Research on Computing Science

Recientes avances en la ciencia de la computación en México

Prólogo de Juan Luis Díaz de León Santiago

Alexander Gelbukh
Grigori Sidorov
Wilbert A. Olán Cristobal
José Ángel Vera Félix
(Eds.)

Instituto Politécnico Nacional
Centro de Investigación en Computación

Recientes avances en la ciencia de la computación en México

Research on computing science

Editorial Board
Hernández Lara, Luis
Torres Frausto, Raquel
Silva Millán, Miguel Ranferi
Ayala Aguilar, Juan Antonio

Cover Design
Macario Hernández Cruz

Volume 7

Recientes avances en la ciencia de la computación en México

Con Prólogo de Juan Luis Díaz de León Santiago

Alexander Gelbukh
Grigori Sidorov
Wilber A. Olán Cristóbal
J. Angel Vera Félix
(Eds.)

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2004

ISBN: 970-36-0149-9

Copyright © by Instituto Politécnico Nacional, 2004

Editores

Alexander Gelbukh, Grigori Sidorov,
Wilbert A. Olán Cristóbal, José Angel Vera Félix

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Col. Zacatenco, CP. 07738, México, D.F., México.
E-mail:gelbukh@cic.ipn.mx, sidorov@cic.ipn.mx,
wolan@sagitario.cic.ipn.mx, javera@sagitario.cic.ipn.mx
http://www.gelbukh.com

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

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

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

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

Printing:500
Printed in Mexico

Prólogo

La computación juega un papel único en el mundo contemporáneo, cambiando profundamente la manera de ser del hombre, su manera de generar e intercambiar el conocimiento, su ambiente de trabajo y de vida cotidiana, así como su forma de convivir con otros seres humanos, creando lo que se llama la sociedad de la información. En los pocos años de su existencia la computación ha revolucionado las áreas más importantes de la ciencia e ingeniería –física, química, biología, medicina, comunicaciones, ingeniería civil... El enumerar las diversas áreas de aplicación de la computación equivaldría a enumerar todas las actividades humanas, todas las ramas de la ciencia, ingeniería, economía, política, cultura y educación.

En México, el papel especial de la computación se entiende completamente en todos los niveles políticos, desde el ambicioso programa de modernización computacional del país llamado e-México hasta el apoyo gubernamental al desarrollo e implantación de nuevo software y la educación técnica en la computación y las áreas afines.

El Instituto Politécnico Nacional está a la vanguardia de este esfuerzo nacional e internacional, tanto en la generación de nuevo conocimiento y nuevas aplicaciones de la computación como en la preparación de los recursos humanos de la más alta calidad en esta rama de ciencia e ingeniería y en la creación y el uso de los sistemas basados en las tecnologías computacionales para la modernización de su propio modelo educativo.

Esta importante actividad de nuestro Instituto se realiza en primer lugar a través de su moderno Centro de Investigación en Computación, con sus amplios enlaces nacionales e internacionales, con sus numerosos proyectos de investigación y desarrollo a escala nacional e internacional, con sus grandes congresos internacionales que abarcan todas las áreas más importantes de la computación, con más de mil artículos científicos publicados por sus investigadores y estudiantes en los siete años de su existencia, con varios miembros de la Academia Mexicana de Ciencias e Investigadores Nacionales de los más altos niveles entre sus profesores.

Esta obra continúa la serie de libros sobre la computación preparados por los especialistas de este Centro. Su profundo dominio del tema ha garantizado la más alta calidad de los artículos científicos incluidos en este libro, para el máximo provecho del lector. Me enorgullezco de los jóvenes mexicanos de toda la República que han demostrado su creatividad, dedicación, capacidad de investigación y madurez evidenciada por los trabajos presentados en este libro. Veo que el futuro de la computación en nuestro país está en manos fuertes y confiables.

El libro presenta al lector los avances en las áreas tan importantes de la computación como el procesamiento de imágenes, sistemas de información, ingeniería de software, Internet, procesamiento de lenguaje natural y análisis de textos, robótica móvil y el hardware de las computadoras. El libro será de gran utilidad para los estudiantes y los profesionales en las áreas afines a la computación, así como para los especialistas en otras áreas, los gerentes, empresarios y los promotores de la educación que buscan nuevas formas de modernizar sus actividades a través de las aplicaciones y herramientas computacionales.

No me queda nada más que felicitar a los editores de esta obra por un buen trabajo, a sus autores por los importantes y palpables avances y logros de desarrollo e investigación que se dan a conocer a la sociedad en este libro, y sobre todo felicitar al lector por tener la oportunidad de conocerlos, de enriquecerse con la experiencia y conocimiento de los autores y de sentir la inspiración, entusiasmo y la pasión a la ciencia del México joven.

Dr. Juan Luis Días de León Santiago
Director del Centro de Investigación en Computación, IPN

Prefacio

Todo libro debe presentar un porqué de su existencia, un objetivo. Si quisiéramos identificar el objetivo de este texto podríamos recordar que todo esfuerzo humano que genera nuevos conocimientos e ideas de interés sólo vale la pena si estos frutos son presentados al mundo, acción que conlleva beneficios tanto al público lector como a los autores que presentan sus resultados, teniendo en cuenta que estos beneficios surgen sólo del análisis y discusión inteligentes del conocimiento presentado. Así, "Recientes avances en la ciencia de la computación en México" –un libro de investigación– fue diseñado para ser un punto de difusión del conocimiento generado por nuestros jóvenes investigadores mexicanos en el campo de la ciencia de la computación, así como también, este libro cuenta con capítulos de algunos investigadores reconocidos que fueron especialmente invitados.

Cumpliendo lo anterior, el libro presenta al lector un panorama de las investigaciones llevadas a cabo en México en el campo de la computación, teniendo en cuenta que este material es solo una pequeña porción del inmenso trabajo que nuestros investigadores realizan día a día.

Los capítulos del libro fueron cuidadosamente seleccionados con la ayuda de un comité editorial internacional, cada capítulo fue revisado al menos por tres miembros del comité editorial, a quienes expresamos nuestro más profundo agradecimiento.

La presente obra está conformada por 19 capítulos, los cuales fueron clasificados dentro de diferentes áreas de investigación, a saber:

- Procesamiento de imágenes,
- Sistemas de información y ingeniería de software,
- Internet,
- Procesamiento de lenguaje natural y análisis de textos,
- Robótica móvil,
- Hardware.

Estamos seguros que el lector encontrará que el trabajo de cada uno de los jóvenes investigadores es por demás excelente, haciendo una muy merecida mención a sus reconocidos tutores. Aunando a lo anterior los trabajos de los doctores especialmente invitados Dr. Juan Luis Díaz de León Santiago (Centro de Investigación en Computación, México) y Dr. Mateo Valero (Instituto Politécnico de Cataluña, España) nos podemos sentir seguros de estar entregando a usted un libro de la más alta calidad.

Agradecemos al IPN, CIC-IPN, CONACyT, ACM, y RITOS-2 por el apoyo brindado para las actividades vinculadas a la realización de este libro.

Alexander Gelbukh,
Grigori Sidorov,
Wilbert A. Olán Cristóbal,
J. Angel Vera Félix

Mayo, 2004, México, D.F.

Contenido

Procesamiento de imágenes

Sistemas de información e ingeniería de software

Internet

Lenguaje natural y análisis de textos

Robótica móvil

Hardware

Web-Mapping Application to Retrieve Spatial Data by means of Spatial Ontologies

Miguel Torres, Juan Luis Díaz de León, Rolando Quintero, Marco Moreno

Geoprocessing Laboratory,
Centre for Computing Research,
National Polytechnic Institute,
Mexico City, Mexico
{mtorres, jdiaz, quintero, marcomoreno}@cic.ipn.mx
http://geo.cic.ipn.mx, http://geopro.cic.ipn.mx

Abstract. Many types of information are geographically referenced and interactive maps provide a natural user interface to such data. However, the process of access and recover spatial data presents several problems related to heterogeneity and interoperability of the geo-information. In this paper, a Tourism Onto-Guide-Web Application (TOGWA) is presented, which is a *web-mapping* system focused on retrieving and representing geo-information on the internet by means of spatial ontologies. Moreover, a Multi-Agent System is proposed in order to deal with the process of obtain the tourism geo-information, which aids in the information-integration task for the several nodes (geographic sites) that are involved in this application. The agent system provides a mechanism to communicate different distributed and heterogeneous Geographical Information Systems and retrieve the data by means of GML description. Also, this paper proposes an interoperability approach based on spatial ontologies matching. The matching is performed by the Multi-Agent System in every node considered in the application. The retrieval mechanism is based on encoding the information in a GML description to link the spatial data with the ontologies that have been proposed. TOGWA is a web-mapping system that is composed by two tiers: Client tier and Spatial Data Server tier, it offers an efficient and user-friendly interface to the clients.

1 Introduction

Nowadays, the use of maps has increased considerably for local, networked and mobile information systems that share or communicate geographically referenced information. This has become possible recently due to the relative widespread availability of digital map data and developments in Geographical Information System (GIS) technology. The application range is wide and includes: local government planning, environmental monitoring, market analysis, navigation and public access to information [1].

Interaction with a digital map is typically based on a cycle of elicitation of user input via menu and dialog boxes, the selection of map areas or features, and the return of information, which may in turn induce a modification to the map content. The maps themselves are often close replicas of traditional paper map cartography.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.1-18.© Centro de Investigación en Computación, IPN, México

This approach is usually found in many commercial GISs and is now being adopted in mapping applications on the Internet.

Developments in human computer interaction with regard to information retrieval and data visualization raise the question of whether or not the conventional approach can be improved. Certainly, there is a motivation to investigate new methods, since the current map interface, particularly on the Internet, often suffers from poor legibility of symbols and text, unnecessary user actions and inadequate adaptation to user interests [1].

Nowadays, the spatial databases are very useful and powerful tools to handle, display, and process the geographical information. These databases usually integrate GISs, which are designed to store and process spatial data. In order to solve some ambiguities in the spatial data processing and interpretation, the geo-information should have good quality, from the input to the representation. The "adequate" representation of spatial data is crucial for improving the decision making process in different environments [2].

In this paper, spatial ontologies based on the spatial semantics are generated, these ontologies can be used to represent geographical objects by means of concepts ("not words"). Such spatial data conceptualization aims to compress the data and to facilitate the knowledge discovery into spatial databases (SDB).

Up-to-date GISs do not extensively explore the spatial data semantics. In that sense, the development of a solid spatial semantics theory is a great challenge in the new trends of the Geocomputation field. For this, the spatial analysis can use alternative methods to represent spatial data. This data representation, together with the semantic rules - both based on data semantics - can be stored in a knowledge-base to generate new concepts that form the spatial ontologies. These concepts are defined by the properties and the behavior of geographical objects, and explored by human experience. In general, we seek to correctly represent geographical objects for their subsequent processing [2, 3], as to retrieve spatial data from different SDBs and to represent them in the TOGWA, for instance.

In our proposal, the first step to generate the spatial ontologies is to obtain the *spatial semantics* of the geographical objects. By obtaining this definition, we can generate spatial ontologies and rely on centralized ontology databases, which are stored in relational database systems. The emergence of Extensible Markup Language (XML) and Geographic Markup Language (GML) allows the ontology metadata to be embedded in the encoded web document, facilitating *semantic matching* by retrieval spatial concepts.

We propose a *Multi-Agent System* (MAS) to perform the communication between the different spatial databases. Although the encoding agents may still refer to centralized ontology databases during the encoding process, the spatial databases can also be encoded in GML because of its openness. Like many systems, we propose a Spatial User Interface Agent (SUIA) in TOGWA to make use of ontologies to validate user inputs and to capture the requests for retrieving spatial data by means of "concepts". In addition, the SUIA works in a web browser providing an easy-use web user interface for online geo-information retrieval. SUAI is characterized by the following features:

- Handle spatial data.

- Retrieve spatial data by means of concepts, considering the spatial subject domain.
- Perform spatial queries according to the generated spatial ontologies.
- Generate new geo-information making spatial analysis.

In this application, the spatial data are associated with different concepts, provided by the spatial ontologies. Moreover, an agent is considered to make several processes, which are divided into different tasks. The tasks that we are considering are the following: representation of geographical phenomena, capability of communicating with the spatial data (different SDB), access to the spatial ontologies, and retrieval of the GML definition according to the user query.

The rest of the paper is organized as follows: In Section 2 we describe the Multi-Agent System proposed to perform a number of tasks in the application, and the spatial ontology that retrieve the spatial data. In Section 3 we present the architecture of the TOGWA web-mapping application. The implementation of the prototype is shown in Section 4. Section 5 shows the preliminary results obtained by TOGWA. Our conclusions are sketched out in Section 6.

2 Multi-agent system and spatial ontology

TOGWA is composed by two basis components to retrieve spatial data. These components are the following:

- A Multi-Agent System (MAS) to perform tasks related to communicate different spatial databases by means of GML definition and to encode the spatial data for retrieving in the SUAI.
- Spatial ontologies to solve *ambiguities* that are presented in the spatial data through concepts ("not words").

2.1 The multi-agent system

According to [4], an agent is a system that tries to fulfill a set of goals in a complex, dynamic environment. It can sense the environment through its sensors and act upon it using its actuators. In this work, we propose a Multi-Agent System (MAS) that provides some services to facilitate the geo-information retrieval mechanism within a tourism system, which is called TOGWA.

There are two main functions of the agent in the TOGWA. One is to provide the user with an intelligent service that communicates different spatial databases and encodes the spatial data for retrieving. The other is to check the GML definition and to link the ontology for knowing whether the concepts fulfill the search criteria.

Several types of agents have been proposed, they are organized in four layers depending on its functionality. The agents are shown in Fig.1.

- *Data Layer.* It is composed by the agents that provide data access services. These services can be, among others: retrieval, storage, adjustment to the communication format.
- *Management Layer.* The agents of this layer handle and coordinate other agents into MAS. Also, they provide the capabilities of communication to other agents.
- *Application Layer.* In this layer the agents perform the tasks of the specific application, such as visualization and functions to the spatial data. Moreover, these agents manage the ontologies (*Trip Package* and *Map*) that provide the needed data to the interface in order to give its own services.
- *Presentation Layer.* Here, the agents provide a user graphic interface to allow the users to obtain the TOGWA services.

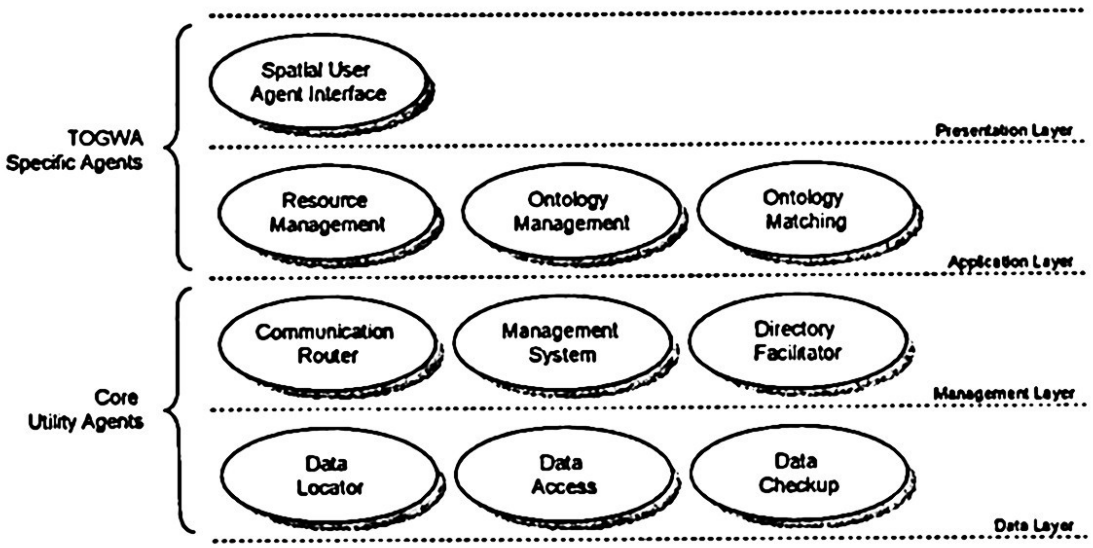


Fig. 1. Classification of the agents used in TOGWA

In Fig. 1 we grouped the layers into two clusters: Core Utility Agents and TOGWA Specific Agents. Core Utility is a set of agents that can be used not only in TOGWA but also in any other application. Its tasks are the following:

- *Data Locator.* It finds the data that better fulfills the description given by its clients. The agent provides, as a result, the address of the agent which can provide the access to the data.
- *Data Access.* It provides the mechanisms to access to the data and metadata of a particular source. The queries and results are given in XML.
- *Communication Router.* This agent provides the capabilities to MAS for communicating with other one, through any suitable way (Internet, others MAS, Virtual Private Networks, etc.).
- *System Management.* This agent handles the process within a MAS. It starts the compute of all other agents in the same MAS.
- *Directory Facilitator.* It maintains a list of all the known agents by MAS as well as the services that each agent provides to the layers.

- *Spatial Facilitator.* This agent retrieves the spatial data from the SDB. According to the client's request. The agent sends the geographical objects to make-up a map in the adequate format (GML description).

The TOGWA Specific Agents is a set of agents that work to accomplish specific TOGWA goals so that they can not be used in other applications. The agents that belong to this cluster are the following:

- *Resource Management.* This agent deals with all the resource assignment tasks, as searching a hotel and flight and finding trip packages, for instance.
- *Ontology management.* This agent keeps the information about the Map Geo-Ontology and uses it to translate the user's request into structured queries, which will be computed into the Ontology Administration Query Module. These queries allow other agents assigning resources to users and finding out geographical objects to provide maps to the clients.
- *Ontology matching.* This agent acts when there are confusions about the concepts in the client's ontology and the MAS ontology. Then the agent attempts to find the closest concept in MAS ontology, according to the concept given by the client.
- *Spatial User Interface Agent.* This agent translates data given by the MAS into a rendered map that the user can understand. The agent is the user interface of the MAS, but it does not belong to it.

The elements that compose the TOGWA-MAS are shown in Fig. 2. We can see that MAS consists of seven agents and an ontology repository.

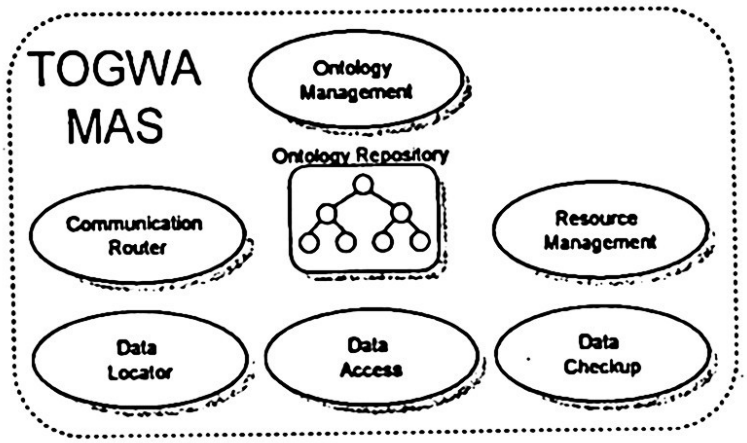


Fig. 2. Features that compose the TOGWA-MAS

In Fig. 3 the interaction model between two TOGWA-MAS is shown. Moreover, Fig. 3 shows the steps to accomplish the TOGWA process, which is outlined as follows.

1. The Client (Spatial User Interface Agent) makes a request to TOGWA (for example, a user in Spain desires to get a road map of the zone of Cancun in Mexico).
2. The MAS in Spain asks to its Directory Facilitator for the MAS that has such information.

3. The Directory Facilitator searches in its database the requested information, and responds to the MAS that the MAS of Mexico has the map.
4. The MAS in Spain asks the MAS in Mexico for the road map of the zone of Cancun.
5. The MAS in Mexico computes the request and determines if it has such information.
6. If it does, then the request passes to the Spatial Facilitator.
7. It makes a spatial query to the SDB for retrieving the geographical objects requested.
8. The Spatial Facilitator returns to the MAS in Mexico the geographical objects needed to make-up the requested map.
9. The MAS in Mexico translates this information using the ontology to a format that the MAS in Spain will understand.
10. Hence, it sends the information to the MAS in Spain.
11. Finally, the MAS in Spain sends the result to the client, and it displays the road map of Cancun to the user with a brief attributive description.

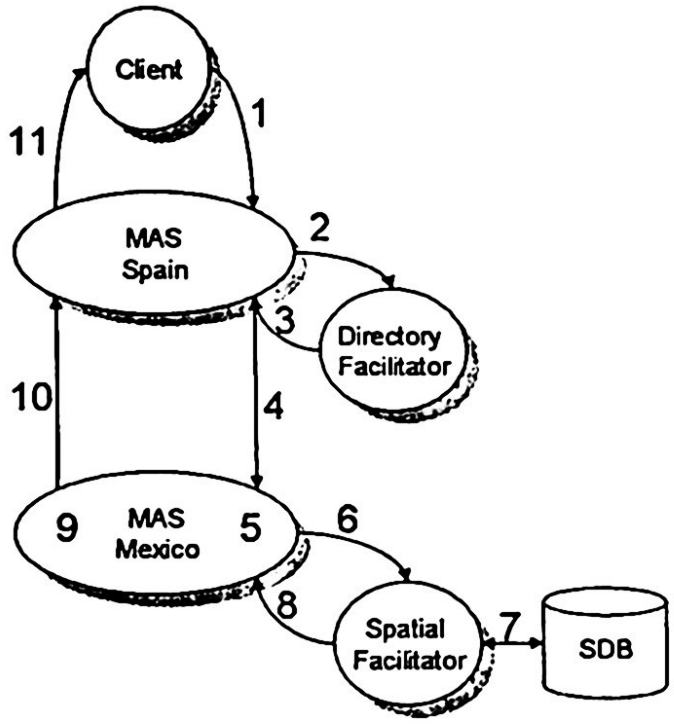


Fig. 3. Interaction model between two existing TOGWA-MAS

2.2 The spatial ontologies

The most widely accepted common conceptualization of the geographical data is based on the description of geographical objects and fields [5, 6]. These objects are not necessarily related to a specific geographic phenomenon, because human-built features are typically modeled as objects [7, 8]. The spatial semantics definition is described in [2] and aim to correctly represent spatial data in an alternative and universal way to generate spatial ontologies.

For this purpose, we will consider a spatial ontology as an explicit, shared and structured specification of conceptualization, that is, a description of properties and relationships that can exist between the geographical objects to form concepts.

Moreover, ontologies can be considered as "languages", which use a specific vocabulary to describe entities, classes, properties and functions related to a certain view of the geographical world [9, 10].

In that sense, our approach is designed to solve the ambiguities that can exist when we deal with single characteristics of the geographical objects. This can be achieved because the spatial ontology is defined by concepts (not by words) according to the geographical objects.

The spatial ontologies can be classified in levels according to their dependence on a specific task or point of view. These levels are generated for a specific spatial ontology (*top-ontology*) and it can be particularized to define a particular ontology (*down-ontology*). There are also different levels of information detail: *Low-level* ontologies correspond to very detailed information and *high-level* ontologies correspond to more general information.

In this situation, the generation of more detailed ontologies should be based on the *high-level* ontologies, so that each new ontology level can incorporate the new knowledge presented in the higher level. These new ontologies are more detailed, because they refine general descriptions of the level from which they have been generated [11].

The levels of ontologies can be used to guide processes for the extraction of more general detailed information. The use of multiple ontologies allows the extraction of information in different stages of classification.

The use of explicit spatial ontologies contributes to better correct spatial representation, because every geographical object description is based on an implicit ontology. By using that, it is possible to avoid explicit *conflicts* and *confusions* between the ontological concepts and the implementation [12].

On the other hand, spatial ontologies play an essential role in the conceptualization of spatial databases, allowing the establishment of correspondences and interrelations among the different domains of geographical objects and relations.

For instance, the ontology "Limit" can be represented in different concepts for diverse spatial databases. "Limit" in some cases represents: "coast boundary", separation between the "ground" and the "sea", "contour of value zero", "boundary" among two regions (states, countries, etc.), and so on (Fig. 4).

Using this approach, we can generate specific spatial ontologies after defining the top-ontology to particularize the conceptualization in other specific ontologies (down-ontology).

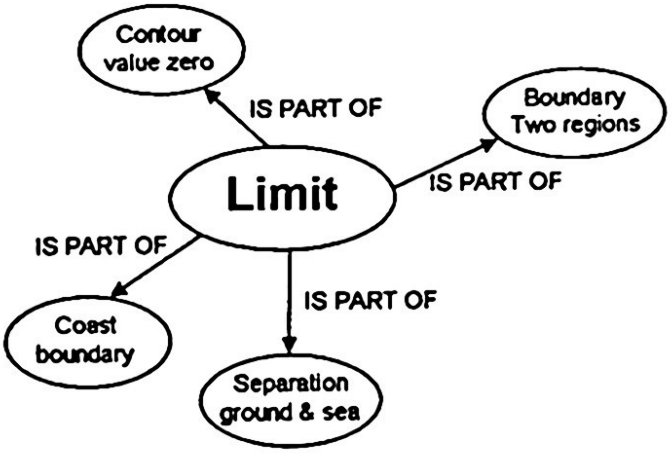


Fig. 4. Ontology "Limit" that is composed by different concepts related to "limit" for subsequent spatial representation

According to our approach, it is obligatory to count with a spatial subject domain. It is defined as a set of "names" that describe the primitives of spatial representation. Thus, we can start with *a priori* knowledge of the geographical objects that appear, e.g. in the map legend. For example, "blue" lines are united under the concept (name) "river" and "black" lines are united under the concept "fracture", etc. In reverse, the different concepts are united under the same description of the spatial representation that is "line" [2]. The interaction between the subject domain and the taxonomy is used to locate concepts into the spatial subject domain that correspond to a case of study, and to compute these concepts in order to generate spatial ontologies [2, 4]. Fig. 5 shows the definition of spatial subject domain.

