pueden acceder. Utilizando el módulo de administradores de Discoverer se creó para cada una de ellas toda la capa de metadatos, esto es, se definieron todos los elementos necesarios para ocultar la complejidad de las estructuras de la base de datos al usuario, y éste pudiera hacer todos los análisis y consultas de información en su propio lenguaje y sin hacer programación, teniendo diferentes niveles de detalle de la misma. También con este módulo se crearon todos los sumarizados necesarios.

Para cada área se definieron, con el módulo de usuario final, una serie de vistas predefinidas que le permiten al usuario revisar y analizar toda la información extraída y resumida de la base de datos. Estas pueden ser modificadas fácilmente por él mismo. Discoverer cuenta con una interfaz muy intuitiva que le permite al usuario crear otras vistas, además de las predefinidas, a la medida de sus necesidades. Esto se logra con un buen diseño de la capa de metadatos a través del módulo de administradores.

3.3 Ventajas y Desventajas

A continuación se describen las ventajas y desventajas que se tienen al utilizar Discoverer como una herramienta ROLAP.

Ventajas:

- Provee una perspectiva de negocios mediante la agrupación de información en áreas de negocios lógicas.
- Las complejas estructuras de la base de datos son vistas como una lista de objetos familiares para el usuario.
- Presenta la información a través de distintos formatos (tablas y gráficos), fáciles de comprender y sólo con la información necesaria para cada nivel directivo.
- El usuario puede acceder a vistas predefinidas, de la información de las diferentes áreas del negocio, y modificarlas.
- Permite la creación y parametrización de nuevas vistas de la información.
- El usuario puede manipular los resultados de las consultas de diferentes maneras: cambiando los ejes (dimensiones), moviéndose entre jerarquías, manejando excepciones, condiciones y cálculos.
- En una misma vista se pueden tener distintas variables (data point's) dimensionadas.
- El usuario puede hacer formateo de datos, exportación de datos, y creación de gráficos.
- Se pueden tener datos sumarizados.
- Proporciona al usuario final acceso de sólo lectura para proteger la importante información de la base de datos fuente.

Desventajas:

- Su forma de presentar los datos depende completamente de la estructura que éstos tengan en la base de datos relacional de la cual parten.
- No construye de manera natural jerarquías, si éstas no existen modeladas en la base de datos como copos de nieve (tablas jerarquizadas).
- No se pueden hacer análisis complejos de los datos. Soporta preguntas tácticas más no estratégicas.
- Si no se usan datos sumarizados (previamente), el tiempo de recuperación de los datos dentro de las vistas del usuario es lento, del orden de segundos a un minuto.
- El control de acceso (seguridad) lo proporciona a través de las áreas de negocio, no a nivel de celdas.
- En las consultas que contienen un promedio de 10 dimensiones el tiempo de recuperación de los datos es lento, del orden de segundos a un minuto.
- Se tiene que adquirir una licencia por cada usuario que lo utilice, pues es una herramienta Cliente/Servidor.

4 Cuadro Comparativo

Tabla 1. Cuadro Comparativo

Característica	Oracle Express	Oracle Discoverer
Uso de la herramienta en Sicorp	MOLAP	ROLAP
Necesidad de Bases de Datos intermedias para dar velocidad a las consultas	Se crean cubos MOLAP	Se crean Bases de Datos de Agregados.
Accede información de otras Bases de Datos	B.D. Express / RDBMS	RDBMS
Plataforma requerida	Win NT / Unix	Win 95 / NT
Responde a cuestionamientos	Estratégicos	Tácticos
Vistas	Tablas y gráficos en Web.	Tablas y gráficos.
Creación de vistas	Predefinidas, no las puede alterar el usuario.	Predefinidas, el usuario puede realizar cambios y puede crear nuevas vistas.
Navegación en las vistas	Drilling down, Slice and dice	Drilling down, Slice and dice
Presentación de los datos	Depende de la estructura de la base de datos Multidimensional.	Depende de la estructura de la base de datos relacional.
Licencias	Únicamente para el administrador de la B.D.	Se requiere licencia por cada usuario.
Máximo número real de dimensiones	Hasta 6	Se probaron hasta 6
Esquema de seguridad	No lo prevé	Sólo por vista (no por celdas)
Tiempo para presentar las vistas al usuario	Hasta 3 minutos	Hasta 1 minuto

5 Conclusiones

El vertiginoso crecimiento de las empresas modernas ha exigido a la informática crear nuevas herramientas para el apoyo a la toma de decisiones, surgiendo así, entre otras, una gran variedad de herramientas OLAP, desarrolladas por distintas empresas de software.

Una de las empresas líderes en el ramo es Oracle. En este artículo se analizaron dos herramientas OLAP de Oracle, detectándose que si bien son herramientas poderosas, aún no cumplen en su totalidad con las necesidades actuales de una organización de gran tamaño, como lo es la CFE.

La apreciación del usuario sobre Oracle Express es que realmente modela una base de datos del tipo multidimensional con todas las ventajas de análisis en línea que esto tiene, a diferencia de otras herramientas (como Brío, de Brío Intelligence) que aunque están orientadas al mismo tipo de uso, en realidad solo crean tablas

pivote sin la funcionalidad completa de un verdadero OLAP. Lo anterior significa que el usuario aprecia tener Oracle Express pues puede realizar muy diversos análisis, a nivel macro y micro, de manera sencilla, con intranet en línea, lo cual asegura la oportunidad de la obtención de respuestas a los cuestionamientos estratégicos.

6 Agradecimientos

Los autores desean agradecer la importante labor de apoyo, organización y promoción del proyecto realizada por los Ings. Jorge Hernández Aguilar, Gerente de la GIT, Ricardo Hernández Rodríguez, Subgerente de Informática de la GIT y Oscar Miranda. También se reconoce la participación en las distintas fases del SICORP del personal del IIE: Dr. Gustavo Arroyo, Gerente de la GSI, Dr. Guillermo Rodríguez, Jorge González, Ricardo Molina, Israel Galván, Rosa Mondragón, Alfredo Martínez, Víctor García, Hilda Escobedo, Felipe López, Javier Martínez, Octavio Reynaga, Humberto Hernández, Norma García, Gustavo Bravo, Raúl García, Luz Ocampo, Yazmín Hernández, Miguel Pérez, Carlos Herrera, Benjamín Zayas y Reynaldo Alanís.

Referencias

[1] Gill, H. S., Data warehousing, Prentice Hall Hispanoamericana, S.A., 1996.

[2] Weldon, J. L., "Data warehouse building blocks", Byte, enero de 1997, pp. 82-102.

[3] Kimball, R., The data warehouse toolkit, John Wiley & Sons, Inc., 1996.

[4] Personal Express, Installation and configuration Guide, Release 6.2, 1998.

[5] (http://www.olapreport.com/Segments.htm).

[6] (http://www.rockportsoft.com/oraclesod.html)

[7] Oracle Discoverer for Administrators, Release 3.1, 1998.

[8] Oracle Discoverer for End Users, Release 3.1, 1998.

[9] http://www.oracle.com/ip/analyze/warehouse/collateral/olap_sod.html.

Sitios Web de interés:

- Olfalito, "Construcción de un data warehouse", 192.41.50.238/rincon/ficheros/dataware.zip
- The data warehousing information center, www.dwinfocenter.org/
- Welch, J.D., An introduction to data warehousing, www.datawing.com/intro2dw/intro.htm

Technical Recommendation Sheets (HRT) for Support of Software Development Process

Gustavo Arellano, Hanna Oktaba

Maestría en Ciencias e Ingeniería en Computación
Universidad Nacional Autónoma de México
Ciudad Universitaria, México D.F. 04510
gustavo.arellano@metasoft.com.mx

Facultad de Ciencias
Universidad Nacional Autónoma de México
Ciudad Universitaria, México D.F. 04510
ho@fciencias.unam.mx

Abstract. Due to the great diversity that exists in the development of software, it is difficult to establish standard criteria of implementation for all projects. Each development has unique characteristics that make it different from any other. A form to acquire development capacity is through the experience and the accumulation of elements or re-usable techniques in new projects. This document proposes an incremental mechanism of acquisition of capacity of development through an artefact (the HRT's) that should be present in every stage of the development.

1 Introduction

A software development process is a guide that defines an organized form to generate software defining development phases, rolls, activities and artefacts to be generated during this process.

In general, a software development process assumes that the development team has certain degree of technical knowledge for the suitable implementation of the functional aspects of the system. Also it supposes that the technology to use independent of the development in terms of the process since this one does not vary the technology changes. In the practice, a development process it is used as administrative guide and as a times control tool for human resources but as reference of how the activities that this one defines, are due to make.

For the accomplishment of this work, three development processes were analyzed (TSPi, Team Software Process[1]; The Unified Software Development Process[3], and eXtremme Programming[2]). This was made in order to find similar elements the proposal of the present article. Only in eXtreme programming[2] was found element called "spike" that suggests to do a previous investigation and test the development or the implementation of certain complex functionality that arises during the development. However, this strategy does not offer a mechanism to the registry

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 276–282, 2003. © Centro de Investigación en Computación, IPN, México.

this information for its reuse and does not propose a mechanism for tracking the proposal.

In the following sections it is proposed to include, to the development process of an organization, an artefact that will register technical proposals and will include information for administrative aims and tracking. Also it is suggested a format for control management and another one for the resultants to these proposals. Finally, the automation of this proposal is recommended and, at the end, it is exposed an example of the utility based in four projects in which this mechanism was used.

2 Proposal of technical recommendations sheets

The technical recommendations sheets (in the successive HRTs) are an addition to the set of artefacts to be generated in a project. These indicate what type of technical activities must be considered or be carried out during certain phase of the development process by a certain roll, rolls or the full team. A HRT can, at any time of the development, be presented / displayed by any member of the team, for its evaluation and later approval for its execution. The main objectives are the following ones:

i. To support to the risk mitigation plan generated in some phase, or to mitigate risks that were identified later to this plan.
ii. To serve as a base or reference for the decisions to be taken in the development of the system.
iii. To serve as a technical reference in later developments and, in this way, decrease times and costs, reusing the knowledge generated previously.
iv. To increase the capacity of development of the organization by reducing the dependency of specialized human resources.

3. Description

The technical recommendations sheets are documents that define a formal frame of administration and tracking proposals that suggest doing activities that will facilitate, in the development, the decision making and/or the mitigation of identified risks and, at the same time, the recovery of knowledge generated in previous developments for their use in the present development. Each HRT will include administrative data:

- Reference. (Sequential number which defines, in a unique way, each technical recommendation sheet).
- Name of the project to which it belongs.
- Phase and cycle in which the recommendation was generated.
- Creation date.
- Deadline for execution.
- Author of the recommendation.

Responsible for execution.

Approval (Team members signatures who agree in consensus about the execution of the recommendation).

In a similar way, each HRT will present/display technical data:

Origin (reason that motivated the proposal).
Proposal (the recommendation itself) This one must meet the following criteria:

a. To suggest activities non contemplated in the scripts of the organization development process.
b. Is due to evaluate the positive impact of the proposal in the project or the development process and to conclude that this one is representative.

Justification (reasons that supports the proposal).
Strategy (explicit way in which the proposal will be done).
Benefits.
Disadvantages.
Impact in quality (Quality characteristics -according ISO/IEC 9126- that the proposal promotes in the development)
Involved Technologies.

Tracking

In order to provide a formal frame for administration and tracking the different recommendations, a control format it is proposed to register the necessary things for such effect. This it is the registry for control and tracking of HRTs (that will be called RCSH format in the successive). Next, the elements that will be included in this format are listed and explained:

ID. Unique Identifier of the HRT.

IDR. Identifier of the document or documents that were product of the recommendation.

Date of creation. Date in which the recommendation was answered.

Responsible for execution. Acronym and roll of the human resource assigned to the execution of the task.

Status. One of the following indicators:

a. Proposed.
b. Approved.
c. Rejected.
d. In process.
e. Finished.

Observations. Additional information that complements the documents that were generated in the execution of the HRT.

Resulting Documents

each technical proposal corresponds one or several resulting documents that must properly organized and classified for their later recovery. In this way, the organization will increase the maturity in the software process production.

Additionally, in these documents it is registered how to make certain activities that could be necessary to repeat in the future. The record format has five attributes:

ID. Unique Identifier that along with the attribute part, forms a key.
Part. Consecutive associated to each ID, useful for the identification of sub documents with a same ID.
Date of generation. Date in which the document were generated.
Location of archives. Physical location of the resulting archives of the execution.
Report of results. A general description of the final result of the recommendation

Automated Control

With the purpose of offer a control automated process for the administration of the HRTs and additionally to generate a knowledge base with information about how they arise and solve certain technical aspects, it is presented a tool whose objective is to support and to facilitate this kind of activities.

The HRTs administrator or AHRTe, (as it will be mentioned from now) it is a tool of authentificated access that permits:

- To define projects and participants.
- To assign rolls.
- To generate, to assign and to reassign HRTs
- To solve HRTs.
- To give pursuit to the development of the HRTs.
- To generate reports.
- To define and to execute consultations on HRTs.

The AHRTe will keep and recover information from a repository central of data and could be acceded from any equipment that has access to network Internet.

7 Application

This proposal has been proven successful with four developments that have similar characteristics. In this document they will be called To, B, C and D respectively. Next, appears a brief description of each one of them:

Project A. Web system for query information based on sheets of definition of search criteria and sheets of presentation of results with hyperlinks of expanding such information.

Project B. Web system for capture and query information for the support in the evaluation of the qualitative and quantitative characteristics of a software product.

Project C. Web system for capture and query information for the support to the generation, tracking and administration of HRTs.

Project D. Web system for capture and edition of information related to the estimate the value of a social interest house.

Next a summary of the information obtained in these projects tries to justify the four objectives of the present work:

- In the four projects were used 100% of the proposed low support classes generated the project A HRTs. (this supports the fulfillment of objective iii)
- The development team for each project was different, but the development process that integrated this proposal was the same one, so this is independent of the knowledge and experience of specialized developers. (this supports the fulfillment of objective iv)
- If the number of HRTs generated for project X is n(X), then it has been observed that:

$$n(A) > n(B) > n(C) > n(D)$$

  (this supports the fulfillment of objective iv)
- If the number of HRTs reused for project X is r(X), then it has been observed that:

$$r(A) < r(B) < r(C) < r(D)$$

  (this supports the fulfillment of objective iii)
- The following HRT, that was generated in the project A, was reused in the other projects, and served to mitigate a risk (what supports the fulfillment of objective i) common to all the these projects:
  - Origin. It is not known which data base management system will be the definitive one in the production environment.
  - Proposal. To prepare the system so it can be adapted to an arbitrary database management system.
  - Justification. This proposal is a form to mitigate the identified risk. It will save codification time at the moment of selecting the definitive DBMS.
  - Strategy. To implement a class whose objective is to translate SQL strings between different DBMS.
  - Benefits
    i. The system acquires portability.
    ii. Facility of scalability, update and maintenance.
    iii. Minimization of integration errors with different DBMS.
    iv. This proposal is re-usable in other projects.

- Disadvantages
    i. Investment of time in the implementation of this automatic process of translation.
    ii. It is required knowledge of several SQL dialects.
- Impact in quality.
    i. Maintainable.
    ii. Robust.
    iii. Scalable.
    iv. Portable.
    v. Modular.
- Involved Technologies
    i. Java
    ii. SQL

Finally, the following HRT served as a base for the decision making on the project A (this supports the fulfillment of objective II) and later, it was reused in projects B, C and D.

- Origin. The architecture of the system through the selection of an suitable architectonic pattern to the technology is due to define that will be used
- Proposal. To use architectonic pattern MVC.
- Justification. Architectonic pattern MVC adapts itself to the technology to use since this one offers feasible elements of being used like Views, Models and Controller of natural way
- Strategy. To implement the Vista with elements JSP and HTML, the Model with Java Beans and the Controller with Servlets
- Benefits
    i. Attachment to a recognized architectonic pattern.
    ii. This pattern can be reused in other similar projects.
- Disadvantages
    i. Investment of time in the assimilation and maturation of a new architectonic pattern.
    ii. It is required of solid knowledge of Java and HTML.
- Impact Quality
- Mantenible
- Robust
- Modular
- Involved Technologies.
    i. Java.
    ii. HTML.

8 Conclusions

The integration of the technical and technological context in a development process will allow developing software in a more and more systematic way avoiding redoing work and reducing the dependency of specialized human resources. With this technique it is expected that, for later projects, the number of HRTs for the analysis, design and implementation phases diminish, if the use or purpose of software is similar in these new developments.

In general it is expected that, after the accomplishment of several projects (of different types) the HRTs of the subsequent developments can be useful in others and thus to generate less HRTs every time. In other words, this generation of recommendations will decrease as the organization is releasing more and more systems and possibly, will be minimum the number of recommendations to generate, given the acquired knowledge.

In the practice, has been observed that this strategy of inclusion of technical recommendations to the course of the development has been successful and it is proposed as a future work to obtain statistics of more developments so they can reaffirm this proposal.

The disciplined inclusion of technical recommendations to the set of documents that integrates a development facilitates the reusability of the same ones in later developments and it reduces the dependency that exists between an organization and the specialized human resources.

Another aspect important to consider in favour of this process of knowledge acquisition is that it reduces in an important way the curve of learning and investigation for the new human resources that are integrated to the organization. The previous thing impacts directly in the quality of the product, since this mechanism allows to analyze and in its case, to update or to improve a proposal in order to generate later, superior products.

Additionally, this mechanism is independent of any process of development. For this reason, it is easy to integrate to the work of development of any organization.

References

1] Introduction to the Team Software Process. Watts S. Humphrey Ed. Addison Wesley 1997.

[2] Extreme Programming. Kent Beck. Ed. Addison Wesley 2000.

[3] The Unified Software Development Process. Ivar Jacobson, Grady Booch, James Rumbaugh. Ed Pearson, 2000.

Diseño de un portal cultural basado en patrones de diseño para el Web

Plutarco Galavíz Carmona [1], Jaime Muñoz Arteaga [2], Carmen Santiago Díaz [3]

[1] Universidad Autónoma de Tlaxcala (UAT), Depto. de Ingeniería y Tecnología, Calzada Apizaquito S/N, Apizaco, Tlaxcala, México
culturatlaxcala@prodigy.net.mx

[2] Instituto Nacional de Astrofísica Óptica y Electrónica(INAOE), Depto. de Ciencias Computacionales, Luis Enrique Erro No. 1, Sta. María Tonanzintla, Puebla, México
jaime@inaoep.mx

[3] Benemérita Universidad Autónoma de Puebla(BUAP), Facultad de Ciencias de la Computación, 14 sur y Av. San Claudio C.U. Puebla, México
carmen@cs.buap.mx

Resumen. La civilización del siglo XXI se define hoy día como la "civilización de la información". Esta civilización se apoya en la utilización masiva, para el conjunto de las actividades económicas, sociales y culturales, de una ciencia, la informática, de una tecnología, la de redes de comunicación, y de un instrumento específico, la computadora. Gracias a las posibilidades que ofrecen las redes, se puede generar un medio alternativo de difusión. Para el sector cultural, los por-tales son una vía importante para publicitar local, regional y mundialmente, la organización, servicios o procesos culturales de una región. En este artículo pro-ponemos la construcción de un portal cultural basado en patrones de diseño para el Web. El objetivo es detectar la problemática que existe actualmente en el diseño de sitios web utilizando de manera adecuada los patrones de diseño, con el fin de satisfacer las necesidades de los usuarios.

1 Introducción

Hay que reconocer, desde un primer momento, que la informática, sobre todo Internet y, en particular, la Web, tienen capacidad para cambiar el mundo y las relaciones humanas, y que el ciberespacio, es decir, el espacio en el que se llevan a cabo estas relaciones, genera cambios en nuestra concepción de espacio-tiempo, en nuestras formas de representación y en nuestro lenguaje. En el ámbito económico existen varios factores que empujan al incremento del uso y de las utilidades de Internet. El primero de ellos es la reducción de costos de los equipos de cómputo y su generalización tanto en el trabajo como en el hogar. El parque de computadoras está creciendo en círculo virtuoso con Internet, cada vez se compran más computadoras con el señuelo de la conexión a la red y ésta a su vez se fortalece con la expansión de las

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 283–293, 2003. © Centro de Investigación en Computación, IPN, México.

terminales capaces de navegar en ella. El segundo factor es la reducción de costos de conexión a Internet. Como es sabido, una de las claves de la difusión del nuevo sistema de comunicación es la posibilidad de transmitir información a larga distancia con el costo de una llamada local.

Todos los sectores de nuestra sociedad: educación, salud, comunicación, economía y por supuesto cultura, se han beneficiado de los avances científicos y tecnológicos. Actualmente en la mayoría de las dependencias gubernamentales tienen dirección de correo electrónico, lo que permite tener una mejor, gestión, coordinación y principalmente mayor comunicación.

Los portales son sitios de gran movilidad de información en donde se aprovecha al máximo las bondades del Internet. La mecánica de los portales es diferente según sea el tipo de información que se ofrece, pero la meta final siempre es la publicidad o servicios. La mayoría de las veces los portales no tienen un solo dueño, sino que se trata de equipos en donde se encuentran distintos niveles de inversionistas y en donde se utilizan todo tipo de recursos on-line y periféricos para llegar a los navegantes [7]. Otro de los aspectos importantes de los portales es referente a la gran cantidad de contenidos y aplicaciones que le dan funcionalidad a los usuarios y los invitan a regresar con frecuencia.

Existe una clasificación básica de portales: Portales generales (horizontales) y los portales especializados (verticales)[6]. Los primeros manejan contenidos muy generales, los cuales son los ideales para los usuarios que comienzan a navegar por ejemplo: Yahoo, Lycos, etc. Los portales especializados son para públicos un poco más específicos sobre un tema especializado por ejemplo: formula1, chistemania, etc.

Podemos decir que para el sector cultura, los portales son una vía importante para publicitar local, regional y mundialmente la organización, servicios o procesos culturales, además no solamente mejoran la calidad y eficiencia en la difusión de la oferta cultural, sino que se reducen costos de impresión, reproducción y distribución de una gran cantidad de información, se permite mantener actualizada e integrada la información para la toma de decisiones y finalmente tienen un gran poder de gestión [1].

Actualmente en el Web nos encontramos con algunos sistemas carentes de un esquema y una estructura de organización, lo que provoca: una navegación inconsistente, uso indebido de la rotulación, vínculos inválidos, uso indebido del hipertexto, etc. Con base en lo anterior actualmente existe una serie de patrones de diseño para el Web, con los cuales si se aplican de manera adecuada, se logrará satisfacer las necesidades cada vez más exigentes del usuario.

Tomando en cuenta que un patrón es una forma de describir una solución a un problema recurrente bajo un determinado contexto [11], en el presente trabajo se propone hacer una categorización de patrones principalmente en los aspectos referentes a navegación, elementos de página, mecanismos de búsqueda y rotulación.

El presente trabajo se compone de 6 secciones: La primera es la introducción, en la segunda se aborda la problemática existente en el diseño de los sistemas en el Web, en la tercera se muestran los módulos que componen la arquitectura donde se ven las particiones funcionales del portal; además se hace una clasificación de pa-

trones de diseño para el Web, en la cuarta se muestra un caso de estudio haciendo una descripción de los patrones aplicados, en la quinta se muestran los trabajos relacionados a través de una tabla, indicando el grupo de patrones que se aplican para cada trabajo y en la sexta sección se dan una serie de conclusiones.

2 Problemática de diseño de los sistemas en el Web

El aspecto de los actuales sistemas en el Web tienen poco que ver con los sistemas de la primera mitad de los 90´s, los años iniciales de la Web. Entonces, casi todos los sistemas tenían un fondo gris, texto escrito de lado a lado en un solo bloque y algunas imágenes. No era, evidentemente, la rica experiencia visual de hoy en día. De hecho, actualmente muchos sistemas en el Web contienen imágenes, sonido y video, por lo que el diseño y desarrollo de este tipo de aplicaciones y de su interfaz dependerá el éxito o fracaso de los sistemas en el Web.