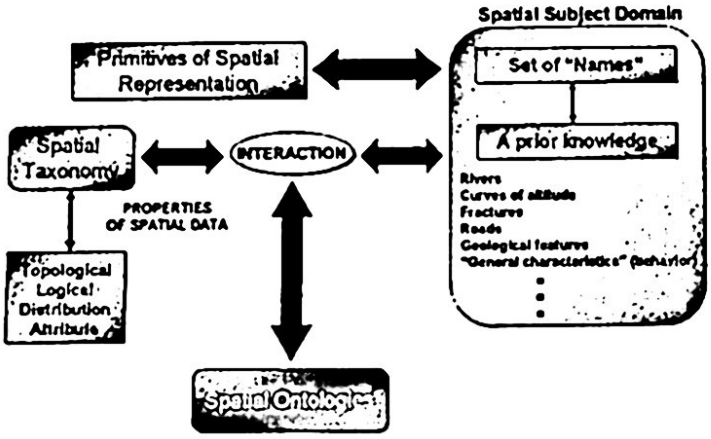


Fig. 5. Interactions of the Spatial Subject Domain

To define the spatial subject domain, it is necessary to elaborate a description of the spatial thematic to be analyzed, considering the main features that compose this theme, such as the data model and the resolution levels of the spatial data.

All characteristics that are considered in the description of the spatial subject domain should represent relationships between themselves too. Spatial subject domain has to recognize the different *semantic levels* of *a priori* knowledge that is stored in this domain.

The use of ontologies in spatial databases enables knowledge sharing and information integration. The proposed approach provides dynamic and flexible information exchange and allows partial integration of spatial data when completeness is impossible.

This can help the next generation of spatial databases to solve *semantic ambiguities* in the available geo-information, because the context of the spatial data can change according to the case of study [11].

The query functionality to retrieve spatial data by means of spatial ontologies is the following: 1) the user makes a request by means of TOGWA; 2) the information is searched in any TOGWA-MAS; 3) when the data are located; it is encoded into the GML definition; 4) this definition is sent to the Ontology Administration Query Module to compute the request for obtaining the spatial and attributive data by means of concepts, which form the ontology; 5) inside TOGWA, *a priori* knowledge that is stored in the spatial subject domain interacts with the spatial taxonomy, considering in this case, the "arcs" as primitive of representation. Fig. 6 shows the mechanism to obtain the spatial ontology by means of the Ontology Administration Query Module.

In addition, Fig. 6 shows the query mechanism to describe the concept "Roads" into the *Map Geo-Ontology*. In this case, the ontology is composed by several "sub-concepts", which are ordered in a hierarchical way. Moreover, we see in Fig. 6 different levels of the concepts, starting with a top-level (Roads) and finishing with down-levels (One rail, two rails, etc.). When the ontologies present more levels of concepts, it is possible to particularize these in sub-concepts, while the level is less (down-level), the concept is more particular. The ontologies that have been proposed by TOGWA are described in Section 4.

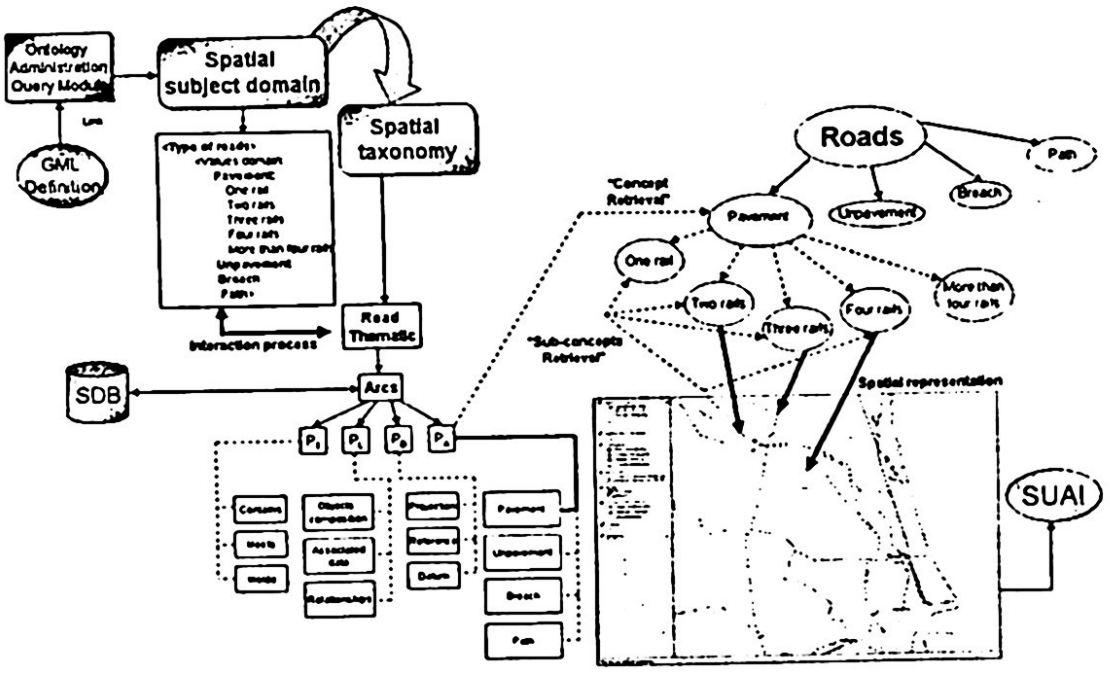


Fig. 6. Interactive process to retrieve geographical objects by means of spatial ontologies

3 Architecture of TOGWA

The Tourism Onto-Guide-Web Application (TOGWA) is a web-mapping system that is composed by two tiers: Client tier and Spatial Data Server tier. These tiers contain the following components: Spatial User Agent Interface (SUAI), Ontology Administration Query Module (OAQM), Spatial Data Server (SDS), Agent Administration Module (AAM) and Spatial Database (SDB) [13].

This application presents client-server architecture. TOGWA is considered a distributed system because it is able to retrieve spatial data from different GIS sites by means of GML definition. Fig. 7 depicts the general architecture of the Web-Mapping system.

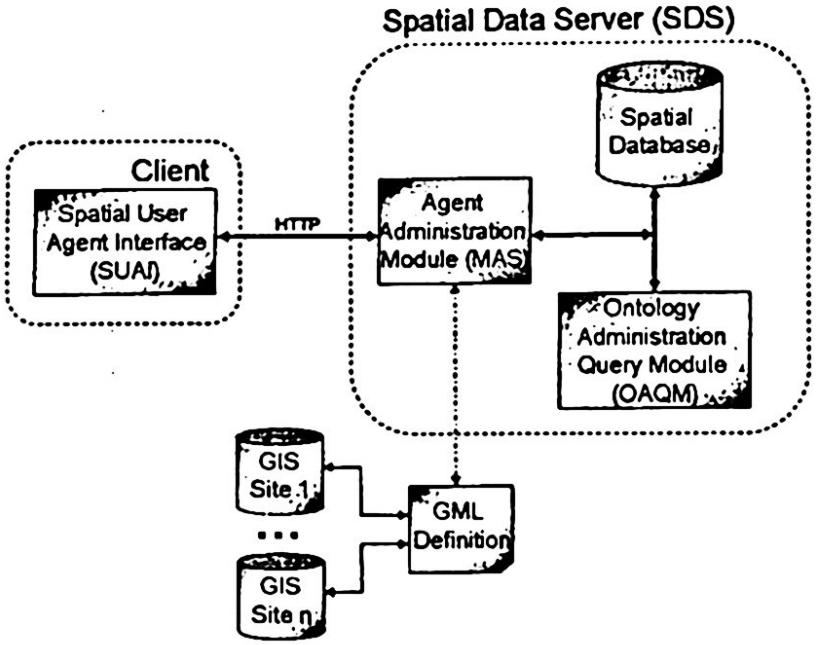


Fig. 7. Architecture of the Tourism Onto-Guide-Web Application

The general process to retrieve spatial and attributive data is the following:

Spatial User Agent Interface (SUAI) receives requests from a user. It assists the client to search, query and manipulate the map in an efficient and user-friendly way. SUAI attempts to understand the subject domain (geographical context of the user), and sends a message to the Spatial Data Server (SDS) to ask more geo-information or to modify the map to change the content and resolution detail. SUAI should keep a concise profile for each user to record his search of interest. The Agent Administration Module (AAM) receives requests from the SUAI and broadcasts the requests of the users to the Ontology Administration Query Module (OAQM) in order to search the concept into the ontologies and to retrieve the geo-information from the Spatial Database (SDB). If the geo-information associated to the concept could not be found in the SDB, the OAQM will send a notification to the AAM to perform a query in different GIS sites linked to TOGWA. This process is made up by means of the GML definition, when the geo-information is found; it is encoded in the GML description and transferred to the AAM to retrieve the spatial data according to the spatial ontology. Finally, the spatial data is sent to the SUAI.

4 Implementation of the prototype

TOGWA prototype has been implemented in Java to keep the distribution and multi-platform execution [14]. TOGWA consists of seven nodes to retrieve spatial and attributive data. The nodes that are considered to this application are the following: Mexico, Spain, Costa Rican, Italy, England, Cuba and Chile. The SUAI is implemented as a Java Applet and runs on the client side to interact with a web user. The AAM has been implemented as a Java servlet using Tomcat 5.0.12. The visualization on the client side is based on *Shapefiles*, which is proposed by Esri, Inc [15].

There are several components in the prototype, a web page, an invisible applet, a servlet and an ontology parser. The data workflow is depicted in Fig. 8.

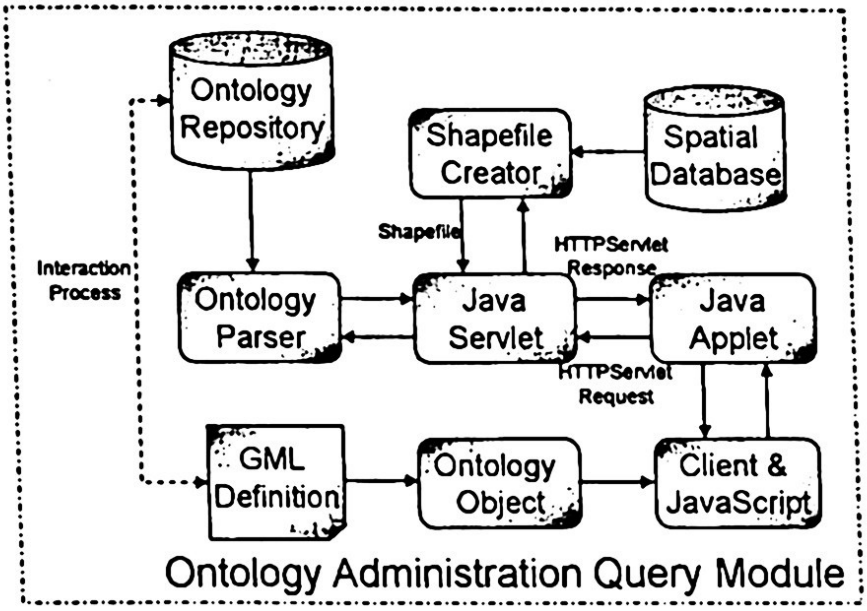


Fig. 8. Data workflow to obtain the spatial data according to the spatial ontology

When a user accesses the web page, the JavaScript embedded in the web page will call a Java applet to send an http request to the Java servlet, which will invoke the ontology parser to create an ontology object from the ontology repository. If the information is not found in the Ontology Repository, the OAQM sends a GML definition to locate the data in any node. When the information is found, it is received by the OAQM for being computed. Later, the OAQM sends the object as a serialized Java object to the applet.

The ontology object contains the entire ontology. The applet uses the ontology object to verify if the user has performed a valid search. If valid, the applet will submit the search to the servlet, which in turn invokes the shapefile generator to obtain a shapefile for the client to refresh the web page and to retrieve the spatial data.

In this context, a spatial ontology is a part of knowledge, concerning a particular spatial subject domain; it describes a spatial taxonomy of concepts for that subject domain, which define the *spatial semantic interpretation* of the knowledge. Spatial ontologies in TOGWA define the *spatial semantic relationships* of the geographical objects. The ontology repository is organized in a tree structure.

We propose two ontologies to obtain the spatial data by means of concepts, in this case the *Map Geo-Ontology* and *Trip Package Ontology.*

These ontologies provide the concepts related to the information retrieval to the user. The retrieval process is performed by constraints, which are defined by the client.

Map Geo-Ontology is focused on retrieving particular maps of the user interest. This spatial ontology can generate four types of maps: Roads, Weather, Urban and Sightseeing. The spatial ontology is generated by the interaction process of the spatial taxonomy and the spatial subject domain.

Trip Package Ontology is proposed to acquire attributive data related to the interest places to visit for the users. A user can obtain relevant information according to his interest and the matching concepts in the definition. The data can be found in any node considered into the application.

The interaction and communication process has been described in previous section (Multi-Agent System). The retrieval of concepts among ontologies is performed considering the *relationships* of them by means of MAS. Fig. 9 and Fig. 10 show the ontologies that we propose to retrieve spatial data by means of concepts.

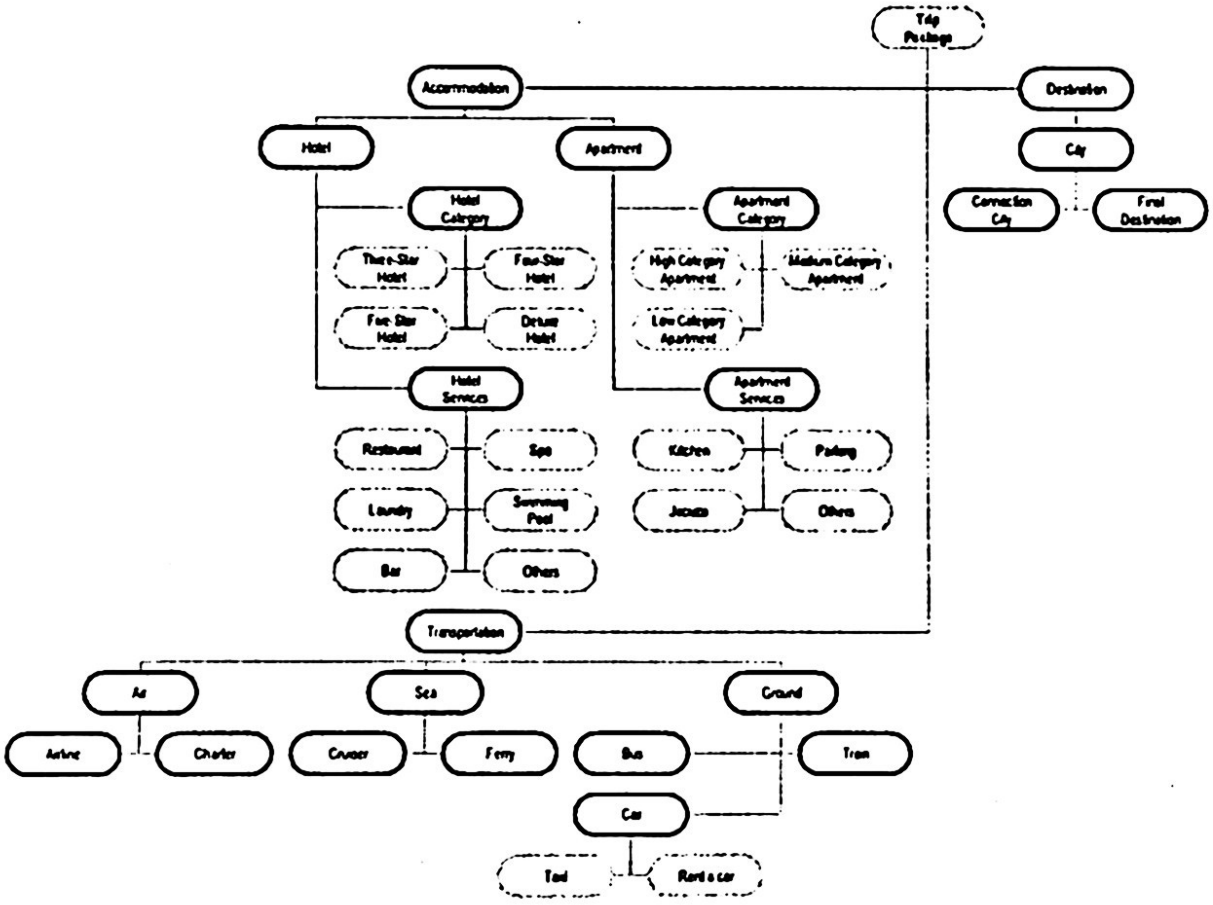


Fig. 9. Trip Package ontology

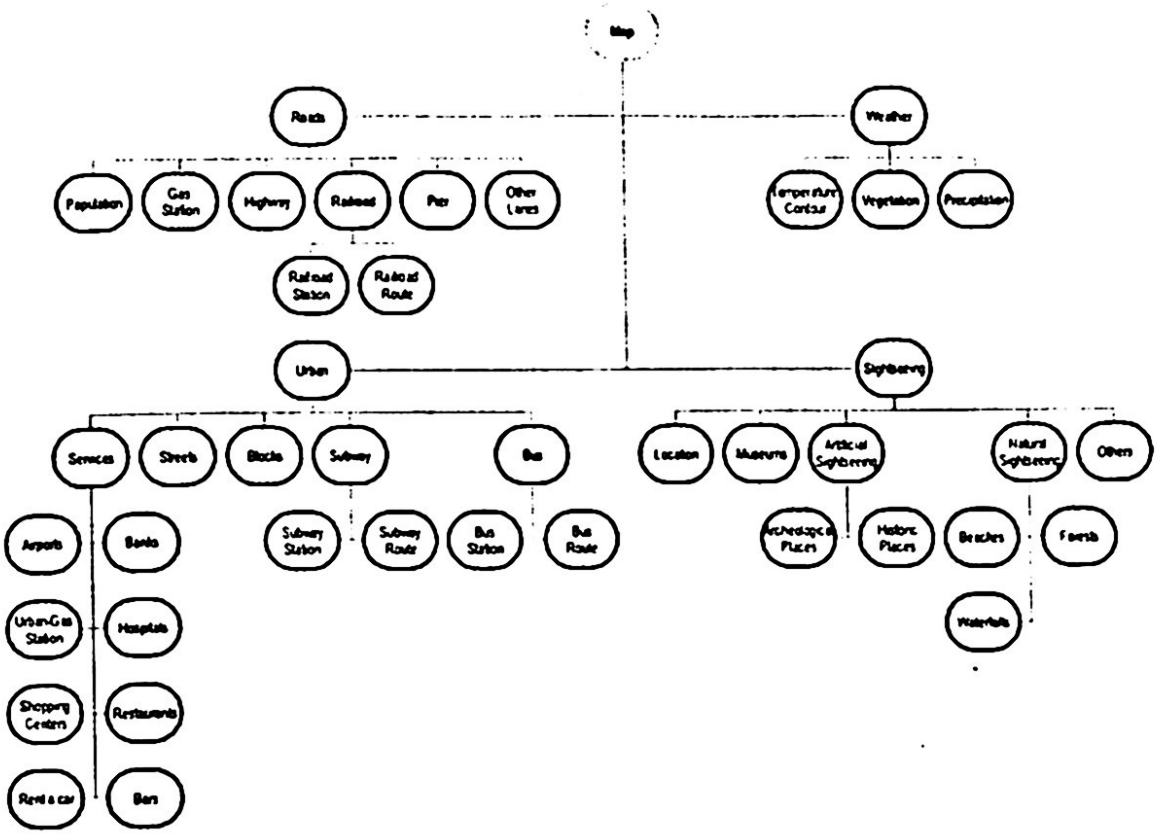


Fig. 10. Map spatial ontology

On the other hand, a GML definition is used to obtain the spatial data from different distributed GIS according to the request of the user. MAS sends this definition to find the specification related to the request. If the data has been found, the GML definition encodes the information, which is sent to the Ontology Repository for matching this information encoded into the GML definition with the ontology structures. Inside the Ontology Administration Query Module, the information is parsed for relating it with the concepts, which integrate the ontologies. We use the *relationships* between concepts that belong to the ontology to communicate the *Map* and *Trip Package Ontologies*. The use of relationships of concepts provides detailed information (spatial and attributive), because we can obtain concepts in certain directions as breath and depth first search.

If the information is valid, it is necessary to generate the *shapefile* by means of the Shapefile Creator in order to send it to the Spatial User Agent Interface. A brief description of the GML definition is shown in Table 1.

5 Preliminary results by TOGWA

By using TOGWA, we have developed roads, city, weather and sightseeing maps. These maps are generated by means of concepts that belong to the *ontologies* (*Trip Package* and *Map*). The data have been retrieved by the GML definition according to the user request. *SUAI* contains an efficient and user-friendly interface, which is composed by some spatial tools. Some results are shown in this section.

Fig. 11 depicts the *map of roads* for Toluca City, Mexico. This map consists of different thematics as Populations, Roads, Urban Areas and Internal Administrative Divisions. The roads presented in this map are classified by its properties: four rails, two rails, tracks and urbanized routes. The goal of this map is to guide the users for knowing their interest places in low level of detail (1:200,000).

Table 1. Brief GML description related to TOGWA

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xml:lang="en"
   xmlns:camb="http://geo.cic.ipn.mx:9090/RDF/VRPexample_profile3_schema.rdfs#"
   xmlns:gml="http://geo.cic.ipn.mx:9090/RDF/VRP/Examples/gml.rdfs#"
<!--The camb and gml namespaces replaced for validation purposes Map Ontology Data-->
   <camb:Map>
      <gml:boundedBy>
         <gml:Box gml:srsName="ROAD:4326">
            <gml:coordinates>
               0.0,0.0 100.0,100.0
            </gml:coordinates>
         </gml:Box>
      </gml:boundedBy>
      <camb:modelDate>
         Dic 2003.
      </camb:modelDate>
      <camb:modelMember>
         <camb:Roads>
            <gml:name>
               I45
            </gml:name>
            <gml:description>
               Federal Highway from San Pablito to Cancun.
            </gml:description>
            <gml:centerLineOf>
               <gml:LineString gml:srsName="ROAD:4326">
                  <gml:coordinates>
                     0.0,50.0 100.0,50.0
                  </gml:coordinates>
               </gml:LineString>
            </gml:centerLineOf>
                  <camb:ModelMember>
                     <camb:Highway>
                           <gml:Name>
                               Interstate 35
                           </gml:Name>
                           <gml:description>
                      Main Highway to connect Cancun and Chetumal.
                           </gml:description>
               <gml:LineString gml:srsName="ROAD:4326">
                  <gml:coordinates>
                     344,552.4,566,763.67,763,234.12, 3,456,655.65, 890,765.31
                  </gml:coordinates>
               </gml:LineString>
                           </camb:Highway>
                        </camb:ModelMember>
         </camb:Roads>
      </camb:modelMember>
   </camb:CityModel> _ </rdf:RDF>
```

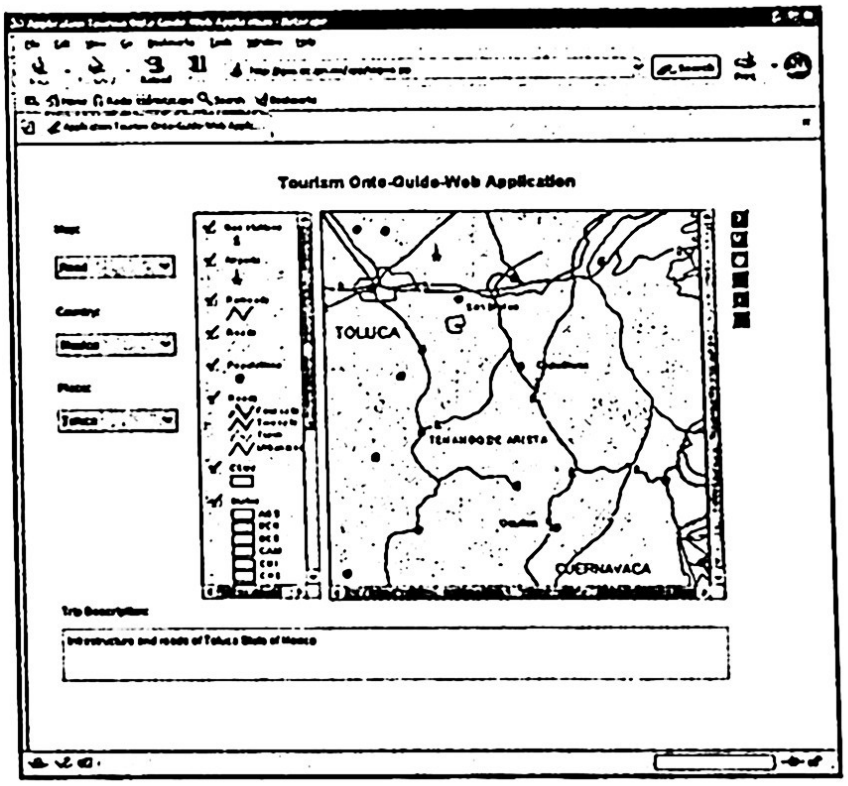


Fig. 11. Map of roads

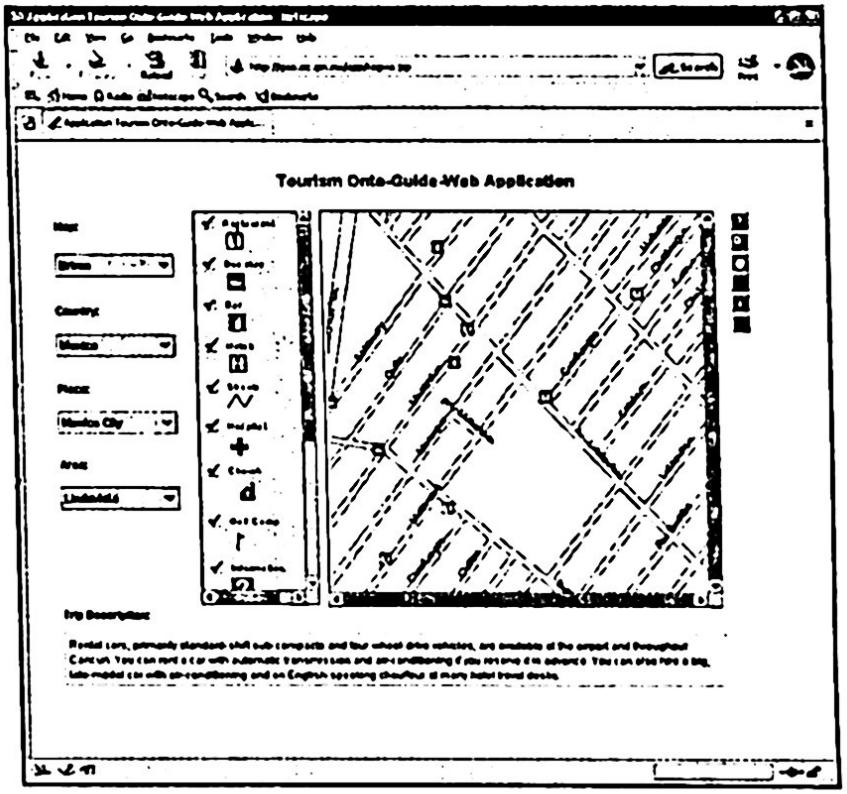


Fig. 12. City Map

City map is composed by streets, avenues and present different interest sites. Fig. 12 shows the city map of Lindavista area in Mexico City, within a scale of 1:5,000. In this map we show the location of different sites as Restaurants, Bus Stations and Hotels in this area. The users can retrieve a *city map* according to their necessities. Also, *Trip Description Box* provides useful information related to the user request.

Fig. 13 shows the *Sightseeing map* of San Pablito in Quintana Roo, Mexico. This map describes general aspects of San Pablito, showing the Information Sites Location, Gas Stations, Camping Zones, Restaurants and Archeological Sites. Moreover, it provides the general structure of the population. Additionally, this map presents the roads that connect with San Pablito (in red color). The map scale depends on the size of the area of interest.

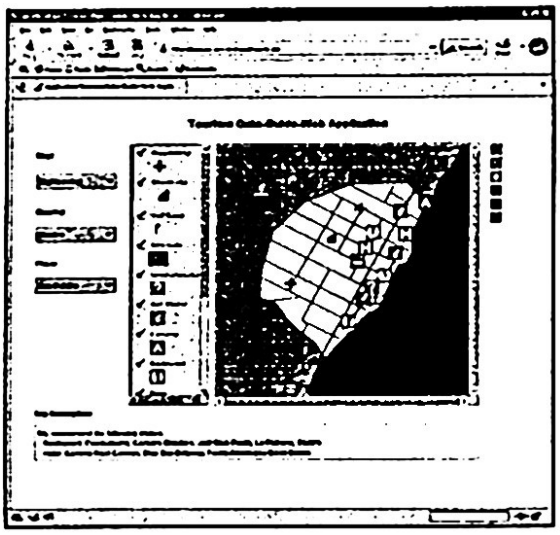


Fig. 13. Sightseeing Map

Fig. 14. Weather Map

Weather map consists of vegetation areas, temperature and precipitation contours. This map guides to the users to know the characteristics of the weather in a particular place, when the users want to travel according to their criteria of retrieval request. In

addition, Fig. 14 depicts attributive information related to the map into the *Trip Description Box*.

6 Conclusions

In the present work, the Tourism Onto-Guide-Web Application (TOGWA) has been proposed. TOGWA is a *web-mapping* system focused on retrieving geo-information by means of spatial ontologies, and on representing it on the Internet. We use the spatial semantics to generate the geo-ontology for representing geographical objects by means of concepts.

TOGWA contains a Multi-Agent System, which performs the following tasks:

- To communicate different spatial databases by means of GML definition.
- To encode the spatial data for retrieval in the SUAI.
- To solve *ambiguities* that can be presented in the spatial data by means of concepts ("not words").

The spatial subject domain definition is oriented towards an interaction with spatial taxonomy to conceptualize the spatial databases. In essence, the spatial subject domain is defined as a set of "names" that describe the primitives of spatial representation. Thus, we can start with *a priori* knowledge of the geographical objects to examine the spatial data, which interact with the spatial taxonomy to generate spatial ontologies.

We attempt to show an alternative approach to represent spatial data on the Internet considering the *relationships* that compose the ontologies to retrieve spatial data according to several search criteria.

In addition, the spatial ontologies catch the semantics of the spatial data to provide relevant information related to the concepts. These ontologies can be used to establish agreements on diverse views of the world and consequently to carry out the "meaning" of the geo-information. In many situations, this geo-information is embedded in the spatial representation of geographical phenomena in the human-mind.

The use of ontologies in spatial databases enables knowledge sharing and information integration. The proposed approach provides dynamic and flexible information exchange and allows partial integration of spatial data when completeness is impossible in the web.

The communication between ontologies is performed by MAS, which seeks the relationships of the concepts to match nodes in the ontologies. This process is iterative and the new generated concepts can be considered in the spatial subject domain.

This approach can aid to solve *semantic ambiguities* between the available geo-information, because the context of the spatial data can change, according to the case of study and the representation state by means of concepts of the spatial data.

Acknowledgments

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

References

1. Li, M., Zhou, S. and Jones, C.B.: Multi-agent Systems for Web-Based Map Information Retrieval. In Egenhofer, M.J. and Mark, D.M. (Eds.), GIScience 2002, Lecture Notes in Computer Science Vol. 2478 (2002) 161-180
2. Torres, M. and Levachkine, S.: Semantics Definition to Represent Spatial Data, In: Levachkine S., Bodanksky E. and Ruas A., (eds.), e-Proceedings of International Workshop on Semantic Processing of Spatial Data (GEOPRO 2002), Mexico City, Mexico (2002)
3. Egenhofer, M. and Frank, A.U.: LOBSTER: Combining AI and Database Techniques for GIS. International Journal of Photogrammetric Engineering and Remote Sensing, Vol. 56, No.6 (1997) 919-926
4. Maes, P.: Modeling Adaptive Autonomous Agents. Artificial Life, No. 1 (1994) 135-162
5. Fonseca, F. and Egenhofer, M.: Ontology-Driven Geographic Information Systems, Proceedings of 7th ACM Symposium on Advances in Geographic Information Systems, Kansas City, United States (1999) 14-19
6. Fonseca, F., Egenhofer, M. and Agouris, P.: Using Ontologies for Integrated Geographic Information Systems. Transactions in GIS, Vol.6, No. 3 (2002) 25-40
7. Egenhofer, M. and Frank, A.: Naive Geography, in Frank A. and Kuhn W., (Eds.) Spatial Information Theory, A Theoretical Basis for GIS, Proceedings of the International Conference COSIT '95, Lecture Notes in Computer Science, Vol. 988, Springer-Verlag, Berlin (1995) 1-15
8. Gooldchild, M.F., Egenhofer, M., Fegeas, R. and Kottman, C. Interoperating Geographic Information Systems, Editorial: Kluwer Academic Publishers (1999)
9. Guarino, N.: Formal Ontology and Information Systems, in Guarino (Ed.), Formal Ontology in Information Systems, Editorial: IOS Press (1998) 3-15
10. Guarino, N.: Formal Ontology, Conceptual Analysis and Knowledge Representation. International Journal of Human and Computer Studies, 43, Vol. 5, No. 6 (1999) 625-640
11. Guzmán, A. Domínguez, C. and Olivares, J. Reacting to unexpected events and communication inspite of mixed ontologies, In: C. Coello, A. Albornoz, L. Sucar, O. Cair and G. Kemper (eds.), Advances in Artificial Intelligence, Proceedings of Mexican International Conference on Artificial Intelligence (MICAI 2002), Vol. 2313, Lecture Notes in Computer Science, Springer-Verlag, Merida, Yucatan, Mexico (2002) 377-386.
12. Levachkine, S. and Guzmán, A.: Relatedness of the elements of hierarchies partitioned by percentages, Lecture Notes in Computer Science, Vol. 2972, Springer-Verlag, Berlin (2004) 135-155
13. Torres, M., Moreno, M., Menchaca, R. and Levachkine, S.: Making Spatial Analysis with a Distributed Geographical Information System, Proceedings of 21st IASTED International Conference on Databases and Applications (DBA' 2003), Innsbruck, Austria (2003) 1245-1250
14. Zhou, S. and Jones, C.: Design and Implementation of Multi-Scale Databases, Proceedings of 7th International Symposium on Spatial and Temporal Databases (SSTD01), Lecture Notes in Computer Science, Vol. 2121, Springer-Verlag, Berlin (1995) 365-384
15. http://www.esri.com, Environmental Systems Research Institute

Recuperación y análisis de datos espaciales usando tecnología .NET

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 este capítulo, se presenta una aplicación basada en Web Mapping para recuperar y analizar información espacial utilizando tecnología .NET, en particular efectuar análisis espacial de buffer en la web. La aplicación esta basada en un componente Net el cual opera sobre un servidor Web, se divide básicamente en dos módulos los cuales son el modulo de análisis Espacial (MAE) y el Módulo de Recuperaciones Espaciales (MRE).Para demostrar la utilidad de la propuesta los análisis y consultas se realizan a partir de una imagen JPG que muestra un mapa de la republica mexicana; además de poder visualizar los datos de forma tabulares trabajo se realizo con base en el modelo cliente-servidor usando un Navegador Web.

1 Introducción

Hoy en día, una de las tendencias de los Sistemas de Información Geográfica (SIG) esta enfocada hacia el web. Donde el concepto de web mapping se ha referido a la capacidad de ofrecer las funcionalidades básicas de navegación de un SIG, como son: Pan, Zoom, Identify etc. Pero actualmente se ha extendido para que se proporcionen las capacidades de análisis espacial mas comunes de un SIG, el caso de análisis de proximidad es uno de ellos, y por ello este trabajo fue desarrollado para implementar dicho análisis usando Web Mapping, El análisis de Buffer es una función disponible en los SIG como Arcview y ArcInfo [1], actualmente, algunos servidores web de mapas solo realizan análisis de *buffer puntual* [2] y sobre formatos raster, como es el caso de [3]. Las aplicaciones de Web Mapping existentes no proporcionan capacidades de análisis espacial, ya que están enfocados a la navegación, presentación y ejecución de consultas tabulares. Es por ello que esa propuesta se enfoca a implantar el algoritmo de buffer lineal para una aplicación de Web Mapping y con la capacidad de obtener resultados en formato raster y vectorial.

En el área de la Geocomputación, aplicar análisis de *buffer o proximidad*, es efectuar un proceso que consiste de generar un polígono, el cual encerrara una primitiva espacial (punto, línea o polígono) a través de un radio de distancia especifica o por medio del valor de un atributo; a esto se le denomina el área de

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.19-29.© Centro de Investigación en Computación, IPN, México

influencia. Los buffer se generan de acuerdo al tipo de primitiva espacial, por lo cual aplicar un buffer a un dato de tipo puntual, resultara en un buffer puntual, para el caso de la línea en un buffer *lineal*, y finalmente para polígonos en un *buffer poligonal* [4].

El análisis de proximidad depende del radio o longitud establecida, de las primitivas espaciales y del tipo de fenómeno natural sobre el que se aplica. El análisis de proximidad, forma parte de los principales análisis espaciales que ofrece un SIG, su funcionalidad se ha desarrollado en ambientes 3D con lenguajes de programación tales como: Java y VRML [5] la desventaja de estos es que no trabajan sobre formatos propietarios y no realizan buffer. También, el proceso de bufferizacion se ha implantado para una herramienta SIG de escritorio [6]. Los SIG como Arcinfo [7] ofrecen opciones de configuración para efectuar bufferizacion, atributos que se encuentran implicados y la distancia del buffer [8].

La aplicación del análisis de Buffer se basa en la longitud o radio de influencia, y una vez aplicado, el resultado consiste en una nueva capa de datos espaciales, la cual esta conformada por los objetos espaciales, de tipo polígono, los cuales cubren o rodean a las objetos espaciales sobre los cuales se aplico el análisis. En esta nueva capa, los objetos que se intersectan, se fusionan para generar un solo objeto de superficie continua, lo cual permite hacer un cálculo del área de proximidad o buffer de los objetos.

Por otro lado, si se considera el enfoque del Web y la gran variedad existente de formatos espaciales que se encuentran tanto en bases de datos orientados a objetos, como archivos de formato propietario y etc. [9] esto resulta en un problema para recuperar y manejar la información , por lo cual se han generado diversas propuestas y modelos para poder compartir y recuperar los datos, todo ello a través de lenguajes como: XML o GML lo cual facilita el manejo e interoperabilidad de los objetos geográficos en Internet [10] aunque no se han enfocado a los análisis de proximidad . Además, teniendo en cuenta que en el Web es prioridad la capacidad y velocidad de respuesta se opto por un modelo bajo el cual las operaciones se realizan tanto en formato *vectorial* como *raster*, pero los resultados se muestran en un formato de imagen JPEG o GIF, lo cual permite agilizar las respuestas del servidor hacia el cliente, cabe señalar que el sistema proporciona al cliente la opción de obtener el resultado en formato vectorial, y ésta es una alternativa de solución a la limitación de aplicar análisis espacial en una pagina HTML [11].

El resto del documento esta organizado en la siguiente manera. En la sección 2 se presenta la arquitectura de la aplicación y funcionamiento de la aplicación; la sección 3 describe el proceso de *proximidad*. Se muestran los resultados en la sección 4 y la sección 5 trata sobre las conclusiones del trabajo.

2 Arquitectura de la aplicación

La arquitectura de la aplicación se basa en el modelo cliente – servidor, y en la parte del servidor residen los siguientes módulos: Módulo de Análisis Espacial (MAE), el Módulo de Recuperaciones Espaciales (MRE) quien recibe peticiones del MAE, el MRE se encarga de seleccionar la función correspondiente a las peticiones recibidas.

El MRE trabaja con base en un componente cuya función es desplegar y convertir las unidades del mapa a unidades de pantalla, y efectuar consultas atributivas y espaciales desde una base de datos geográfica o un archivo de formato especifico, cabe señalar que estos módulos fueron desarrollados con tecnología Net.

Cuando el MRE ha recuperado los objetos espaciales requeridos para el análisis, estos son enviados al MAE; y de acuerdo a estos resultados, se construye la página HTML, incrustando la imagen que muestra el resultado de la petición. El modelo se ilustra en la Fig. 1.

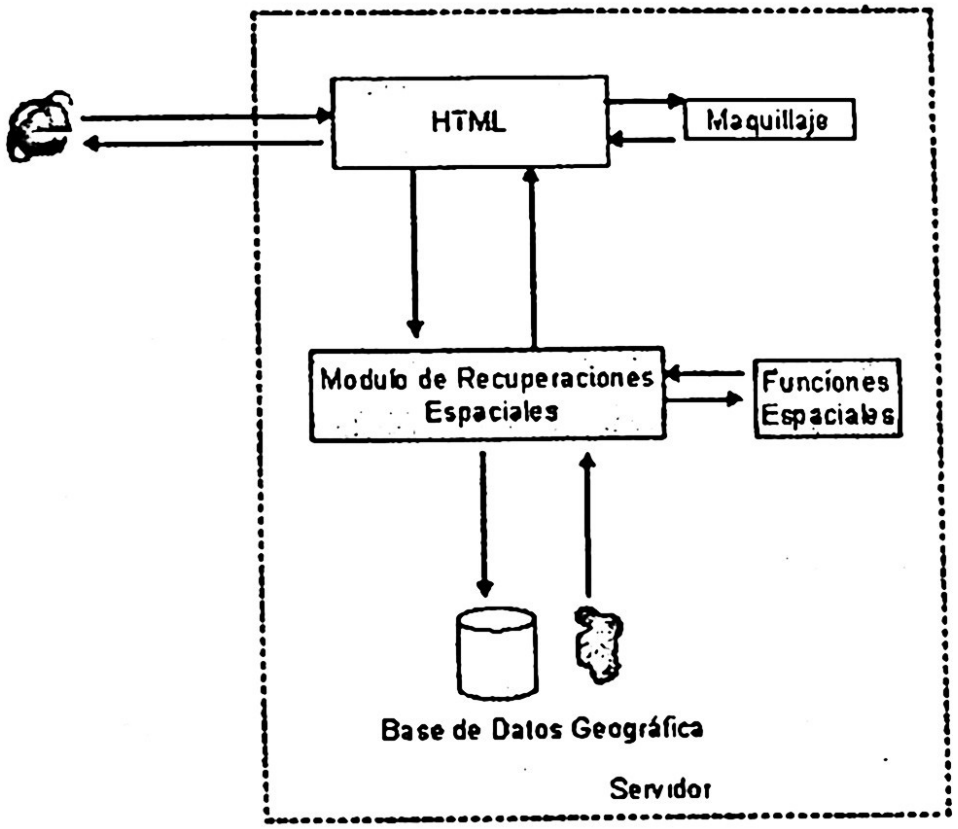


Fig. 1. Arquitectura del Servidor Web de Mapas

En la Fig. 1 se puede apreciar que el Módulo de Recuperaciones Espaciales (MRE) es la parte principal en el modelo. Entre sus funciones esta el proceso para hacer el análisis espacial requerido por el cliente. Una vez que se ha seleccionado la función, se recuperan los datos, construyendo la consulta espacial necesaria para acceder a la información, geométrica o topológica, que esta en alguna fuente de datos. La consulta es ejecutada y el MRE, envía los resultados obtenidos generando la imagen del mapa y construyendo la página HTML.

El modelo para análisis de buffer se muestra en la Fig. 2. El MRE una vez que selecciona el tipo de operación espacial a realizar, hace la llamada a las funciones implicadas para obtener los resultados requeridos y enviarlos.

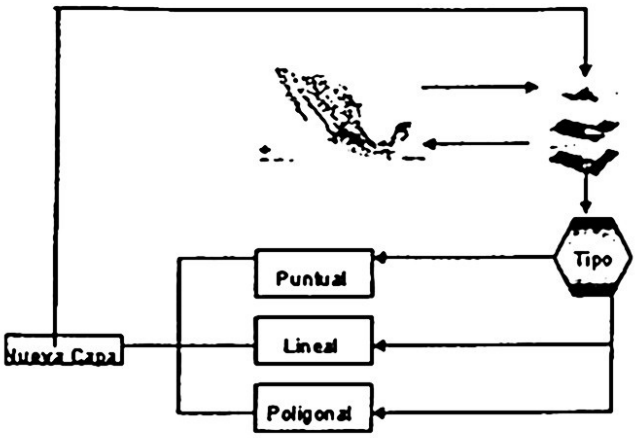


Fig. 2. La operación de MROE

En la Fig. 2, Se muestra el funcionamiento del MRE respecto al análisis de proximidad, donde lo primero que se realiza es determinar el tipo de capa sobre la cual se pide efectuar el buffer, el tipo de capa se determina a través del tipo de primitiva de representación espacial y una vez que se conoce esto se aplica el proceso de *bufferizacion* correspondiente. El proceso de *bufferizacion* genera una nueva capa espacial en base a los parámetros de influencia que sean considerados en el análisis. Este mecanismo se encarga de generar la nueva capa de datos (capa buffer de datos), en base al siguiente proceso:

- Se aplica el buffer a cada capa de datos seleccionada.
- Se genera una nueva capa de datos buffer.
- Se exporta a formato shapefile para que pueda presentarse como una capa de datos nueva en el sistema y sea posible procesarla de manera independiente.

3 Proceso de bufferización

Para efectuar el proceso de *bufferizacion* es necesario conocer el tipo de primitiva de representación: puntos, líneas o polígonos. A continuación se describe el proceso que se efectúa para realizar la *bufferizacion* lineal, ya que la puntual consiste únicamente de generar un circulo cuyo radio esta en función de un atributo o distancia especifica, siendo el proceso muy sencillo, además la bufferizacion lineal es la base de la bufferizacion poligonal.

Para efectuar la bufferizacion es necesario trabajar con la topología de las capas. Para iniciar, se asignan a cada arco de línea (delimitado por nodos) el ancho de buffer apropiado, (puede ser fijo o mediante una tabla de pesos). Dicho ancho de buffer lo denominaremos radio Buffer b; es necesario que cada arco se procese de manera individual, para hacer un proceso iterativo. Entonces considerando la Fig. 3, tendremos que cada segmento de línea tiene un nodo inicial definido por (E_1, N1) y un nodo final definido por (E_2, N2).

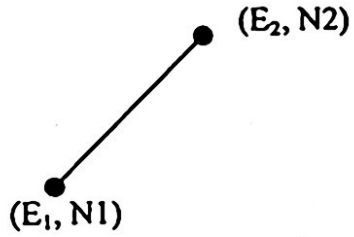


Fig. 3. Nodos iníciales de un arco de línea

En el siguiente paso, utilizaremos las coordenadas de cada nodo, para calcular Δx y Δy entre los dos puntos a ser computarizados, esto se muestra en la Fig. 4.

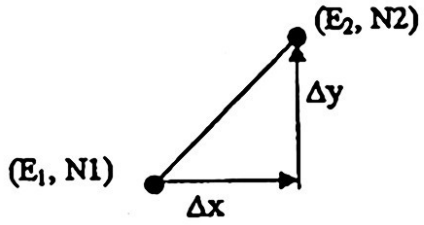


Fig. 4. Cálculo de Δx y Δy

En el tercer paso se determinan los puntos finales de las líneas buffer, las cuales se posicionan a los costados del segmento de línea original, a una distancia perpendicular b. En la Fig. 5 se ilustra este proceso; las ecuación 1 se aplica para determinar los valores correspondientes.

$$E_1 \pm b * Sen\left[Tan^{-1}\left(\frac{\Delta x}{\Delta y}\right)\right] \qquad (1)$$

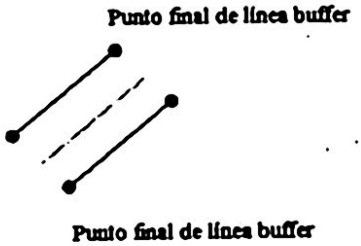


Fig. 5. Puntos finales de las líneas buffer

Para el caso en el que la línea sea horizontal o vertical con respecto al eje de coordenadas la formula se simplifica. Es entonces cuando se efectúa el siguiente paso: 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 de buffer, han sido identificadas,

se repite el proceso al siguiente segmento de línea. El quinto paso consiste en identificar las intersecciones de las líneas paralelas de buffer de cada segmento y las nuevas coordenadas son asignadas a los vértices comunes.

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. 6 muestra estos puntos.

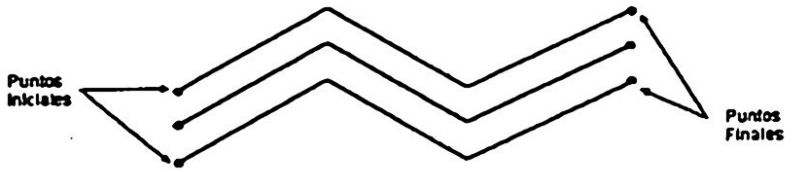


Fig. 6. Puntos iniciales y finales de las líneas de buffer

Para finalizar el buffer, se procede a 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. 7.

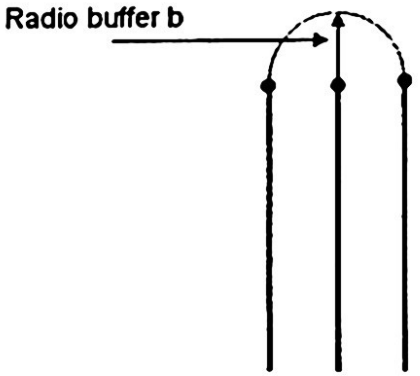


Fig. 7. Buffer de radio b

3.1 Buffer para datos tipo punto

Dentro de las 3 primitivas espaciales existentes para un Shapefile, aplicar buffer puntual, es el proceso más sencillo, ya que solo se necesita generar un polígono circular alrededor de cada punto con un radio igual al ancho del buffer. Es posible que existan múltiples puntos en la capa a la que se le aplicará el buffer, por lo que se deben verificar las sobreposiciones que se produzcan. Las secciones que se sobrepongan deberán ser fusionadas, 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 conocidas como: *Intersección* y *Disolución.*

En la Fig. 8 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. 9 muestra la operación de eliminación.

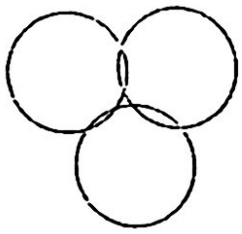


Fig. 8. Operación de interseccion

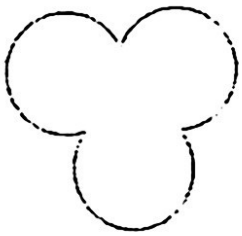


Fig. 9. Operación de eliminación a partir de la intersección

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. 10 ilustra el resultado de esta operación.

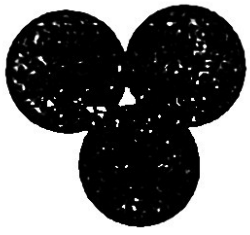
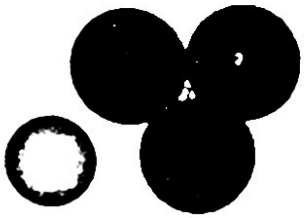


Fig. 10. Operación de disolución

El proceso de buffer da como resultado una nueva capa de datos de tipo polígono, los 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 en 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. 11 se muestra este proceso, donde 1 indica que el polígono está dentro de la zona de buffer y 0 indica lo contrario.



Puntos	Distancia
1	1
2	1
3	0

Fig. 11. Resultado del buffer final

4 Resultados

Haciendo una solicitud al servidor Web de mapas podemos consultar datos espaciales de acuerdo a los parámetros requeridos. Y utilizar la herramienta de buffer para aplicar *bufferizacion* en las capas de datos sobre las cuales trabajemos. Los datos espaciales de la Republica Mexicana fueron utilizados para realizar las pruebas correspondientes. A continuación se ilustran resultados de los análisis efectuados. La Fig. 12 muestra una capa espacial de México, donde se aprecian las carreteras del estado de Baja California y de Sonora.

Fig. 12. Carreteras en el estado de Baja California y Sonora

Sobre la capa de datos espaciales presentada en la figura 12 se aplica un análisis de buffer para determinar las áreas de afectación por ruido, el resultado se aprecia en la Fig. 13.

Fig. 13. Poblaciones (puntos en color rojo) que se verían afectadas por el ruido ante la cercanía de una carretera, el área buffer esta en color magenta

En la Fig. 13 se pueden apreciar las poblaciones que caen dentro del área de fluencia del buffer, estas puede ser identificadas fácilmente con ayuda de la erramientas de identificación o consultando la tabla para saber los datos relativos a ada una de ellas

Fig. 14. Mapa del la republica mexicana que muestra el radio de afectación (círculos verdes) pro la contaminación de refinerías (puntos en rojo)

En la Fig. 14 se muestra el radio de afectación de las refinerías situadas en Tampico, Salamanca, Tula y Minatitlán de la Republica Mexicana. Aquí se aplicó el análisis espacial de buffer para conocer el radio de influencia por contaminación de la refinería.

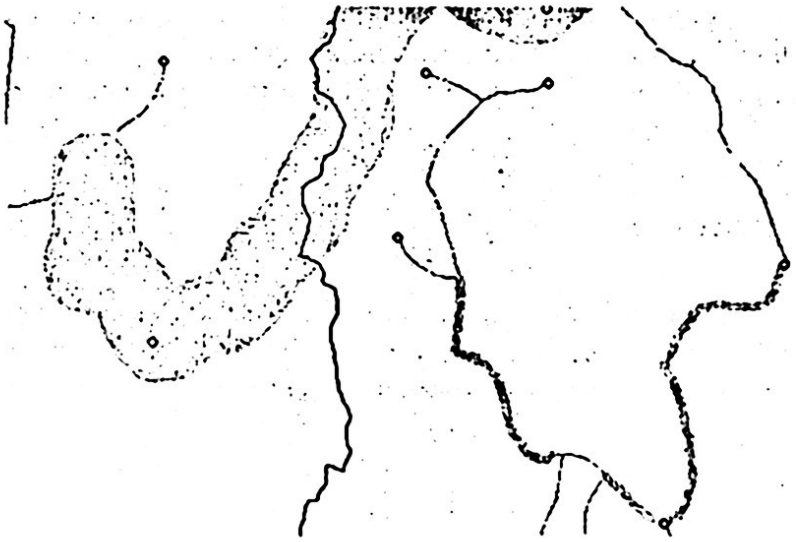


Fig. 15. Resultado del buffer (sobrepuesto en la capa original) donde se muestran las áreas de afectación (polígonos en color verde) de acuerdo a pesos

Otros resultados están enfocados a los segmentos de línea, los cuales pueden representar avenidas o carreteras. 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 carreteras. (Por ejemplo el promedio del número de autos por minuto). Cada línea es identificada por un número o color y la tabla (conocida como tabla de pesos) indicara los pesos mencionados, ver Fig. 15.

La Fig. 15 muestra el buffer aplicado a una capa de carreteras, donde cada carretera tiene asignado un peso específico, como se observa la carretera con un mayor peso, en consecuencia un mayor índice de tráfico, implica que el buffer genere un polígono de radio mayor mientras que en las de menor peso el radio es menor, y entonces también el área de afectación es menor.

5 Conclusiones

En el presente trabajo se ha presentado una aplicación en Web con capacidad de hacer análisis espacial, en particular análisis de proximidad y algunos de sus casos de uso. Utilizando la aplicación es posible analizar datos espaciales en formato shapefile o base de datos geográfica y los resultados obtenerlos en una nueva capa de datos, la cual puede ser obtenida en formato vectorial para posteriores análisis. Es asi, como la aplicación puede acceder, manipular y analizar datos en formato vectorial, raster y desde una base de datos geográfica. La especificación definida en la aplicación de buffers existe solo en herramientas SIG de escritorio y las comerciales, por lo cual este modelo tiene como ventaja que los datos pueden ser consultados y manejados vía un navegador Web de los mas comunes.

La facilidad de acceder a shapefiles o bases de datos geográficas, nos evita problemas de interoperabilidad, ya que la información puede ser convertida a otras especificaciones para su análisis.

El uso de un componente de tecnología Net da una mayor capacidad de respuesta para el usuario, y el navegador no necesita de requerimientos adicionales, y permite que las operaciones de accesos y recuperación de datos se realice de manera sencilla.

Agradecimientos

Los autores de este artículo agradecen al Instituto Politécnico Nacional y a la Centro de Investigación en Computación por el apoyo brindado a este trabajo; así como a los revisores de este trabajo por sus pertinentes comentarios.

Referencias

1. ArcInfo y Arcview, ESRI, Environment research institute , http://www.esri.com
2. Moreno M., "La Generalización Automática de la Información Geográfica Multiescala", Centro de Investigación en Computación, Instituto Politécnico Nacional, México, D.F. (2001) 123-145
3. Kyong-Ho K., Kiwon 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

4. 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
5. ArcInfo de ESRI, htpp://www.esri.com
6. Hong Y., "Exploring Spatial Analysis in GIS", OnWordPress, Thompson Learning, USA (1997) 89-117
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

Sistema clasificador de materiales basado en el reconocimiento de imágenes

Luis Topiltzin Domínguez Butrón, Luciano González Rodríguez,
Víctor Hugo García Ortega

Instituto Politécnico Nacional, Escuela Superior de Cómputo,
Av. Juan de Dios Batiz s/n, esq. Miguel Othón de Mendizábal,
Unidad Profesional "Adolfo López Mateos", 07738,
México, D.F.
topiltzin_jm@yahoo.com.mx, chanogr@yahoo.com.mx
vgarciaortega@yahoo.com.mx

Resumen. En el presente documento se describe el desarrollo del SCM (Sistema Clasificador de Materiales), basado en el procesamiento de imágenes y análisis espectral, para realizar la clasificación de un número especificado de materiales, contando para ello con una banda transportadora donde se tendrán los objetos y se realizará la clasificación. Así mismo contiene la propuesta de solución y el diseño de los módulos que componen el sistema.

1 Introducción

Cada vez es más común ver que las empresas que realizan algún tipo de proceso industrial, deseen automatizar la mayoría de los procesos, sino es que todos. Si hablamos de procesos que involucren el uso de una computadora podemos nombrar muchísimos, incluyendo los que suceden en la misma, esto es, existe un gran campo de aplicación y los procesos de clasificación no son la excepción. Si ahora hablamos de la clasificación de materiales, podríamos enfocarnos a un conjunto de objetos de características de interés para nosotros y aún más, podríamos también nombrar las diferentes formas que tiene el ser humano para reconocer un objeto, pero aún no sabemos a ciencia cierta que proceso sigue para distinguir un material de otro, es decir, la visión computacional aún se encuentra lejos de emular con precisión el sistema de visión del humano. En la actualidad se sigue trabajando sobre algoritmos existentes para mejorarlos u optimizarlos y así tener mejores resultados en el menor tiempo posible, así como también la combinación de herramientas para evaluar resultados y mejorar o proponer métodos alternativos. Enfocándonos al reconocimiento de materiales que es el tema medular, el ser humano toma en cuenta la forma de los objetos y su textura para asociarles a este un nombre o tipo, en este caso limitaremos el sistema solo a formas cuadradas y centraremos el análisis de datos a la textura de los objetos. Los materiales que identificaremos serán 3: Madera, Metal y Plástico. Cada uno con naturaleza distinta como es el color y la textura que serán los patrones a identificar y así poder diferenciar entre ellos.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.30-37.© Centro de Investigación en Computación, IPN, México

2 Descripción general del sistema