De acuerdo con los trabajos [2] y [9] algunos de los problemas que se pueden cometer en la construcción de un sistema en el Web son:

- uso indebido del sistema de rotulación,
- ausencia de un esquema de organización,
- ausencia de una estructura de organización, lo que provoca una navegación inconsistente,
- vínculos inválidos (internamente y externamente),
- sitios que requieren de un navegador específico,
- no poner información de contacto
- no encontrar la información del sitio web,
- diseño deficiente de la interfaz gráfica,
- diseño del sistema sólo para satisfacer la necesidades de los que construyen el sistema y no para los usuarios finales,
- uso indebido del hipertexto.
- uso indebido de sonidos y animaciones,
- sitios que están en construcción.

En el Web no es difícil encontrar documentos que ofrezcan a los interesados en el tema de la construcción de sistemas en el Web, una amplia variedad de recomendaciones para la implementación de sistemas en el Web [10]. Para evitar los problemas anteriores, en los trabajos[3] y [4] describen una serie de patrones de diseño para la construcción de sistemas en el Web.

El presente trabajo recomienda el uso de patrones de diseño para la construcción de entornos web enfocados a los sectores culturales y turísticos, en este caso se aplican a un portal cultural, porque se han detectado una serie de problemas tales como:

- *Programación Cultural.* Existe información cultural que tiene una gran dinámica, la cual se le denomina programación cultural (cartelera), esta se genera mensualmente. La población que se entera de dichas actividades es a través de los medios de comunicación tradicional y estas acuden a ciertas actividades cultura-

les, obviamente dependiendo del gusto y la inclinación hacia ciertas disciplinas de las bellas artes(danza, teatro, literatura, música y artes plásticas). En base a lo anterior por ejemplo, cuando una persona tiene interés sobre un curso o taller de enseñanza artística, la persona tiene que trasladarse a la organización, dependencia o institución para pedir información sobre el trámite o gestión que esta debe hacer para beneficiarse del servicio.

- *Proyección de los creadores de una región.* Por otro lado tenemos la falta de proyección de los creadores(teatreros, danzantes, *músicos*, pintores, escultores, artesanos, titiriteros, escritores). Actualmente no todos tienen las posibilidades de dar a conocer su trayectoria o el tipo de obra que realizan.

- *Costumbres y tradiciones.* En algunos lugares existe una serie de riquezas en costumbres y tradiciones por lo que no existen los *mecanismos* para que se den conocer de manera local, regional, nacional y mundialmente.

- *Costo de la difusión de los eventos.* Actualmente algunas instituciones públicas privadas que realizan alguna actividad, tienen la necesidad de difundirla a través de los medios tradicionales. En el caso de la *presentación* de una obra de teatro, independientemente de los diferentes requerimientos técnicos que se necesiten para el evento, se realiza un proceso de difusión. Para esto se tiene que generar toda una campaña publicitaria a través de carteles, volantes, invitaciones, spot's inserciones en periódicos, lo cual genera un gasto excesivo.

El conjunto de problemas muestran la necesidad de hacer un esquema de organización y una estructura organizacional para contextualizar los contenidos culturales, apoyándose en el diseño y uso de patrones de diseño tal como lo recomiendan algunos trabajos [3], [4] y [12].

En la siguiente sección proponemos una clasificación de patrones de diseño para entornos web.

3 Patrones de diseño para el Web

El desarrollo de sistemas interactivos en la Web implica una necesidad del usuario través de los componentes interactivos de la interfaz gráfica del sistema, lo que ha motivado el surgimiento de modelos arquitecturales que especifiquen los componentes de la Interfaz de Usuario. Darle el control al usuario sobre una aplicación es una tarea difícil, por tal razón han surgido varios modelos arquitecturales con la intención de modelar los componentes involucrados en el diálogo humano-computadora. Durante la etapa de especificación de diseño en el proceso de concepción, se define el modelo arquitectural que describe la manera como se llevará a cabo el diálogo en la Interfaz de Usuario. Toda Aplicación Interactiva se compone de un grupo de particiones funcionales, cada partición tiene una tarea específica, independientemente del modelo arquitectural empleado es posible identificar las particiones funcionales tales como: núcleo funcional, adaptador del núcleo funcional y el diálogo.

Existe una variedad de modelos arquitecturales para modelar aplicaciones, en este caso aplicaremos un modelo trifásico inspirado en el modelo de Seeheim, el cual se presentó en 1985 dentro de una nueva categoría de herramientas denominada UIMS (User Interface Management Systems)[5]. El modelo plantea la comunicación entre el usuario y la aplicación estructurada en tres niveles, como se muestra en la figura 1:

- *Primer nivel (información).* En este nivel se encuentran toda la información que se ofrece en el sistema.
- *Segundo nivel(servicios).* En este nivel están los servidores necesarios para el funcionamiento del sistema.
- *Tercer nivel(interacción).* En este nivel se lleva a cabo la activación y retroalimentación a través de los navegadores.

Fig. 1. Modelo trifásico inspirado en el modelo arquitectural de Seeheim [5].

El portal propuesto entra dentro de la clasificación de portales especializados[6], es decir, en este se maneja información específicamente cultural y turística, donde la tarea principal del portal es satisfacer las necesidades del usuario a través del modelo propuesto en la figura 1; es decir la información que está en el primer módulo de alguna manera se tiene que gestionar para ofrecerla al usuario. Para esto en el módulo de servidor tenemos todos los servicios que se consideran necesarios para atender las necesidades de información cultural, todo esto a través de los navegadores.

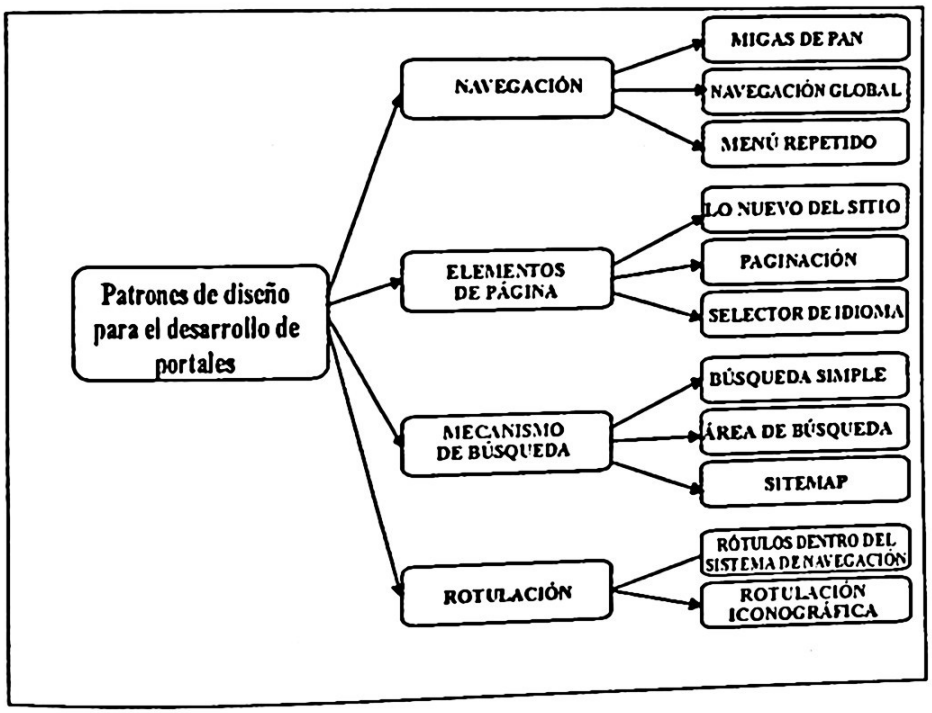


Fig. 2. Taxonomía de patrones para el Web

En el presente trabajo, se hace una revisión de los patrones de diseño para el Web que se proponen en los trabajos[3] y [4], para lo cual se realiza la categorización que se muestra en la figura 2, todo esto con la finalidad de eficientar el funcionamiento de los elementos del modelo trifásico propuesto.

Enseguida describimos el primer nivel de la taxonomía mostrada en la figura 2.

. *Navegación.* Con el diseño de los sistemas de navegación se debe apoyar el aprendizaje asociativo al presentar recursos relacionados con el contenido que el usuario esta viendo. En base a esto el reto más importante es equilibrar la flexibilidad de movimiento con el riesgo de abrumar al usuario dándole demasiadas opciones[3]. Van Welie[4] propone los siguientes patrones en el aspecto de navegación: a) migas de pan, b) navegación global y c) menú repetido.

2. *Elementos de página.* Los elementos de página son complementarios a lo que son los sistemas de navegación en general, ya que con estos se muestra lo nuevo del sistema, cuando se tienen una serie de páginas se tiene que mostrar el índice o mapa de estas para lo cual se aplican los siguientes patrones: a) lo nuevo del sitio, b) paginación y c) selector de idioma.

3. *Mecanismos de búsqueda.* Para ofrecer un motor de búsqueda en un sistema, lo que tenemos que hacer primero es evaluar si el sistema lo requiere o no, porque si el sistema es pequeño no tiene caso ofrecer esta herramienta, de lo contrario si el sistema cuenta con una gran cantidad de contenidos si sería necesario considerar aplicar este mecanismo, por lo que en seguida mostramos algunos patrones para este aspecto: a) búsqueda simple, b) área de bú squeda, y c) sitemap.

4. *Rotulación.* Los rótulos son una consecuencia natural de la creación de los sistemas de organización y navegación[2] sin los cuales un sistema no puede funcionar, porque poner rótulos a las cosas es inherente a los seres humanos. En seguida mostramos algunos patrones que corresponden a este aspecto: a) rótulos dentro del sistema de navegación, b) rotulación icnográfica.

Enseguida mostramos la interpretación o el formato de un patrón sugerido por Martijn van Welie [4].

Nombre del patrón: Barra de navegación

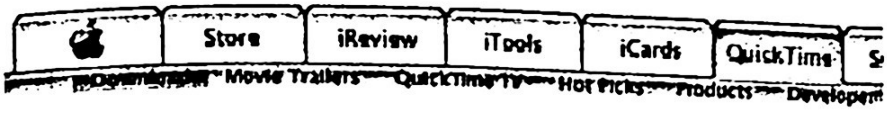
Problema: Los usuarios necesitan que las partes del sitio web estén estructuradas y organizadas para que sean fáciles de entenderse y utilizarse.

Contexto: La mayoría de los sitios web, especialmente para los sitios comerciales que desean conseguir contactarse con sus visitantes. Además, el sitio ofrece la funcionalidad navegacional que debe estar disponible en cada página. Una cierta funcionalidad es relevante en cada pagina. El usuario puede desear ir a diverso lugar en la página

Solución: La barra de la navegación consiste en tener los elementos comunicativos y generales de navegación. Los elementos de la navegación incluyen pagina principal, el mecanismo de búsqueda, el mapa del sitio e índice. Los elementos comunicativos dicen a los usuarios algo sobre la organización y cómo

contactarse. Normalmente la barra de la navegación se coloca en la región superior de la página de modo que sea siempre visible.

Ejemplo:



4 Caso de estudio

Con el fin de poner a disposición la información cultural del Estado de Tlaxcala, se construye el portal cultural denominado "expedición cultural", en el cual se muestra la aplicación de los patrones de diseño en el sistema de navegación. De manera que describimos los pasos a seguir para localizar el contenido del Museo Nacional del Títere ubicado en la ciudad de Huamantla, Tlaxcala. Para esto en la pantalla principal, a la izquierda, tenemos la estructura de navegación principal, figura 3.

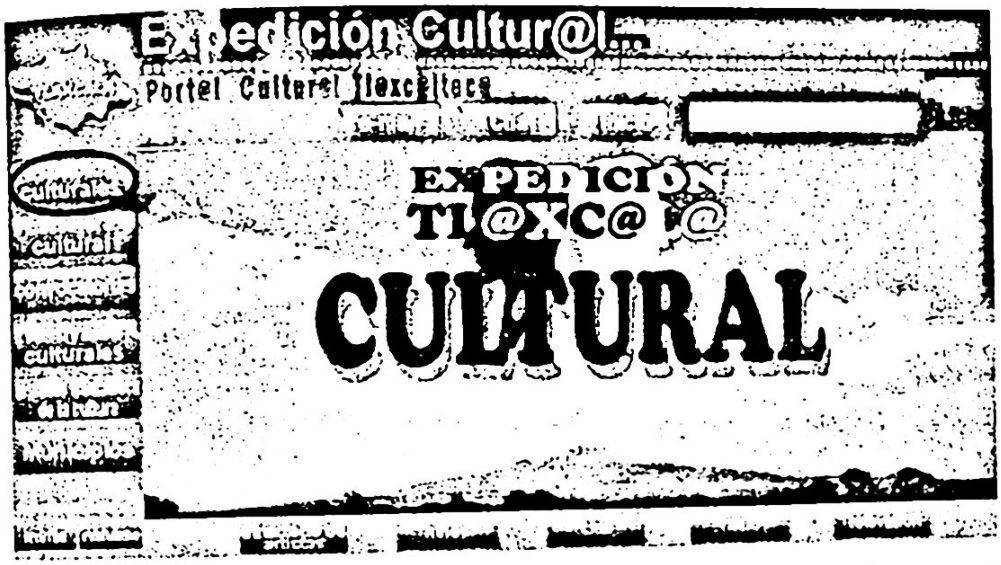


Fig. 3. Página principal de "Expedición Cultural". Se muestra el sistema de navegación izquierdo, también denominado sistema de navegación vertical.

Los módulos de la navegación vertical(espacios, cartelera, convocatorias, festivales, ITC y municipios) son la base para resolver problemas de la sección dos.

De manera que en el caso de estudio propuesto se requiere utilizar el modulo de espacios culturales, en el cual muestra toda la infraestructura cultural (museos, zonas arqueológicas, centros culturales, librerías, teatros, bibliotecas públicas, artesanías, parques y jardines), para lo cual elegimos el submodulo de museos como se muestra en la figura 4.

Al entrar al submodulo museos y galerías como se muestra en la figura 5, se ingresa al tercer nivel de la estructura jerárquica del sistema, por lo que es necesario describir el uso de los patrones en términos de sistemas de navegación.

Estando en este nivel de profundidad existe el riesgo de que el usuario no sepa en que parte del sistema se encuentra. Para esto tenemos la opción inicio dentro de las

opciones de navegación superior, la cual independientemente del nivel en que se encuentre el usuario, la opción lo conducirá al sistema de navegación principal, con lo que resolveremos la problemática de orientación para el usuario.

Fig. 4. Desglose del submódulo de "espacios culturales". En esta figura se muestra lo correspondiente a profundidad, el segundo nivel de la estructura jerárquica del portal.

Fig. 5. Infraestructura museística. Tercer nivel de la estructura jerárquica del portal.

Otra problemática en el sistema de navegación que se resuelve en este caso, es que el rango de niveles de profundidad en las estructuras jerárquicas de los sistemas el Web es de 2 a 3 como máximo, para evitar abrumar al usuario. En este caso contenido del Museo Nacional del Títere se encuentra en el tercer o máximo nivel permitido, como se muestra en la figura 6.

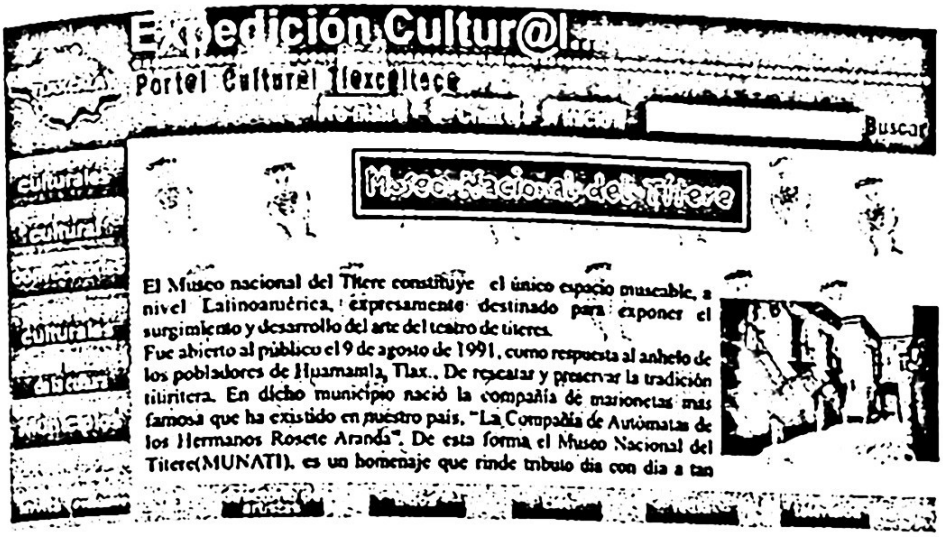


Fig. 6. Contenido del Museo Nacional del Títere.

5 Trabajos Relacionados

La tecnología avanza a pasos agigantados, por lo que resulta indispensable que las instituciones gocen de las ventajas que ofrecen las Tecnologías de Información(TI), para facilitar su trabajo de difusión y principalmente para favorecer su contacto con el público. En seguida mostramos los trabajos relacionados con este proyecto.

- *Culturama.* En la ciudad de México, el 17 de agosto de 1999, el sistema de difusión cultural "Culturama" inicia sus operaciones, formando una Red de Información Cultural, como uno de los proyectos del Instituto de Cultura de la Ciudad de México, basado en la iniciativa del Gobierno del Distrito Federal para difundir la gran diversidad de actividades culturales que existen en la ciudad.
- *Morelos.* Por otro lado tenemos el Estado de Morelos con "El Instituto Morelense de Cultura", este sistema esta desarrollado para difundir las actividades culturales del estado de Morelos, ya sean eventos artísticos como conciertos, teatro, títeres, exposiciones, cine; así como la educación para la cultura y las artes, el desarrollo comunitario, la promoción de la cultura y la preservación del patrimonio cultural.
- *Quintana Roo.* En cuanto al Estado de Quintana roo con el "Instituto Quintanarroense de Cultura"; es una entidad joven, cuya realidad cultural la integran una amplia gama de elementos provenientes de diferentes fuentes, producto del mosaico etnográfico que conforma su sociedad. La política cultural del Instituto Quintanarroense de la Cultura en el estado se encamina a fortalecer los procesos identitarios de los grupos sociales que a diario construyen la cultura popular, donde las expresiones de la plástica, de la danza, de la literatura, del teatro y de la música son alentadas a través de escuelas en las que se forjan quienes habrán de ser los creadores y difusores de la cultura de nuestro tiempo. De todo esto surge la necesidad de aprovechar las tecnologías de comunicación para dar una ma-

yor cobertura a toda su actividad cultural, lo que conlleva a crear el sistema de información cultural en el Web "Instituto Quintanarroense de Cultura".

Enseguida mostramos un cuadro comparando el uso y aplicación de patrones de diseño en los trabajos antes mencionados, con el portal cultural propuesto, tomando en cuenta la taxonomía mostrada en la sección tres.

Tabla 1. Cuadro comparativo con el portal "Expedición Cultural"

Patrones	Trabajos Relacionados			Trabajo Propuesto
	Culturama	Morelos	Quintana Roo	Expedición
Navegación				
- Migas de pan	+	–	–	+
- Navegación global	+	+	+	+
- Menú repetido	+	–	–	+
Elementos de página				
- Lo nuevo del sitio	–	–	–	+
- Paginación	+	–	–	+
- Selector de idioma	–	–	–	–
Mecanismos de búsqueda				
- Búsqueda simple	+	–	–	+
- Área de búsqueda	+	–	–	+
- SiteMap	–	+	–	+
Rotulación				
- Rótulos en el sistema de navegación	+	+	+	+
- Rotulación iconográfica	+	+	+	+

En los resultados de la comparación de la tabla 1, el signo + indica que en el trabajo relacionado se aplica ese grupo de patrones y el signo – indica la ausencia de ese grupo de patrones. Conforme al resultado de esta tabla, el trabajo cumple con mayor número de patrones, el único problema esta dentro del grupo de patrones de elementos de página, específicamente el selector de idioma, el cual se aplicará en otra etapa.

6 Conclusiones

Actualmente existe una gran variedad de patrones de diseño, por lo que en este trabajo se presenta una categorización de estos y se aplican a un portal cultural, lo que permitirá que los contenidos ofrecidos en este sean gestionados por el usuario manera eficiente.

Cada vez los sistemas en el Web son más grandes y complejos, por lo que los usuarios sufren para encontrar la información que necesitan. Es por esto que actual-

mente es mayor el desafío para el desarrollo y mantenimiento de los sistemas de información en el Web.

Los patrones de diseño están relacionados directamente con la arquitectura de software, diseño e instrumentación de programación y diseño de entornos web. En estos últimos, los patrones vienen a resolver problemas recurrentes, tales como: navegación y rotulación inconsistentes, vínculos erróneos, entre otros.

Referencias

1. Ledo Mezquita, Yoel, Damián Bergantiños Hidalgo, Alejandro Machado Cento, William Azcuy Morales, Yadenis Piñero Pérez, Camilo Núñez Morfa. *"Desarrollo de Portales Dinámicos. Experiencia Universitaria"*, En CLEI, Merida, Venezuela, 2001.
2. Rosefell, Louis, and Peter Morville, *"Arquitectura de la Información para el WWW"*, O'Reilly Editores, 1998.
3. van Duyne, Douglas K., James A. Landay, and Jason I. Hong, *The Design of Sites*, Addison Wesley; 1st edition, 2002.
4. van Welie, Martijn, and Hallvard Trætteberg, *Interaction Patterns in User Interfaces*, En: 7th. Conf. Pattern Languages of Programs, 13-16 August, Allerton Park Monticello, Illinois, USA, 2000.
5. López, Victor, Francisco Montero, Antonio Fernández, María Dolores Lozano. *"Towards Adaptive User Interface Generation: One Step Closer To People"*. En ICEIS 2003, Angers, France, 2003.
6. García Gómez, Juan Carlos, and José Luis González Olivares, *"Los Portales Web Ante el Reto de la Generación de Negocio en Internet"*, In INDOEM proceeding, Murcia, España (2002)
7. Grimán, Anna, Luis Mendoza, María Pérez, Teresita Rojas. "Evaluación de Ambientes de Desarrollo de Portales utilizando el Estándar Internacional ISO/IEC 9126", En CLEI, Merida, Venezuela, 2001.
8. Orcero Santo, David. *Linux en Internet*. En SOLO Linux review, vol 27 (2002) 67-68.
9. Ruiz Castro, Roberto, *"Propuesta de metodología para el desarrollo de sitios web, basada en la ingeniería de software"*, En ANIEI, Guadalajara, México, 2002.
10. World Wide Web Consortium, www.w3c.org, visitado 1 de marzo de 2003.
11. Alexander, Christopher, Sara Ishikawa, Murria Silverstein, Max Jacobson, Ingrid Fiksdahl-King, Shlomo Angel, *A Pattern Language*, Oxford University Press, New York, 1977.
12. Muñoz Arteaga, Jaime, Gustavo Rodríguez Gómez. *"Patrones de Interacción: Una Solución para el diseño de la Retroalimentación Visual de Sistemas Interactivos"*. En XI International Computing Conference, Mexico, DF, Mexico, 2002, 165-176.

Interacción háptica de alto desempeño con objetos virtuales dinámicos deformables

Omar A. Domínguez-Ramírez, Vicente Parra-Vega

Centro de Investigación y de Estudios Avanzados del IPN
Departamento de Ingeniería Eléctrica - Sección Mecatrónica
Av. IPN #2508, San Pedro Zacatenco, México, D.F., CP.07300
(omar, vparra)@mail.cinvestav.mx

Resumen En este trabajo se presenta al dispositivo háptico PHANToM 1.0, sus características electromecánicas y de programación, el modelo cinemático y dinámico, el método de generación de fuerza de reacción utilizado y un nuevo método que permite incrementar el realismo durante la interacción operador humano-ambiente virtual, considerando la dinámica no lineal de interacción. Se dan a conocer los experimentos y el estudio comparativo. El experimento consiste en la navegación aleatoria sobre la superficie de una esfera virtual.