A grandes rasgos, el diagrama del sistema se puede apreciar en la figura 1.

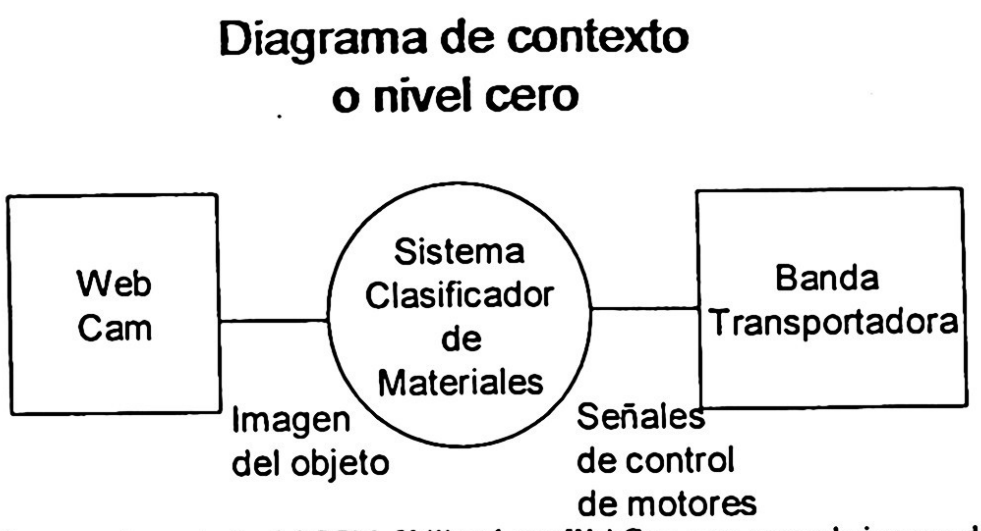


Fig. 1. Diagrama de contexto del SCM. Utilizará una WebCam para tomar la imagen del objeto, la cual se procesará y analizará en la PC, posteriormente la banda transportadora y las implementaciones necesarias realizarán la clasificación.

El objetivo entonces del trabajo es: Diseñar un sistema que constará de una banda transportadora que presentará objetos a una cámara, para que esta tome una fotografía, la envíe a una PC, la cual debe analizar e interpretar la imagen para determinar el material mostrado y así poder hacer la clasificación de objetos para aplicaciones como en una planta recicladora.

3 Descripción del sistema de reconocimiento

A continuación se describen los módulos que conforman la resolución del problema, así como los resultados obtenidos en cada uno de ellos, que serán utilizados a su vez por el siguiente modulo.

3.1 Captura de la imagen

Para la captura de la imagen se utilizó una WebCam Mitzu, con una velocidad de captura de 30 muestras por segundo, enfoque manual, arroja una imagen digital de 352x288 píxeles en formato bmp (mapa de bits) e interfaz USB.

El lente de la cámara es colocado a 9cm de altura sobre la banda transportadora para tener una toma superior del objeto, con iluminación controlada para crear un ambiente similar en cada toma.

3.2 Cambio a escala de grises

Existen razones para afirmar que no se necesita una imagen a color para realizar el procesamiento y análisis de la imagen, así pues, debemos cambiar a escala de grises la imagen que arroja la cámara, ya que esta es a color.

El proceso de cambio a escala de grises es simple, se toman los 3 bytes de color, se promedian y se asignan a como nuevos valores de esos mismos 3 bytes a cada píxel. A continuación se muestra una imagen a color entregada por la cámara y su imagen en escala de grises.

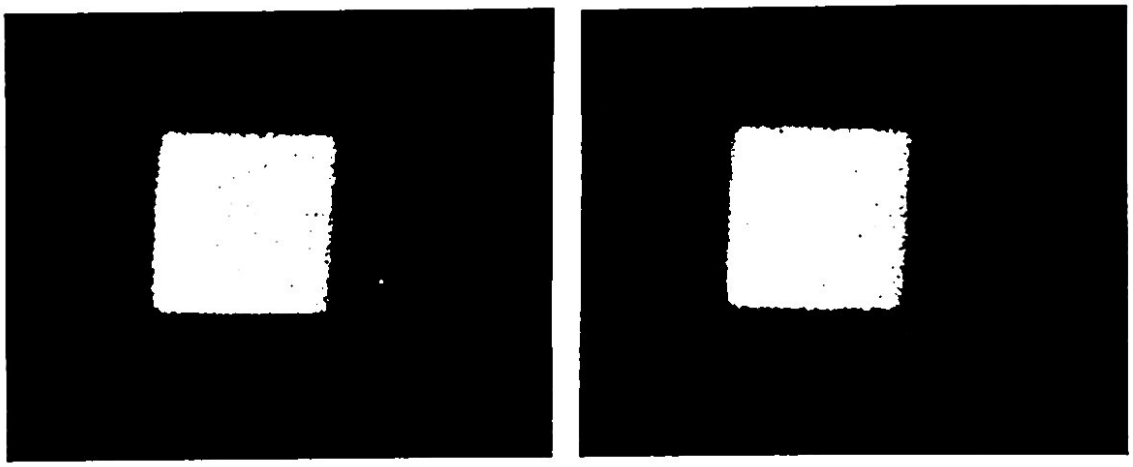


Fig. 2. Imagen original a color en formato de mapa de bits y su correspondiente imagen en escala de grises.

3.3 Filtrado de la imagen

El principal objetivo de las técnicas de mejora de una imagen digital es de forma que resulte más adecuada que la original para una aplicación específica.

Para efectos de este proyecto nos dedicaremos al filtrado en el dominio espacial ya que, el tipo de ruido presentado en nuestras imágenes que tenemos como prototipo nos dimos cuenta de que el tipo de ruido generado es el llamado *sal y pimienta* este tipo de ruido es fácilmente eliminado por las técnicas en el dominio espacial como lo es por ejemplo *el filtro por la mediana.*

El filtro por la mediana es un filtro no lineal, que preserva la agudeza de los bordes, es decir, reduce más el ruido que el difuminado de los bordes, elimina estrechos de picos de intensidad aislados. Podemos ver ahora la imagen filtrada en la figura 3:

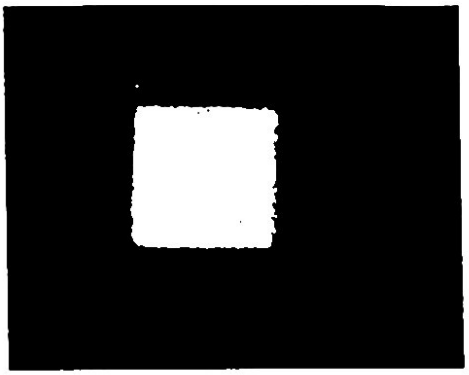


Fig. 3. Imagen filtrada mediante el filtro de la mediana.

3.4 Segmentación

El primer paso del análisis de imágenes consiste generalmente segmentar la imagen. La segmentación subdivide una imagen en sus partes constituyentes u objetos. El nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. Esto es, la segmentación deberá detenerse cuando los objetos de interés de una aplicación hayan sido aislados.

El objetivo de la segmentación por regiones es dividir una imagen en regiones. Se ha planteado el problema encontrando límites entre regiones basándose en discontinuidades de la intensidad, en las propiedades de distribución de los píxel, tales como la intensidad o el color. Al tratar de segmentar por regiones debemos ocupar una técnica que es la de umbralización, que define un parámetro de nivel de intensidad de gris para poder definir la posición del objeto y diferenciarlo del entorno. Esto es, si el nivel de gris del píxel está por debajo del umbral quiere decir que estamos analizando el entorno y para el caso en el que el nivel de gris del píxel esté por encima del umbral quiere decir que nos encontramos posicionados en el objeto. Esto considerando que tendremos un ambiente controlado de iluminación y de distancia de la cámara invariante.

Fig. 4. Distinción de la imagen segmentada.

Podemos notar la segmentación en la figura 4:

3.5 FFT

FFT (Fast Fourier Transform o Transformada Rápida de Fourier)

La principal ventaja de utilizar este tipo de transformada es el número de operaciones, que es considerablemente menor si los datos de entrada son mas grandes y potencias de 2, además para programar el algoritmo se utilizo el método de "*Decimation in time*". Otra ventaja es que la transformada bidimensional puede obtenerse en dos pasos aplicando sucesivamente la transformada unidimensional, esto lo podemos ver en las siguientes expresiones:

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)\exp\left\{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right\} \tag{1}$$

Así mismo la transformada inversa se define como:

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)\exp\left\{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right\} \tag{2}$$

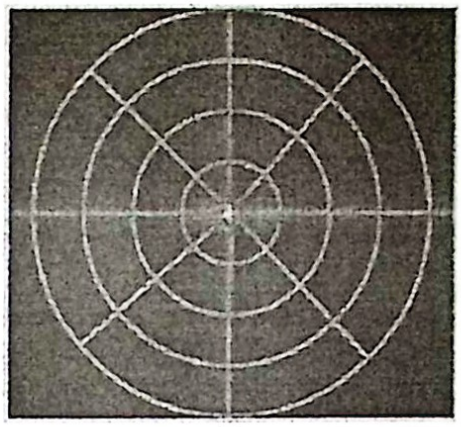
En la figura 5 se muestra la transformada de Fourier de la imagen original de la figura 1:

Fig. 5. Imagen original y su transformada de Fourier.

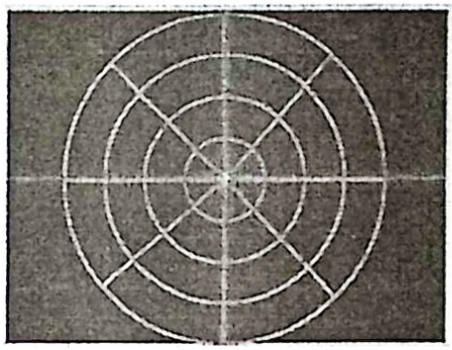
3.6 Reconocimiento del material

La extracción de los rasgos característicos del espectro derivado de la FFT se hará por medio de una división en regiones del espacio de frecuencias, que será de forma radial con 4 círculos y el plano se dividirá en 8 regiones en lugar de cuatro y el centro se dejará intacto para tener entonces 33 regiones. Las cuales serán componentes espectrales después de haber normalizado cada región por medio de la sumatoria de

todos los componentes por región, para tener una idea más clara de lo anterior observemos las comparaciones en la figura 6:



(a)



(b)

(c)

Fig. 6. Espectro de Fourier y regiones a utilizar para su reconocimiento. (a) Plástico, (b) Madera y (c) Metal.

3.7 Clasificación de materiales

Las redes neuronales son la técnica más adecuada para llevar a cabo el reconocimiento, al menos en nuestro caso, ya que no conocemos a ciencia cierta el comportamiento de nuestros datos y precisamente las redes neuronales resuelven problemas tanto linealmente separables como no linealmente separables, implementación relativamen-

te sencilla y de un diseño bastante flexible, así como tolerancia a fallos. En este trabajo utilizaremos un perceptrón multicapa.

En la figura 7 se muestra la arquitectura de la red neuronal a utilizar para hacer el reconocimiento.

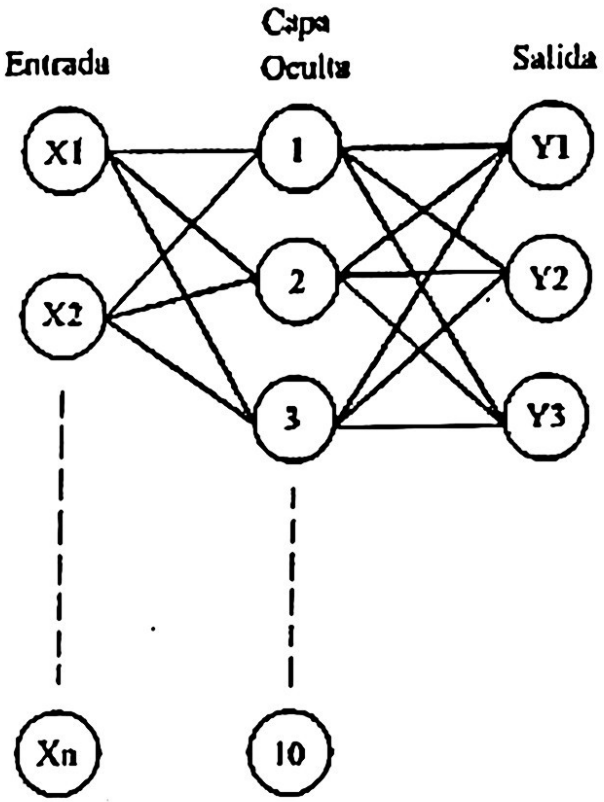


Fig. 7. Arquitectura de la red neuronal, en este caso un Perceptrón multicapa.

4 Descripción del sistema de clasificación

En este trabajo se utiliza una banda transportadora, en esta se presentan los objetos y cuando se detecta la presencia de alguno, se captura la imagen por medio de la cámara, posteriormente se realiza todo el análisis anteriormente nombrado y una vez que se determina el tipo de material del que se trata, se realiza la clasificación.

La clasificación se realiza con ayuda de un servomotor modelo HS-311 de Hitec, que trabaja de 4.8 a 6 V, se manejan para esto tres posiciones, debido a los tres materiales ya mencionados, estos materiales van al contenedor del tipo de material del que se trate; a su vez la banda transportadora cuenta con un motor de CD que trabaja a 12V y cuenta con un juego de engranes reductor de velocidad, para un mayor control. El control se realiza por medio del puerto paralelo de la PC.

5 Resultados y conclusiones

Existen numerosas técnicas de reconocimiento, y podría no ser una novedad la realización del trabajo, sin embargo, la aplicación a una planta recicladora donde los materiales a clasificar están ya definidos, tales como metales, cartón, vidrio, etc., es una buena propuesta debido a que se podría realizar el sistema que realice la clasificación y simplemente se tomen los contenedores dependiendo del material que se desee realizar el reciclado, debido a esto consideramos pues este trabajo como la primera parte de un proyecto que podría ser de mayor magnitud, importancia y ayuda para procesos industriales.

Existen así mismo restricciones para este trabajo, podemos mencionar por el momento que los materiales que se utilizaran serán de dimensiones conocidas e idénticas, deberán estar en buen estado, esto es no presentar rayones, cortes, pintura, etc., estas son algunas de las limitaciones del trabajo, pero debemos mencionar que se buscará dar seguimiento para poder tanto clasificar más materiales, como también presentar materiales de otras formas

El sistema de software tendrá una interfaz visual en C++ Builder y a la fecha se encuentra en un 80% de realización, a su vez el hardware del mismo sistema esta en construcción y podríamos considerar que se encuentra en un 70% realizado.

Referencias

[1] Phillip John Mc Kerrow "Introdution to robotics" Editorial Addison Wesley, 1993.
[2] Ming-Kuei Hu "Visual Pattern Recognition by Moments Invariants" IRE Trans. Inform. Theory, 1962.
[3] Escalera Hueso, Arturo de la "Visión por computador: Fundamentos y Métodos" España, Pearson Educación, 2001
[4] Inigo Madrigal, Rafael "Visión artificial por computador: Fundamentos, sistemas y aplicaciones en la industria y robotica" Madrid, Paraninfo, 1986.
[5] González Rafael, Woods Richard "Tratamiento Digital de Imágenes" Addison-Wesley E.U.A., 1996
[6] Mano, M. Morris "Diseño Digital" Trad. Julio Fournier González. México, 1987
[7] "Redes Neuronales Artificiales", José R. Hilera y Víctor J Martínez. 2000. Alfaomega. Madrid. España
[8] "Digital Neural Networks", S. Y. Kung, 1993. PTR Prentice Hall, Inc.

Mobile Nested Transactions Monitor based on Multi-Agent Systems: Workflow Layer

Jorge Martinez[1], Matias Alvarado[2]

[1] Artificial Intelligence Lab., CIC. National Technical University.
J. Batiz esq. O. De Mendizabal s/n., Mexico, DF, C.P. 07738
george@correo.cic.ipn.mx
[2] PIMAyC, Mexican Petroleum Institute
Eje Central Lazaro Cardenas 152. Mexico, D.F., C.P. 07730
matiasa@imp.mx

Abstract. Transaction Processing Monitors featuring Nested Transactions are in the core model for mission-critical applications. In this paper, a workflow layer for Mobile Nested Transactions is presented together with a multi-agent system implementation. The main advantage to introduce is a fault-tolerant mechanism. This is aimed to deal with both the loss of communication –very usual in mobile environments- and concurrent client access.

Introduction

The mobile computing paradigm has introduced new issues and challenges in data processing. Users are able to access their information with the help of mobile phones, personal digital assistants and portable computers. However these devices are prone to power outages, network disconnections and memory overflows. Then, control mechanisms are required in order to preserve data and information consistency. Transaction Processing is a Distributed Systems branch related to the study of data consistency. Beyond the traditional flat models –where objects are stored at the same host, the distributed approach for transaction processing encloses the scenario where there's more than one host.

The above-mentioned paradigms are distributed by definition. And so are Multi-Agent systems. Considering such shared condition, it is proposed in this paper a Transaction Processing Monitor based on a Multi-Agent system. It is also the objective of this work, to take a formal model to the ground of a real implementation. That is the case of the Logic of Interaction [1, 2], a BDI framework aimed to deal with concurrent actions interaction within a distributed agents group.

In the next section, the multi-agent and mobile computing paradigms are explained under an integrated scope. Section 3 presents a historical evolution of Transaction Processing ending at the Mobile Nested approach, for which a Monitor is under development. Section 4 describes the workflow layer for the Transaction Processing Monitor and a control mechanism for transactional behavior in agents. Conclusions and undergoing work are addressed in Section 5.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.38-47.© Centro de Investigación en Computación, IPN, México

Multi-agent framework for mobile computing

Multi-agent systems have arisen as an alternative approach for solving distributed problems. In fact, these kinds of systems are derived from research efforts in the field Distributed Artificial Intelligence. It was the development of message exchange mechanisms among smart-like computer systems, which pushed the idea for modelling each entity as an agent.

The basic features in the expected behaviour of an agent are: autonomy, that implies that the agent (to some extend) keeps control over its internal state and the way it behaves over an environment; social ability, the agent must be capable of interact and communicate with other entities during the problem-solving process; and learning, that is, evolve through its life-cycle and acquire both new knowledge and abilities. Additional features of agent and multi-agent systems include those related to reasoning, mobility and persistence. It is also expected that agents perform sensing operations over their environment as well as adaptability to deal with unexpected events. Agents are designed to reach their goals under conditions described for their information (knowledge/beliefs) [21].

Advances in wireless networking and portable information applications have introduced a new paradigm known as mobile computing. Users are no longer restricted to working at fixed sites. With mobile devices, such like phones, personal digital assistants and laptop computers, users can access their information despite of their geographical location. The multi-agent approach has been firmly related to dealing with problems that are distributed in nature and located at fast evolving environments. One example for these kinds of problems can be found in the field of so called mobile computing [21]. A mobile computing environment is shown in Figure 1; it is basically composed of: Fixed Hosts, Mobiles Hosts, Mobile Support Stations. These latter provide wireless network access to a limited scope range named Cell.

A Mobile User is that one who gets connected to the fixed network through his/her mobile host. It would be fairly desirable to maintain a connection alive while the user is on the road. The problems with a mobile environment arise when the device loses its connection by any reason. The most frequent are low batteries, physical location of the device (e.g. the user enters subway station) and links instability. From these issues, it becomes feasible to implement multi-agent systems over mobile groups, that is, one or more agents per device, which interact in order to fulfil a set of goals. The main advantage to achieve is the development of a fault-tolerant mechanism that deals with the lost of communication that may occur between two devices in a mobile environment. The underlying issues of combining these two technologies have been treated in [9] and [15]. This latter also mentions a mobile database approach like the one introduced later in the present work. Examples of multi-agent systems over mobile groups have been implemented for securing an electronic marketplace [10] and as schedulers for travellers [8].

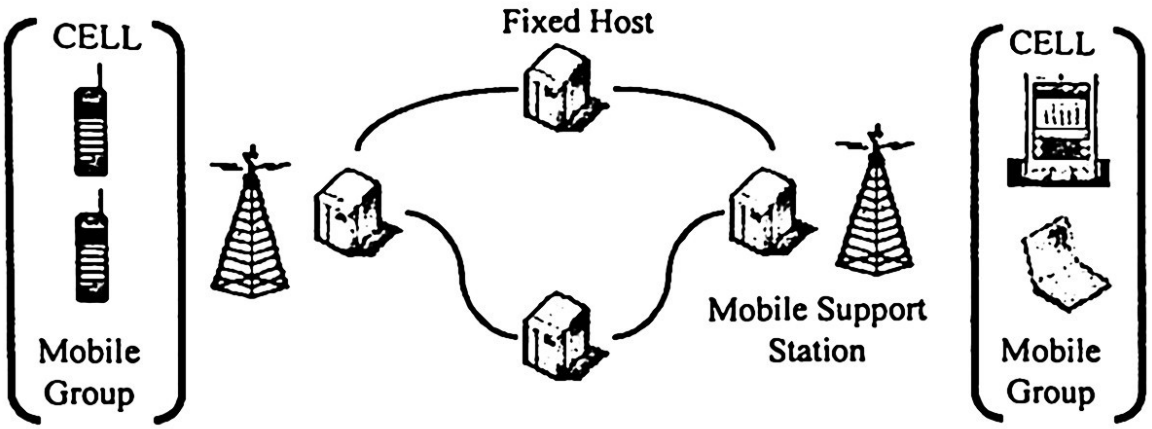


Fig. 1. Mobile Computing environment featuring two service Cells

3 Mobile nested transactions

The concept of Transaction can found in almost any type of business process currently modelled by object-oriented techniques. In [5], it is paid special attention to Transaction Processing as a hallmark for the next Client/Server technology generation.

A definition from [7] mentions that a transaction works as a mechanism for preserving consistency in the set of working objects. A common belief about the origin of transactions, establishes that they come from Database Management Systems theory. They were born from the user requirement for executing a set of operations over a database as a unit. The transaction idea was introduced in the theory of Distributed Systems under the form of Transactional File Servers [16]. There, the objective was to provide file access service to concurrent users over a network. File consistency required protection against multiple requests for both reading and writing.

In the present work, the goal is not only to control database access, but also to coordinate a mobile group that involves logical units and physical devices.

3.1 ACID properties

Transaction models are deeply based on the preservation of four basic properties. *Atomicity* stands for "all or none" and implies that the set of operations comprising a transaction must be executed as a whole. Although *consistency* is considered as a transaction property, it actually refers to data state from the database; however it is the responsibility for a transaction to take the database from a consistent state and to leave it in a similar one. *Isolation* means that a transaction must think of itself as the only one in execution at a given moment. *Durability* stands for data persistence; once a transaction has been committed, changes over database objects cannot be undone but by the execution of a second transaction.

Safekeeping the ACID properties, is the ultimate task to be achieved in transaction processing. However, recent works [11] in the concurrency control side propose the relaxation of isolation conditions. The benefit is an improvement in the number of

concurrent clients; they are allowed to access a shared resource without introducing further conflicts in the read/write operations.

3.2 Concurrency control and distributed transactions

A client is responsible of requesting the execution of operations to a server. The natural consequence of multiple clients requests, arises as a concurrency control issue over the objects to be used. It depends on whether a transaction needs to read or write an object that the server blocks the resource for shared or exclusive access respectively.

A flat transaction can be thought of a set of atomic operations over a group of database objects. These operations are organized under a partial order [14]. Three bracket operations mark the context of a transaction: *Begin, Commit, Abort*. These latter are invoked at the end in order to indicate whether the transaction execution succeeded or not.

Every request belonging to a flat transaction is executed over objects from the same server. In a distributed transaction, there are two or more servers holding the resources that will be read or modified by a transaction. Thus, there is necessary to have a Coordinator in charge of managing the success or failure result from the other participants. This is commonly achieved under the Two-Phase Commit Protocol (2PC) [19].

3.3 Nested transactions

The idea of Nested Transactions extends the distributed concept by allowing other transactions (known as children) to be born under the context of a parent transaction [7, 17, 18]. In this way, there will be a tree of nodes executing read/write operations and capable of spawning inner sub-transactions. Available computing resources are the unique limitation for the tree deepness; however, it is no longer required to complete all the operations in one place. Each child could be executed in different locations, that is, nested transactions are distributed. In [12, 13], it has been concluded that nested transactions feature special conditions under which it is not possible to comply with the four ACID properties. Except for the root node, children and leaf nodes fail ensuring *Durability*. The reason is that any changes over the database are actually performed until the root node commits or aborts. A transaction tree example is depicted in Figure 2.

Nested Transactions suit in mobile applications involving wireless or cellular connections for the following reasons. A compound transaction with nested children, offers better performance in concurrency and fault tolerance. Children could be executed in a parallel fashion and each one holds local responsibility for committing or aborting. However, none of the operations are actually reflected in the database content until the root transaction commits or aborts. Some other Nested Transactions features include the optionally of success for children nodes. According to this, a root transaction is able to commit even if some of its branches fail during execution.

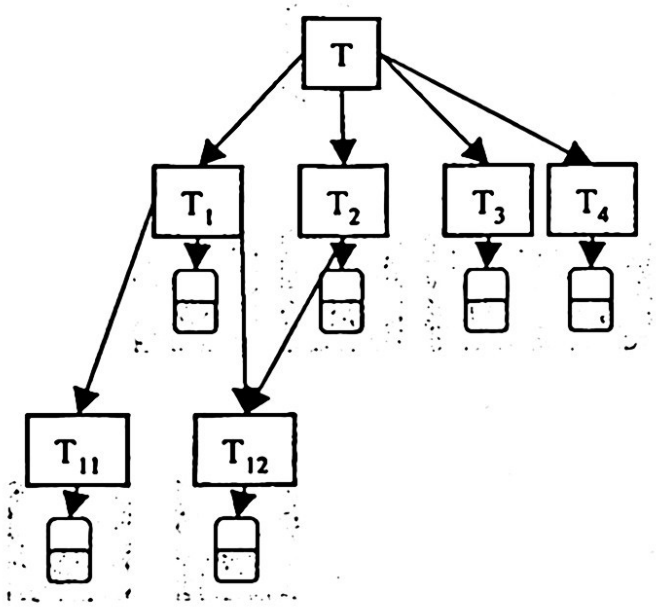


Fig. 2 Nested transaction. The Root node T spawns four children. T_1 spawns two more children resulting in four leaf nodes. It can be seen that T_{12} actually works as sub-transaction for both T_1 and T_2. T_3 and T_4 work at the same node

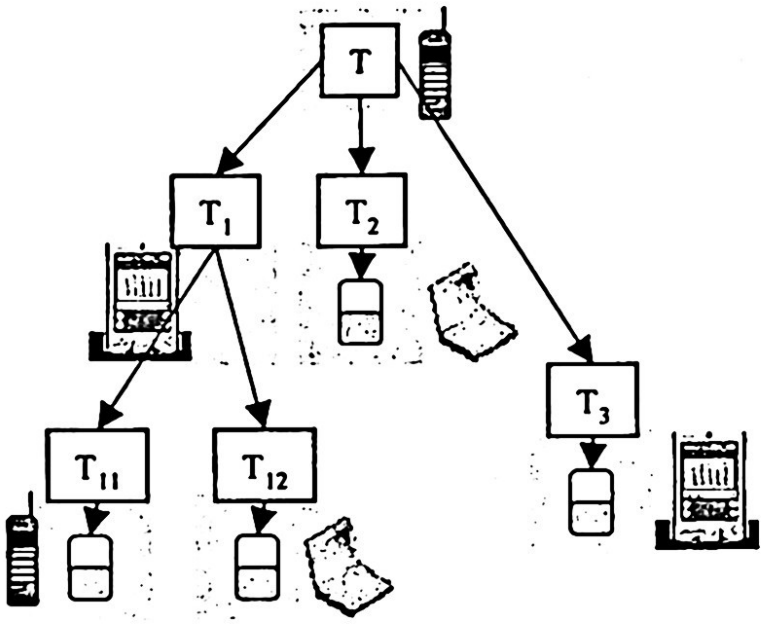


Fig. 3. Mobile Nested Transaction

3.4 Mobile nested transactions

In [12] the concept of *Mobile Nested Transactions* is introduced. These are the result of combining the concepts of Nested Transactions with Mobile Computing, particularly, a group of mobile hosts. There, a mobile device represents each node in the transaction tree. A noticeable difference between the general model and the above proposal is found in the read/write access. In [17], only leaf nodes are able to access database objects whereas the [14] definition enables intermediate nodes to execute such operations as well. In contrast with the general definition from 3.3, the next

remarks are considered in the Mobile Nested approach: child nodes have only one parent, read/write operations are actually done at the lowest (leaf) level and, each transaction node is located at one mobile host. This can be seen in Figure 3.

4 Mobile nested transactions monitor

4.1 Underlying formal model

Logic of Interaction [1], [2] was introduced as a formal model for multi-agent systems, aimed to deal with the balance agent's knowledge and actions. The central issue is that individual agent actions do interact. So, an action representation must make these interactions explicit and need to explicitly model different processes. The important aspect is the way in which agents perform the actions. In this sense, provides a model for synchronized, parallel, sequential and concurrent actions carried out by a single agent and by a group. Modal temporal logic formalism is used for action representation and modelling.

The key elements from the Logic of Interaction are related to Beliefs (pre-conditions), Goals (post-conditions) and Action execution itself. There has been defined a set of operators for interactions control. These are aimed to help in differentiating actions based on their space and time nature. Based on this, agents can perform: sequential, parallel, synchronized and concurrent actions. The full definitions concerning the Logic of Interaction are out of the scope of this paper.

In order to pave the way that connects a formal model with the actual implementation, only some elements from the logic of interaction have been translated to code. The system has been implemented in an open environment using JADE (Java Agent DEvelopment Framework) as the agent platform [6]. Agents use ontologies in order to represent internally pieces of information [3]. This is used in the exchanged messages and for inner control and operations.

4.2 Workflow layer

A Transaction Processing Monitor is conceived as the software in charge of managing simple requests from users that will be scaled over a distributed system [4]. It is also responsible for safekeeping the ACID properties during concurrent transactions execution. In a few words the monitor: receives some request, translates it into a system understandable language, triggers the transaction beginning, controls the commit or abort operations and finally, reports the result to the user.

According to [5], Transaction Processing Monitors featuring Nested Transactions as the core model for mission-critical applications will better reflect the business process nature. Three layers define the gross structure of Transaction Processing monitor:

Presentation. It receives instructions from the user and translates them into a system understandable language. These instructions are sent to the workflow layer. The result of the transaction is later presented to the user.

Workflow. As the name implies, this layer is responsible for routing, managing and answering the requests received from the presentation layer. Instructions are turned to the third layer and results are sent back to the first layer.

Database. The actual access to database objects is achieved at this level. The results, either successful or not, are informed to the previous layer. Currently, the implementation features an embedded database at each device: Pointbase Micro Edition [20]. Since the agents development framework is Java-based, it was required a compatible database tool that could run over mobile devices.

At this point, it has been implemented a basic workflow layer for the Mobile Nested Transactions Monitor. In particular, it has been implemented a Nested Two-Phase Commit. In this mechanism, a set of participants join the transaction started by some user at a root node. Descendant nodes may join each participant until a tree is completed. Once the tree is ready, each leaf informs its parent whether they failed or succeeded in executing their assigned set of operations. Eventually, all this information reaches the root node, then:

Phase 1. The root node asks all the successful nodes to get ready for commit. Since intermediate nodes may abort locally, only those descendants that are sons of successful nodes receive the *canCommit* request sent by the root node.

Phase 2. With the retrieved answers, the root node decides to commit or abort and informs the final decision to all of the involved nodes in the 2PC.

The above process is shown in Figure 4.

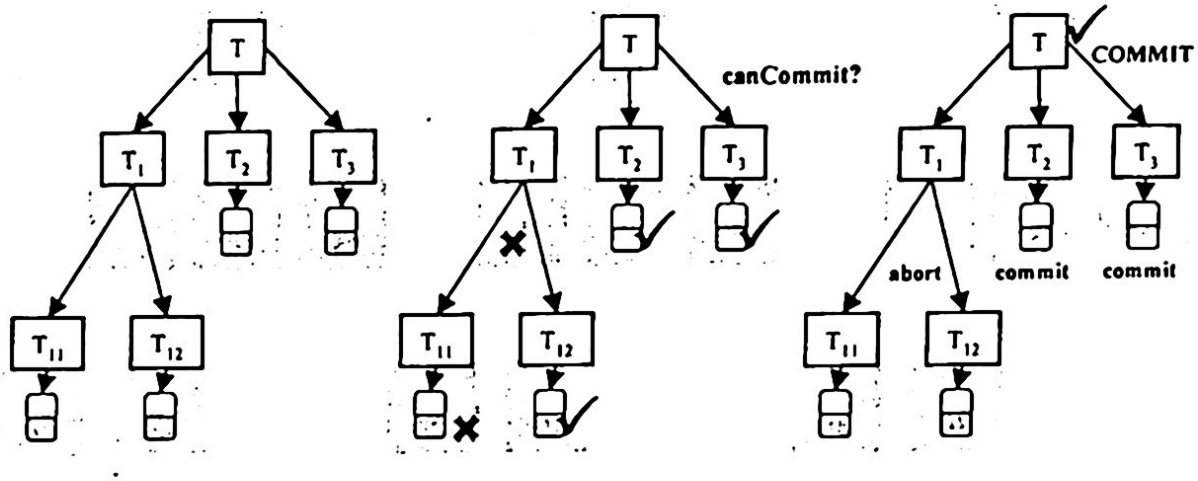


Fig. 4. Twb-Phase Commit protocol in a Nested Transaction. Since T_{11} and T_{12} do not receive a final commit or abort, they abort by default after a timeout

4.3 Control mechanism for transactional agents behaviour

During transaction processing, agents may find that some data objects are already in use due to concurrent access by other clients. In order to address this sort of events, a control mechanism has been proposed. This is an embedded behavior in each agent taking part in a given transaction. Agent's actions are classified into mandatory and

optional, strong and weak; the combination of these traits and the expected behavior can be found in Table 1.

Table 1. Actions classiffication supporting concurreny control in agents interaction and nested transactions processing. The description column shows what to do in case of action failure

Type	Sub-Type	Description
Mandatory	Strong	The transaction is aborted.
Mandatory	Weak	A new attempt is completed unless the user cancels.
Optional	Strong	It is ignored for this transaction but another transaction is scheduled for later attempts.
Optional	Weak	No more attempts are done.

The above classification helps an agent-node in the transaction tree in decision-making. This may happen while trying to access a data object that is already locked, or even when communication is lost with one of the agent's siblings. This is achieved by introducing alternative ways for the node to operate with a different set of descendants from the original one. In this way, an agent is not fully required to wait for an object to get unlocked. If the failed operation is not mandatory, a new attempt can be done, either in the same transaction or by scheduling a future transaction. There is introduced a notion of action *persistence* in the sense that the agent, will attempt to complete their assigned set of instructions –whenever it is possible. Previous user feedback is required in order to set the configuration parameters that will rule the agent's behavior for this mechanism.

As host crashes and breaks in communications are expected events in a mobile environment, the proposed mechanism works as a base for logging and recovery controls. On the other hand, better concurrency control is achieved through a Lock Manager [11, 7]. These two mechanisms are part of an undergoing effort and currently out of the scope of this paper.

5 Conclusions

The multi-agent systems approach is firmly devoted to problems that are distributed in nature and located at fast evolving environments, such as those found in the so called mobile computing. Under such context, users are able to access and process information with the help of mobile devices. However, these latter are prone to operation outages that may endanger data consistency.

The Mobile Nested Transaction concept was introduced as an alternative to deal with the ACID safekeeping challenge in the operation of a set of mobile hosts. There, sub-transactions are allowed to born inside the context of a parent transaction. Children nodes failures no longer imply that the root parent has to abort the complete transaction. A Transaction Processing Monitor is under development; it is based in a multi-agent system. In this paper, it was presented the Workflow layer featuring a Mobile Two-Phase Commit protocol.

The Logic of Interaction is used to rule the agent's action coordination. It is implemented in an open environment (JADE). A control mechanism was introduced

in order to achieve a transactional-like behaviour in the agent's participation within a Mobile Nested Transaction.

Pending work is heavily related to implement the other two layers from the monitor: Presentation and Database access. In this latter, a Lock Manager mechanism is also considered. It will help to improve concurrency control in shared data objects.

References

1. Alvarado, M., Sheremetov, L.: Modal Structure for Agents Interaction Based on Concurrent Actions. In V. Maik, J. Müller, M. Pchouek (eds.): Multi-Agent Systems and Applications III: 3rd International Central and Eastern European Conference on Multi-Agent Systems. Lecture Notes in Computer Science, Vol. 2691. Springer-Verlag, Berlin Heidelberg New York (2003) 29-39
2. Alvarado, M., Sheremetov, L., German, E., Alva, E.: Logic of Interaction for Multiagent Systems. In: C.A. Coello Coello, A. de Albornoz, L.E. Sucar, O.C. Battistutti (eds.): MICAI 2002: Advances in Artificial Intelligence: Second Mexican International Conference on Artificial Intelligence. Lecture Notes in Computer Science, Vol. 2313. Springer-Verlag, Berlin Heidelberg New York (2002) 378-396
3. Bellifemine, F., Poggi, A., Rimassi, G.: JADE: A FIPA-Compliant agent framework, Proc. Practical Applications of Intelligent Agents and Multi-Agents, April (1999), 97-108.
4. Bernstein, P.A., Newcomer, E.: Principles of Transaction Processing. Morgan Kaufmann Publishers Inc. (1997)
5. Byte Archive at http://www.byte.com/art/9504/sec11/art1.htm
6. Caire, G.: JADE Tutorial: Application-defined content languages and ontologies. TILab S.p.A. (2002)
7. Coulouris G., Dollimore J. , Kindberg T.: Distributed Systems. Addison Wesley (2002)
8. van Eijk, R.J., Ebben, P.W.G., Bargh, M.S.: Implementation of a scheduler agent system for traveling users. In proc. of Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices. Bologna (2002)
9. Finin, T., Joshi, A., Kagal, L., Ratsimor, O., Avancha, S., Korolev, V., Chen, H., Perich, F., Cost., S.: Intelligent Agents for Mobile and Embedded Devices. International Journal of Cooperative Information Systems (2002)
10. Fischer, K., Hutter, D., Klush, M., Stephan, W.: Towards secure mobile multiagent based electronic marketplace systems. Electronic Notes in Theoretical Computer Science. Vol. 63. Elsevier Science (2002)
11. Gama, L.A., Alvarado, M.: Concurrency control for Read-Only in Mobile Nested Transactions. In proc. of the 2nd Workshop on Intelligent Computing in the Petroleum Industry ICPI (2003)
12. Gama, L.A., Alvarado, M.: Mobile Nested Transactions for Nomadic Teams. In: Alvarado, M., Sheremetov, L., Cantu, F.: Special Issue on Intelligent Computing for Pretoleum Industry. Elsevier. (2003)
13. Gama, L.A., Alvarado, M.: Transacciones para Cómputo Móvil: presente y perspectiva futura. Revista Digital Universitaria, Vol. 3. No. 4. http://www.revista.unam.mx (2002)
14. Gray, J., Reuter A.: Transaction Processing: Concepts and Techniques. Morgan Kaufmann Publishers, Inc. (1993)
15. Loke, S.W.: Supporting Intelligent BDI Agents on Resource-Limited Mobile Devices - Issues and Challenges from a Mobile Database Perspective. In proc. of Workshop on Ubiquitous Agents on embedded, wearable, and mobile devices. Bologna (2002)
16. Mitchell, J.G., Dion J.: A Comparison of two network-based file servers. Comms. ACM, Vol. 25, No. 4. (1982) 233-45

17. Moss, J. E. B.: Nested Transactions: An Approach to Reliable Distributed Computing. MIT Press, Cambridge, MA (1985)
18. Nested Trans.at http://www.cs.panam.edu/~meng/Course/CS6334/Note/master/node89.html
19. Özsu, M.T., Valduriez, P.: Principles of Distributed Database Systems. 2nd Ed., Prentice-Hall, Inc.(1999) 381-401.
20. Pointbase Micro Developer's Guide at http://www.pointbase.com/support/docs/pbmicro.pdf
21. Wooldridge, M.: An Introduction to Multi-Agent Systems. John Wiley & Sons. England (2001)
21. Zaslavsky, A., Tahir Z: Mobile Computing: Overview and Current Status. Australian Computer Journal. Vol. 30. No. 2 (1998)

Ambiente ubicuo para el trabajo colaborativo de edición oportunista

Emilio Gutiérrez Arias, Rolando Menchaca Méndez

Centro de Investigación en Computación, Instituto Politécnico Nacional, México, D. F.
egutierrez@sagitario.cic.ipn.mx, rmen@cic.ipn.mx

Resumen. En el presente capitulo se presenta una arquitectura de un ambiente ubicuo para el trabajo colaborativo de edición entre un grupo de revisores y el propio autor del documento. De igual forma se intenta explotar el paradigma de mensajería instantánea para colaboración, donde pueden estar en contacto los integrantes de un equipo de trabajo por medio de una lista de contactos que los mantiene informados del estatus de cada participantes y de esa manera poder establecer una colaboración oportunista con los miembros activos. Así mismo se establece que los dispositivos móviles por sus características pueden ser una buena herramienta para realizar trabajo en grupo. La base de las comunicaciones serán los protocolos JXTA para redes Peer to Peer que pueden facilitar la interoperabilidad de los elementos de nuestra arquitectura.

1 Introducción

1.1 Trabajo colaborativo

El Área de trabajo Colaborativo asistido por Computadora (CSCW) es la guía de desarrollo de los sistemas de software multiusuario conocidos como *Groupware* [1,2]. Esta tecnología puede ser usada por ejemplo para facilitar la comunicación y el trabajo entre un grupo de personas, cooperar en el trabajo en grupo, coordinar las actividades del grupo de trabajo, resolver problemas en conjunto, optimizar la competencia de grupo o para facilitar las negociaciones [1].

Por su parte [3] define a *Groupware* como: un "Sistema de computadora que apoyan a grupos de personas involucradas en una tarea en común y que proporcionan una interfaz para el ambiente compartido".

1.2 Computación ubicua

Mark Weiser a quien se le considera el padre del Computo Ubicuo dice: "La tecnología más trascendente es aquella que pasa desapercibida"[4]. Es decir, lo importante es enfocarse sobre la tarea y sobre la información no sobre la herramienta. Aunque con suficiente práctica se puede lograr hacer que varias cosas aparentemente difíciles desaparezcan, pero una buena herramienta puede mejorar la invisibilidad.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la pp.48-55.© Centro de Investigación en Computación, IPN, México

A largo plazo las PC's y estaciones de trabajo decaerá por que el acceso a las computadoras estará por todas partes: por las paredes, en las manos, en los ojos, en cualquier lugar físico. Esto es a lo que se conoce como "Computación Ubicua" o "Ubicomp".

1.3 Mensajería instantánea para colaboración

Los sistemas de Mensajería Instantánea están basados en una comunicación sincronizada a través del Internet [5,6], estos pueden asistir a la comunicación uno a uno sobre el mismo sistema o comunicación uno a varios donde a una persona se le puede invitar a integrarse a una comunicación, sin embargo, la comunicación va a depender de las características de la red. Para la comunicación utilizan una interfaz de forma de ventana y un cuadro de diálogo por donde los participantes teclean los mensajes que son enviados hacia la otra persona, se parece a un sistema de comunicación por teléfono que se basa en un modelo de llamadas en forma dinámica [5], los mensajeros instantáneos soportan lo que se le llama un grupo chat, donde cada participante puede entrar a un salón de chat donde se pueden encontrar a varias personas con quienes entablar una conversación.

Las personas disponibles con las que se puede interactuar se organizan en forma de lista, la lista puede ser organizada dependiendo del ámbito o por la relación que existe, esta puede ser organizada por ejemplo: Por amigos, compañeros de trabajo ó familiares con los que en dado momento se puede establecer una interacción, esto es posible, por que en la lista se despliega el estado de las personas que la integran. Los sistemas más populares son por ejemplo AOL [7], Yahoo messenger [8], MSN messenger [9], ICQ [10] y Jabber [11]. Aunque existen varios en el mercado que no son tan populares. Es preciso mencionar que estos sistemas además de servir como un medio para enviar mensajes instantáneos a las personas disponibles, también es posible enviar mensajes en forma de correo electrónico a las personas que no se encuentran en ese momento en línea, con esto se busca que de cualquier forma se pueda estar en contacto en cualquier momento.

1.4 Redes "Peer To Peer"

También recientemente, se ha profundizado en el estudio de los sistemas distribuidos denominados de igual-a-igual (P2P) cuya principal característica es que están compuestos por elementos de software con funcionalidad equivalente y que no requieren de ningún tipo control centralizado. Estas propiedades dotan a los sistemas de igual-a-igual con características como la escalabilidad [12], flexibilidad, tolerancia a fallas [13], simplicidad en la administración, que pueden ser sumamente útiles en el diseño de una infraestructura para el desarrollo de sistemas de cómputo ubicuo.

El resto del articulo esta dividido de la siguiente forma: después de haber dado la introducción de los paradigmas y tecnologías en la sección 2 presentamos un estado del arte de los sistemas parecidos al nuestro, después en la sección 3 se pasará a la descripción del prototipo ECO y por último en la sección 4 terminaremos con algunas conclusiones.

2 Sistemas de trabajo colaborativo de edición

En la actualidad existen varios sistemas que contemplan algunas de las características de nuestro sistema **ECO** (Edición Colaborativa Oportunista) de los cuales a continuación se describen sus características mas importantes.

El sistema COARSY es un sistema de edición y revisión colaborativo [14]. Este sistema tiene una arquitectura Cliente-Servidor. El cliente es la aplicación que el usuario usa para revisar y modificar un documento. Cada cliente trabaja independiente de los demás y no hay forma de comunicarse con los demás clientes.

Este sistema usa dos servidores: un servidor de base de datos y un servidor de aplicación. El servidor de datos introduce y extrae información de la base de datos. El servidor de aplicación es usado para comunicarse con el servidor de la base de datos y así obtener o proporcionar la información requerida y generada por el cliente. El servidor de aplicación recibe mensajes del cliente y determina si la información es para algún documento o es información para la base de datos.

Entonces una vez dentro del sistema cada participante puede revisar o editar algún documento. El documento y las contribuciones son desplegados en una interfaz, esta interfaz se divide de tal forma que en una ventana se despliegue el documento original y en la otra las contribuciones hechas, a su vez esta será visualizada por cada participante. Se esta pensando en desarrollar un cliente COARSY sobre dispositivos móviles.

Otro proyecto es el editor PREP "trabajo en preparación" [15] tiene tres consideraciones importantes (1) Asistencia en la interacción social entre autores y los comentarios, (2) Asistencia a aspectos cognoscitivos de los autores y comentarios externos y (3) Asistencia a prácticamente ambos tipo de interacción.

La idea del proyecto es desarrollar un editor en un entorno multiusuario para asistir a una gran variedad de trabajo colaborativo, y en particular, entre autores y la comunicación de estos. No se enfoca en una interacción entre participantes en un mismo tiempo (es asíncrono). Mas bien la preocupación esta en mejorar la participación colaborativa sobre una red de trabajo.

El editor PREP aborda las ideas de trabajo colaborativo enfatizando la comunicación, planeación y anotaciones organizadas. El enfoque central del editor PREP es proporcionar una útil representación visual de la información. Este editor intenta resolver las necesidades de una herramienta de dibujo durante el proceso de planeación cuando los conceptos están empezado hacer formulados.

Para facilitar la inteligibilidad mutua, el editor PREP proporciona convenciones del espacio de trabajo para las comunicaciones.

En un sistema CSCW los sistemas son usados por múltiples grupos traslapados esto incrementa la cantidad de información y las modificaciones de la información son muy constantes. Por lo tanto se necesitan mecanismos para filtrar la información y presentar la información minimizada de los datos compartidos así como actualizada.

Un sistema de mensajería instantánea para el trabajo de revisión de documentos en forma colaborativa es AIDA [16] este sistema esta pensado en dispositivos móviles mas específicamente para PDA's (Asistente Personal Digital) donde se conoce de la presencia y el estatus de los usuarios.

AIDA tiene una interfaz que despliega el nombre de la persona así como su estatus. En esta interfaz se presenta un cuadro de diálogo que es por donde se pueden teclear

los mensajes que serán enviados a las demás personas. AIDA implementa el API DOC2U [17] en donde se extiende el concepto de asimilitud de presencia asociada al usuario, esta también es una aplicación de mensajería instantánea donde se comparten recursos, ya que esta pensada para asistir en la coordinación de las actividades de edición colaborativa introduciendo el concepto de presencia de documentos. AIDA usa un servidor web como un repositorio de documentos y el servidor de mensajería instantánea Jabber para conocer la presencia de los participantes así como la presencia de documentos compartidos en el repositorio web.

De ahí que AIDA gracias al enfoque de mensajería instantánea, ofrece nuevas oportunidades de colaboración por medio de encuentros casuales o improvisados en una comunidad de autores y revisores de documentos.

3 Arquitectura del prototipo ECO

Como ya se menciono anteriormente este prototipo esta pensado para el trabajo de revisión de un documento entre los revisores y el autor del documento, los cuales forman un grupo de trabajo. Como estas personas necesitan estar en constante comunicación se implementa un cliente Jabber, el cual es un protocolo que nos proporciona el conocimiento de presencia por parte de los colaboradores, de ahí que podamos saber cuando alguien esta en línea o cuando no, cuando alguien esta en línea (disponible) esto se vuelve una oportunidad para entablar una comunicación o colaboración con ese integrante.

Gracias al enfoque Peer to Peer podemos formar una infraestructura de red virtual donde los elementos principales van hacer los peer que representan al autor y a los revisores, los peers que representan el documento y el repositorio, los peers Relay que servirán para entablar la comunicación con lo demás peers y por ultimo los peers Rendezvous para encontrar los documentos y los participantes.

La arquitectura que se presenta en la figura 1.

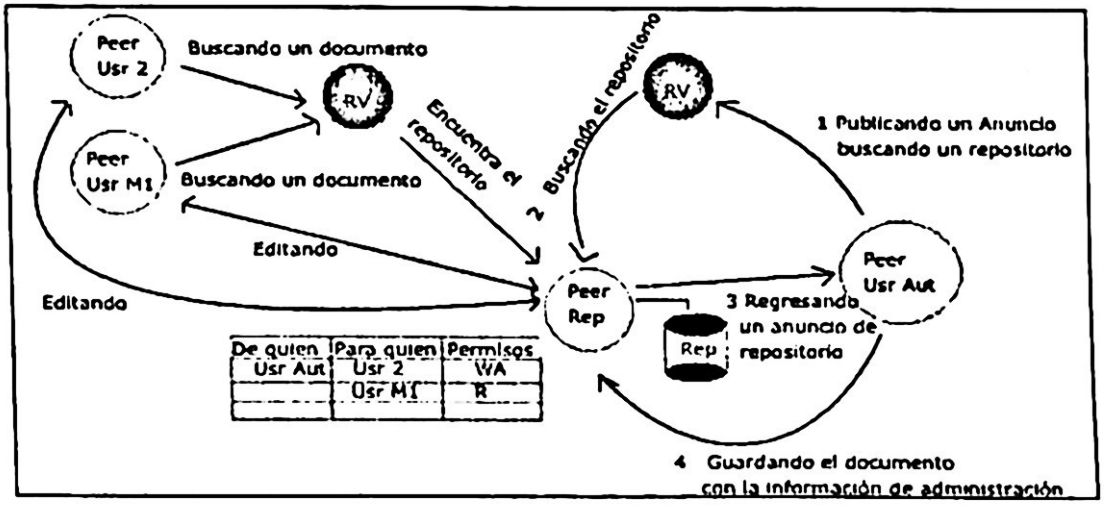


Fig. 1. Se presenta la arquitectura del sistema ECO en un diagrama donde se enumeran los estados.

- En este diagrama en el primer paso el Autor publica el anuncio a un Rendezvous para buscar quien implementa un servicio de almacenamiento de documentos.
- En el segundo paso el Peer Rendezvous se encarga de buscar el peer quien implementa el repositorio para pasarle la solicitud del *Peer Usr Aut.*
- En el Tercer paso una vez que el Peer-Rep conoce el anuncio del *Usr Aut* le regresa su anuncio indicando que ya vio la solicitud y se establece una relación entre *Repositorio* ↔ *Usr Aut.*
- En el Cuarto paso entonces el *Usr-Aut* le pasa el documento al repositorio para que este lo almacene y lo administre. Al pasar documento este contiene información para la administración del mismo como permisos y disponibilidad.
- Como se puede ver en la Figura 1. El Peer que implementa el repositorio contiene una tabla de administración con tres columnas, en la primer columna contiene el propietario del documento, en la segunda los elementos relacionados con el documento y en la tercera los permisos que se tiene hacia el documento.
- En un quinto paso los peers buscan al documento preguntando a su Peer Rendezvous, este los contacta con el Peer que implementa el repositorio y se establece una interacción entre revisores y el documento.

ECO presenta una interfaz en forma de mensajería instantánea que indica el estatus de los elementos que están involucrados en la edición del documento. Así como el estatus de otros contactos proporcionado por las características de Jabber sin que tengan que ver con el documento. La figura 2 muestra la interfaz en una estación de trabajo y en un dispositivo móvil en este caso una PDA, El llevar el sistema hasta una PDA hace que los participantes puedan interactuar con el sistema independientemente del lugar físico haciendo mas flexible la colaboración.

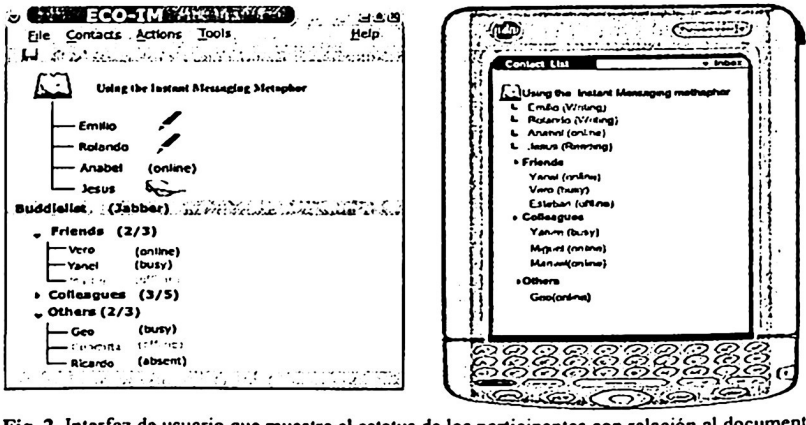


Fig. 2. Interfaz de usuario que muestra el estatus de los participantes con relación al documento

La parte inferior muestra el estatus de otros contactos, pero que no tienen nada que ver con el documento, solo sirve para ilustrar que ECO puede a parte de presentar una interfaz de colaboración puede hacer el papel de cualquier mensajero instantáneo comercial. Para el proceso de revisión y edición se tienen que establecer permisos y prioridades para los usuarios como se menciono en la arquitectura, de tal forma que pueda haber un control del documento, el autor del documento es el responsable de establecer estos permisos, él decide a quienes les da mas prioridad para el caso en que dos revisores quisieran hacer modificaciones al mismo tiempo. El sistema ECO esta pensado para que todos puedan leer el documento una vez que esta disponible, pero para el proceso de escritura hay quien va a tener mas prioridad. La figura 3 muestra la pantalla del autor en donde él puede establecer las prioridades de los participantes y la pantalla de un revisor con un permiso asignado.

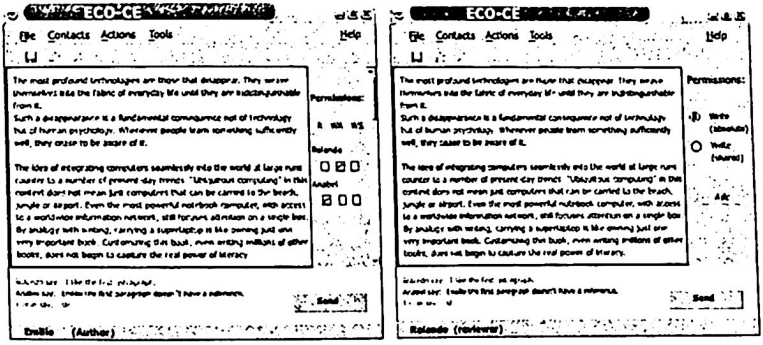


Fig. 3. La interfaz de la izquierda es del autor que muestra la forma de poder asignar permisos sobre el documento y la interfaz de la derecha muestra que el revisor Rolando tiene asignado el proceso de revisor absoluto lo cual indica que él es el único que puede hacer modificaciones sobre el documento y nadie mas tiene ese permiso.

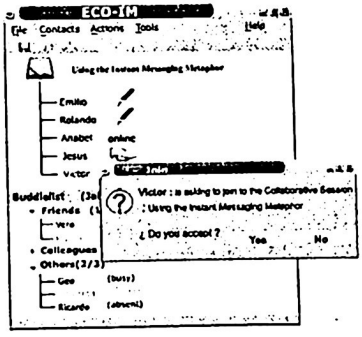


Fig. 4. Interfaz que presenta la solicitud de colaboración de un usuario que acaba de conectarse.

Entonces los revisores que inicien una sección podrán ver quienes se encuentran en esos momentos editando el documento y pueden encontrar una oportunidad para editar colaborativamente el documento (ver figura 4), por lo tanto puede solicitar participar en el proceso de revisión colaborativa. O gracias a la interfaz que también presenta una ventana de conversación escrita el autor puede unir a los revisores en una revisión múltiple.

Con respecto al documento, éste se encuentra almacenado en el repositorio que es donde el autor lo almacenó para así ponerlo a disposición de los revisores, cabe mencionar que pueden haber uno o varios repositorios, en si a los revisores no les interesa donde se encuentra almacenado a ellos les vasta con saber que existe y que esta disponible para ellos. Entonces el autor es el encargado de definir quienes serán las personas que tendrán acceso a ese documento.

4 Conclusiones

Como se puede ver en este trabajo se introducen varios conceptos y tecnologías que fundamentan su participación. Entonces se forma ECO un sistema el cual facilita el trabajo de revisión de documentos en un grupo de trabajo de revisores y autores, la principal aportación en este sistema es la introducción de los protocolos JXTA para formar una Red Peer to Peer para trabajo colaborativo. Por que en el momento en que se forma una red virtual se le asignan a cada uno de los elementos de la red su rol y se hace una distribución del trabajo sin caer en la computación centralizada como los demás sistemas que tienen los mismos objetivos. De ahí que formemos nuestro ambiente ubicuo por que nunca se sabe exactamente donde se encuentra el documento y no es de gran importancia saberlo, lo importante es conocer que existe un documento y que podemos entrar al documento en cualquier lugar en que nos encontremos y en cualquier tiempo, desde una estación de trabajo estática o desde un dispositivo móvil.

Referencias

[1] Baecker, R.M.(1993). Reading in Groupware and Computer Supported Cooperative Work. Morgan Kaufmann Publisher, San Mateo. CA, pp 873.

[2] Grudin, J. (1994). "Computer-Supported Cooperative Work: History and Focus". IEEE Computer, 27(5), pp 19-26.

[3] van Leeuwen, J. (ed.): Computer Science Today. Recent Trends and Developments. Lecture Notes in Computer Science, Vol. 1000. Springer-Verlag, Berlin Heidelberg New York (1995)

[4] Mark Weiser. "The world is not a desktop". Interactions; January 1994; pp. 7-8.

[5] Grinter, R. & Palen, L. (2002). *Instant Messaging in Teen Life*, Proc. CSCW '02,New Orleans, LA.

[6] Nardi, B., Whittaker, S. and E. Bradner. "Interaction and Outeraction: Instant Messaging in Action." Proceedings of CSCW 2000. (2000), ACM Press, 79-88.

[7] AOL Messenger http://www.aim.com/

[8] Yahoo messenger http://messenger.yahoo.com/

[9] MSN messenger http://messenger.msn.com/
[10] ICQ messenger http://web.icq.com/
[11] Jabber http://www.jabber.org/
[12] Bala Balakrishnan, H. et al. Looking Up Data in P2P Systems. Communications of the ACM, Vol. 46, No. 2, February 2003.
[13] Kubiato Kubiatowics, J. Extracting Guarantees from Chaos. Communications of the ACM, Vol. 46, No. 2, February 2003.
[14] Favela, J., & Ruiz, D. (2001). Collaborative Authoring and Reviewing over the Internet. WebNet Journal: Internet Technologies, Applications & Issues 3(3), 26-34.
[15] Neuwirth, C., Kaufer, D., Chandhok, R. And Morris, J. (1990). Issues in the design of computer support for co-authoring and commenting. Proceedings of the third conference on CSCW'90. Baltimore, MD : ACM Press, pp. 183-195.
[16] Favela, J., C. Navarro, M. Rodriguez, "Extending Instant Messaging to Support Spontaneous Interactions in Ad-hoc Networks," in *Proceedings of ACM 2002 Conference on Computer Supported Cooperative Work* (CSCW '02), New Orleans, Louisiana, 2002.
[17] Moran, L., Favela, J., Martinez, A., Decouchant, D.: Document Presence Notification Services for Collaborative Writing. Proc. of CRIWG'01 (2001) 125-133

Descubrimiento dinámico de servicios Web en nodos UDDI mediante USML

Giner Alor Hernández , César Sandoval Hernández, José Oscar Olmedo Aguirre

Centro de Investigación y de Estudios Avanzados del IPN.
Departamento de Ingeniería Eléctrica. Sección de Computación,
Av. Instituto Politécnico Nacional 2508, Col San Pedro Zacatenco. 07360,
México, D. F.
{gineralor, csandoval}@computacion.cs.cinvestav.mx,
oolmedo@delta.cs.cinvestav.mx

Resumen. Para la publicación de servicios Web, organizaciones, consorcios y grandes empresas han desarrollado el *Universal Description, Discovery and Integration* (UDDI), un servicio de búsqueda y recuperación de información sobre servicios Web. Desafortunadamente, las consultas actuales en nodos UDDI se enfocan en un solo criterio de búsqueda como el nombre de negocio, situación comercial, categorías comerciales o tipos de servicio. Para resolver esta problemática, en este trabajo se desarrolló un portal de servicios Web que permite el descubrimiento dinámico de servicios Web mediante la utilización de *UDDI Search Markup Language* (USML), un lenguaje basado en XML desarrollado por IBM, que permite realizar consultas en diversos nodos UDDI utilizando múltiples criterios de búsqueda. Además, se describen las características principales del lenguaje USML para la creación de consultas en diversos nodos UDDI. Finalmente, se propone un caso de estudio en comercio electrónico donde se describe la funcionalidad del portal desarrollado en la búsqueda y localización de servicio Web.

1 Introducción

El descubrimiento dinámico de los servicios Web se basa principalmente en encontrar los servicios que ofrecen las organizaciones u empresas para satisfacer ciertas necesidades en base a un criterio de búsqueda dado por el usuario. Para esto, es necesario utilizar un registro distribuido, conocido como *Universal Description, Discovery and Integration* (UDDI) [1], en donde se encuentran publicadas las descripciones de los servicios Web. En un nodo UDDI solo se pueden localizar negocios cuyas identidades son muy conocidas, con el fin de que los usuarios puedan averiguar qué servicios Web se están ofreciendo y cómo integrarse con ellos. Además, las consultas actuales en nodos UDDI se enfocan en un solo criterio de búsqueda como el nombre de negocio, situación comercial, categorías comerciales, tipo de servicio e identificador comercial. Mas aun, estos tipos de consultas son muy imprácticas cuando la población de servicios Web publicados en un nodo UDDI es muy grande debido a que el resultado de una consulta podría devolver miles de negocios que satisfacen un criterio particular, por lo cual lo hace inmanejable. Entonces, aquí surgen dos grandes problemáticas de localización de servicios Web en

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.56-67.© Centro de Investigación en Computación, IPN, México

nodos UDDI: 1) búsqueda eficaz de los servicios Web apropiados, 2) búsqueda exacta de los servicios Web.

Para resolver estas dos problemáticas, IBM desarrolló *UDDI Search Markup Language* (USML), un lenguaje basado en XML que permite realizar consultas en nodos UDDI utilizando múltiples criterios de búsqueda. Sin embargo, las características provistas por el lenguaje USML todavía no se encuentran implementadas en ningún nodo UDDI. Como una solución a lo anterior, hemos desarrollado un portal de servicios Web que permite el descubrimiento dinámico de servicios Web publicados en nodos UDDI mediante el lenguaje USML. Así también, proponemos un caso de estudio situado en el contexto del comercio electrónico B2B, donde se ejemplifica la funcionalidad del portal de servicios Web desarrollado. Además, realizamos un análisis comparativo del portal con los diversos nodos UDDI, con el fin de mostrar las ventajas y desventajas del uso de USML. El caso de estudio y la funcionalidad del portal de servicios Web desarrollado se describe en la sección 4. El análisis comparativo se describe en la sección 5.

2 UDDI Search Markup Language

UDDI Search Markup Language (USML) es un lenguaje basado en XML que permite realizar la búsqueda y localización de servicios Web con múltiples criterios en diversos nodos UDDI. Según la especificación de UDDI [1], hay tres tipos de datos principales que pueden consultarse en un nodo UDDI: Negocio, Servicio, y Tipo de Servicio (*tModel, Technology Model*). Un *tModel* define los tipos de servicios que utiliza el servicio Web, al tiempo que incluye las definiciones de operación, la estructura y protocolos de los mensajes, y protocolos de seguridad de dichos servicios. Partiendo de un *tModel* concreto, se puede saber qué operaciones de servicio Web realiza la entidad que implementa dicho *tModel* y cómo obtener acceso a ellas. Además, en un *tModel* se especifica información como: nombre del *tModel*, nombre de la organización que publicó el *tModel*,. una lista de categorías que describen el tipo de servicio. En USML, estos tres tipos de datos pueden consultarse de forma conjunta o independiente. El esquema general de un documento USML se muestra en la figura 1.