1. Introduccción

PHANToM 1.0 es un dispostivo electromecánico que permite la interacción un operador humano con un ambiente virtual. La háptica estudia la manera como articular el tacto con un mundo generado por una computadora. Uno los problemas actuales de la realidad virtual es la limitación de estímulos par- el sentido del tacto. La retroalimentación de fuerza o kinestética, es un campo de investigación de la háptica que trata con dispositivos que interactúan con músculos y tendones, y proporcionan al operador humano la sensación de que se aplica una fuerza. La retroalimentación táctil, trata con dispositivos que teractúan con los nervios terminales en la piel los cuales indican la presencia calor, presión y textura. El dispositivo háptico PHANToM 1.0 permite la retroalimentación táctil y kinestética. El método para generar la retroalimentación kinestética es el método de penalización, y consiste en asignar la propiedad de resorte (rigidez) a la superficie del objeto virtual en el punto de contacto (ley Hooke). Este método es utilizado en los dispositivos hápticos existentes debido a su fácil implementación, sin embargo, por no considerar la dinámica no lineal de interacción definida por la dinámica del dispositivo háptico, la dinámica objeto virtual y la participación del operador humano es considerada como una interfaz háptica pasiva, es decir, la retroalimentación de fuerza no representa dinámica de interacción del fenómeno real que se intenta representar. La programación de PHANToM 1.0, para generar aplicaciones, esta basada en una interfaz programable de aplicaciones llamada GHOST SDK, GHOST permite detectar el contacto entre el operador humano (representado en el ambiente virtual) y

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 294–310, 2003. © Centro de Investigación en Computación, IPN, México.

objeto virtual, a partir de ello y de la deformación define la retroalimentación de fuerza, enviandola al operador humano para recrear la interacción. Sin embargo, y de manera implícita GHOST efectúa la conversión de fuerza a par, el par es definido con el conocimiento de la matriz jacobiana del dispositivo háptico, y posteriormente es enviado a los servomotores para recrear la fuerza de reacción en el efector final de PHANToM 1.0. Por esta razón se presenta en este trabajo el modelo cinemático de posición y diferencial, necesario para operar a PHANToM 1.0 en arquitectura abierta. En este trabajo proponemos un método nuevo e innovador, que garantiza el realismo de una interfaz háptica activa. Este método es denominado como Lagrangiano Restringido y consiste en definir la fuerza de reacción en función de la dinámica no lineal del dispositivo háptico, del modelo dinámico del objeto virtual definido con masa, amortiguamiento y rigidez, y de la participación del operador humano. Como este método requiere del conocimiento del modelo dinámico del dispositivo háptico, este es presentado, así como los parámetros dinámicos utilizados.

1.1. Planteamiento del problema

El problema a resolver gira en torno a la pregunta: ¿Cómo obtener una retroalimentación kinestética que garantice el realismo del fenómento que se desea representar con la interacción operador humano-ambiente virtual?.
Con el conocimiento del bajo desempeño del método de penalización (interfaz háptica pasiva) para generar la fuerza de reacción y que la interfaz de aplicación programable de GHOST SDK 3.1 de PHANToM 1.0 opera en arquitectura cerrada, se propone definir la fuerza de reacción considerando la dinámica restringida del dispositivo háptico y el modelo del objeto virtual que representa la superficie holónoma (ecuación algebráico-diferencial). Para operar al dispositivo háptico PHANToM 1.0 en arquitectura abierta, se requiere del conocimiento del modelo cinemático de posición y diferencial, situación que permitirá validar experimentalmente el algoritmo propuesto.

1.2. Solución propuesta

Para obtener el algoritmo que permita calcular la fuerza de reacción (Lagrangiano Restringido) que garantice una interacción realista entre el operador humano y el ambiente virtual, se plantea la siguiente solución:

- Obtener el modelo dinámico restringido del dispositivo háptico PHANToM 1.0.
- Identificar los parámetros dinámicos de PHANToM 1.0 mediante métodos de identificación en línea o aproximación basada en mediciones físicas.
- Definir un modelo dinámico de segundo orden que represente al objeto virtual de manera análoga a un sistema masa-resorte-amortiguador.

- Determinar el par de contacto (interacción de PHANToM con la superficie del objeto virtual).

- Resolver el multiplicador de Lagrange (fuerza de reacción) en términos del par de contacto.

Para operar al dispositivo háptico PHANToM 1.0 en arquitectura abierta, se propone la siguiente solución:

- Obtener el modelo cinemático directo e inverso de posción y velocidad del dispositivo háptico PHANToM 1.0.

- Definir un detector de contacto descrito por una función definida en términos del modelo cinemático directo de posición y de la ecuación del objeto virtual.

- Determinar la matriz transpuesta del jacobiano del dispositivo háptico PHANToM 1.0.

- Crear una clase que interactúe con GHOST SDK y que permita enviar los pares obtenidos a los servomotores del dispositivo háptico PHANToM 1.0.

1.3. Organización

En la sección II se presentan las características electromecánicas y de programación del dispositivo háptico PHANToM 1.0, así como recomendaciones de operación. En la sección III se define el modelo cinemático del dispositivo háptico PHANToM 1.0. En la sección IV se presenta la formulación de Euler-Lagrange que define la dinámica. En la sección V se presenta el modelo dinámico restringido de PHANTOM 1.0 y el modelo del objeto virtual deformable, en esta sección se define el Lagrangiano Restringido. En la sección VI se consideran las condiciones y características del experimento así como las gráficas experimentales del método de penalización y del método del Lagrangiano Restringido, y el estudio comparativo entre estos dos. Y finalmente se concluye y se definen perspectivas de este trabajo.

2. PHANToM 1.0: Un dispositivo háptico de última generación

El dispositivo háptico PHANToM 1.0, es un dispositivo electromecánico que permite la inmersión virtual, y percibir durante la interacción, sensaciones kinestéticas y táctiles. Es un mecanismo de eslabones articulados de tres grados de libertad con articulaciones de revolución y cuyo efector final es un dedal que permite la interacción del operador humano . La posición nominal es de 0,03 mm, su espacio de configuraciones admisibles $130 \times 180 \times 250$ (mm^3), la fricción por

articulación es de 0,04 N y la máxima fuerza aplicada por el operador humano es de 8,5 N. PHANToM 1.0 requiere de un amplificador de potencia de 10 A con voltajes de alimentación de 120 V o 220V a 60 $Hertz$ de frecuencia. En la Figura 1, se presenta la estación de trabajo PHANToM 1.0. El estímulo visual se efectúa

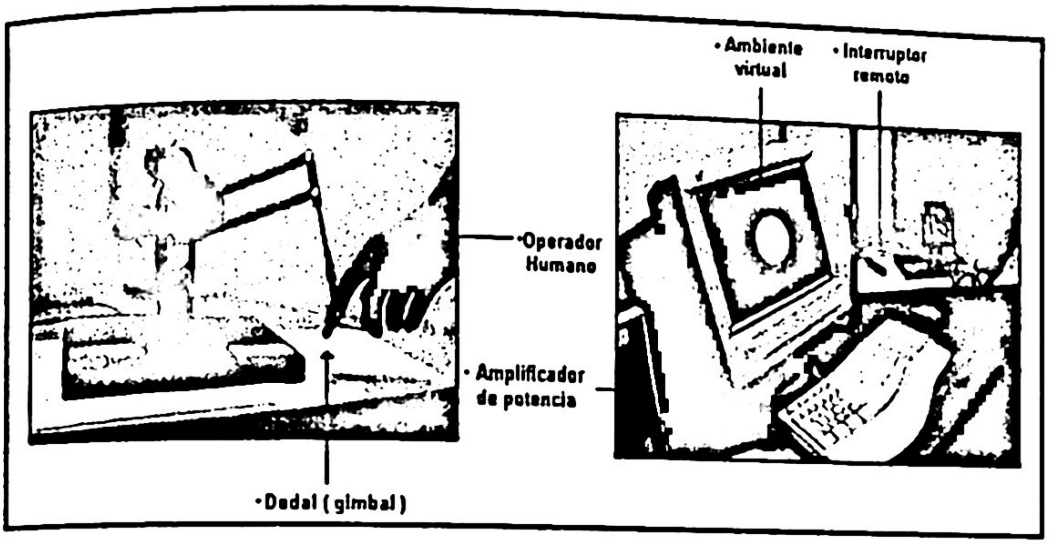


Fig. 1. Interfaz Háptica PHANToM 1.0 (Estación de trabajo)

mediante un ambiente virtual modelado con GHOST SDK 3.1. Esta interfaz de aplicaciones programable es compatible con Visual C++ 6,0 con sistema operativo Windows NT 4,0/SP6 o bien Windows 2000/SP2. Es recomendable una velocidad de procesamiento mayor a 300 MHz y un espacio disponible de disco duro de 500MB.

3. Modelo cinemático

En esta sección, se presenta el modelo cinemático directo e inverso de posición, así como el modelo cinemático directo e inverso de velocidad y aceleración (cinemática diferencial). Para ello se utilizó la metodología Denavit-Hartemberg. El modelo cinemático es utilizado de manera importante para operar a PHANToM en arquitectura abierta y para realizar tareas de entrenamiento remoto (guiado háptico). La cadena cinemática de PHANToM, representando los marcos ortonormales y articulaciones de revolución, de acuerdo a metodologías convencionales, es definida en la figura 2. Donde o_1,o_2 y o_3, corresponden a los orígenes

de los marcos ortonormales en la segunda y tercera articulación, L_0, L_1 y $L2$ son las longitudes de los eslabones 1, 2 y 3. Los parámetros Denavit-Hartemberg, conforme a la cadena cinemática, están definidos en la tabla (1). El producto de

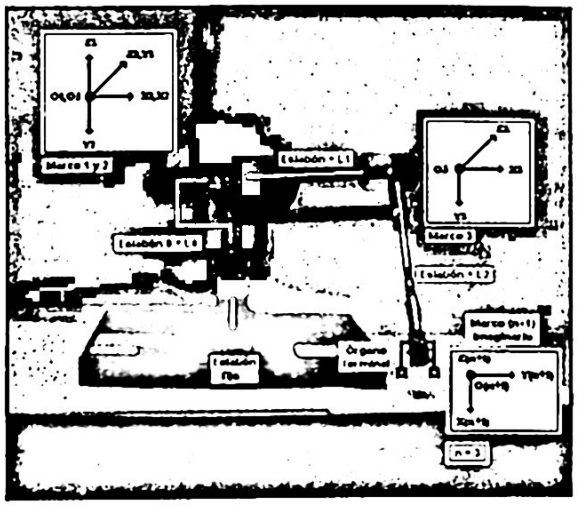


Fig. 2. Cadena Cinemática de PHANToM 1.0

i	α	a	d	θ
1	$-\pi/2$	0	0	θ_1
2	0	L_2	0	θ_2
3	0	L_3	0	θ_3

Cuadro1. Parámetros Denavit-Hartemberg de PHANToM 1.0.

las tres matrices elementales derivadas de los parámetros Denavit-Hartemberg constituyen una matriz de transformación homogénea que permite definir la posición y orientación del órgano terminal respecto al marco ortonormal de referencia en la base del robot, y haciendo la transformación correspondiente al dedal. La matriz de transformación homogénea de PHANToM 1.0 esta definida como,

$$ {}^{1}_{4}T = \begin{bmatrix} C_1 & -S_1S_3 & C_3S_1 & S_1\left(L_1C_2 + L_2S_3\right) \\ 0 & C_3 & S_3 & L_2 - L_2C_3 + L_1S_2 \\ -S_1 & -C_1S_3 & C_1C_3 & -L_1 + C_1\left(L_1C_2 + L_2S_3\right) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

Donde $C_i = \cos(\theta_i)$ y $S_i = \sin(\theta_i)$. La matriz (1) define el modelo cinemático directo de posición, ecuaciones que permiten identificar las coordenadas opera-

cionales del extremo final de PHANToM (dedal), respecto al marco de referencia base (condición inicial de PHANToM 1.0, figura (3)).

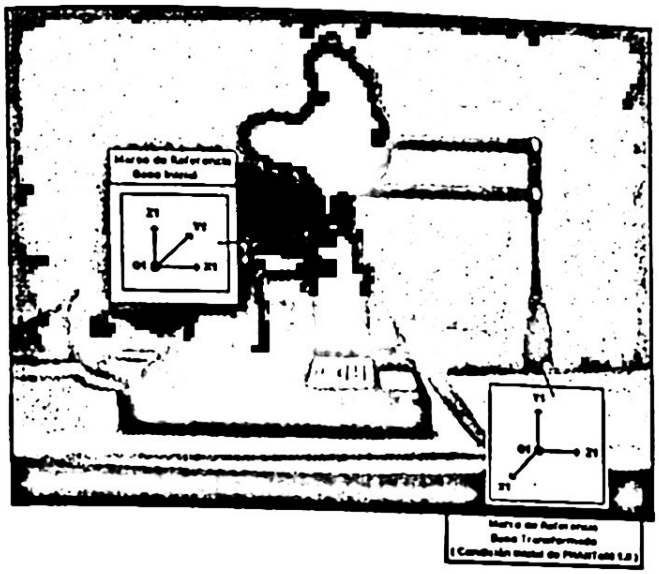


Fig. 3. Marco de Referencia Base y Condición Inicial de PHANToM 1.0

1. Modelo cinemático de posición

Las ecuaciones que definen el modelo cinemático directo de posición, se describen (2).

$$\begin{aligned} X &= S_1 \left(L_1 C_2 + L_2 S_3\right) \\ Y &= L_2 - L_2 C_3 + L_1 S_2 \\ Z &= -L_1 + C_1 \left(L_1 C_2 + L_2 S_3\right) \end{aligned} \tag{2}$$

modelo cinemático inverso de posición representa al conjunto de ecuaciones que permiten conocer las coordenadas generalizadas, con respecto a las coordenadas operacionales del extremo final de PHANToM (dedal). Para ésto, es necesario considerar que la posición de inicio de PHANToM que define el marco referencia base, donde $\theta_1 = 0$ rad, $\theta_2 = 0$ rad y $\theta_3 = 0$ rad esta definido en la figura (3). Las ecuaciones estan definidas a continuación,

$$\theta_1 = a\tan 2\left(x_1, z + L_1\right) \tag{3}$$

$$r = \sqrt{x^2 + \left(y - L_2\right)^2 + \left(z + L_1\right)^2}$$

$$d = \sqrt{x^2 + \left(z + L_1\right)^2}$$

$$\begin{aligned}
\beta &= a\tan 2\,(y - L_2, d) \\
L_2^2 &= L_1^2 + r^2 - 2L_1 r Cos\,(\gamma) \\
\gamma &= Cos^{-1}\left(\frac{L_1^2 + r^2 - L_2^2}{2L_1 r}\right)
\end{aligned}$$

$$\theta_2 = \gamma + \beta \tag{4}$$

$$\begin{aligned}
r^2 &= L_1^2 + L_2^2 - 2L_1 L_2 Cos\,(\alpha) \\
\alpha &= Cos^{-1}\left(\frac{L_1^2 + L_2^2 - r^2}{2L_1 L_2}\right) \\
\alpha &= \frac{\pi}{2} - \theta_2 + \theta_3
\end{aligned}$$

$$\theta_3 = \alpha + \theta_2 - \frac{\pi}{2} \tag{5}$$

De esta manera se obtienen las coordenadas generalizadas θ_1, θ_2 y θ_3, en función de las coordenadas operacionales del dedal de PHANToM 1.0, representando al modelo cinemático inverso de posición.

3.2. Modelo cinemático de velocidad

El modelo cinemático directo de velocidad de PHANToM 1.0, esta definido por $\dot{x} = J\dot{\theta}$, donde $\dot{x}$ representa la velocidad operacional del extremo final de PHANToM, J representa la matriz Jacobiana y $\dot{\theta}$ corresponde a la velocidad articular. De ésta manera J esta definido por la seguiente expresión

$$J = \begin{bmatrix} L_1C_1C_2 + L_2S_3C_1 & -L_1S_1S_2 & L_2S_1C_3 \\ 0 & L_1C_2 & L_2S_3 \\ -(L_1S_1C_2 + L_2S_1S_3) & -L_1S_2C_1 & L_2C_1C_3 \end{bmatrix} \tag{6}$$

El modelo cinemático directo de velocidad de PHANToM 1.0 a partir de (6),esta definido por

$$\begin{aligned}
\dot{x} &= \{L_1C_1C_2 + L_2S_3C_1\}\,\dot{\theta}_1 - \{L_1S_1S_2\}\,\dot{\theta}_2 + \{L_2S_1C_3\}\,\dot{\theta}_3 \\
\dot{y} &= \{L_1C_2\}\,\dot{\theta}_2 + \{L_2S_3\}\,\dot{\theta}_3 \\
\dot{z} &= -\{L_1S_1C_2 + L_2S_1S_3\}\,\dot{\theta}_1 - \{L_1S_2C_1\}\,\dot{\theta}_2 + \{L_2C_1C_3\}\,\dot{\theta}_3
\end{aligned} \tag{7}$$

El modelo cinemático inverso de velocidad de PHANToM esta definido por $\dot{\theta} = J^{-1}\dot{x}$, donde J^{-1} esta definida por

$$J^{-1} = \frac{adj\,(J)}{\det\,(J)} \tag{8}$$

donde

$$\det\,(J) = L_1L_2\left\{L_1S_2S_3C_2 + L_1C_2^2C_3 + L_2S_2 - L_2S_2C_3^2 + L_2S_3C_2C_3\right\} \tag{9}$$

Con la condición de que $\theta_3 \neq \theta_2 + \frac{\pi}{2}$ para evitar que la matriz J sea singular,

$$adj\,(J) = \begin{bmatrix} L_1L_2C_1C_{2-3} & 0 & -L_1L_2S_1C_{2-3} \\ -L_2S_1S_3K & L_2C_3K & -L_2S_3C_1K \\ L_1S_1C_2K & L_1S_2K & L_1C_1C_2K \end{bmatrix} \tag{10}$$

donde $K = L_1C_2 + L_2S_3$.

3.3. Modelo cinemático de aceleración

El modelo cinemático directo de aceleración define la aceleración operacional en el extremo final de PHANToM, cuya expresión general es $\ddot{x} = J\ddot{\theta} + \dot{J}\dot{\theta}$, donde $\ddot{\theta}$ representa la aceleración articular y $\dot{J}$ es la derivada temporal de la matriz jacobiana analítica, definida a continuación,

$$\dot{J} = \begin{bmatrix} \dot{J}_{11} & \dot{J}_{12} & \dot{J}_{13} \\ \dot{J}_{21} & \dot{J}_{22} & \dot{J}_{23} \\ \dot{J}_{31} & \dot{J}_{32} & \dot{J}_{33} \end{bmatrix} \tag{11}$$

donde

$$\begin{aligned}
\dot{J}_{11} &= -\left\{L_1S_1C_2 + L_2S_1S_3\right\}\dot{\theta}_1 - L_1S_2C_1\dot{\theta}_2 + L_2C_1C_3\dot{\theta}_3 \\
\dot{J}_{12} &= -L_1S_2C_1\dot{\theta}_1 - L_1S_1C_2\dot{\theta}_2 \\
\dot{J}_{13} &= L_2C_1C_3\dot{\theta}_1 - L_2S_1S_3\dot{\theta}_3 \\
\dot{J}_{21} &= 0 \\
\dot{J}_{22} &= -L_1S_2\dot{\theta}_2 \\
\dot{J}_{23} &= L_2C_3\dot{\theta}_3 \\
\dot{J}_{31} &= -\left\{L_1C_1C_2 + L_2S_3C_1\right\}\dot{\theta}_1 + L_1S_1S_2\dot{\theta}_2 - L_2S_1C_3\dot{\theta}_3 \\
\dot{J}_{32} &= L_1S_1S_2\dot{\theta}_1 - L_1C_1C_2\dot{\theta}_2 \\
\dot{J}_{33} &= -L_2S_1C_3\dot{\theta}_1 - L_2S_3C_1\dot{\theta}_3
\end{aligned}$$

El modelo cinemático inverso de aceleración, es el conjunto de ecuaciones que permite conocer la aceleración articular en términos de la aceleración operacional y de la velocidad articular, definida como: $\ddot{\theta} = J^{-1}\left\{\ddot{x} - \dot{J}\dot{\theta}\right\}$.

4. Modelo dinámico

El modelo dinámico de un robot manipulador, se puede obtener a partir de las leyes de la mecánica newtoniana y lagrangiana. Existen diversos métodos convencionales como Euler-Lagrange (E-L) y Newton-Euler (N-E). El uso del método E-L, para la obtención del modelo dinámico es simple y sistemático, proporciona las ecuaciones de estado en forma explicita y estas pueden ser utilizadas para analizar y diseñar estrategias de control. Las ecuaciones de movimiento de E-L son

$$\frac{d}{dt}\frac{\partial}{\partial\dot{\theta}_i}L - \frac{\partial}{\partial\theta_i}L = \tau_i \qquad i = 1, 2, ..., n \tag{12}$$

donde el Lagrangiano es

$$L = \sum_{i=1}^{n} (K_i - P_i) \tag{13}$$

con K_i es la energía cinética del i-esimo eslabón, P_i es la energía potencial del i-ésimo eslabón, θ_i es la i-ésima coordenada generalizada o variable articular, $\dot{\theta}_i$ es la primera derivada respecto al tiempo de la coordenada generalizada θ_i y τ_i corresponde a la fuerza generalizada aplicada a la articulación i para mover el eslabón i. Utilizando (12) se obtiene la representacón del modelo dinámico

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{14}$$

dado que $n = 3$, se tiene que (14) se expresa como

$$\begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix} + \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & 0 & c_{23} \\ c_{31} & c_{32} & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} + \begin{bmatrix} 0 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} \tag{15}$$

donde los elementos de la matriz de inercias son

$$\begin{aligned}
m_{11} &= \left\{ \left(0{,}5L_1^2 + 0{,}125L_2^2\right) m_a + \left(0{,}125L_1^2 + 0{,}5L_3^2\right) m_c \right\} + \\
&\quad 0{,}125L_1^2 \left(4m_a + m_c\right) Cos\left(2\theta_2\right) + m'_{11} \\
m'_{11} &= -0{,}125 \left(L_2^2 m_a + 4L_3^2 m_c\right) Cos\left(2\theta_3\right) + \\
&\quad 0{,}125L_1 \left(L_2 m_a + L_3 m_c\right) Cos\left(\theta_2\right) Sen\left(\theta_3\right) \\
m_{22} &= L_1^2 \left(m_a + 0{,}25m_c\right) \\
m_{23} &= -0{,}5L_1 \left(L_2 m_a + L_3 m_c\right) Sen\left(\theta_2 - \theta_3\right) \\
m_{32} &= m_{23} \\
m_{33} &= 0{,}25L_2^2 m_a + L_3^2 m_c
\end{aligned}$$

Los elementos de la matriz de fuerzas de Coriolis son

$$\begin{aligned}
c_{11} &= 0{,}125 \left\{ -2Sen\left(\theta_2\right) \left[L_1^2 \left(4m_a + m_c\right) Cos\left(\theta_2\right) + \right. \right. \\
&\quad \left. \left. 2L_1 \left(L_2 m_a + L_3 m_c\right) Sen\left(\theta_3\right) \right] \dot{\theta}_2 \right\} + c'_{11} \\
c'_{11} &= 0{,}25Cos\left(\theta_3\right) \left\{ 2L_1 \left(L_2 m_a + L_3 m_c\right) Cos\left(\theta_2\right) \right. \\
&\quad \left. + \left(L_2^2 m_a + 4L_3^2 m_c\right) Sen\left(\theta_3\right) \right\} \dot{\theta}_3 \\
c_{12} &= -0{,}125 \left\{ L_1^2 \left(4m_a + m_c\right) Sen\left(2\theta_2\right) + \right. \\
&\quad \left. 4L_1 \left(L_2 m_a + L_3 m_c\right) Sen\left(\theta_2\right) Sen\left(\theta_3\right) \right\} \dot{\theta}_1 \\
c_{13} &= -0{,}125 \left\{ -4L_1 \left(L_2 m_a + L_3 m_c\right) Cos\left(\theta_2\right) Cos\left(\theta_3\right) - \right. \\
&\quad \left. \left(L_2^2 m_a + 4L_3^2 m_c\right) Sen\left(2\theta_3\right) \right\} \dot{\theta}_1 \\
c_{21} &= -c_{12} \\
c_{23} &= 0{,}5L_1 \left(L_2 m_a + L_3 m_c\right) Cos\left(\theta_2 - \theta_3\right) \dot{\theta}_3
\end{aligned}$$

$$c_{31} = -c_{13}$$
$$c_{32} = 0{,}5L_1 \left(L_2 m_a + L_3 m_c\right) Cos\left(\theta_2 - \theta_3\right) \dot{\theta}_2$$

Los elementos del vector de fuerzas gravitacionales son

$$g_1 = 0$$
$$g_2 = g \left\{L_1 \left(m_a + 0{,}5m_c\right) + L_5 m_{be}\right\} Cos\left(\theta_2\right)$$
$$g_3 = g \left\{0{,}5L_2 m_a + L_3 m_c - L_6 m_{df}\right\} Sen\left(\theta_3\right)$$

fácil verificar que la matriz de inercia es simétrica $M(q) = M(q)^T$ y definipositiva $\det\{M(q)\} > 0$, ademas que se cumple la propiedad de la matriz antisimétrica $x^T \left(\dot{M}(q) - 2C(q,\dot{q})\right) x = 0 \quad \forall x \in \Re^n$. Los parámetros aproximados del modelo dinámico que representan masas y longitudes de los eslabones elementos del dispositivo háptico PHANToM 1.0 son

Parámetro	Valor	Unidad
m_a	$17{,}5x10^{-3}$	Kg
m_c	$10{,}4x10^{-3}$	Kg
m_be	0,2214	Kg
m_df	0,1106	Kg
L_1	139,7	mm
L_2	139,7	mm
L_3	32,5	mm
L_4	36,8	mm
L_5	52,7	mm
g	9,81	m/s^2

Cuadro2. Parámetros del Modelo Dinámico de PHANToM 1.0.