```
<Query>
    <Source/>
    <SourceURL/>
    <BusinessName/>
    <Identifier/>
    <Category/>
    <ServiceName/>
    <ServiceTypeName/>
    <DiscoveryURL/>
    <FindBy/>
    <AggOperator/>
    <RequestTypeName/>
</Query>
```

Fig. 1. Esquema general de un documento en USML

Donde:

- Query: especifica las condiciones de búsqueda. Combina la búsqueda de la palabra clave, la búsqueda basada en los identificadores, y la búsqueda basada en las categorías.
- Source: es el nodo UDDI donde se realiza la búsqueda. Puede ser un nodo público o privado.
- SourceURL: es el URL del nodo UDDI donde se realiza la búsqueda.
- BusinessName: es el nombre del negocio a buscar.
- Identifier: es el nombre del identificador y su valor asociado. Se aceptan dos tipos de identificadores: *Data Universal Numbering System* (D-U-N-S) [2] y Thomas Register [3].
- Category: es el nombre de la categoría y su valor asociado. Actualmente, se aceptan cinco tipos de categorías: *North American Industry Classification System* (NAICS) [4], *United Nations Standard Products and Services Code* (UNSPSC) [5], GEO [6], UDDITYPE, y *Standard Industrial Classification* (SIC) [7].
- ServiceName: es el nombre del servicio. Se usa cuando la búsqueda es por el nombre de servicio.
- ServiceTypeName: es el nombre del tipo de servicio (*tModel*). Se usa cuando la búsqueda es por el tipo de servicio.
- DiscoveryURL: es el URL para el descubrimiento.
- FindBy: especifica el tipo de datos a buscar (Negocio, Servicio o Tipo de Servicio).
- AggOperator: especifica el tipo de operador lógico a utilizar en la búsqueda. Se soportan dos tipos de operadores lógicos: AND y OR. Si se utiliza el operador OR, la información especificada en la etiqueta *FindBy* de cada búsqueda es devuelta. En caso de utilizar el operador AND, solo información relacionada al tipo de datos especificado en la etiqueta *RequestTypeName* será devuelta. Más de 3 búsquedas (una búsqueda para cada uno de los tipos de datos) no se permite.
- RequestTypeName: especifica el tipo de datos que se desean como resultado en la consulta (Negocio, Servicio o Tipo de Servicio).

Un ejemplo de una solicitud de consulta en un nodo UDDI en USML se muestra en la figura 2.