Dinámica restringida

PHANToM 1.0, al interactuar con el objeto virtual, la dinámica presenta restricción en su movimiento, definido por el siguiente sistema de ecuaciones algebráico-diferencial,

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) = \tau + \frac{J_\varphi^T}{\left\|J_\varphi J_\varphi^T\right\|} f_r \tag{16}$$

$$\varphi(q) = 0 \tag{17}$$

donde $M(q) \in \Re^{n \times n}$ denota una matriz de inercia definida positiva y simétrica, $C(q,\dot{q}) \in \Re^{n \times n}$ corresponde a la matriz de fuerzas de Coriolis y centripetas, $g(q) \in \Re^n$ modela el vector de fuerzas gravitacionales, $\tau \in \Re^n$ corresponde al vector de torque de entrada, $f_r \in \Re^r$ es, (r=1 es un escalar que indica un punto de contacto) el Lagrangiano restringido y corresponde a la magnitud de la fuerza de contacto, $\frac{J_\varphi^T}{\|J_\varphi J_\varphi^T\|}$ es la proyección normalizada del jacobiano restringido sobre la normal al plano tangente del punto de contacto $\varphi(q) = 0$, $J_\varphi \in \Re^n$, $J_\varphi = J_\varphi(q) = \left[\frac{\partial}{\partial q_1}\varphi(q)\ \frac{\partial}{\partial q_2}\varphi(q)\ \frac{\partial}{\partial q_3}\varphi(q)\right]$, y la superficie de contacto esta definida como $\varphi(q) = 0$. Con base en el modelo dinámico restringido, y en ausencia de un sensor de fuerza que permita sensar la fuerza de interacción de PHANToM con el objeto virtual, se propone definir la fuerza de reacción f_r que garantiza una interfaz háptica activa.

5.1. Interfaz háptica activa

Una interfaz háptica activa involucra el conocimiento de la fuerza de reacción mediante un sensor físico o virtual, y la dinámica no lineal de interacción. En esta sección se presenta la propuesta para implementar una interfaz háptica activa. La superficie de contacto del objeto virtual es considerada como la restricción cinemática de movimiento de PHANToM, cuyo modelo es

$$\varphi(X) = 0 \ \in \Re^r \tag{18}$$

donde $r = 1$ es el número de puntos de contacto con $r < n$ y $n = 3$ el número de grados de libertad de PHANToM. Del modelo cinemático directo de posición $X = f(q)$ se tiene que $\varphi(f(q)) = 0 \ \in \Re^r$, donde $q \in \Re^n$ corresponde a las coordenadas generalizadas de PHANToM, por consiguiente es posible formular el ambiente virtual de restricción como

$$\varphi(q) = 0 \tag{19}$$

A partir de esto, se propone definir la fuerza de reacción f_r mediante la metodología del lagrangiano restringido, es decir

$$\tau_r = \frac{J_\varphi^T}{\|J_\varphi J_\varphi^T\|} f_r \tag{20}$$

donde $J_\varphi = \nabla_q \varphi(q) \in \Re^{n \times 1}$ representa la matriz jacobiana en el espacio de la tarea, definida por $J_\varphi = J_\varphi(q) = \left[\frac{\partial}{\partial q_1}\varphi(q)\ \frac{\partial}{\partial q}\varphi(q)\ \frac{\partial}{\partial q_3}\varphi(q)\right]$, y f_r corresponde al lagrangiano restringido que define la fuerza de contacto de PHANToM y la restricción cinemática definida por el objeto virtual. El modelo dinámico queda

representado por el sistema siguiente

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + \tau_r \tag{21}$$

$$\varphi(q) = 0 \in \Re^r \tag{22}$$

Las derivadas temporales de la restricción, están definidas como

$$\dot{\varphi}(q) = J_\varphi \dot{q} \triangleq 0 \tag{23}$$

$$\ddot{\varphi}(q) = J_\varphi \ddot{q} + \dot{J}_\varphi \dot{q} \triangleq 0 \tag{24}$$

Se propone un estabilizador de la restricción, normalizado a un sistema masa-resorte-amortiguador

$$\ddot{\varphi}(q) \triangleq -k_v \dot{\varphi}(q) - k_p \varphi(q) \tag{25}$$

donde las ganancias proporcional (k_p) y derivativa (k_v) dependen de la inercia m, la flexibilidad b y la elasticidad k

$$k_p = {}^{k}/_{m} \quad k_v = {}^{b}/_{m}$$

de (23), se obtiene

$$\ddot{\varphi}(q) + {}^{b}/_{m} \cdot \dot{\varphi}(q) + {}^{k}/_{m} \cdot \varphi(q) = 0 \tag{26}$$

Despejando la aceleración articular $\ddot{q}$ en (21) y sustituyendo (20)

$$\ddot{q} = M(q)^{-1} \left\{ -C(q,\dot{q})\dot{q} - G(q) + \tau + \frac{J_\varphi^T}{\|J_\varphi J_\varphi^T\|} f_r \right\} \tag{27}$$

En lazo cerrado con el estabilizador de la restricción (23) y (25) se tiene

$$J_\varphi M(q)^{-1} \left\{ -C(q,\dot{q})\dot{q} - G(q) + \tau + \frac{J_\varphi^T}{\|J_\varphi J_\varphi^T\|} f_r \right\} + \dot{J}_\varphi \dot{q} = \ddot{\varphi}(q) \tag{28}$$

despejando el lagrangiano f_r de (28), resulta

$$f_r = \frac{\|J_\varphi J_\varphi^T\|}{J_\varphi M(q)^{-1} J_\varphi^T} \left\{ \ddot{\varphi}(q) - \dot{J}_\varphi \dot{q} + J_\varphi H(q)^{-1} \left[C(q,\dot{q})\dot{q} + G(q) - \tau \right] \right\} \tag{29}$$

Considerando el control que estabiliza la restricción (25), finalmente tenemos que

$$f_r = \frac{\left\| J_\varphi J_\varphi^T \right\|}{J_\varphi M(q)^{-1} J_\varphi^T} \left\{ -kp \cdot \varphi(q) - kv \cdot \dot{\varphi}(q) - \dot{J}_\varphi \dot{q} + J_\varphi H(q)^{-1} \left[C(q,\dot{q})\dot{q} + G(q) - \tau \right] \right\} \tag{30}$$

La expresión (30), representa la fuerza de reacción en términos de la dinámica de PHANToM 1.0, en función de las propiedades dinámicas de inercia, rigidez y amortiguamiento del objeto virtual y un controlador τ.

6. Desempeño experimental en movimiento restringido

En esta sección se presentan los experimentos en arquitectura abierta de una interfaz háptica pasiva considerando las propiedades de rigidez y amortiguamiento para el cálculo de la fuerza de contacto. También son presentados los experimentos de una interfaz háptica activa empleando el algoritmo propuesto del Lagrangiano restringido (30) validando la conveniencia de su uso mediante el estudio comparativo con el método de penalización.

6.1. Modelo matemático de los objetos virtuales y detección de la colisión

El modelo cinemático de la restricción (superficie de contacto), con base en (18), del plano virtual paralelo a X-Z es

$$\varphi(x) = Y - y_0 \triangleq 0 \tag{31}$$

$$x = f(q) \tag{32}$$

$$\varphi(f(q)) = \varphi(q) \triangleq 0 \tag{33}$$

$$\varphi(q) = l_2 - l_2 C_3 + l_1 S_2 - y_0 \triangleq 0 \tag{34}$$

donde $f(q)$ corresponde al modelo cinemático directo de posición. $J_\varphi(q)$ y u son empleados para dar dirección a la fuerza de reacción f_r calculada por cualquier método, y se presentan a continuación,

$$J_\varphi(q) = \left[\frac{\partial}{\partial q_1}\varphi(q) \ \frac{\partial}{\partial q_2}\varphi(q) \ \frac{\partial}{\partial q_3}\varphi(q) \right] \tag{35}$$

$$= \left[0 \ l_1 C_2 \ l_2 S_3 \right] \tag{36}$$

$$u = \frac{J_\varphi(q)}{\sqrt{J_\varphi(q) J_\varphi^T(q)}} \tag{37}$$

Utilizando (18), el modelo cinemático de la restricción (superficie de contacto), que define a la esfera virtual es:

$$\varphi(x) = (x-\alpha)^2 + (y-\beta)^2 + (z-\gamma)^2 \triangleq r^2 \tag{38}$$

$$\varphi(x) = (x-\alpha)^2 + (y-\beta)^2 + (z-\gamma)^2 - r^2 \triangleq 0 \tag{39}$$

$$x = f(q) \tag{40}$$

$$\varphi(f(q)) = \varphi(q) \triangleq 0 \tag{41}$$

$$\varphi(q) = \varphi(q)_1 + \varphi(q)_2 + \varphi(q)_3 \triangleq 0 \tag{42}$$

$$q)_1 = 2l_1^2(1 - C_1C_2) + 2l_2^2(1 - C_3) + 2l_1l_2(S_2 - S_3C_1 - S_{2-3} + S_1^2S_3C_2)$$
$$q)_2 = 2l_1\{\gamma(1 - C_1C_2) - \alpha S_1C_2 - \beta S_2\} + 2l_2\{\beta(C_3 - 1) - (\alpha S_1 + \gamma C_1)S_3\}$$
$$q)_3 = \alpha^2 + \beta^2 + \gamma^2 - r^2$$

donde $c(\alpha, \beta, \gamma)$ representa el centro de la esfera y, r su radio. Las ecuaciones) y (38) son empleadas como detector de la colisión en arquitectura abierta de manera importante en la obtención de la fuerza de reacción con base en el método propuesto del Lagrangiano restringido. Con tolerancia de acercamiento $< |\xi|$ con $\xi = 1x10^{-6}\ m$

6.2. Experimentos de fuerza de reacción basado en el lagrangiano restringido

Con $K = 200\ N/m$, $b = 100\ Nseg/m$ y $M = 12{,}5\ Kg$, el comportamiento sobre superficie de contacto es estable con frecuencia de 0 Hz. Se logra deformación del objeto de manera adecuada, ver figuras (4) y (5). Con $K = 2000\ N/m$, $= 100\ Nseg/m$ y $M = 1{,}25\ Kg$, el comportamiento sobre la superficie de contacto es estable con frecuencia de 2.3 Hz. La deformación es nula en el objeto, figuras (6) y (7). Con $K = 500\ N/m$, $b = 100\ Nseg/m$ y $M = 1{,}25\ Kg$, comportamiento sobre la superficie de contacto es estable con frecuencia de 0 Se logra deformación controlada y estable en el objeto virtual, ver figuras (8) (9). En la tabla siguiente se presentan los comentarios del estudio comparativo entre las metodologías de fuerza de reacción por penalización, rigidez y amortiguamiento simultáneos y Lagrangiano restringido, los beneficios del método propuesto se basan en un control sobre la superficie de contacto, dada la baja frecuencia que el operador induce por la reacción de la superficie de contacto, como deformación controlable por el operador humano.

Conclusiones y perpectivas

este trabajo es posible apreciar el beneficio de implementar el método del Lagrangiano Restringido para asignar realismo a la interacción operador humano-

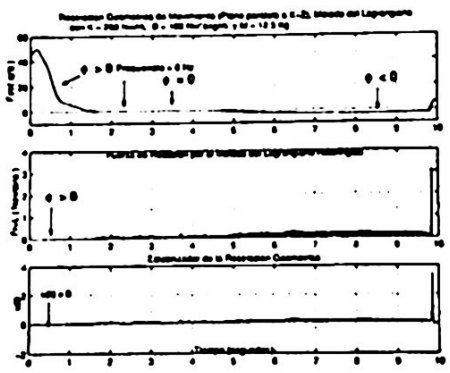


Fig. 4. Lagrangiano Restringido en una pared virtual, con $K = 200\ N/m$, $b = 100\ Nseg/m$ y $M = 12{,}5\ Kg$

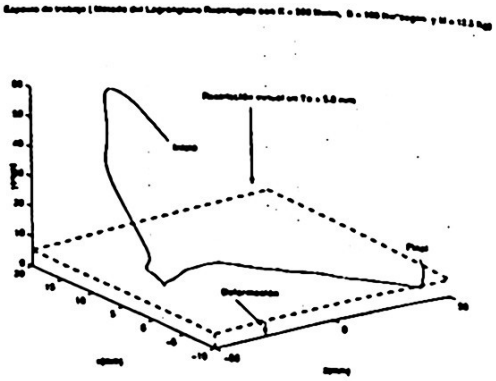


Fig. 5. Lagrangiano Restringido en una pared virtual

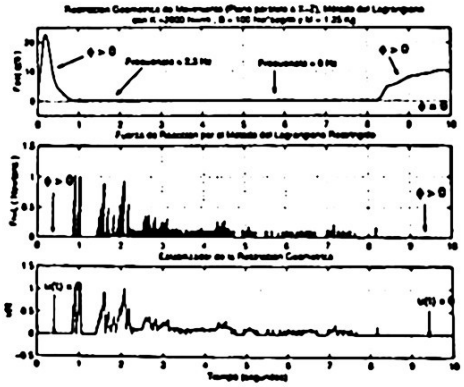


Fig. 6. Lagrangiano Restringido en una pared virtual, con $K = 2000\ N/m$, $b = 100\ Nseg/m$ y $M = 1{,}25\ Kg$

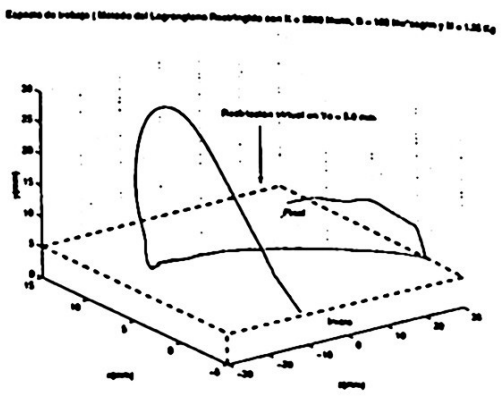


Fig. 7. Lagrangiano Restringido en una pared virtual

ambiente virtual, debido a que esta considerando la dinámica no lineal que representa al sistema. Los métodos pasivos de penalización y de amortiguamiento rigidez simultáneos no permiten la percepción de las propiedades dinámicas del objeto virtual bajo condiciones de deformación, situación apreciable en el método propuesto. Esta metodología es posible de aplicarse en entrenamiento remoto o guiado háptico libre y restringido, en telepresencia realista y cirugía de mínima invasión.

Referencias

1. G.C. Burdea, *"Force and Touch Feedback for Virtual Reality"*, John Wiley and Sons, Inc., Electrical and Computer Engineering Department, The State University of New Jersey, 1996.

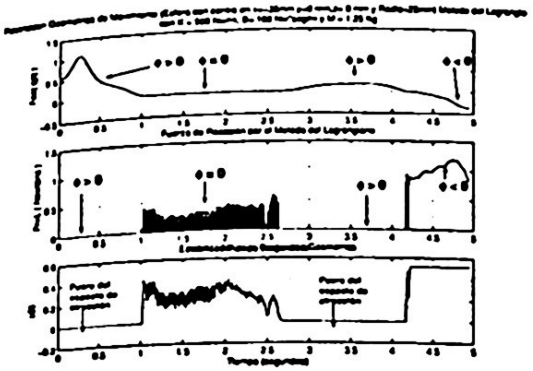


Fig. 8. Lagrangiano Restringido en una esfera virtual, con $K = 500\ N/m$, $b = 100\ Nseg/m$ y $M = 1{,}25\ Kg$

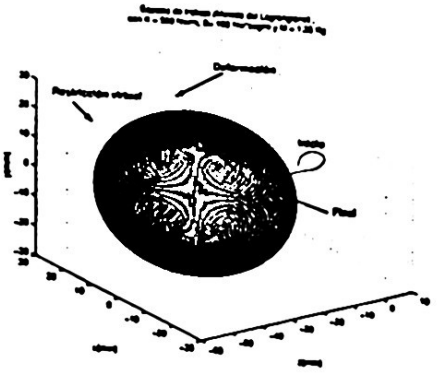


Fig. 9. Lagrangiano Restringido en una esfera virtual

2. M.A. Srinivasan, *"Haptics Research at the MIT Touch Lab"*, The First Phantom user's Group Workshop, Massachusetts Institute of Technology, Cambridge, MA, pp. 10-13, 1996.
3. K. Salisbury, *"An Overview of Haptics Research at MIT's AI Lab"*, The First Phantom user's Group Workshop, Massachusetts Institute of Technology, Cambridge, MA, pp. 6-9, 1996.
4. Salisbury, J.K. and Mandayam A. Srinivasan. *"Phantom-Based Haptic Interaction with Virtual Objects"*, IEEE Computer Graphics and Applications; Volume 17, Number 5, pp. 6-10; September-October 1997.
5. SensAble Technologies, Inc., *"GHOST SDK Version 3.1: Programmer's Guide"*, Rev.1.31/3/00.
6. SensAble Technologies, Inc., *"GHOST SDK Version 3.1: API Reference"*, Rev.1.31/3/00.
7. M. Moreyra, B. Hannaford, *"A Practical Measure of Dynamic Response of Haptic Devices"*, International Conference on Robotics and Automation, Proceedings of the 1998 IEEE, pp. 369-374, Leuven Belgium, 1998.
8. T. Yoshikawa, A. Nagura, *"Development and Control of Touch and Force Display Device for Haptic Interface"*, IFAC, pp. 127-132, 2000.
9. V. Parra-Vega, *"Telepresence with Master-Slave Robots using a Nonlinear Formulation without Time Delay"*, Machatronics Division, CINVESTAV, México.
   S. Arimoto, Yun-Hui Liu, V. Parra-Vega, *"Design of Model-Based Adaptive Controllers for Robot Manipulators Under Geometric Constraints"*, The American Society of Mechanical Engineers, Proceesings of the Japan U.S.A., Symposium on Flexible Automation, Book No. I0338A-1992.
   Omar A. Domínguez-Ramírez, Vicente Parra-Vega, *"Constrained Lagrangian-Based Force-Position Control Haptic Guidance"*, EuroHaptics 2003, Dublin, Ireland, 2003.
   Omar A. Domínguez-Ramírez, Vicente Parra-Vega, *"Active Haptic Interface with Purposes of Remote Training"*, The 11th International Conference on Advanced Robotics , University of Coimbra, Portugal, 2003.
   Omar A. Domínguez-Ramírez, Vicente Parra-Vega, *"Texture, Roughness, and Shape Haptic Perception of Deformable Virtual Objects with Constrained La-*

Método \ Objeto virtual	Plano virtual		Esfera virtual	
Método de Penalización	K=200 N/m	K=2000 N/m	K=500 N/m	K=2000 N/m
	Inestable en SC DSC no controlada	Estabilidad en SC (Fsc = 14.28 Hz) DSC = 0	Inestable SC (11.11Hz<Fsc<500Hz) DSC no controlada	Inestabilidad en SC (500Hz<Fsc<1KHz) DSC = 0
Método de Rigidez y Amortiguamiento Simultáneos	K=200 N/m B=100 Ns/m	K=2000 N/m B=100 Ns/m	K=500 N/m B=100 Ns/m	K=2000 N/m B=100 Ns/m
	Baja estabilidad en SC. Fsc = 7.14Hz DSC = controlada	Estabilidad en SC Fsc = 11.11 Hz DSC = 0	Estabilidad en SC Fsc=8.4 Hz DSC = controlada	Estabilidad en SC (50Hz<Fsc<300Hz) DSC = 0
Método del Lagrangiano Restringido	K=200 N/m B=100 Ns/m M=12.5 Kg	K=200 N/m B=100 Ns/m M=1.25 Kg	K=500 N/m B=100 Ns/m M=1.25 Kg	K=2000 N/m B=100 Ns/m M=1.25 Kg
	Estabilidad en SC Fsc = 0 Hz DSC = controlada (condición óptima)	Estabilidad en SC Fsc = 2.3 Hz DSC = baja	Estabilidad en SC Fsc = 0 Hz DSC = controlada (condición óptima)	Estabilidad en SC Fsc = 22.2 Hz DSC = baja

SC: Superficie de Contacto
Fsc: Frecuencia sobre la superficie de contacto
DSC: Deformación de la superficie de contacto

Fig. 10. Tabla comparativa de métodos para generar fuerza de reacción

grangian Formulation", The 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, USA, 2003.

Control remoto universal a distancia DTMF por línea telefónica

Andrés Gerardo Fuentes Covarrubias, Fermín Pascual Estrada González, Ricardo Fuentes Covarrubias

Facultad de Ingeniería Mecánica y Eléctrica – Universidad de Colima
Km. 9 Carretera Colima-Coquimatlán; C.P. 28400 – Coquimatlán, Col., México
fuentesg@ucol.mx; fermineg@venus.ucol.mx; fuentesr@ucol.mx

Resumen. Este trabajo presenta una versión personalizada de control remoto doméstico inteligente usando la línea telefónica como medio de transmisión para gobernar un máximo de cuatro aparatos eléctricos, basada en el intercambio tanto de señales DTMF (*Dual Tone Multiple Frecuency*) como de frases verbales, aprovechando para este propósito los recursos tecnológicos disponibles y la base instalada de la telefonía actual.

Introducción

codificación por tonos dobles de frecuencia múltiple es la más difundida para el intercambio de información en la forma de cantidades binarias representadas con grupos de cuatro bits, relacionadas principalmente con las 16 teclas de los aparatos telefónicos de tonos ó digitales.

La señalización telefónica sigue siendo una forma confiable de transmisión de datos la cual ofrece las siguientes ventajas únicas:

Una red mundial ya instalada y de uso económico.
Terminales de usuario económicas y fáciles de usar (aparatos telefónicos).
Hardware accesible.

Planteamiento del proyecto

Las aplicaciones para este sistema de control pueden ser tanto domésticas (figura 1) como industriales, ya que solo es necesario contar con una línea telefónica para implementarlo y puede controlar cargas tanto de corriente directa como de corriente alterna, limitadas en este trabajo a un consumo máximo de corriente de 10 amperes para cada dispositivo, debido a que esta es la cantidad máxima de corriente que demanda una aplicación doméstica crítica.

Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 311–324, 2003. © Centro de Investigación en Computación, IPN, México.

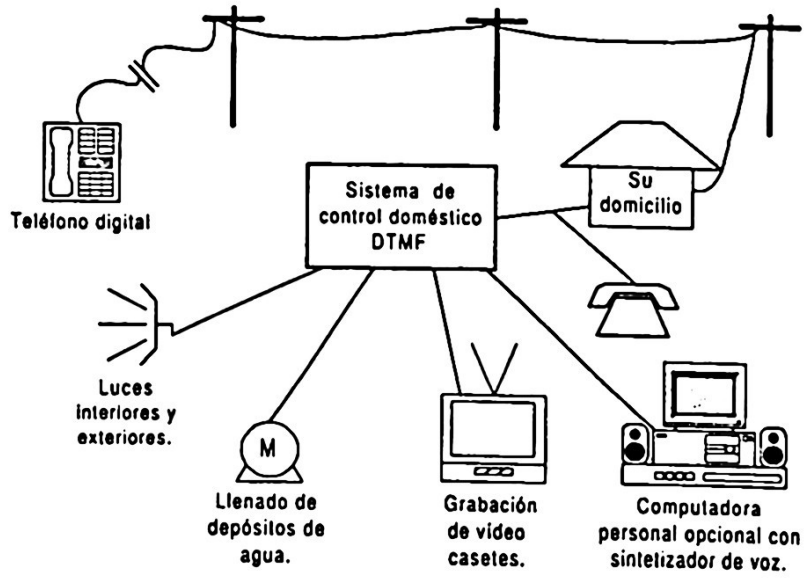


Fig. 1. Sistema de control remoto DTMF doméstico.

El sistema de control DTMF propuesto consta de 5 módulos principales mostrados en diagrama de bloques de la figura 2.

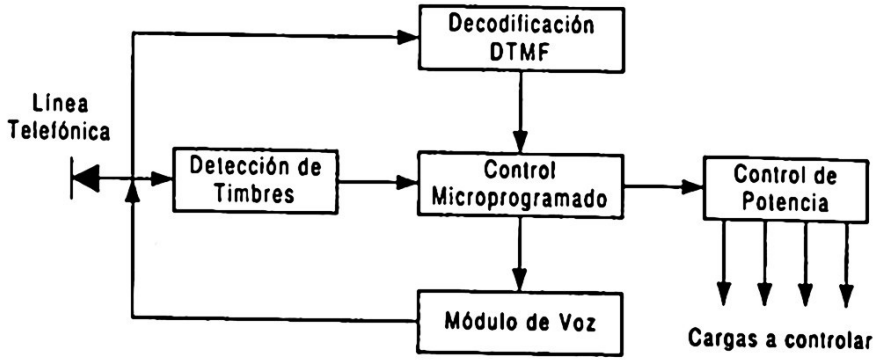


Fig. 2. Diagrama de bloques del control remoto DTMF doméstico.

Como se puede observar, cada uno de éstos tiene una tarea particular que mpeñar dentro del sistema, estas funciones son:

Detector de timbres: Es el encargado de traducir las señales de requerimiento de respuesta por parte del abonado remoto y convertirlas a pulsos de nivel TTL para que éstas a su vez puedan ser reconocidas por el módulo de control microprogramado.

Decodificador DTMF: Lleva a cabo la función de detectar los tonos dobles de frecuencia múltiple en la línea telefónica y reportar las cantidades binarias relacionadas con dichos tonos al módulo de control.

Módulo de voz: Este módulo tiene como función principal, la de proporcionar guía verbal al usuario sobre las acciones que tienen lugar en el sistema de control

remoto. Éste es controlado en su totalidad por el módulo de control microprogramado.

Control de potencia: Esta etapa del sistema tiene como finalidad controlar las cargas de manera aislada a todos los demás módulos para protegerlos de sobrevoltajes, cortocircuitos y transitorios en la línea de suministro de alto voltaje.

Control microprogramado: Es el cerebro del sistema ya que es aquí donde tiene lugar el procesamiento de las señales provenientes de los módulos de detección de timbres, decodificación DTMF y final de reproducción de mensaje, para convertir estos estímulos externos en acciones concretas tales como; descolgar / colgar la línea telefónica, activar / desactivar una carga, emitir una frase de guía.

Señalización en el intercambio de información

intercambio de información en este sistema se da en dos sentidos; entre el usuario y stema de control principal por medio de los tonos producidos por el teclado de un arato telefónico digital (señales DTMF) el cual puede ser remoto o local y la guía rbal como respuesta del sistema para llevar a cabo las acciones, esto último permite usuario interactuar con el sistema de manera natural.

Las respuestas del sistema se basan en frases completas para indicar al usuario la ción que ha elegido realizar, para confirmar que dicha acción ha sido efectuada o ra indicar que la acción ha sido rechazada. Las respuestas e indicaciones verbales sistema han sido previamente almacenadas en un circuito integrado que permite el ceso a cada una de las palabras y frases.

Los comandos DTMF tienen como finalidad provocar una acción de programación requerir información de parte del sistema de control principal. Estos comandos son siguientes:

(Desactivar Dispositivo). Permite inactivar una de 4 cargas posibles.
(Activar Dispositivo). Comando que permite activar una de 4 cargas posibles.
(Consultar estado del sistema). Consultar el estado de dispositivos controlados por el sistema.
(Cambiar número de timbres). Para programar el nuevo número de timbres antes de descolgar para conectar el sistema de control principal a la línea telefónica.
(Cambiar clave de acceso). Permite programar una nueva clave de acceso.
(Salir del sistema). Comando para colgar la línea telefónica y quedar a la espera de una nueva sesión.
(Repetir opciones del sistema). Permite que el sistema repita para el usuario las opciones disponibles y su descripción.
(Omitir el menú principal). Comando que permite saltar el menú principal y pasar directamente a la consulta o programación.

2.2 Consulta y programación del sistema

El estado del sistema puede ser consultado por medio del comando 2 una vez que se haya iniciado la sesión de programación o consulta.

Al seleccionar este comando el sistema emitirá la frase "Consultar estado del sistema, Ingrese número de dispositivo" y se pondrá a la espera de un dato comprendido entre el 1 y el 4 para informar de inmediato por medio de la frase "Dispositivo activo / inactivo" si el dispositivo o carga en cuestión se encuentra activo o inactivo respectivamente, después de esto el sistema queda en espera del siguiente comando.

Cuando el usuario requiera consultar o programar el sistema de control remoto necesariamente tendrá que marcar el número del abonado donde se encuentra conectado este sistema. Una vez que el número de timbres programados es detectado y validado, el sistema descuelga y emite la frase "Ingrese la clave de acceso" quedando a la espera de una combinación de 4 dígitos los cuales serán comparados contra la clave con una cantidad igual de dígitos programada previamente, en caso de que no exista coincidencia entre estas dos combinaciones el sistema emitirá la frase "Clave de acceso no válida, intente de nuevo", una vez que la clave correcta haya sido ingresada el sistema emitirá de manera verbal el menú con las ocho opciones, después del cual quedara iniciada la sesión de programación y consulta.

La clave de acceso se encuentra conformada por cuatro dígitos (por defecto 0-0-0-0) y solo permite tres intentos para introducir el código correcto, después de los cuales el sistema colgará si ninguna combinación coincide con la programada, esto permite un nivel de seguridad de 11,880 combinaciones ya que en un teléfono estándar existen 12 teclas diferentes (indicadas en la tabla 1 junto a su correspondiente código binario).

Tecla	Código binario	Tecla	Código binario
1	0001	7	0111
2	0010	8	1000
3	0010	9	1001
4	0100	0	1010
5	0101	*	1011
6	0110	#	1100

Tabla 1. Relación del teclado con su correspondiente código binario.

Para colgar el usuario deberá introducir el comando 5, después de esta acción el sistema emitirá la frase "Para salir del sistema presione el número uno, o cualquier otro para cancelar" quedando a la espera de la opción que tomará el usuario, si cualquier número diferente del uno es presionado el sistema emitirá la frase "salida del sistema cancelada", pero si por otra parte, la opción de salir es seleccionada el sistema colgará sin mas preámbulos.

Una vez realizados los cambios o consultas al sistema de control remoto y si el usuario desea colgar podrá emplear el comando 5, de no ser así el usuario podrá colgar sin teclear éste comando o si esto llegará a ocurrir por descuido, el sistema colgará de manera automática. Para prevenir que la línea quede descolgada y no permitir accesos posteriores, el sistema se encuentra provisto de un temporizador que

verifica la actividad DTMF, de no detectar dicha actividad contabiliza 30 segundos para después colgar la línea y regresar al modo de ahorro de energía.

2.3 Cambios en el sistema y control de las cargas

Algunos parámetros del sistema de control remoto por señalización DTMF pueden ser modificados para permitir al usuario la personalización de éste, los cambios posibles son:

- Control de los dispositivos o cargas.
- Clave de acceso.
- Número de timbres.

Las cargas pueden ser activadas o desactivadas, una a la vez empleando los comandos 1 o 0 respectivamente. Cuando el comando elegido haya sido introducido al sistema, éste responderá con la frase "Activar / Desactivar carga, ingrese número de dispositivo" después de esto quedara a la espera de un dato comprendido entre el 1 y el 4 para indicar en cual carga se llevará a cabo la modificación.

Una vez introducido el número correspondiente a la carga en cuestión el sistema emitirá la frase "Dispositivo activado / inactivado" para indicar que el cambio ha tenido lugar, o la frase "Dispositivo activado / desactivado previamente" para indicar que la carga se encuentra en el estado que se desea programar.

En el caso de que se introduzca un valor diferente del intervalo 1 a 4 el sistema emitirá la frase "Selección fuera de rango".

En cualquiera de los casos anteriores el sistema regresará al estado de espera de un nuevo comando.

La clave de acceso puede ser modificada con el comando 4. Una vez tecleado el comando, el sistema instruirá al usuario con la frase "Cambiar clave de acceso, ingrese nueva clave de acceso" después esperará el nuevo código de acceso de cuatro dígitos. Ya introducido el código tendrá que ser confirmado para verificar que no haya errores en la clave deseada por el usuario, es por esta razón que el sistema solicitará la confirmación por medio de la frase "Confirme clave de acceso", una vez que el par de códigos han sido introducidos se verifica la correspondencia y solo en caso de coincidencia se almacena la que será la nueva clave de acceso, de haber algún error en el par de claves introducidas el sistema responderá con la frase "La clave de acceso no actualizada" sin llevar a cabo ninguna modificación. Los dígitos permitidos para la clave son: 1, 2, 3, 4, 5, 6, 7, 8, 9, 0, * y #.

El sistema tiene programado por defecto 3 timbres para descolgar la línea, el número de estos puede ser de 1 a 12 de acuerdo a las necesidades particulares del usuario.

El número de timbres puede ser reprogramado dentro de la sesión de programación y consulta con el comando 3 desde el teclado del teléfono remoto o local, cuando este comando sea validado, el sistema emitirá la frase "Ingrese el nuevo número de timbres" quedando a la espera de un solo dígito comprendido entre el 1 y el numeral (#). En la tabla 2.1 se encuentran indicadas las teclas y su correspondiente valor en binario, de manera que el 0 codifica al 10, el asterisco al 11 y el numeral al 12. Teniendo en cuenta esto se puede programar una cantidad máxima de 12 timbres.

Cuando el usuario seleccione el nuevo número de timbres, el sistema emitirá la frase "Número de timbres actualizado" dando por terminada esta acción y quedando a la espera de un nuevo comando.

2.4 Consulta y programación en sitio

El control remoto por señalización DTMF también puede ser consultado y programado desde el aparato telefónico del domicilio en que se encuentra instalado. Para tal efecto solo es necesario descolgar el auricular y activar el interruptor de presión en el sistema de control, el cual ha sido dispuesto para simular el timbre e indicar requerimiento para acceder al sistema por parte del usuario.

El interruptor deberá ser presionado tantas veces como número de timbres hayan sido programados en la última sesión, antes de que el sistema descuelgue, se ponga en comunicación con el usuario y le solicite la clave de acceso.

3 Codificación y decodificación DTMF

Una señal DTMF válida es la suma de dos tonos, uno de un grupo bajo y el otro de un grupo alto, con cada grupo conteniendo cuatro tonos individuales. Las frecuencias de los tonos fueron cuidadosamente seleccionadas de tal forma que sus armónicos no se encuentran relacionados y que los productos de su intermodulación produzcan un deterioro mínimo en la señalización. Este esquema permite 16 combinaciones únicas. Diez de estos códigos representan los números del cero al nueve, los seis restantes (*, #, A,B,C,D) son reservados para señalización especial. La mayoría de los teclados en los teléfonos contienen diez interruptores de presión numéricos mas el asterisco (*) y el símbolo de numeral (#). Los interruptores se encuentran organizados en una matriz, cada uno selecciona el tono del grupo bajo de su fila respectiva y el tono del grupo alto de su columna correspondiente.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2 ABC	3 DEF	A
770 Hz	4 GHI	5 JKL	6 MNO	B
852 Hz	7 PRS	8 TUV	9 WXYZ	C
941 Hz	*	0 oper	#	D

Tabla 2. Relación del teclado con su correspondiente código binario.

3.1 Codificación DTMF

esquema de marcado DTMF fue diseñado por los laboratorios BELL e introducido los Estados Unidos a mediados de los años 60 como una alternativa para a la marcación por pulsos o rotatoria. Ofreciendo incremento en la velocidad de marcado, mejorando la fiabilidad y la conveniencia de señalización de punto a punto.

Existen varias especificaciones que han sido resultado de el estándar original las cuales parten de los estándares de AT&T, CEPT, NTT, CCITT y la ITU, etc. Las variaciones de un estándar a otro son típicamente tolerancias en las desviaciones de frecuencia, niveles de energía, diferencia de atenuación entre dos tonos e inmunidad al habla.

Los estándares CCITT se encuentran localizados en las recomendaciones Q.23 y Q.24 en la sección 4.3 del libro rojo de la CCITT, volumen VI, fascículo VI.1.

En conclusión, DTMF es el sistema de señales usado en los teléfonos para el marcado por tonos, estos son el resultado de la suma algebraica en tiempo real de dos senoides de diferentes frecuencias, la relación de teclas con su correspondiente par de frecuencias se muestra en la tabla.

El sistema de señales DTMF son generadas por un codificador, y son la suma algebraica en tiempo real de dos tonos; uno de baja frecuencia y otro de alta, el tono alto normalmente es de + 1.5 % (2db) con respecto del tono bajo (como se muestra en la figura 1) para compensar perdidas de señal en las largas líneas de conexión con la central telefónica.

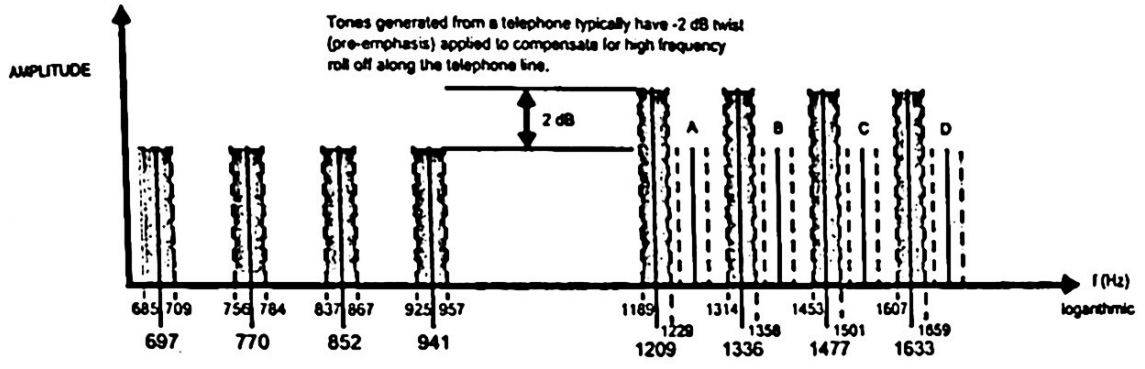


Fig. 3. Espectro de las señales DTMF.

3.2 Descodificación DTMF

La división de frecuencias en los grupos alto y bajo simplifica el diseño de receptores DTMF como se muestra en la figura 4. Este diseño particular incluye una aproximación estándar. Cuando se encuentra conectado a una línea telefónica, receptor de radio o cualquier otra fuente de señal DTMF, el receptor filtra el ruido del tono, separa la señal en los componentes de grupos de alta y baja frecuencia para luego medir el cruce por cero promediando los periodos para producir la decodificación de un dígito.

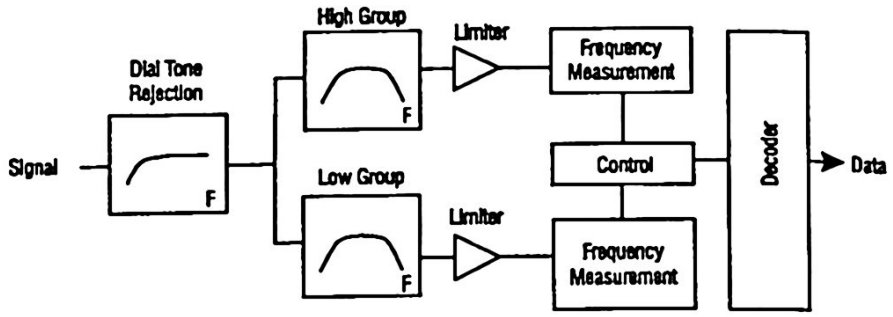


Fig. 4. Arquitectura típica de un receptor DTMF.

La detección DTMF se puede ver complicada por la presencia de ruido de línea de 50/60 Hz , tonos de varias frecuencias, ruido aleatorio y otras fuentes de interferencia. Tratar con estos problemas mientras permanece inmune a la simulación de tonos por voz presenta el mas grande reto para los diseñadores de receptores DTMF.

3.3 Sistema de recepción para las especificación 1151 de British Telecom

El circuito mostrado en la figura 3.4 ilustra el uso de un dispositivo MT8870 en un sistema de recepción típico. Las especificaciones de British Telecom definen las señales de entrada menores de –34 dBm como niveles no operativos. Esta condición puede ser evitada seleccionando valores adecuados de R_1 y R_2 para proporcionar una atenuación de 3 dB, de tal forma que la señal entrada de –34 dBm corresponderá a –37 dBm en la terminal de programación de ganancia (GS). En el diagrama se muestran los valores de los componentes de R_3 y C_2 para los requerimiento de tiempo de guarda cuando la tolerancia total de los componentes es del 6%.

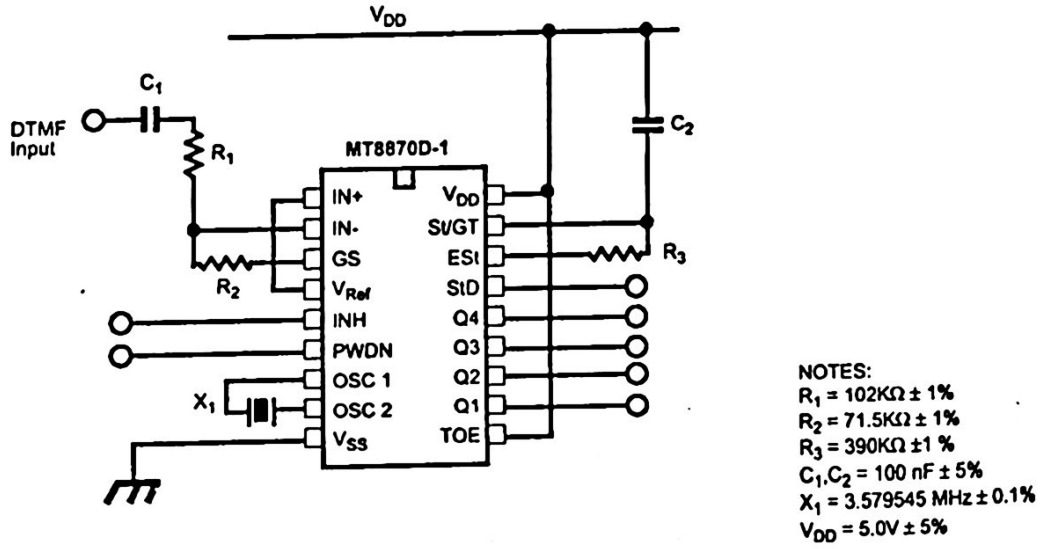


Fig. 5. Configuración de entrada de ganancia unitaria para las especificaciones BT o CEPT.

4 Integración del sistema

El sistema de control se encuentra conformado físicamente por los cinco módulos principales indicados en la figura 1. Cada uno de estos se compone a su vez de submódulos operativos que tienen funciones tan importantes como: Acondicionamiento de señales, aislamiento físico de la línea telefónica con el sistema de control digital principal, descolgado / colgado de la línea y control de bajo voltaje de los módulos de potencia solo por mencionar algunos.

4.1 Interfaz con la línea telefónica

El microcontrolador requiere de señalización que le indiquen cuando ocurre un timbre (señal de requerimiento de respuesta por parte del abonado llamado), dicha señalización necesariamente debe primero, ser aislada eléctricamente y acondicionada para después ser entregada a una terminal de entrada del microcontrolador programada previamente como entrada y sensible a cambios de estado (habilitada para provocar interrupciones por cambio de estado de la señal de entrada) esto con la finalidad de poder contabilizar dichos timbres. Esta etapa es la de detección de timbrado.

El sistema de control se encuentra permanentemente conectado a la línea telefónica, esta condición hace necesaria una etapa crucial para la prevención de daños accidentales debidos principalmente a fenómenos meteorológicos, ya que estos son la causa de la mayor parte de los estragos en los sistemas que emplean la infraestructura telefónica instalada, esta etapa es la de protección por sobrecarga.

Una etapa de colgado / descolgado de la línea telefónica es necesaria debido a que la mayor parte del tiempo el sistema de control se encuentra sin conexión directa con la línea en cuestión, para esto, el sistema debe tener forma de conectarse a esta (descolgar) y desconectarse (colgar) cuando la consulta de estado o programación haya sido terminada.

4.1.1 Señales eléctricas en la línea telefónica

Este módulo es el responsable de acondicionar la señal de timbre presente en la línea telefónica, su función es la de mantener un nivel lógico TTL alto cuando esta señal se encuentra presente en la línea, los cambios de estado generados por este módulo indican que se requiere de respuesta a una petición remota de atención por parte del sistema de control local, estos pulsos se generan de forma tal que puedan ser reconocidos por el sistema de microcontrol dedicado y efectuar el descolgado automático.

4.1.2. Detección y acondicionamiento del timbre.

Este proceso requiere primeramente de una etapa de reducción de corriente proporcionada por dos resistencias de 100KΩ a ¼ de watt, seguido de un sistema de separación de las componentes de corriente directa y alterna presentes en el proceso de timbre, mismo que se lleva a cabo con una capacitor que permite el paso de

uficiente energía producida por el voltaje pulsante proveniente de la línea para ser osteriormente rectificada y proveer un nivel de voltaje de operación seguro para los omponentes de estado sólido que traducirán la actividad de esta señal alterna. [3].