```
<?xml version="1.0"?>
<!DOCTYPE Search SYSTEM "UDDISearch.dtd">
<Search>
   <Query>
   <Source>Public UDDI</Source>
   <SourceURL>
   http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
   </SourceURL>
   <BusinessName>IBM</BusinessName>
   <FindBy>Business</FindBy>
   </Query>
   <Query>
```

```
    <Source>Public UDDI</Source>
    <SourceURL>
    http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
    </SourceURL>
    <ServiceTypeName>UDDI</ServiceTypeName>
    <FindBy>ServiceType</FindBy>
    </Query>
    <Query>
    <Source>Public UDDI</Source>
    <SourceURL>
    http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
    </SourceURL>
    <BusinessName>IBM</BusinessName>
    <ServiceName>UDDI</ServiceName>
    <FindBy>Service</FindBy>
    </Query>
    <AggOperator>AND</AggOperator>
    <RequestTypeName>Business</RequestTypeName>
</Search>
```

Fig. 2 Estructura de una solicitud de consulta en USML

En el documento USML de la figura 2, se realiza una solicitud a una consulta en donde se establecen tres criterios de búsqueda utilizando el operador AND. En esta solicitud, los tres criterios de búsqueda corresponden a los tres tipos de datos principales en un nodo UDDI:

1. Negocios (situado en la etiqueta *BusinessName* cuyo contenido es "IBM");
2. Tipos de servicios (situado en la etiqueta *ServiceTypeName* y cuyo contenido es "UDDI");
3. Servicios comerciales (situado en la etiqueta *ServiceName* y cuyo contenido es "UDDI").

Así, utilizando USML para realizar consultas a diversos nodos UDDI, podemos establecer múltiples criterios de búsqueda como son: negocios, servicios y tipos de servicio. En este caso, en el documento de la Fig. 2, se especifica el nodo UDDI y el URL asociado (situados en las etiquetas *Source* y *SourceURL*, respectivamente), en donde se desea realizar la consulta. En caso de que el URL no se especifique, un URL predefinido, asociado con el nombre del nodo UDDI se toma del archivo de configuración en donde se tienen los nombres de los nodos UDDI con sus respectivos URLs.Este archivo de configuración permite almacenar un gran número de URLs asociados con diversos nodos UDDI. A este archivo, pueden agregarse nuevos nodos sin la necesidad de modificar algún código o documento USML. Un ejemplo del archivo de configuración se muestra en la figura 3.

En la siguiente sección se describe la manera en la cual se realiza la búsqueda y localización de los servicios Web utilizando los operadores lógicos soportados por USML.

```
<?xml version="1.0"?>
<!DOCTYPE Registries SYSTEM "config.dtd">
<Registries>
    <Registry>
        <Name>Microsoft UDDI</Name>
        <URL> http://uddi.microsoft.com/inquire </URL>
    </Registry>
    <Registry>
        <Name>Private UDDI</Name>
        <URL> http://127.0.0.1/servlet/uddi </URL>
    </Registry>
    <Registry>
        <Name>IBM Public UDDI</Name>
        <URL>
    http://www-w3.ibm.com/services/uddi/testregistry/inquiryapi
    </URL>
    </Registry>
</Registries>
```

Fig. 3 Ejemplo del archivo de configuración

3 Búsqueda y localización de servicios Web mediante operadores lógicos

Con la ayuda de USML se pueden combinar diferentes criterios de búsqueda en una consulta y así conseguir eficacia y exactitud al realizar una sola búsqueda. Esto se debe a que en USML se pueden utilizar dos tipos de operadores lógicos en las consultas: AND y OR. Estos operadores están situados en la etiqueta *AggOperator*.

En el caso del operador OR, considere que se quiere realizar una consulta a un nodo UDDI buscando un negocio cuyo nombre contenga la cadena "IBM" o un negocio cuyo nombre de servicio contenga la cadena "Web". En esta solicitud, podemos especificar 2 consultas separadas: una para el negocio o una para el servicio. El documento USML de esta solicitud utilizando el operador OR, se muestra en la figura. 4.

En el caso de utilizar un operador AND, suponga que se desea realizar una consulta a un nodo UDDI buscando tipos de servicio cuyos nombres comiencen con la cadena "Web" y estos tipos de servicio pertenezcan a negocios cuyos nombres comiencen con la cadena "White". En esta solicitud, podemos especificar dos consultas: una para el tipo de servicio y una para el negocio. El documento USML de esta solicitud utilizando el operador AND, se muestra en la Fig. 5.

```
<?xml version="1.0"?>
<Search>
   <Query>
   <Source>Private UDDI</Source>
   <SourceURL>
   http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
   </SourceURL>
   <BusinessName>IBM</BusinessName>
   <FindBy>Business</FindBy>
   </Query>
   <Query>
   <Source>Private UDDI</Source>
   <SourceURL>
   http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
   </SourceURL>
   <BusinessName>IBM</BusinessName>
   <ServiceName>Web</ServiceName>
   <FindBy>Service</FindBy>
   </Query>
   <AggOperator>OR</AggOperator>
   <RequestTypeName>Business</RequestTypeName>
</Search>
```

Fig. 4. Ejemplo de una solicitud de búsqueda en USML utilizando el operador OR