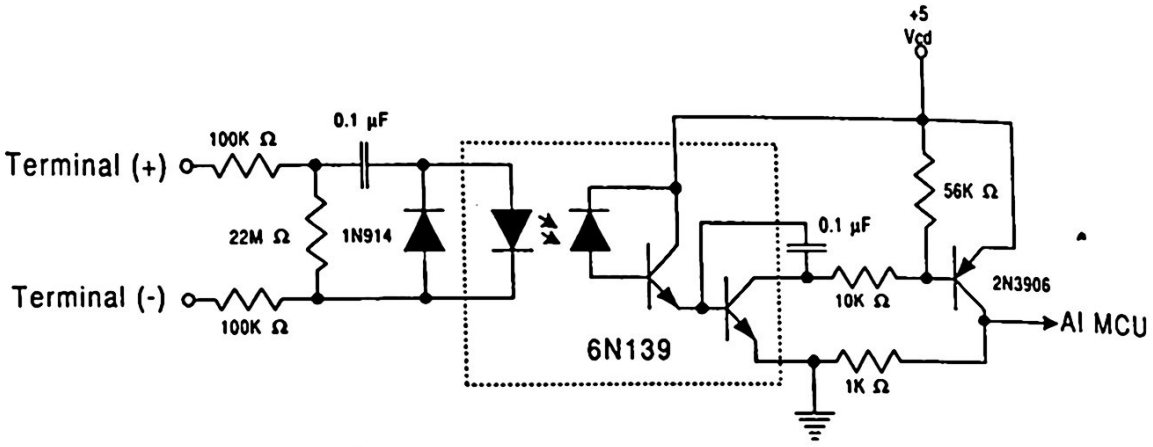


Fig. 6. Diagrama del sistema de detección y acondicionamiento de timbre.

1.3 Protección por sobrecarga

La amplitud máxima de las señales requeridas en la comunicación vía telefónica no sobrepasa los 140 voltios en condiciones normales, de tal manera que toda señal en la nea que exceda ese nivel deberá ser considerado como sobrecarga en la misma.

El uso de un varistor puede solucionar este requerimiento en gran medida ya que estos dispositivos reducen su resistencia conforme el voltaje de ruptura en sus terminales se incrementa, operan en paralelo con la línea de entrada y su resistencia va de unos cientos de Kilo ohmios hasta varios Mega ohmios dependiendo del voltaje de bloqueo para el que estén diseñados, una buena elección es el MOV150 que opera un bloqueo de 150 voltios el cual dará protección al circuito de detección durante una descarga sostenida no mayor a 2 segundos.

4.1.4. Colgado / descolgado de la línea.

Esta función se encuentra bajo el control de una de las líneas de E/S del microcontrolador principal y está a cargo de un relevador de circuito doble, con un voltaje de conmutación de 12V de corriente directa, el cual opera de acuerdo al diagrama de la figura 7. El objetivo principal de esta etapa es seleccionar uno de dos circuitos posibles:

1. Circuito de detección de timbrado con aislamiento óptico.
2. Circuito de entrada / salida de señales del sistema de control principal, aislada de forma inductiva.

En su estado de reposo el relevador se encuentra conectado al primer circuito, debido a que, es en este estado en el que se encuentra la mayor parte del tiempo. De esta manera, no requiere de polarización permanente por parte del microcontrolador.

Cuando el sistema es requerido por un usuario remoto o local tendrán que ocurrir el número de timbres especificado en la última programación del sistema, para que éste descuelgue. La acción de descolgado es llevada a cabo por la línea de control del microcontrolador principal programada como salida y polarizando con una corriente de 2.55mA el transistor T1 y éste a su vez activando el relevador.

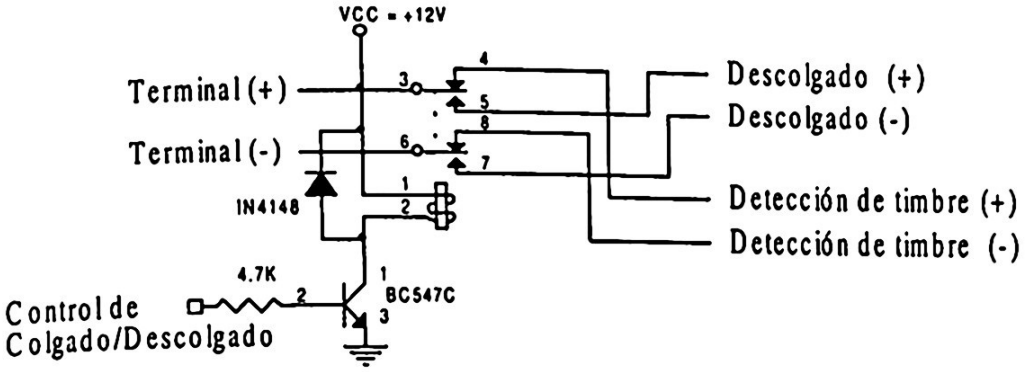


Fig. 7. Sistema de Colgado / Descolgado de la línea telefónica.

4.1.5 Aislamiento eléctrico del sistema digital

Para que el sistema de control pueda manipular y traducir las señales provenientes de la línea telefónica requiere de un sistema de aislamiento eléctrico para evitar el contacto directo con ésta, ya que como se ha dicho; la diferencia de potencial, polaridad y sobrecargas pueden causar daños al sistema digital.

Para esta función en particular se emplea un transformador MTLM 1200E de 600Ω de impedancia en primario y secundario como se indica en la figura 8.

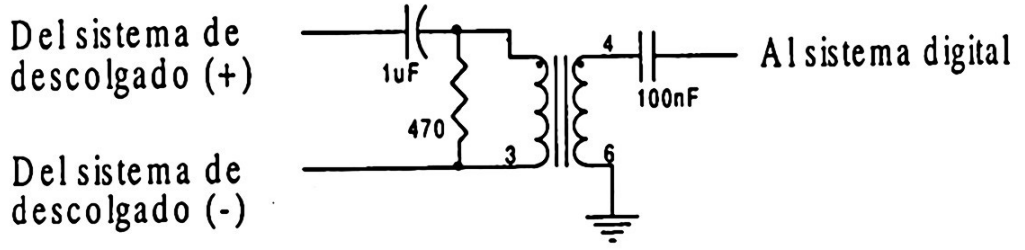


Fig. 8. Sistema de aislamiento eléctrico de la línea telefónica.

4.1.6 Decodificación DTMF

Los ingenieros de Laboratorios Bell eligieron pares de frecuencias específicas (tabla 2) para evitar armónicos y otros problemas que pudieran presentarse cuando estos tonos son enviados o recibidos, además de la dificultad que presentan para ser imitados por la voz humana y puedan accidentalmente activar el decodificador del otro lado de la línea.

En este caso particular la decodificación se encuentra a cargo de un circuito integrado MT8870CE fabricado por MITEL, el cual reconoce los tonos dobles generados por el teclado de un teléfono en modalidad digital, estos tonos son decodificados y después convertidos a cantidades binarias de 4 bits que corresponden

a un valor numérico determinado. Estos valores binarios funcionan comocomandos o datos de acuerdo al orden con que son ingresados al sistema de control.

Este decodificador de señales DTMF se encuentra provisto además de las características ya mencionadas, de una terminal que indica si una señal DTMF válida ha sido recibida, decodificada y su correspondiente código binario capturado en el registro de salida, esta terminal es llamada StD y permite llevar a cabo una interacción en tiempo real con el sistema de control empleándola para provocar una interrupción de manera tal que el controlador atiende este acción prioritaria ejecutando la rutina de atención a interrupción, misma que se limita a leer el código binario presente en el registro de salida del decodificador DTMF, para que posteriormente el programa principal le de el tratamiento de comando o dato según sea el caso y efectuar la acción correspondiente dentro del sistema de control remoto.

4.1.7 Microcontrol dedicado

El control del sistema puede ser implementado de diferentes maneras; una opción podría ser el control ejecutado por una computadora personal, la cual cuenta con recursos hardware y software que la hacen fácil de emplear para esta tarea, pero con la desventaja significativa de que el costo del equipo y de operación es elevado considerando que un equipo de cómputo promedio consume aproximadamente 350 Watts y que sería necesario mantenerlo en operación permanentemente, además de que las dimensiones y peso no permiten el cambio de ubicación fácilmente.

Para salvar estas desventajas, el controlador principal del sistema en este caso es implementado con un microcontrolador PIC16F84 fabricado por Microchip® y programado para este propósito particular, el cual se encuentra a cargo de realizar las acciones de respuesta correspondientes a cada uno de los comandos ó impulsos de entrada generados por los módulos de acondicionamiento de señal anteriores y controlar el encendido ó apagado de cada uno de los cuatro dispositivos a gobernar.

Este microcontrolador fue seleccionado de entre una gran variedad de dispositivos y marcas debido a sus características principales: fácil implementación, disponibilidad y costo accesible.

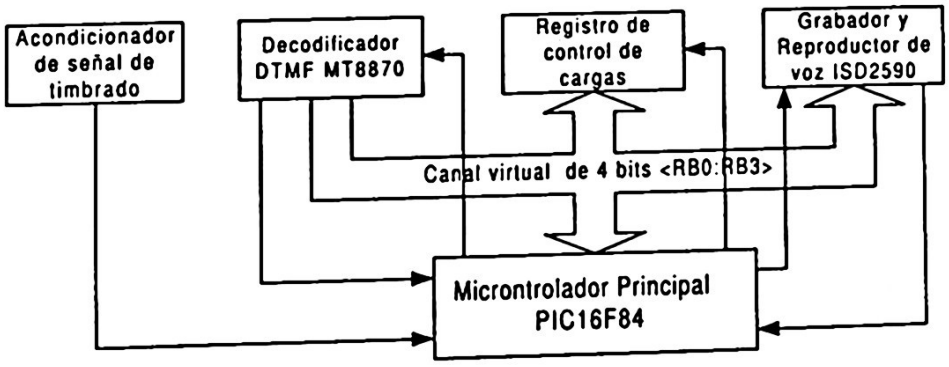


Fig. 9. Diagrama de bloques del control microprogramado.

La disposición funcional de los elementos que integran el sistema de control remoto, depende directamente de las prestaciones con que cuenta el microcontrolador que es el elemento de control principal, como se indica en el la figura 9. En este diagrama de bloques también se muestra la distribución de terminales y la función de cada una de ellas dentro del proceso de comunicación con los demás circuitos integrados. Debido al bajo número de terminales del controlador, se aprovecharon

algunos de los recursos de comunicación similares entre los circuitos integrados, como el caso de la transferencia de datos en paralelo que permite la implementación de un canal virtual de 4 bits.

4.1.9 Control de potencia

Este módulo es la interfaz entre el sistema de control digital principal y las cargas a controlar, el cual es implementado con relevadores que soportan una carga máxima de 0 amperes cada uno.

Es importante que esta etapa permanezca aislada por completo del sistema digital para prevenir descargas accidentales sobre este último, para este propósito se dispone de un relevador por cada carga, el cual tiene la función de separar eléctricamente los dos módulos. El sistema de control digital provee una línea de control independiente para cada uno de los cuatro relevadores (como se muestra en la figura 10) que serán accionados de acuerdo a la programación del usuario.

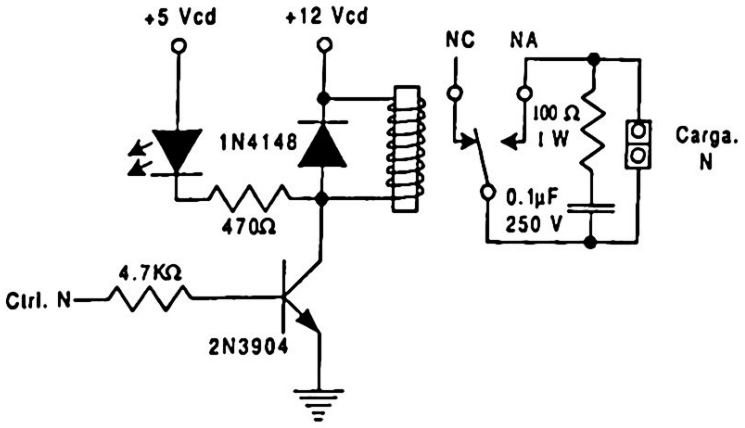


Fig. 10. Control de potencia individual.

Conclusiones

Durante el proceso de investigación y documentación del presente proyecto se ncontró que las aplicaciones de la transmisión de tonos dobles de frecuencia múltiple DTMF tienen un número de aplicaciones cada vez mayor debido a la creciente emanda de servicios que requieren de confiabilidad en el intercambio de información una gran variedad de formatos. El número creciente de fabricantes de circuitos ntegrados especializados provee a los fabricantes y diseñadores de sistemas ectrónicos que requieren procesar información en forma de tonos dobles de una mplia gama de opciones, estos circuitos integrados la mayoría de ellos de fácil nfiguración y costos accesibles que llevan a cabo la tarea de decodificar, generar y ansmitir los tonos DTMF también se encuentran dotados de las interfaces mas opulares para microcontroladores como son: paralelas con tercer estado, SPI y icrowire® solo por mencionar algunos.

Además, es importante tener en consideración que varias de las propiedades seables en los diferentes circuitos integrados que conforman el sistema de control moto tienen efectos sobre el desempeño de éste.

Referencias

[1] Deosthali, Amey. "Dual tone multiple frecuency (DTMF) detection on microcontrollers", Shlumberger Technology Corporation, February, 1999.

[2] Teltone. "Aplications for DTMF and pulse telephone dialing, including remote access and control", www.teltone.com

[3] www.homeautomator.com

Diseño y construcción de una caja de actividad motora con tecnología de microcontroladores para la captura y análisis de datos de ratas en experimentación

Francisco J. Heredia López [1], Karen Z. Navarrete Kao [2]

[1] Universidad Autónoma de Yucatán, Centro de Investigaciones Regionales "Dr. Hideyo Noguchi". Laboratorio de Neurofisiología.
hlopez@tunku.uady.mx

[2] Universidad Autónoma de Yucatán, Facultad de Matemáticas.
karnava@todito.com

Resumen. Este proyecto describe el funcionamiento de un sistema electrónico basado en un microcontrolador "RISC" que controla una caja para registrar la actividad motora en ratas. Es un medio eficaz para obtener de manera automática información de estos animales en experimentación sin necesidad de dedicar de manera exclusiva una computadora PC para dicha tarea.

Abstract. This project describes the operation of an electronic system based on a microcontrolador RISC that controls a box to register the motor activity in rats. It is an efficient method to obtain in an automatic way information of these animals in experimentation with no dedicate a PC computer for that task.

Introducción

uso de computadoras personales se ha hecho común hoy en día, gracias al costo, facilidad de adquisición y capacidad de proceso para el desarrollo científico y tecnológico, permitiendo utilizarlas como estaciones de trabajo en las cuales es posible con los programas y lenguajes adecuados poder realizar la captura de mas de una fuente de información de manera casi simultanea.

Dentro de la investigación biomédica, las neurociencias se ocupan del estudio del sistema nervioso, tratando de comprender mejor los fenómenos que rigen el funcionamiento del mismo. Para ello se realizan, entre otros, estudios conductuales de animales vivos, como ratas y conejos. Así, es posible valorar el funcionamiento del sistema nervioso al establecer una correlación entre los cambios en la actividad motora exhibida por el animal en estudio y la naturaleza del fármaco empleado. Entre las metodologías de estudio de alteraciones en la conducta motora, se encuentra el uso una caja dotada con sensores en las paredes de la misma, o bien con una cámara de video en su parte superior, para capturar la actividad motora y por medio de programas de análisis evaluar las alteraciones conductuales.

En el Laboratorio de Neurofisiología del Centro de Investigaciones Regionales "Dr. Hideyo Noguchi" de la UADY se llevan acabo diversas investigaciones de este tipo. En nuestro caso se emplea una caja dotada con sensores en las paredes con la

Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 325–332, 2003. © Centro de Investigación en Computación, IPN, México.

cual se puede ubicar una posición en x, y, z para poder la actividad motora del animal experimentación.

Antecedentes

Geyer M.A.[1] y Sandra E[2], reportan el uso de un sistema basado en haces frarrojos para medir la actividad espontánea en ratas con el propósito de determinar el efecto en su conducta debida a la administración de drogas y otras manipulaciones. Es un dispositivo automatizado que permite determinar la cantidad y la cualidad de la actividad conductual en ratas.

Neil T. Brockwell [3] describe un sistema computarizado para el monitoreo del condicionamiento de estancia y actividad de desplazamiento en ratas, estos stemas pueden controlar varios gabinetes de experimentación a la vez. Cada gabinete consiste de dos cajas conectadas por un pequeño túnel en los cuales se pueden insertar puertas tipo guillotina. En las paredes del gabinete se fijan pares de sensores infrarrojos. En ambos casos se utilizan como elementos sensores emisores y detectores infrarrojos.

Un segundo grupo de metodologías para sensar conducta motora contempla el uso de sistemas de video como los descritos por Ilan Golani et al [4], Schwarting et al [5], Frank Sams Dodd [6]. En estos sistemas se miden la conducta de giro, exploración, y desplazamiento en ratas. Todos estos sistema requieren del uso de programas de computación para realizar el análisis de la conducta observada, representando a la rata como un patrón de puntos blancos.

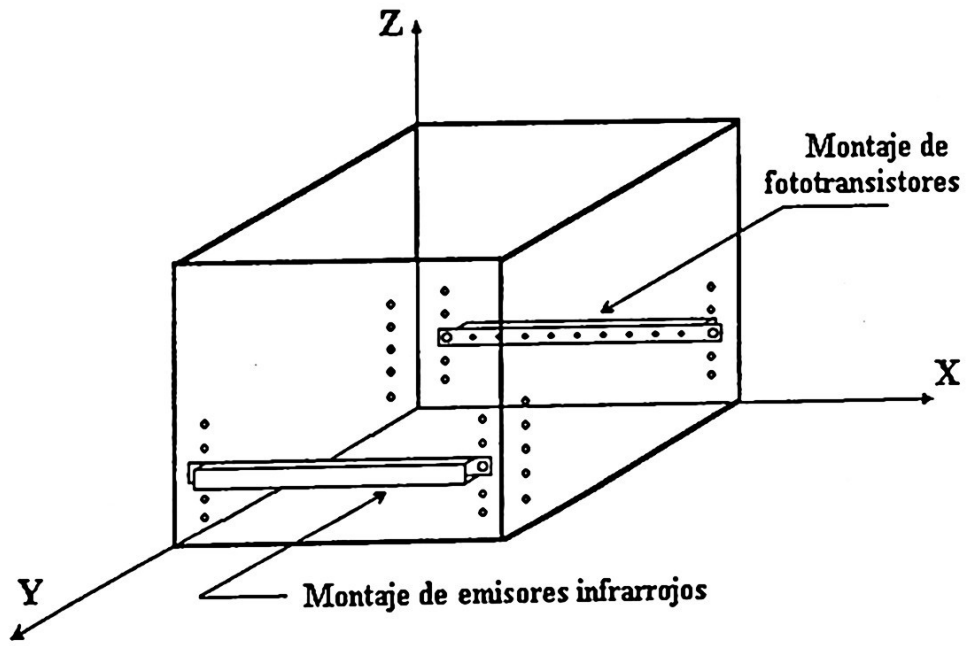


Fig. 1. Esquema de la caja de acrílico con un par de rieles que forma un eje coordenado

Para el caso de nuestro laboratorio se optó por utilizar elementos frarrojos [14] como el medio para poder detectar cambios en la conducta

motora en ratas, para ello se construyó una caja de acrílico transparente (Ver la figura 1) con piso y sin tapa de 50 x 50 cm, dotada de pares de optoacopladores infrarrojos (emisor-receptor) en sus paredes, los cuales permiten ubicar al animal en un plano cartesiano X-Y (nueve coordenadas por eje), y en una coordenada del eje Z, dentro de un espacio de experimentación, con el objeto de tener la información de su comportamiento motor en el tiempo, y almacenarla para su análisis posterior.

Las 90 coordenadas se barren secuencialmente, seleccionando la misma pareja emisor-receptor en los tres ejes por medio de circuitos multiplexores-demultiplexores[9][10]. Para evitar problemas de ruido por luz parásita en el enlace emisor-receptor, la emisión infrarroja es modulada con la frecuencia del oscilador local de un circuito PLL ("phase-locked loop") [11], cuya entrada recibe la señal del receptor [7] correspondiente a través del circuito demultiplexor, para compararla con su propia frecuencia de referencia. Este circuito es reproducido tres veces, uno para cada eje coordenado. Las salidas TTL compatibles de dichos PLL son leídas para determinar la ubicación del animal.

3 Métodos

Para realizar el control de la caja previamente descrita fue necesario desarrollar un circuito de control basado en un microcontrolador, esto debido a que un microcontrolador posee todas las características de un computador completo y puede ser programado para que cumpla una tarea determinada a un muy bajo costo. También fue necesario desarrollar los programas para el mismo y para recuperar la información colectada dentro de una PC. En dicha PC se realiza el análisis de los cambios en conducta por medio de un sistema de programas realizados en un lenguaje de programación gráfica (LabVIEW 4.1 de National Instruments) [15].

Así el trabajo desarrollado puede centrarse en dos grandes secciones: Desarrollo del "hardware" y desarrollo del "software".

3.1 Desarrollo del hardware

Se diseño y construyó un circuito de control para la caja basado en un microcontrolador RISC ("Reduced Instruction Set Computer"). En este proyecto se seleccionó un microcontrolador de 8bits con tecnología "Enhanced Flash" modelo PIC16F877 [14], el cual utiliza cinco puertos digitales: A, B, C, D, E. Dispone de 8Kx14bits de memoria FLASH mejorada, 256 "bytes" de memoria de datos EEPROM, rango amplio de voltaje de funcionamiento de 2,0V a 5,5V, comunicación IIC, dos temporizadores de 8 bits y uno de 16 bits. Se empleó para el circuito de reloj un cristal XT de 4MHz.

Este microcontrolador (Ver la figura 2) envía una secuencia binaria (del 0 al 8) para seleccionar de manera secuencial una de las nueve coordenadas de cada eje, espera 500 ms y lee el estado de las salidas binarias de los circuitos PLL para determinar en qué coordenadas se encuentra el animal. La información de la ubicación del animal es almacenada previa validación en un banco de memorias no volátiles del tipo EEPROM ("Electrical Eraseable Read Only Memory") 24LC65 [15] con una capacidad total de 64,536 "bits".

Este mismo microcontrolador controla una pantalla LCD ("Liquid Cristal Display") [9] alfanumérica, un teclado de tres teclas que permite seleccionar funciones como introducir el tiempo de experimentación, iniciar o abortar un experimento o transmitir los datos colectados a una PC a través del puerto serial RS232 del propio microcontrolador, por medio de un circuito cambiador de nivel MAX232.

También se señala que los sistemas descritos están alimentados por medio de una fuente de alimentación externa (eliminador de baterías de 9V) y cuenta con un banco de baterías NiCd de respaldo para evitar la pérdida del experimento por apagones de corta duración.

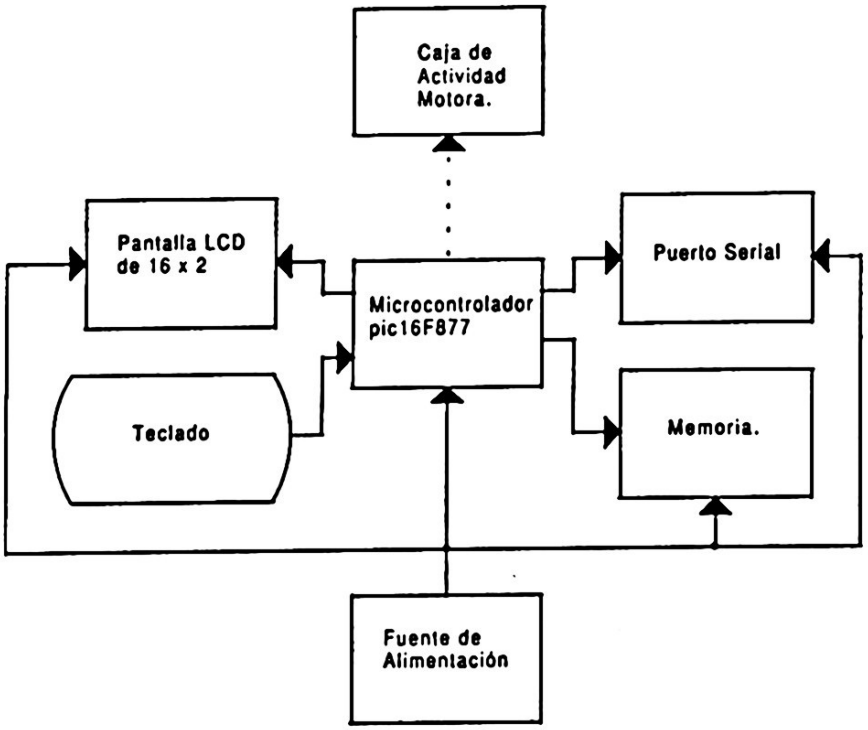


Fig. 2. Diagrama a bloques del circuito desarrollado

3.2 Desarrollo del software

El programa desarrollado para el microcontrolador ejecuta las siguientes tareas:

1) Programa el puerto B como entrada, los puertos A, D y E como salidas, y el puerto C sus "bits" RC0 y RC1 como entradas y el resto como salidas.
2) Inicializa la pantalla LCD para un modo de bus de datos de 8 bits y un desplegado de 5 x7 puntos.
3) Pregunta al usuario si es un nuevo experimento (Ver figura 3), en caso negativo el programa salta a preguntar si transmite los datos colectados por el puerto serial (paso 9). En caso afirmativo:
4) Pregunta el tiempo de duración del experimento en minutos.

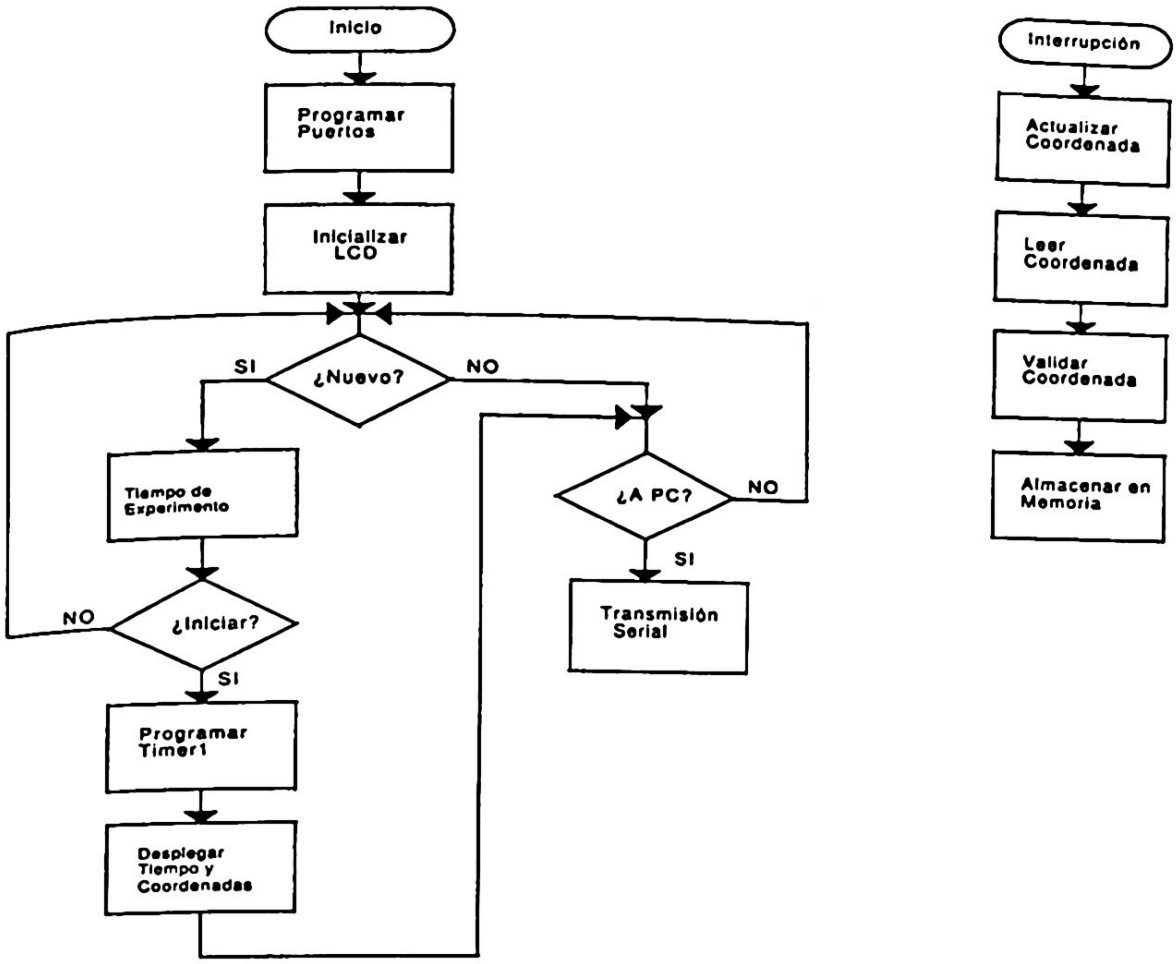


Fig. 3. Diagrama de flujo del programa desarrollado para el microcontrolador

5) Pregunta si inicia el experimento, en caso negativo el programa salta a preguntar si transmite los datos colectados por puerto serial (paso 9). En caso afirmativo: pasos 6. 7. 8.
6) Programa el "Timer1" en modo reloj de tiempo real para generar una interrupción cada 10 ms.
7) El programa permanece en un ciclo desplegando el tiempo transcurrido y las coordenadas detectadas, hasta que el tiempo de experimentación termine o el usuario aborte el experimento por oprimir una tecla ("enter").
8) Durante este ciclo las interrupciones del "Timer1" hacen que el programa cada 30 ms actualice el valor de la coordenada a barrer y leer de manera secuencial. También dentro de este proceso en la interrupción se guardan los datos de tres barridos previos de las nueve coordenadas para compararlos entre si y detectar cuando hay una diferencia entre ellos para así validar un movimiento y almacenarlo en la memoria.
9) Al concluir el experimento o abortarlo el programa pregunta si los datos colectados se transmiten por el puerto serial, en caso negativo el programa salta al paso 3. En caso afirmativo se programa el puerto serial a 2400 bps, 8 "bits2 de datos, no paridad, un "bit" de "stop" y se leen de las primeras dos memorias el total de datos colectados y memorias utilizadas, con esta información el programa empieza a leer los datos colectados, convertirlo a "ASCII" y transmitirlos con el

siguiente formato: tiempo(s), tabulador, coordenada X, tabulador, coordenada Y, tabulador, coordenada Z, "enter". Esto con el fin de que el programa receptor de datos en la PC los almacene en un archivo compatible con el formato que los programas de análisis previamente desarrollados manejan.

10) Concluida la transmisión de todos los datos el programa salta al paso 3.

4 Resultados

La integración de la caja de actividad motora con el módulo del microcontrolador desarrollado se ve de la manera siguiente (Figura 4):

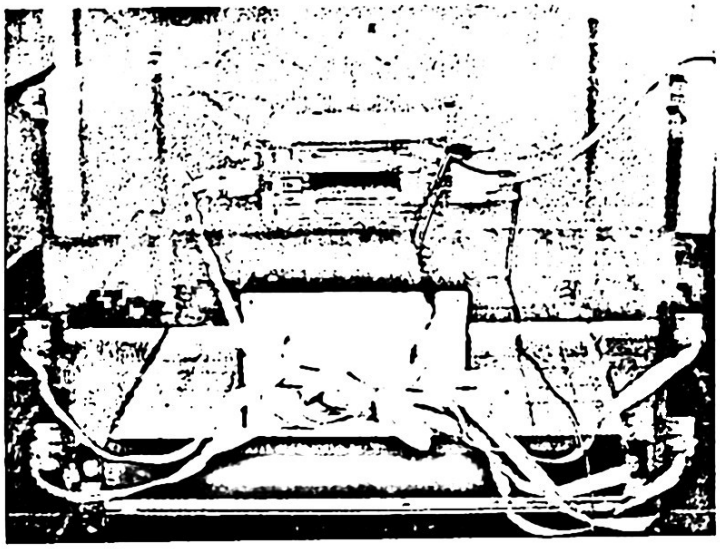


Fig. 4. Caja de Actividad Motora con el Módulo desarrollado incorporado

La caja permite la captura de la actividad motora del animal de manera autónoma y permite transmitir serialmente dicha información a una PC. (Ver tabla 1)

Tabla 1. Ilustración de una sección del archivo de datos (El archivo se genera en ASCII puro, se presenta en formato tabla por razones de legibilidad).

Tiempo	X	Y	Z
108	3	6	1
109	3	7	0
110	2	8	0
111	3	9	0
112	4	9	0
114	5	9	0
115	6	9	0
116	7	9	0
118	7	8	0
119	8	8	0

Esta información se evalúa con los programas: de análisis de velocidad, distancia recorrida, erguimientos y permanencia en determinadas zonas de la caja. (Ver las figuras 5 y 6). El sistema ya ha sido utilizado para estudiar los cambios conductuales esperados por la aplicación de fármacos que intervienen en la modulación de la actividad motora.

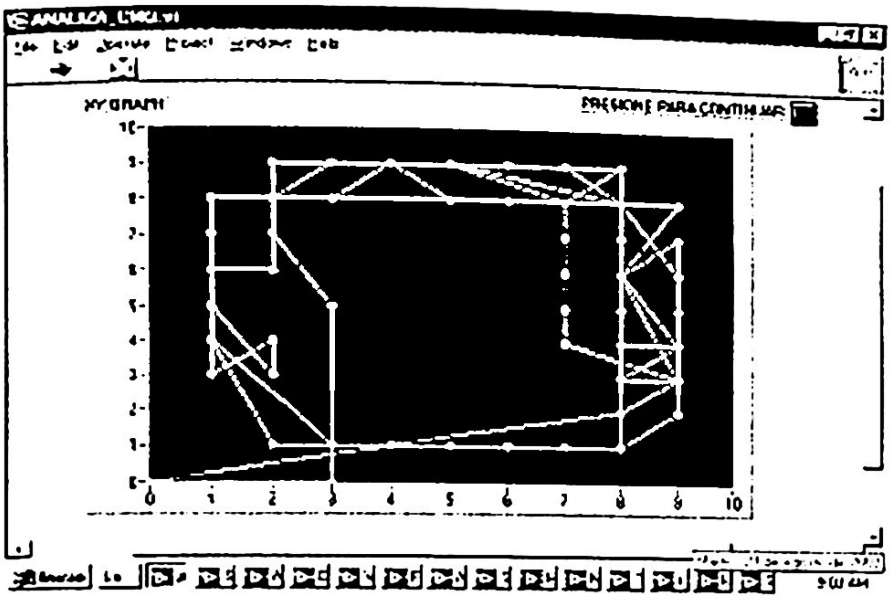


Figura 5. Gráfica que ejemplifica un análisis de conducta motora.

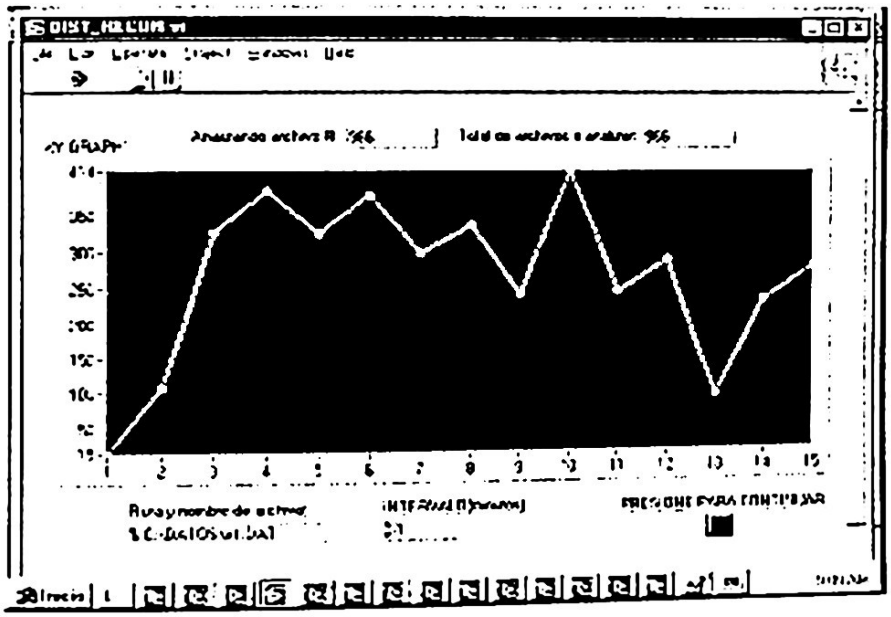


Figura 6. Gráfica que ejemplifica un análisis de distancia recorrida

5 Discusiones

Hasta este momento los datos se recuperan con un programa desarrollado en Basic pero el proceso es ineficiente por el número de pasos que implica, esto hace necesario

migrar este código a Labview para incluir este proceso dentro de los programas de análisis previamente desarrollados.

Referencias

1. Geyer, M.A., Russo, P.V., Masten V.L.: Pharmacol Biochem Behav Vol. 25. (1986) 277-288
2. File, E.S., Baldwin, H.A., Johnston, A.L., Wilks, L..J.: Pharmacol Biochem Beba Vol. 30. (1988) 809-815
3. Brockwell, N.T., Ferguson, D.S., Beninger, R.J.: J Neurosci Meth Vol. 64. (1996) 227-232
4. Golani, I., Benjamin, Y., Eilam, D.: Behavioral Brain Research Vol. 53. (1993) 21-23
5. Schwarting, R.K., Goldenberg, R., Steiner, H., Fornaguera, J., Huston, J.P.: J Neurosci Meth, Vol. 49. (1993) 199-210
6. Sams, F.: J Neurosci Meth Vol. 59. (1995) 157-167
7. Diefenderfer, A.J.: Instrumentación Electrónica (1987) 137
8. AND Display Products Catalog. AND 491 LCD Display (1991) 4-14 4-15
9. CD4051. CMOS Data Book.National Semiconductors Corporation (1989) 1-120 1-125
10. CD4066. CMOS Data Book.National Semiconductors Corporation (1989) 1-126 1-131
11. LM567 Tone Decoder. Special Purpose Linear Devices DataBook. National Semiconductors Corporation (1989) 5-281 5-288
12. Microchip Microcontroller PIC16F877 (2001) 1-218
13. Non-Volatile Memory Products Data Book. 24LC65 Serial EEPROM 3-277 3-288
14. STEREN Catálogo No. 9 Electrónica STEREN S.A de C.V., México D.F. (1998) 99
15. User Manual Lab View Ver. 4.1. National Instruments Corporation (1996)

Algoritmo de cálculo de la carga atendida para redes digitales multiservicio de banda angosta

Julio Carlos Delgado Hernández

Universidad del Valle de México Plantel Lomas Verdes,
Departamento de Tecnociencias
Paseo de las Aves No. 1, Naucalpan, Estado de México, 53220
jcdelgado@uvmnet.edu

Resumen. A fin de elaborar un algoritmo de enrutamiento dinámico para las redes multiservicio de banda angosta basado en el análisis del estado de la red y de cada uno de los arcos de unión entre sus nodos, se vuelve necesario elaborar una metodología dirigida a calcular la capacidad de transmisión existente entre los nodos que conforman la subred de transmisión. En este artículo se describe un algoritmo capaz de calcular la capacidad de transmisión de los arcos de la red. El algoritmo contempla la existencia de diferentes clases de usuarios, y determina la forma del cálculo de las probabilidades de bloqueo para cada una de ellas.

1 Introducción

La tendencia a la digitalización de las redes de comunicación observada en los últimos 35 años, conllevó a la creación de Redes Digitales de Servicios Integrados (RDSI). La integración de servicios que dichas redes presentan está basada en la unificación de principios científicos, técnicos, metodológicos y organizacionales [1].

Las redes digitales de servicios integrados son objetos de telecomunicación de estructura compleja, caracterizados principalmente por su estructura topológica, sus algoritmos de funcionamiento, los parámetros del medio en el cual desarrollan su funcionamiento y por las respuestas que ofrecen ante estímulos exteriores. El principal problema a resolver en el momento de su diseño lo constituye la evaluación de la efectividad de su funcionamiento y una implementación racional de la misma [2].

2 Método probabilístico

La magnitud de carga atendida por un nodo de conmutación puede ser determinada de diferentes formas, una de las cuales se basa en la distribución matemática del número total de canales ocupados.

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 333–337, 2003. © Centro de Investigación en Computación, IPN, México.

Con éste fin, es posible determinar la distribución matemática del número de canales ocupados a un mismo tiempo en un sistema ($[x]$), haciendo uso de la siguiente expresión [7]:

$$[x]=\frac{\frac{1}{x!}\prod_{j=1}^{x}\mu(j-1)\sum_{i=1}^{u}A_i}{\sum_{k=0}^{V}\frac{1}{k!}\prod_{j=1}^{k}\mu(j-1)\sum_{i=1}^{u}A_i}. \tag{1}$$

donde A_i simboliza la carga atendida por cada uno de los canales de salida del nodo i, y k el número total de canales con que cuenta el nodo en observación. Otro método de cálculo de la carga total atendida por un grupo de canales se basa en el uso del concepto de esperanza matemática, como se desprende de la siguiente expresión [3]:

$$Y=[x]\cdot x=\sum_{x=0}^{V}\frac{\frac{1}{(x-1)!}\prod_{j=1}^{x}\mu(j-1)\sum_{i=1}^{u}A_i}{\sum_{k=0}^{V}\frac{1}{k!}\prod_{j=1}^{k}\mu(j-1)\sum_{i=1}^{u}A_i}. \tag{2}$$

Si, paralelamente, se introduce un mecanismo de reserva de canales para ciertas clases de abonados en los arcos de cada uno de los nodos que conforman la red [5], con miras a alcanzar niveles predeterminados de calidad de servicio, entonces en la fórmula (1) deberán de ser introducidos cambios en el tipo de distribución de probabilidad de ocupación de canales. De acuerdo a modelos conocidos de reserva de canal, tenemos que

$$[x]=\frac{\frac{1}{x!}\prod_{j=1}^{x}\mu(j-1)\sum_{i=1}^{u}A_i\tau(T_i-x)}{\sum_{k=0}^{V}\frac{1}{k!}\prod_{j=1}^{k}\mu(j-1)\sum_{i=1}^{u}A_i\tau(T_i-k)},\quad x=\overline{0,V}, \tag{3}$$

donde

$$\tau(c)=\begin{cases}0, & si \quad c<0\\ 1, & si \quad c\geq 0\end{cases}. \tag{4}$$

Para un tipo de conexión del módulo de conmutación sin restricciones, tenemos que

$$\mu(i)=1,\quad i=\overline{0,V}.$$

3 Método determinístico

Otra forma posible de determinar la magnitud de la carga atendida consiste en expresarle como una función de vectores de probabilidades de pérdida, cargas de entrada del sistema y coeficientes de tipos de abonado determinados de antemano. Esta forma de representación de la carga atendida por un sistema es relativamente sencilla de determinar si se representa como una intensidad de carga convertida a bits para cada tipo de abonado.

La magnitud de carga por abonado de esta manera expresada se define por medio de la siguiente expresión [4]:

$$y_i = Mark_i \cdot A_i \cdot \frac{(1 - P_i)}{V}, \tag{5}$$

donde $Mark_i$ representa un coeficiente de uso de canal para cada tipo de abonado, y P_i representa la probabilidad de pérdida para el abonado i, definida de acuerdo con la siguiente expresión:

$$P_i = \sum_{x=V-Mark_i+1}^{V} [x]_i, \quad i = \overline{1,u}. \tag{6}$$

Así, es posible determinar la magnitud total de la carga atendida total haciendo uso de la expresión siguiente:

$$Y = \sum_{i=1}^{u} y_i. \tag{7}$$

Por otro lado, la velocidad de transmisión de bits resultante en el grupo de canales de salida puede ser representada por medio de una variable cualquiera C, y la velocidad de transmisión utilizada por cada tipo de usuario por medio de la variable C_i. Podemos entonces expresar tenemos que

$$C_i = Mark_i \cdot C. \tag{8}$$

Consideración especial merece el hecho de que la velocidad de transmisión de bits resultante puede interpretarse como la velocidad de transmisión entre dos nodos de conmutación de la subred.

A fin de convertir la intensidad de carga atendida a intensidad de carga entrante expresada en bits para cada tipo de abonado de la red, se deberá hacer uso de la expresión siguiente:

$$\alpha_i = \frac{C_i}{C} \cdot A_i, \quad i = \overline{1,u}. \tag{9}$$

Al sustituir (9) en (5), puede replantearse la ecuación de la carga normalizada para cada tipo de abonado, expresada en bits, como sigue:

$$\rho_i = \frac{C_i}{C} \cdot A_i \cdot \frac{(1 - P_i)}{V} . \qquad (10)$$

Es obvio que esta variable tendrá como condición que

$$0 \leq \rho 1_i \leq 1 . \qquad (11)$$

De manera semejante se determina la carga atendida normalizada total en el arco de transmisión, de acuerdo a la siguiente expresión:

$$\rho = \sum_{i=1}^{u} \rho_i . \qquad (12)$$

Las relaciones (8 – 12) permiten evaluar la capacidad de transmisión restante del recurso de canal (V) en cada arco de transmisión según la siguiente expresión:

$$C_r = V \cdot (1 - \rho) . \qquad (13)$$

En caso de precisarse la introducción de un mecanismo de reserva de canales para ciertas clases de usuarios, la expresión (6), que representa la probabilidad de pérdida para el abonado i tomará la forma siguiente:

$$P_i = \sum_{x=T_i - Mark_{i+1}}^{V} [x], \qquad (14)$$

donde la variable T_i representa los niveles de reserva adoptados para cada tipo de abonado que así lo requiera.

El algoritmo aquí desarrollado permite calcular la capacidad de transmisión existente entre los nodos de una subred de transmisión basado en su conversión a velocidad de transmisión de bits, lo que permite eliminar de sus fórmulas de determinación complejas fórmulas matemáticas, sin afectar el resultado final de los cálculos.

Este algoritmo, que se basa en el manejo de coeficientes predefinidos de participación por tipo de usuario en la carga total que ingresa al sistema, permite además contemplar la reserva de canales para ciertas clases de usuarios, en dependencia de su prioridad en el sistema, y elimina la división de la carga total de manera homogénea entre todos los canales del sistema, a diferencia del primer algoritmo detallado, lo que le acerca aún mas al funcionamiento de las redes reales.

Referencias

1. Gvozdenko A.A., Osipov G.B. Redes Digitales de Servicios Integrados. En Radioelectrónica extranjera (en ruso), No.9, Vol. XXXIV, Moscú, (1989) 23 – 29.
2. Delgado J. C. «Valuación de la calidad de servicio en los nodos de las redes multiservicio». En: Cuadernos de Investigación del Instituto de Investigaciones Científicas de la UTCIM, Moscú, (2001) 2 – 8.
3. Ershov V. A., Kuznetzov N. A. «Fundamentos teóricos de la construcción de Redes Digitales de Servicios Integrados (ISDN)» (en ruso). Instituto de problemas de transmisión de la información, Academia de Ciencias Rusa, Moscú (1995).
4. Delgado J. C. «Investigación y desarrollo de un método de enrutamiento para las RDSI de banda angosta» (en ruso). Disertación de grado doctoral, Universidad Técnica de Comunicaciones e Informática de Moscú, Moscú (2002).
5. Ershov B. A., Ershov D. B. «Gestión del recurso de canal de las RDSI con reservación de canales». En: Electrocomunicaciones No. 12, Vol. XLII, Moscú (1994) 11 – 17.
6. Weber A., Fischer W., Huber M. N. Multichannel Circuit Switching – Performance Evaluation of Switching Netwoks. IEEE Journal on selected areas in communications. Vol. 9, N 2, Feb. 1991
7. Pinski E.A. Simple approximation for the Erlang loss function. Performance evaluation, No. 5, 1990, pp. 131- 136.

Robot Command Center (Roc2)

Jesús Savage [1], Emmanuel Hernández [2], Gabriel Vázquez [3]

Laboratorio de Interfaces Inteligentes, Universidad Nacional Autónoma de México
[1] savage@servidor.unam.mx, [2] deyael@mezcal.fi-b.unam.mx,
[3] bowser@mezcal.fi-b.unam.mx

Resumen. En este documento se describe un sistema que permite simular y controlar robots móviles virtuales. El sistema consiste en una interfaz 3D que permite controlar y probar múltiples robots usando un lenguaje de programación creado para este propósito. Dicho lenguaje incorpora comandos de movimiento, lectura de sensores, instrucciones de flujo de control de programa, entre otros.

Introducción

El proceso de planear y construir un robot móvil suele ser lento y costoso. Además, este proceso está sujeto a errores tanto de diseño como de construcción, los cuales, en varias ocasiones, se deben a circunstancias externas como por ejemplo, los materiales defectuosos.

Una vez terminada la construcción del robot, éste debe someterse a una etapa de pruebas con el propósito de verificar su correcto funcionamiento. Pasada la etapa de pruebas, el robot puede ser utilizado para el propósito que fue creado.

Ya que se cuenta con el robot adecuado, es necesario programarlo con ciertas instrucciones para que ejecute las tareas que le fueron delegadas. En general, la estructura de cada robot es diferente por lo que la portabilidad de programas entre robots es mínima; de manera que si se desea que dos robots diferentes ejecuten el mismo programa, será necesario escribir dicho programa para cada configuración de robot en su propio lenguaje de programación. Note que esta fase de programación puede ser tardada y estar sujeta a errores.

Por consiguiente, el Robot Command Center tiene como objetivo presentar una alternativa a la etapa de desarrollo y prueba de algoritmos para diferentes configuraciones de robots móviles. El Robot Command Center consiste en una interfaz gráfica, que permite al usuario simular varios robots móviles virtuales dentro de un espacio tridimensional. Los robots móviles virtuales cuentan con diversos tipos de sensores (contacto, infrarrojos, reflectivos y sonares) que les permiten percibir información referente al mundo virtual en donde se encuentran. Adicionalmente, los robots son controlados utilizando un lenguaje de programación de alto nivel, lo que permite desarrollar programas complicados de manera rápida y sencilla. Es importante

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 338–340, 2003. © Centro de Investigación en Computación, IPN, México.

mencionar que este lenguaje de programación puede utilizarse tanto con robots virtuales como con robots reales, ya que es independiente de la plataforma.

2 Características del Robot Command Center (Roc2)

El Robot Command Center le permite al usuario interactuar con robots móviles virtuales, para ello cuenta con diferentes maneras de especificar instrucciones a un robot, y emplea diferentes puntos de vista con gráficas en 3D para visualizar los resultados de las operaciones.

La siguiente es una lista de características principales que presenta el Robot Command Center:

- Utiliza tres puertos de vista con gráficos 3D para visualizar los resultados obtenidos.
- Es posible navegar libremente el ambiente de simulación.
- Incorpora una consola de comandos.
- El usuario tiene control sobre uno o más robots móviles virtuales.
- Es posible utilizar distintos modelos y configuraciones de robots.
- Los robots cuentan con un brazo mecánico con dos grados de libertad, que les permite tomar ciertos objetos dentro de la simulación.
- Los robots cuentan con una cámara para capturar imágenes de su entorno. Estas imágenes se almacenan localmente y pueden ser transmitidas por Internet.
- El usuario tiene la capacidad de cambiar el ambiente de simulación cuando lo desee.
- Es compatible con el control de juegos del Super Nintendo (SNES joypad).
- Permite entradas de comandos vía remota (sockets) o local (consola de comandos).
- Tiene soporte Text-To-Speech.
- Incorpora un sistema experto capaz de planear las trayectorias de un robot. El sistema experto le permite evadir obstáculos al robot mediante el uso de campos potenciales.
- Es posible modificar los métodos de lectura de sensores y de movimientos de los robots usando librerías dinámicas (DLLs); de esta manera el usuario puede crear nuevos sensores y tipos de movimiento.
- Es compatible con Robel.

3 Características de Robel (Robot Behaviour Language)

Robel es un lenguaje de programación orientado al control de robots móviles. Desde un principio se planeó como una herramienta para llevar a cabo el control de forma rápida y sencilla haciendo uso de instrucciones de alto nivel. Está dirigido al control

de movimientos en robots, sin embargo, la posibilidad de utilizar subrutinas y sensores permite modelar comportamientos más complicados.

Las características principales de Robel son las siguientes:

- Incorpora instrucciones para el control de movimiento.
- Incorpora instrucciones para la lectura del estado lógico del robot (lectura de sensores).
- Incorpora instrucciones para la comunicación vía Internet.
- Control de flujo de programa haciendo uso de ciclos while, if y saltos a etiquetas.
- Soporta los siguientes tipos de variables: enteros y flotantes.
- Es independiente de la plataforma[1].
- Recursividad.
- Puede invocar otras subrutinas creadas por el usuario. Estas subrutinas pueden ser scripts (archivos de texto con instrucciones de Robel), o bien, subrutinas programadas utilizando librerías dinámicas (DLLs).

[1] Para que un programa escrito en Robel se pueda interpretar y ejecutar sobre casi cualquier plataforma o configuración de robot real, requiere de una máquina virtual. La máquina virtual traduce cada instrucción de Robel al lenguaje propio de cada robot. Desarrollar una máquina virtual para un diseño de robot en específico puede limitarse a modificar las rutinas de comunicación y movimiento del robot real; sin embargo, esto no garantiza que todo el conjunto de instrucciones pueda ser ejecutado. Actualmente, existen máquinas virtuales para 3 plataformas: un robot Real World Interface, mini-robots basados en el microcontrolador HC11 y los robots virtuales del Roc2.

Prototipo de un sistema multiagentes para el comando oral de un robot móvil

Oscar Pérez Márquez [1], Luis Villaseñor Pineda [2]

[1] Benemérita Universidad Autónoma de Puebla
Av. San Claudio y 14 Sur
Puebla, México.
osmarp79@hotmail.com

[2] Laboratorio de Tecnologías del Lenguaje
Instituto Nacional de Astrofísica, Óptica y Electrónica
Calle Luis Enrique Erro #1.
Sta. María Tonanzintla, Pue., México
villasen@inaoep.mx

Resumen. El prototipo desarrollado es un sistema de software para el comando oral de un robot móvil. Este sistema está basado en una arquitectura multiagentes, consta de tres procesos principales: el receptor de la orden, el intérprete de la orden, el controlador de movimientos. El diagrama siguiente muestra nuestro sistema.

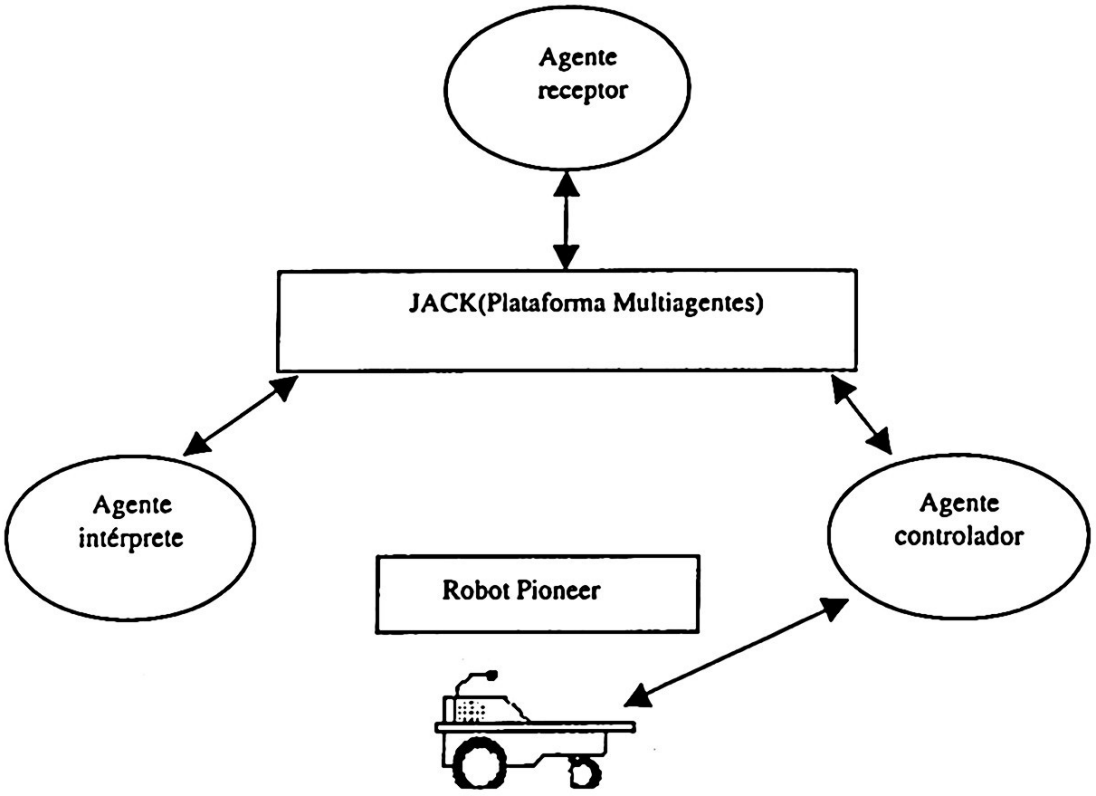
1 Introducción

El presente trabajo presenta el diseño y los resultados de la implementación de una arquitectura multiagentes para el comando oral de un robot móvil. La idea central fue la búsqueda de una plataforma flexible capaz de integrar todos los elementos posibles para poder brindar una solución satisfactoria. En [1] encontrará el lector una descripción teórica detallada de este trabajo.

2 Características y funcionamiento del sistema

Cada uno de los tres procesos que constituyen el sistema es modelado a través de agentes por lo que las funciones de cada proceso van a llevarse a cabo por un agente. Para ello se utilizó la plataforma multiagentes JACK. Esta plataforma brinda las operaciones básicas de un ambiente orientado a agentes: creación de agentes, manejo de directorio de servicios, paso de mensajes a través de la red. JACK esta completamente integrado con el lenguaje de programación JAVA.

A. Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 341–343, 2003. © Centro de Investigación en Computación, IPN, México.



La función del agente receptor es capturar la frase que pronuncio el usuario y enviarla al agente intérprete. Para el reconocimiento de voz se utilizó un sistema de reconocimiento de habla comercial (Dragon Natural Speaking), el cual a partir de una señal acústica entrega el texto correspondiente.

El agente intérprete se encarga de transformar el texto de la orden, expresado en lenguaje natural, a comandos expresados en el lenguaje de control del robot. El intérprete está programado en Prolog y utiliza el formalismo de DCGs (Definite Clause Grammars) para representar las estructuras gramaticales manejadas. Al terminar la transformación de la orden, si se trata de una orden válida, se entregará el comando correspondiente al controlador de movimientos.

El controlador de movimientos se encarga de ejecutar la orden que recibe del intérprete. El agente controlador es un agente en JAVA que encapsula el software de comando del robot Pioneer el cual está programado en C. Esto se hizo a través de métodos nativos. Un método nativo es un método de JAVA cuya implementación está escrita en otro lenguaje de programación. Los métodos nativos se integran a Java utilizando JNI (Java Native Interface). JNI es una interfaz de programación nativa y permite que código de JAVA que corre dentro de la máquina virtual, interopere con aplicaciones y librerías escritas en otro lenguaje de programación, tales como C, C++ ensamblador.

Los agentes utilizados en este sistema pueden ejecutarse dentro de una misma máquina o en máquinas distintas, el máximo número de máquinas que se pueden utilizar para la ejecución de nuestro sistema debe ser el mismo que el número de agentes utilizados.

Por el momento, las órdenes que puede entender el robot son simples y básicamente se realizan acciones de desplazamiento, rotación y modificación de velocidad del robot. Las órdenes que puede entender el robot las clasificamos en los tres tipos siguientes:

Ordenes simples:
Ejemplos: camina, avanza, retrocede, alto, camina más rápido, vete un poco más despacio.

Ordenes compuestas:
Ejemplos: camina dos metros hacia adelante, camina rápido 4 metros hacia delante, camina hacia atrás, gira a la derecha 90 grados, gira 180 grados hacia la derecha, camina hacia la izquierda. Es también posible construir frases donde la frase verbal se da por sobreentendida (elipsis), por ejemplo: hacia adelante un metro, 5 metros hacia delante, a la derecha 90 grados, 90 grados a la izquierda.

Ordenes para la manipulación de la cámara:
Ejemplos: mira hacia la derecha, mira hacia la izquierda, mira hacia el frente.

Referencias

1. Perez Marquez, Oscar y Luis Villaseñor Pineda, *Sistema Multiagentes para el Comando Oral de un Robot Móvil* en este mismo libro.

Visualización y control de una computadora desde varios equipos: Enseñanza virtual

Juan Carlos May Pat, Alejandro de Jesús Ortiz Cervera

Facultad de Matemáticas
Universidad Autónoma de Yucatán.
ocervera@yahoo.com

Resumen. El objetivo de este proyecto es proporcionar una herramienta a los profesores que utilizando una red local permita ejecutar una aplicación, para controlar y visualizar otras computadoras utilizadas por alumnos. Esta aplicación utiliza funciones Hook, algunas A.P.I.´s de Windows y fue desarrollada en el lenguaje Builder C++ 5.0, utilizando algunos componentes VCL. Se enfoca para sistema operativo Microsoft Windows 9.x en adelante.

Introducción

En muchas instituciones los equipos de cómputo son utilizados como herramienta de enseñanza o en su caso como apoyo a materias que los profesores imparten. Una vez que el profesor termina sus clases o en el transcurso de la clase, los alumnos pueden cometer uno de los siguientes errores, que puede implicar la presencia del profesor:

Activar un botón de forma accidental con la acción de desencadenar un evento inesperado
Cerrar accidentalmente la aplicación
Activar otro programa accidentalmente.
Alguna acción inesperada del programa como instalar un componente del programa.

Ante esta situación se sugiere la creación de un programa que permita:

Visualizar el contenido del escritorio de la computadora del alumno en la computadora del profesor.
Poder controlar la computadora del alumno, para que éste pueda recibir la atención necesitada
Que el profesor pueda administrar el control de los equipos de los alumnos.

Este sistema de asistencia virtual se encuentra formado por 2 aplicaciones, una para el cliente (computadora de los alumnos) y otra para el servidor (computadora del profesor).

Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 344–345, 2003. © Centro de Investigación en Computación, IPN, México.

De esta forma un profesor puede tener el control de los equipos de los alumnos, visualizar lo que en este momento se encuentren realizando y los errores que puedan cometer.

2 Aplicación Cliente

La función de las aplicaciones clientes consiste en enviar información a la aplicación servidor y recibir órdenes desde la aplicación servidor simulando el uso del usuario (movimiento del puntero del mouse y teclas presionadas).

La información que se envía al servidor consiste en la imagen del escritorio y los cambios que en él ocurren.

3 Aplicación Servidor

Esta aplicación será instalada en la computadora del maestro que permitirá visualizar y en su momento controlar, el software que el alumno en este momento se encuentre aprendiendo, al igual que el contenido del escritorio.

De esta forma el alumno podrá aprender en línea virtual las instrucciones adicionales que el maestro considere necesario para su aprendizaje.

4 Comentarios

El programa se encuentra desarrollado en el lenguaje Builder C++ 5.0 para Windows, el proyecto puede ser utilizado como base para la creación de otras aplicaciones que se enfoque en visualizar el contenido de uno o varios equipos de cómputo al mismo tiempo.

Utiliza los conceptos de hook, capturando mensajes del sistema, funciones A.P.I. para la modificación de las imágenes y eventos de teclado-ratón y los componentes VCL para el despliegue de las imágenes.

Programa de control en lenguaje ensamblador para el módulo de control de una caja de actividad motora con tecnología de microcontroladores para la captura y análisis de datos de ratas en experimentación

Francisco J. Heredia López [1], Karen Z. Navarrete Kao [2]

[1] Universidad Autónoma de Yucatán, Centro de Investigaciones Regionales "Dr. Hideyo Noguchi". Laboratorio de Neurofisiología.
hlopez@tunku.uady.mx

[2] Universidad Autónoma de Yucatán, Facultad de Matemáticas.
karnava@todito.com

Resumen. Este artículo describe el programa en ensamblador del microcontrolador "RISC" que controla una caja para registrar la actividad motora en ratas.

Abstract. This paper describes the software developed in assembler for the RISC microcontroller that controls a box to register the motor activity in rats.

1 Introducción

Con el objeto de controlar el barrido de las coordenadas de la caja de actividad motora y la adquisición y almacenaje en memorias de las coordenadas donde la rata se desplaza durante los experimentos, se escribió un programa en el lenguaje ensamblador del microcontrolador PIC16F877 de Microchip [1].

Este programa instruye al microcontrolador para barrer las coordenadas del eje cartesiano de la caja, detectar y validar los movimientos del animal, almacenar dichos movimientos en memorias seriales no volátiles tipo EEPROM [2], desplegar dicha información en una pantalla alfanumérica LCD [3], contabilizar el tiempo que el usuario indique por medio de un teclado para la duración del experimento y enviar dicha información a solicitud del usuario a una PC por medio de comunicación serial.

Métodos

El algoritmo correspondiente se explica en la sección 3.2 y la figura 3 del artículo «Diseño y construcción de una caja de actividad motora con tecnología de microcontroladores para la captura y análisis de datos de ratas en experimentación» de los mismos autores en este mismo libro.

Gelbukh, M. Hernández Cruz (Eds.) Avances en la ciencia de la computación en México, CORE-2003, pp. 346–347, 2003. © Centro de Investigación en Computación, IPN, México.

3 Resultados

El software desarrollado cumple con el objetivo de instruir al microcontrolador RISC para que éste realice las tareas de monitorear y validar los movimientos del animal y almacenar dicha información en memorias no volátiles. También permite al usuario programar la duración del experimento y poder enviar la información almacenada a una computadora PC para su análisis.

4 Discusiones

Con este software se puede lograr que la caja vigile la conducta motora de ratas en experimentación sin necesidad de dedicar una computadora PC a esta tarea, lo cual abre la opción de tener más de una caja trabajando de manera simultanea (por el bajo costo del sistema).

Esto permitiría que se realicen varios experimentos de manera simultanea, lo cual ayudaría a que el trabajo de investigación en neurofarmacología sea más eficiente y expedito.

La relevancia de este punto radica en el hecho que se requiere de bastante tiempo y recursos para preparar un solo experimento (el crecer a los animales, habituarlos, mantenerlos alimentados y en las condiciones luz-oscuridad, temperatura y humedad requeridos, los procesos quirúrgicos para administrar los fármacos, la recuperación postoperatoria, y el experimento mismo). Así con varias cajas se podrían obtener más y mejores resultados a menor costo, ya que los procesos mencionados se pueden hacer para varios animales de manera simultanea.

También el hecho que la caja posea su propio sistema de control permite que pueda ser enlazada fácilmente a un sistema más complejo, donde no sólo se capture la conducta motora sino también actividad eléctrica y conducta estereotipada lo cual proporcionaría a los fisiólogos una cantidad mayor de información y con un panorama más completo de cómo los grupos neuronales en estudio participan y ejercen sus funciones para controlar todas las actividades motoras en estudio.

En resumen, el haber desarrollado este software para el nuevo hardware de control permitió tener una caja de actividad motora auto controlada que es fácilmente actualizable y enlazable a otros sistemas, y que para su aplicación en fisiología proporciona una herramienta de muy bajo costo y alta versatilidad.

Referencias

1. Microchip Microcontroller PIC16F877 (2001) 1-218.
2. AND Display Products Catalog. AND 491 LCD Display (1991) 4-14 4-15.
3. Non-Volatile Memory Products Data Book. 24LC65 Serial EEPROM 3-277 3-288.

Índice de autores

Altamirano Robles, Leopoldo 209, 220
Angles Domínguez, Manuel Iván 44
Arellano, Gustavo 276
Argotte, Liliana 266
Arias Estrada, Miguel 198

Balladares Ocaña, Leandro 78, 89
Bautista Thompson, E. 171, 181

Camarillo, Edmundo 89
Cano, Francisco 89
Carlos May Pat, Juan 344
Castro, Lizette P. 193
Cortés Poza, Yuriria 14
Cruz Enriquez, Janeth 209
Cuéllar Mejía, Guillermo Adolfo 21
Chim, Manuel 78

Delgado Hernández, Julio Carlos 333
Domínguez Ramírez, Omar A. 294

Espinosa, Humberto 1
Estrada González, Fermín Pascual 311

Figueroa Nazuno, J. 171, 181
Frausto Solis, Juan 68
Fuentes Covarrubias, Andrés Gerardo 311
Fuentes Covarrubias, Ricardo 311

Galavíz Carmona, Plutarco 283
Galicia-Haro, Sofía N. 159
Germán Soto, Ernesto 107
González Cortés, Dorian Omar 255
Guerra Vásquez, J. Juan 118
Guzmán Ramírez, E. 181

Heredia López, Francisco J. 325, 346
Hernández, Emmanuel 1, 338

Jácome, Norma 266
Juárez, Fernando 89

Li, Xiaoou 56

Manjarrez Sánchez, J. 135
Márquez, Edna 1
Martínez Muñoz, Jorge 118
Mata, Miguel F. 244
Medina Marín, Joselito 56
Mejía, Manuel 266
Menéndez Domínguez, Víctor 232
Montero Valverde, José A. 32
Muñoz Arteaga, Jaime 283

Navarrete Kao, Karen Z. 325, 346
Noguerón, Claudia 78
Nuño Maganda, Marco Aurelio 198

Oktaba, Hanna 276
Olivas Zazueta , Omar 107
Ortega, Servando 89
Ortiz Cervera, Alejandro de Jesús 344

Parra Vega, Vicente 294
Pérez Márquez, Oscar 126, 341
Prado de la Paz, Olga Leticia 68
Pruneda Martínez, Arturo 255

Quiané Ruiz, J. 135

Rivera Lopez, Rafael 68
Rivera Mancera, V. 171

Santiago Díaz, Carmen 283
Savage, Jesús 1, 338
Soancatl Aguilar, Venustiano 220
Sosa Tzec, Omar 232

Terashima Marín, Hugo 44
Torres Castañeda, Jesús Eduardo 255
Torres González, Areli 149
Torres, L. A. 193
Torres, Miguel 244
Tovilla, Juan Carlos 78

Vargas Villamil, Luis 232
Vázquez, Gabriel 1, 338
Velásquez Elenes, Rigoberto 107
Villaseñor Pineda, Luis 126, 149, 341

Zaprianov Doychev, Petar 255
Zárate Silva, Víctor Hugo 32

Avances en la ciencia de la computación en México
se terminó de imprimir en mayo de 2003
en los talleres gráficos de IMPRESOS RIVERO,
Rumania 1363, San Simón Ticumac,
México, D.F., C.P. 03660
Su tiraje fue de 500 ejemplares

Computación en México, desde los calendarios milenarios, desde la primera computadora hace 45 años, desde el primer Centro de Investigación en Computación hace 6 años. Los jóvenes mexicanos, continuando con la gran tradición, presentamos al mundo nuestras obras, con pasión y amor.

ISBN 970-36-0026-3

9 789703 600267