```
<?xml version="1.0"?>
<Search>
   <Query>
   <Source>Private UDDI</Source>
   <SourceURL>
   http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
   </SourceURL>
   <BusinessName>White</BusinessName>
   <FindBy>Business</FindBy>
   </Query>
   <Query>
   <Source>Private UDDI</Source>
   <SourceURL>
   http://www-3.ibm.com/services/uddi/v2beta/inquiryapi
   </SourceURL>
   <BusinessName>IBM</BusinessName>
   <ServiceTypeName>Web</ServiceTypeName>
   <FindBy>ServiceType</FindBy>
   </Query>
   <AggOperator>AND</AggOperator>
   <RequestTypeName>Business</RequestTypeName>
</Search>
```

Fig. 5 Ejemplo de una solicitud de búsqueda en USML utilizando el operador AND

Cuando se utiliza el operador lógico AND, es necesario llenar con algún valor la etiqueta *RequestTypeName* en donde se especifica qué tipo de dato se quiere como resultado de la consulta.

Para esto, se pueden realizar 9 posibles combinaciones de consultas utilizando el operador AND:

1. BusinessServiceType: El operador AND con los tres tipos de datos (negocio, servicio y tipo de servicio), el resultado de la búsqueda devuelve un negocio.
2. ServiceBusinessType: El operador AND con los tres tipos de datos (negocio, servicio y tipo de servicio), el resultado de la búsqueda devuelve un servicio.
3. TypeBusinessService: El operador AND con los tres tipos de datos (negocio, servicio y tipo de servicio), el resultado de la búsqueda devuelve un tipo de servicio.
4. BusinessService: El operador AND con dos tipos de datos (negocio y servicio), el resultado de la búsqueda devuelve un negocio.
5. ServiceBusiness: El operador AND con dos tipos de datos (negocio y servicio), el resultado de la búsqueda devuelve un servicio.
6. BusinessType: El operador AND con dos tipos de datos (negocio y tipo de servicio), el resultado de la búsqueda devuelve un negocio.
7. TypeBusiness: El operador AND con dos tipos de datos (tipo de servicio y negocio), el resultado de la búsqueda devuelve un tipo de servicio.
8. ServiceType: El operador AND con dos tipos de datos (servicio y tipo de servicio), el resultado de la búsqueda devuelve un servicio.
9. TypeService: El operador AND con dos tipos de datos (tipo de servicio y servicio), el resultado de la búsqueda devuelve un tipo de servicio.

A continuación, se propone un caso de estudio en donde se describe y ejemplifica la funcionalidad de un portal de servicios Web en la búsqueda de servicios Web en nodos UDDI mediante USML.

4 Caso de estudio

El caso de estudio describe como el portal facilita el descubrimiento dinámico de los servicios Web que ofrecen las empresas registradas en nodos UDDI. A continuación se describe el siguiente escenario:

1. Suponga que una compañía A desea comprar componentes electrónicos, sin embargo, desconoce quienes podrían ser sus potenciales proveedores.
2. Suponga que una serie de proveedores de componentes electrónicos, han registrado sus servicios Web de manera dispersa en diferentes nodos UDDI.

En éste escenario, ¿cómo puede la compañía A buscar y encontrar los servicios Web de todos los potenciales proveedores dado que éstos, han registrado sus servicios en forma dispersa en diferentes nodos UDDI?

Para esto, se desarrolló un portal de servicios Web. En este portal de Internet, se abarcan las principales tecnologías de servicios Web, como son WSDL [8], SOAP [9], UDDI [1], WSIL [10], BPEL4WS [11] y USML. Este portal de Internet fue construido con base en los trabajos [12], [13] y [14].

Para contestar la pregunta descrita en el escenario, es necesario que la compañía A utilice el portal de servicios Web. Dentro del portal, hay una opción en el menú principal llamada USML. En esta opción, la compañía A puede realizar la búsqueda y

localización de los servicios Web de los potenciales proveedores mediante algunos criterios de búsqueda como son: nombre del negocio, nombre del servicio y tipo de servicio. Para ello, el portal ofrece 2 modalidades de consulta: 1) consulta en todos los nodos UDDI, y 2) consulta en nodo particular.

En la primera modalidad, la compañía A no necesita saber cuáles son los nodos UDDI y cuáles son los URLs de dichos nodos. En este sentido, la compañía A delega al portal la tarea de buscar los servicios Web de los proveedores en los diferentes nodos UDDI. Para ello, el portal primero realiza una búsqueda de los nodos UDDI existentes en el archivo de configuración descrito en la sección 2. Cabe recordar, que en dicho archivo se encuentran registrados los nombres y URLs de los diversos nodos UDDI en los que se pueden llevar a cabo las consultas. Una vez recuperados los nombres de los nodos UDDI y sus respectivos URLs, y una vez establecidos los criterios de búsqueda de la compañía A, el portal construye un documento USML para cada nodo UDDI en donde se va a realizar una consulta. Después, el portal empaqueta en SOAP cada documento USML. El protocolo SOAP permite el intercambio libre y estructurado de mensajes a través de Internet, definiendo un mecanismo estándar de comunicación e intercambio de datos entre aplicaciones [9,15]. Una vez construidos los documentos USML empaquetados bajo SOAP, se envían los documentos a los diferentes nodos UDDI por medio de *Java Web Service Developer Pack* (Java WSDP) que permite construir, probar y desplegar servicios y aplicaciones Web. Cabe señalar, que el envío de los documentos USML a los diversos nodos UDDI, se realiza en forma paralela, para esto, el portal crea un *thread* o hilo de ejecución por cada consulta en un nodo UDDI. Ya realizadas las consultas, es decir, enviados los documentos USML, el portal obtiene las respuestas de las consultas de cada nodo UDDI. El portal analiza estas respuestas y extrae la información que le es útil y construye un nuevo documento USML, el cual será la respuesta a la solicitud hecha por la compañía A. Una vez creada la respuesta, el portal despliega los resultados en una interfaz gráfica, y le asocia un hipervínculo a cada negocio, servicio o tipo de servicio encontrado. Todo esto, con el fin de obtener información adicional en el nodo UDDI en donde se encuentra registrado el negocio, servicio o tipo de servicio. Aquí, la compañía A ya conoce los servicios Web que ofrecen los potenciales proveedores registrados en los diferentes nodos UDDI.

En lo que respecta a la segunda modalidad, ésta se puede ver como un caso particular de la primera. En esta modalidad, la compañía A debe conocer los diversos nodos UDDI existentes. Para esto, el portal realiza una búsqueda en el archivo de configuración con el fin de obtener nombres y URLs de los diversos nodos UDDI existentes

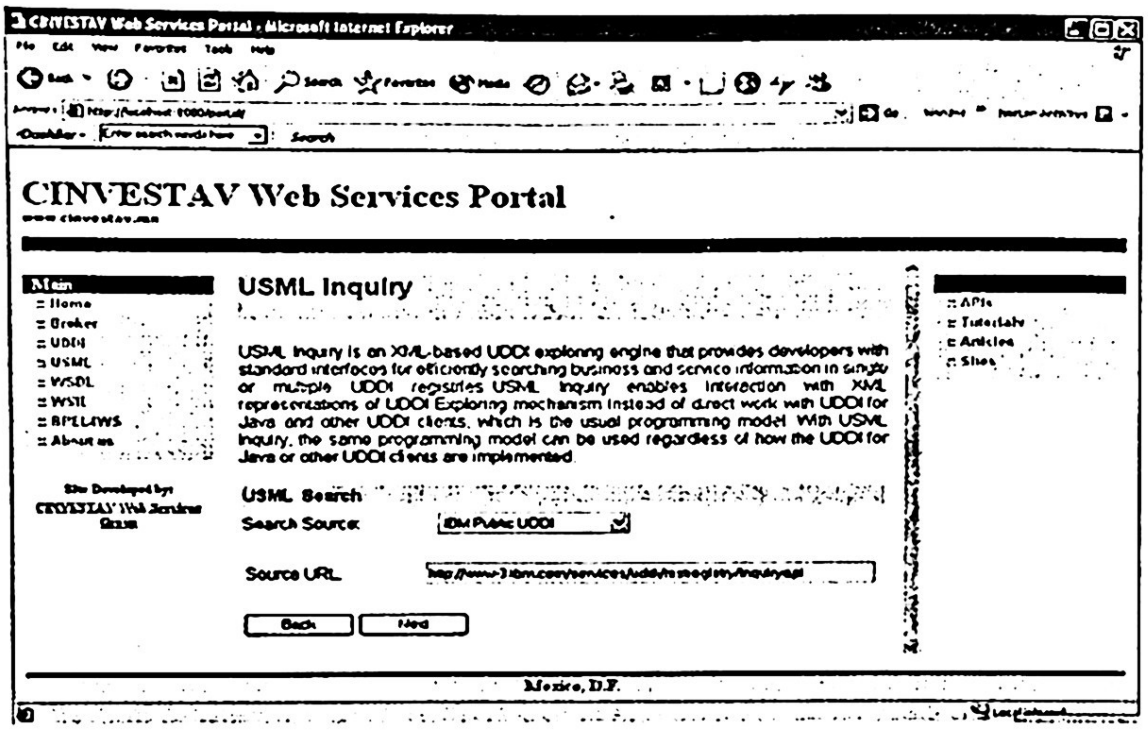


Fig. 6. Interfaz gráfica de la selección del nodo UDDI en donde se pretende realizar la búsqueda

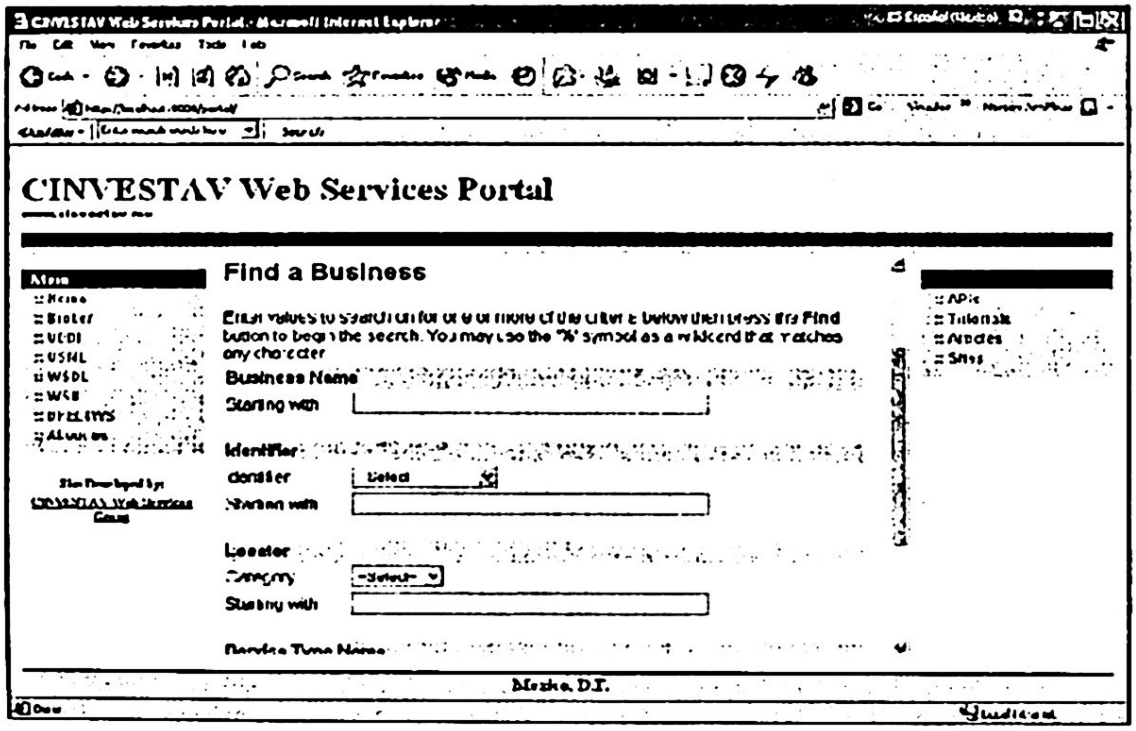


Fig. 7. Interfaz gráfica de la selección del tipo de búsqueda en USML

Una vez recuperados los nombres y URLs de los nodos UDDI, el portal despliega los resultados en una interfaz gráfica, para que la compañía A seleccione el nodo

UDDI donde desea realizar la consulta. Una interfaz gráfica de la selección del nodo UDDI se muestra en la figura 6.

Una vez seleccionado el nodo UDDI, la compañía A debe establecer los criterios de búsqueda mediante una interfaz gráfica. Al establecer un criterio de búsqueda, el portal despliega información pertinente a ese criterio, es decir, si la compañía A selecciona el criterio de búsqueda por negocio, se despliega información como: nombre del negocio a buscar, identificador del nombre de negocio a buscar, categoría del nombre de negocio a buscar, entre otros. La información que el portal despliega varía de acuerdo al criterio de búsqueda que se está seleccionando. Una interfaz gráfica del establecimiento de los criterios de búsqueda se observa en la figura 7.

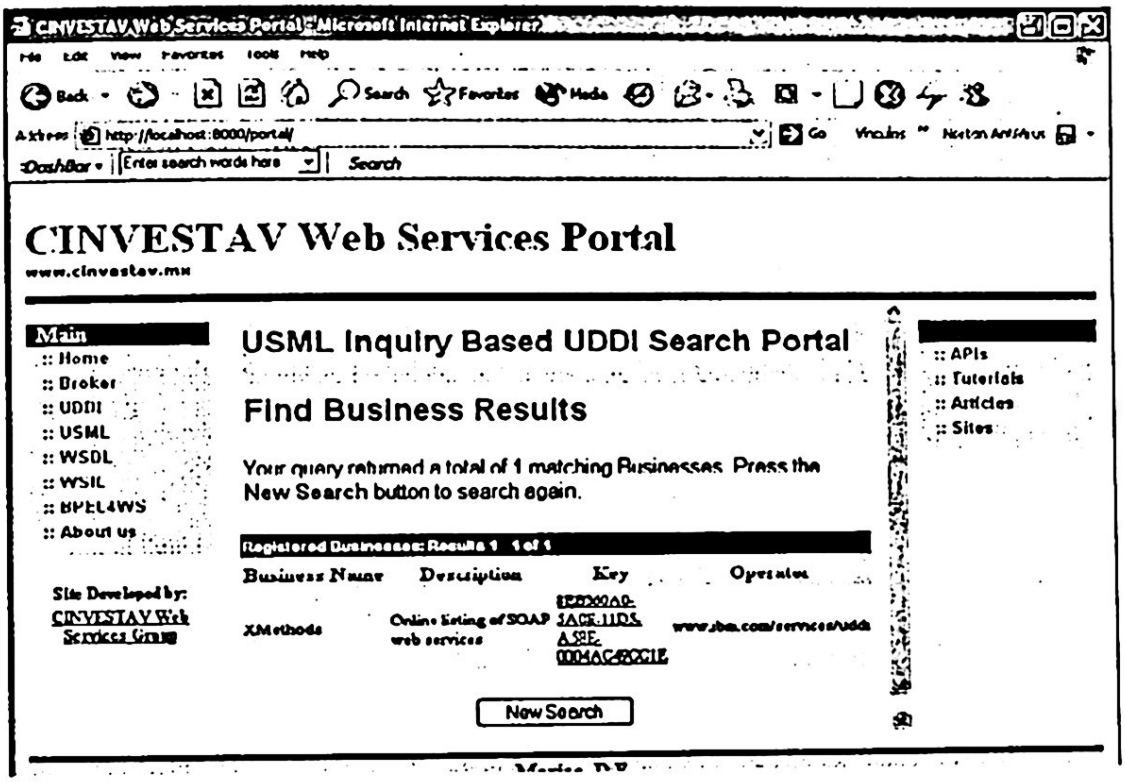


Fig. 8. Interfaz gráfica que muestra los resultados de una consulta en USML

Después de ser establecidos los criterios de búsqueda, el portal muestra una interfaz gráfica donde la compañía A debe seleccionar el operador lógico a utilizar en la consulta (OR o AND). Posteriormente de establecerse el operador lógico, para esto, el portal construye el documento USML que corresponde a la solicitud de consulta de la compañía A y lo muestra en otra interfaz gráfica. Posteriormente, el portal empaqueta en SOAP el documento USML y lo envía al nodo UDDI por medio de Java WSDP. Finalmente, el portal recibe la respuesta y despliega el resultado en una interfaz gráfica. Dicha interfaz se muestra en la figura 8. De igual forma que en la primera modalidad, el portal asocia un hipervínculo a cada negocio, servicio o tipo de servicio encontrado. Todo esto, con el fin de obtener información adicional en el nodo UDDI en donde se encuentra registrado el negocio, servicio o tipo de servicio.

En la búsqueda y localización de servicios Web, el portal es de gran ayuda a la compañía A, ya que la compañía A desconoce el nombre, la ubicación geográfica y

electrónica, los servicios y los tipos de servicios que ofrecen los potenciales proveedores.Para llevar a cabo un proceso de integración de la compañía A con los potenciales proveedores, es necesario llevar a cabo la invocación de los servicios Web ofrecidos por los proveedores. El portal de servicios Web, ofrece ésta capacidad adicional que también no esta presente en ningún nodo UDDI. La manera en que el portal lleva a cabo la invocación de servicios Web es mediante el análisis de documentos WSDL [8]. Para ello, el portal genera de forma dinámica interfaces gráficas para llevar a cabo el consumo del servicio Web. Sin embargo, esto esta fuera del alcance del artículo. Una descripción más detallada de esto, se puede encontrar en [16].

A continuación, se describe un análisis comparativo realizado del portal, con los diversos nodos UDDI mostrando ventajas y desventajas en el uso de USML.

5 Análisis comparativo del portal con nodos UDDI

El análisis comparativo describe las ventajas y desventajas encontradas del portal con los diversos nodos UDDI existentes. Este análisis fue realizado en comparación con los nodos UDDI [17], [18], [19], [20], [21].

Los resultados de este análisis se presentan en la figura 9

	Portal de Servicios Web	[17]	[18]	[19]	[20]	[21]
Búsqueda por negocio	X	X	X	X	X	X
Búsqueda por servicio	X	X	X	X	X	X
Búsqueda por tModel		X	X	X	X	X
Búsqueda por Identificadores NAICS, SIC, UNSPSC		X	X	X	X	X
Uso de USML	X					
¿Permite invocar servicios Web?	X					

Fig. 9. Resultado del análisis del portal de servicios Web en comparación con los principales nodos UDDI

Como se puede observar, el portal de servicios Web presenta ventajas significativas en comparación con los principales nodos UDDI. A diferencia de los nodos UDDI, el portal se enfoca principalmente a la búsqueda eficaz y exacta de los servicios Web, así también, como su respectiva invocación. Mientras que pasa a segunda importancia, la búsqueda de servicios mediante el uso de identificadores y *tModel* de manera singular, el cual, este tipo de consulta es provisto por los nodos UDDI.

6 Conclusiones

USML nos ayuda en gran medida a conjuntar varios criterios para la búsqueda y localización de servicios Web publicados en diversos nodos UDDI. Esto lo hace a través del uso de operadores lógicos como son AND y OR. Además, en USML podemos especificar el tipo de datos que se desea obtener como resultado de una consulta dada. Todas estas características, son muy importantes ya que los sistemas de información que lo utilicen, como el portal de servicios Web desarrollado, obtendrán exactitud y eficacia en las consultas a diversos nodos UDDI.

Referencias

1. UDDI, UDDI Version 3.0, Published Specification, July 19, 2002.
2. Data Universal Numbering System, D-U-N-S Homepage, http://www.dnb.com/.
3. Thomas Register, Thomas Register Homepage, http://www.thomasregister.com/
4. North American Industry Classification System, NAICS Homepage, http://www.naics.com/.
5. United Nations Standard Products and Services Code, UNSPSC Homepage, http://www.unspsc.org/.
6. International Organization for Standardization (ISO) - Language codes, ISO Homepage, http://www.iso.ch/
7. Standard Industrial Classification, SIC Homepage, http://www.sec.gov/info/edgar/siccodes.htm
8. Web Services Description Language Specification, WSDL Home, http://www.w3.org/TR/wsdl12
9. Simple Object Access Protocol Specification, SOAP Home, http://www.w3.org/TR/SOAP/
10. Web Services Inspection Language (WS-Inspection) 1.0. November 2001. From International Business Machines Corporation (IBM).
11. Business Process Execution Language for Web Services. BPEL4WS 1.1 Specification. IBM May 5 2003.
12. Giner Alor Hernández, José Oscar Olmedo Aguirre, "Automatización de la Cadena de Suministro usando UDDI", Proceedings CNCIIC-ANIEI 2003.
13. Giner Alor Hernández, José Oscar Olmedo Aguirre, "Sistema de Intermediación para el Comercio Electrónico B2B basado en Servicios Web", Proceedings CIC 2003.
14. Giner Alor Hernández, José Oscar Olmedo Aguirre, "Automatización de la Cadena de Suministro en Comercio Electrónico B2B", Proceedings CIE 2003.
15. Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi and Sanjiva Weerawarana, "Unraveling the Web Services Web An Introduction to SOAP, WSDL, and UDDI". IEEE Internet Computing. April 2002. Pages. 86-93
16. César Sandoval Hernández, Giner Alor Hernández, José Oscar Olmedo Aguirre. *"Generación Dinámica de GUIs para la Invocación de Servicios Web publicados en nodos UDDI"*. Proceedings CORE-2004. Por publicarse.
17. IBM UDDI Business Production Registry, https://uddi.ibm.com/ubr/registry.html
18. IBM UDDI Business Test Registry, https://uddi.ibm.com/testregistry/registry.html
19. Microsoft UDDI Business Test Registry, http://test.uddi.microsoft.com/
20. Microsoft UDDI Business Production Registry, http://uddi.microsoft.com/
21. SAP UDDI Business Registry, http://uddi.sap.com/

Generación dinámica de GUIs para la invocación de servicios Web publicados en nodos UDDI

César Sandoval Hernández, Giner Alor Hernández,
José Oscar Olmedo Aguirre

Centro de Investigación y de Estudios Avanzados del IPN,
Departamento de Ingeniería Eléctrica. Sección de Computación,
Av. Instituto Politécnico Nacional 2508, Col San Pedro Zacatenco, 07360,
México, D. F.
{csandoval, gineralor}@computacion.cs.cinvestav.mx,
oolmedo@delta.cs.cinvestav.mx

Resumen. Actualmente, la forma como se conducen las transacciones comerciales está sufriendo cambios significativos. Los mercados están usando el Internet como una forma eficaz de conducir prácticas comerciales. Esto ha provocado el surgimiento de un nuevo paradigma computacional llamado servicios *Web* el cual tiene como principal aplicación el comercio electrónico. Los servicios *Web* hacen amplio uso de UDDI el cual es básicamente un servicio públicamente disponible en el que se puede publicar y recuperar información sobre negocios. Sin embargo, actualmente no existen herramientas públicamente disponibles que permitan ubicar, analizar e invocar servicios *Web* que se encuentran en los diferentes nodos UDDI a partir del nombre del negocio o del servicio. El presente trabajo ofrece una interfaz de usuario que resuelve esta situación permitiendo no sólo hacer búsquedas por palabras clave sobre los negocios y servicios que se encuentran en repositorios UDDI, sino que también permite analizar la información sobre los negocios que se encuentra en los nodos UDDI y ejecutar los correspondientes servicios.

1 Introducción

El uso de los servicios *Web* en el *World Wide Web* se expande rápidamente debido a la creciente necesidad de comunicación entre aplicaciones y de interoperabilidad. Estos servicios proveen esencialmente un medio estándar de comunicación entre diferentes aplicaciones de software. Los protocolos de comunicación que los servicios *Web* utilizan están basados en el *eXtensible Markup Language* (XML) [1] el cual, como su nombre lo indica, es un lenguaje de marcado extensible que permite describir cualquier tipo de información. En los servicios *Web* se hace uso de lenguajes derivados de XML que ayudan a describir las operaciones que se van a ejecutar o la información a intercambiar con otro servicio. El mayor logro de los servicios *Web* es que han permitido a las aplicaciones comunicarse entre ellas de una manera independiente del lenguaje de programación y de la plataforma, es decir, producen

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.68-79.© Centro de Investigación en Computación, IPN, México

una mayor interoperabilidad. Los servicios *Web* también definen una interfaz de software basada en dialectos de XML que describe una colección de operaciones que pueden ser accedidas remotamente.

Existe una gran cantidad de tecnología relacionada con los servicios *Web* dentro de la que destaca el directorio *Universal Description, Discovery and Integration* (UDDI) [2] y el *Simple Object Access Protocol* (SOAP) [3]. La primera define un registro y protocolos asociados para la búsqueda y localización de servicios *Web*. Mientras que la segunda permite codificar documentos XML para que puedan ser transmitidos en la red usando protocolos de transporte como HTTP.

En el presente trabajo se describe la funcionalidad de una interfaz de usuario orientada a la *Web* construida de forma dinámica para visualizar información relacionada con los servicios *Web* ofrecidos por diversos negocios que se encuentran publicados en nodos UDDI. El principal interés para construir dicha interfaz es tener una herramienta que sea fácil de usar y que además ayude en la invocación de cualquier tipo de servicio *Web*. En la primera parte de este documento se muestra la interfaz que permite realizar búsquedas de servicios *Web* en nodos UDDI dados algunos criterios de búsqueda. En la segunda parte se describen los procedimientos usados para la inspección de los servicios *Web*. Y finalmente se presenta la forma en que se realiza la invocación dinámica de los servicios *Web* a través de mensajes SOAP.

2 Búsqueda de servicios Web en nodos UDDI

Los servicios *Web* son un conjunto de tecnologías de comunicación que proporcionan interfaces a través de las cuales se puede tener acceso a servicios públicamente disponibles. Estos servicios *Web* pueden ser publicados y consultados en nodos UDDI como los que provee IBM [4], Microsoft [5], SAP [6], entre otros. UDDI provee dos tipos de servicio: publicación y consulta. El primero define operaciones que permiten el registro, modificación y eliminación de negocios y servicios. Algunos ejemplos de esto son *save_business*, *save_service*, *delete_business*, *delete_service*, por mencionar algunos. Mientras que el servicio de consulta define operaciones que permiten realizar búsquedas sobre negocios y servicios en un nodo UDDI como *find_business*, *find_service*, *find_tModel*, *get_businessDetail*, *get_serviceDetail*, entre otros. Los negocios interesados en el registro de sus servicios *Web* en algún nodo UDDI deben usar el servicio de publicación. La información del negocio y sus servicios se almacena en una entidad llamada BusinessEntity. Esta entidad es un documento XML que según la especificación de UDDI es la más importante dentro de su estructura. En la Fig. 1 se muestra la entidad BusinessEntity y sus elementos.

Al igual que un directorio telefónico el elemento BusinessEntity divide la información en tres categorías: páginas blancas, páginas amarillas y páginas verdes. Como se puede ver en la Fig. 1, las páginas blancas contienen información de contacto como nombre, teléfono, dirección o *e-mail*. Mientras que las páginas amarillas proveen información relacionada con la categoría del negocio basada en ontologías como las establecidas por el *North American Industry Classification System* (NAICS) [7] o el *United Nations Standard Products and Services Code*

(UN/SPSC) [8], por mencionar algunos. Finalmente las páginas verdes ofrecen información de los servicios en la cual se especifica como otros negocios pueden integrarse.

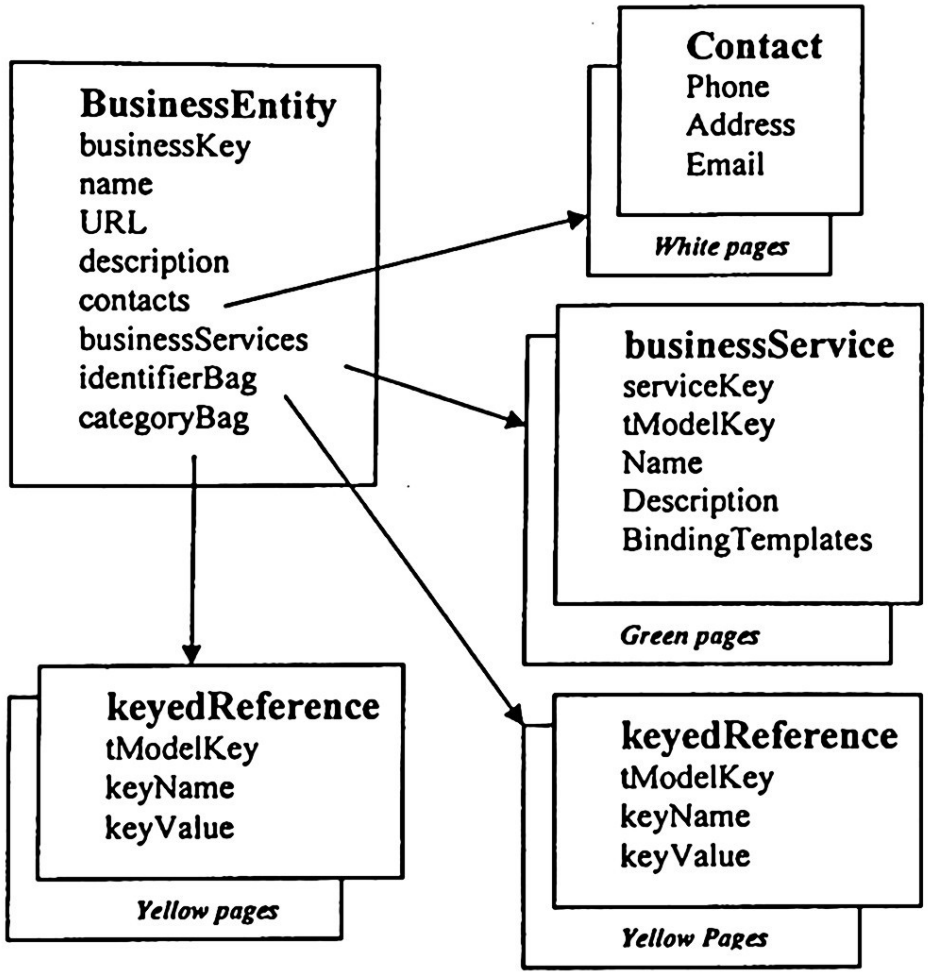


Fig. 1. Información necesaria para el registro de un negocio en un nodo UDDI.

El portal que se desarrolló hace uso de UDDI4J (implementación de UDDI en Java) [9] para obtener los servicios *Web* contenidos en algún repositorio UDDI que cumplan con ciertas características. En particular, las características son el nombre del negocio y servicio, aunque es posible hacer consultas más elaboradas como se explica en [10].

Dentro del portal existe una opción llamada UDDI que permite la búsqueda y localización de servicios Web publicados en nodos UDDI. Para realizar la búsqueda el portal ofrece dos criterios los cuales son: por servicio o por negocio tal como se muestra en la Fig. 2. Supongamos que se desea buscar uno o más servicios *Web* cuyo nombre de negocio contenga la palabra Amazon en algún nodo UDDI de Microsoft. Entonces el usuario introduce estos criterios en la interfaz de la Fig. 2.

Una vez establecidos los criterios, el portal utiliza las operaciones *find_business* o *find_service* (según sea el caso) del servicio de consulta de UDDI. Entonces construye un documento XML y lo empaqueta en SOAP. A continuación envía el mensaje al nodo UDDI. El portal despliega una lista de negocios y sus correspondientes servicios que cumplen con los criterios de búsqueda. Aquí el usuario puede elegir alguno de los

negocios o servicio para obtener información adicional que resulte de su interés. Al seleccionar un servicio el portal utiliza la operación *get_serviceDetail* mediante la cual se puede obtener información como el nombre del negocio, su descripción, los servicios, los puntos de acceso y la ubicación del documento escrito en el *Web Services Description Language* (WSDL) [11] el cual está relacionado con las páginas verdes. Esta información se muestra en la Fig. 3.

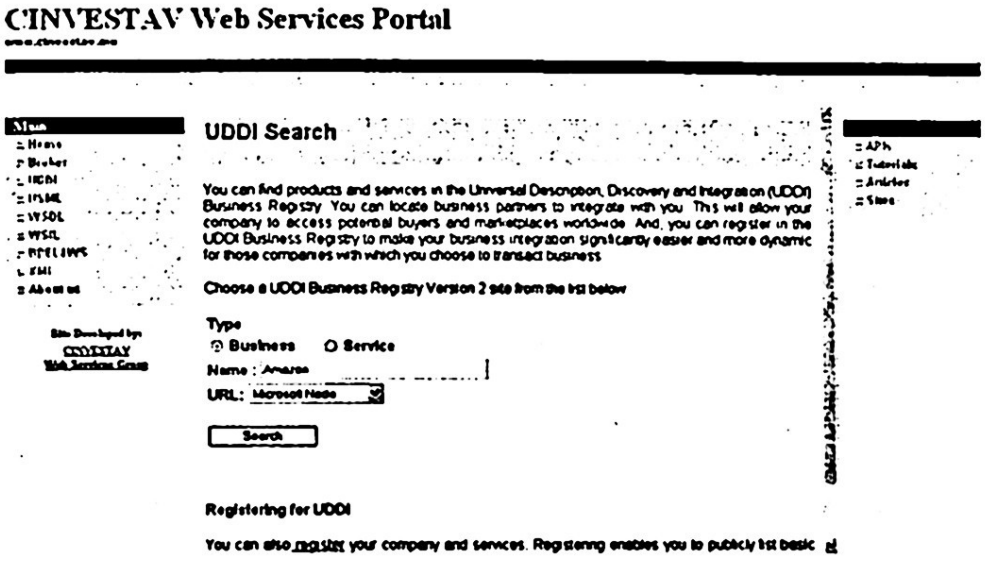


Fig. 2. Interfaz gráfica de usuario mediante la cual se pueden hacer consultas a nodos UDDI dado el nombre de negocio o servicio

En particular, el documento WSDL es de gran importancia porque provee información suficiente para realizar posteriormente la invocación de los servicios ofrecidos por la organización. La posibilidad de invocar un servicio *Web* no es provista en ningún nodo UDDI. Como solución el portal permite efectuar la invocación mediante la generación dinámica de interfaces gráficas de usuario (GUIs)

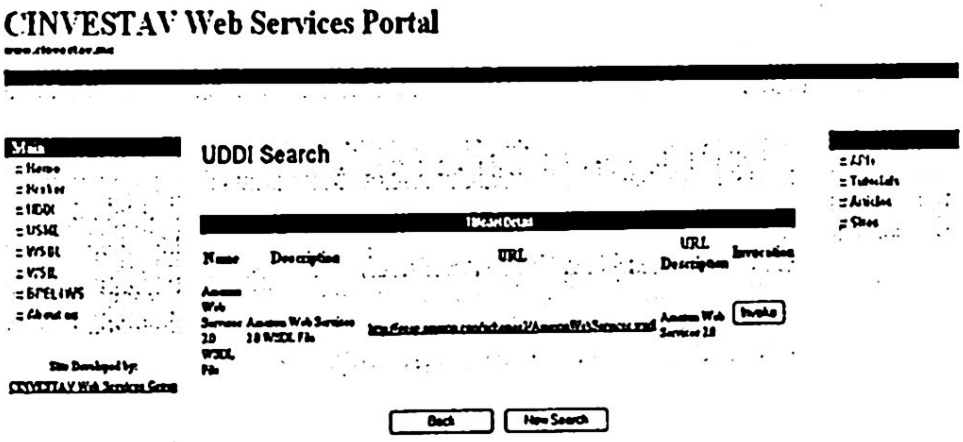


Fig. 3. Información desplegada cuando se selecciona un negocio en específico. Se puede ver tanto el WSDL como proceder a su invocación.

A continuación se describen los procedimientos aplicados para llevar a cabo la inspección de los servicios *Web*.

3 Inspección de los servicios Web

Un documento WSDL representa una serie de sentencias XML que constituyen la definición para las interfaces de cada servicio lo que ayuda a obtener una descripción precisa acerca de los servicios que se ofrecen. Una vez ubicado el documento WSDL correspondiente a un servicio *Web* el portal procede a su análisis. Para realizar el análisis el portal usa el *Web Services Invocation Framework* (WSIF).

El WSIF es una API en Java para la invocación de los servicios *Web*, no importando dónde y cómo los servicios se proveen [12]. WSIF contiene algunas utilerías con las cuales se pueden obtener objetos del *Web Services Description Language for Java* (WSDL4J). El WSDL4J es una API en Java de WSDL que permite mapear la información encontrada en documentos WSDL a objetos Java que pueden manipularse [13].

En la Fig. 4 se describen los pasos seguidos por el portal para obtener objetos Java representativos del documento WSDL. La información en un documento WSDL que se requiere para llevar a cabo la invocación de un servicio *Web* se encuentra en los elementos *Service*, *PortTypes* y *Bindings*.

El elemento *Service* contiene información para acceder al servicio *Web*, la cual está contenida en el atributo *AccessPoint*. El elemento *PortType* define un conjunto de operaciones provistas por el servicio. El elemento *Binding* define los protocolos asociados a las operaciones descritas en el elemento *PortType*. El portal analiza el documento WSDL y extrae los elementos antes mencionados. Para mostrar el resultado, el portal genera una GUI y despliega los elementos PortTypes encontrados. El usuario puede entonces seleccionar el PortType que le sea de interés.

Para el caso de Amazon, sólo se tiene un elemento PortType como se puede ver en la Fig. 5. Cabe mencionar aquí que en general se pueden encontrar uno o más elementos PortType que sirven para hacer diversos tipos de petición como HTTP o SOAP. El portal únicamente puede enviar mensajes a través de SOAP.

Al seleccionar un *PortType*, el portal internamente recupera un enlace entre el servicio y las operaciones. A través del WSDL4J se obtienen las operaciones asociadas a dicho enlace las cuales son presentadas en una nueva GUI como se muestra en la Fig. 6. El usuario puede entonces seleccionar alguna de las operaciones. Supongamos que quiere hacer una búsqueda en el servicio *Web* de Amazon de su película favorita "*Matrix*", puede entonces escoger la operación *KeywordSearchRequest* que se encuentra dentro del conjunto de operaciones disponibles por Amazon. Una vez seleccionada la operación, el portal obtiene el conjunto de parámetros asociados con esa operación los cuales son presentados en la GUI de la Fig. 7. Los parámetros pueden ser de tipo primitivo (string, float, boolean, etc.) o de tipo complejo (contiene a tipos primitivos). El usuario debe entonces llenar los campos con los datos generales de la película y enviarlos al servidor *Web*.

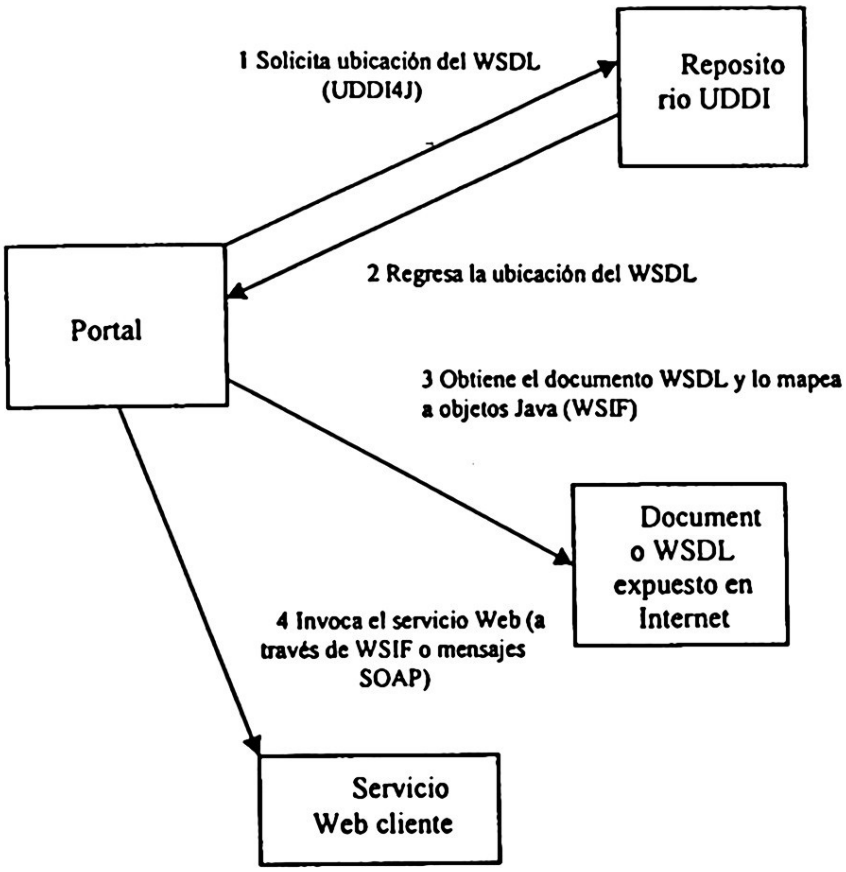


Fig. 4. Esquema general que muestra los pasos efectuados por el portal para obtener objetos representativos del documento WSDL.

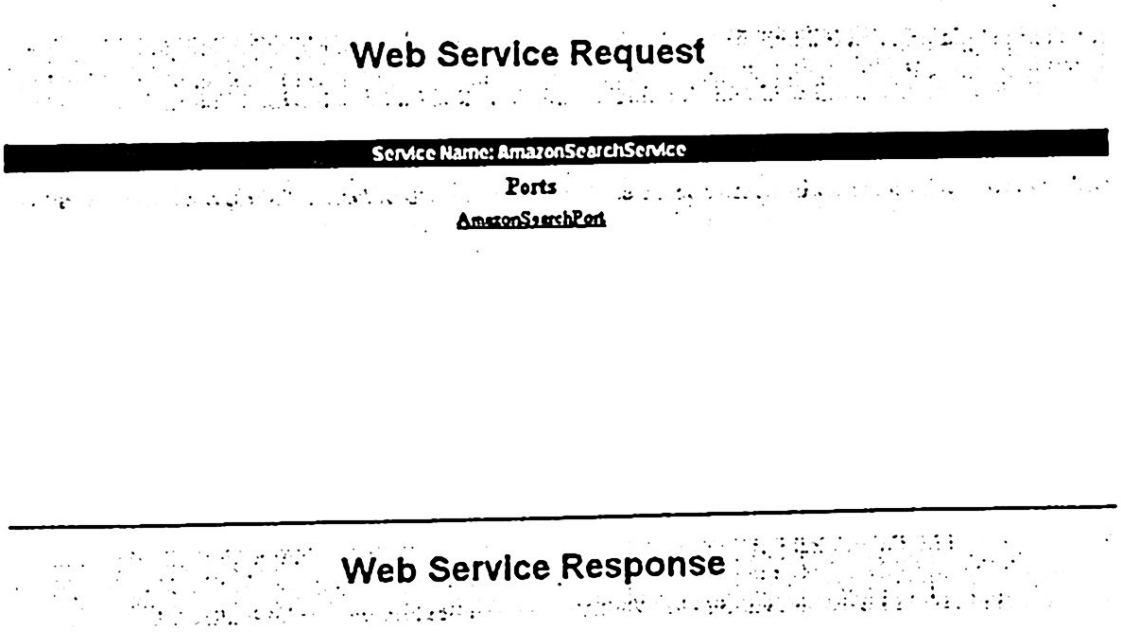


Fig. 5. Desplegado de los puertos disponibles en el servicio *Web* de Amazon.

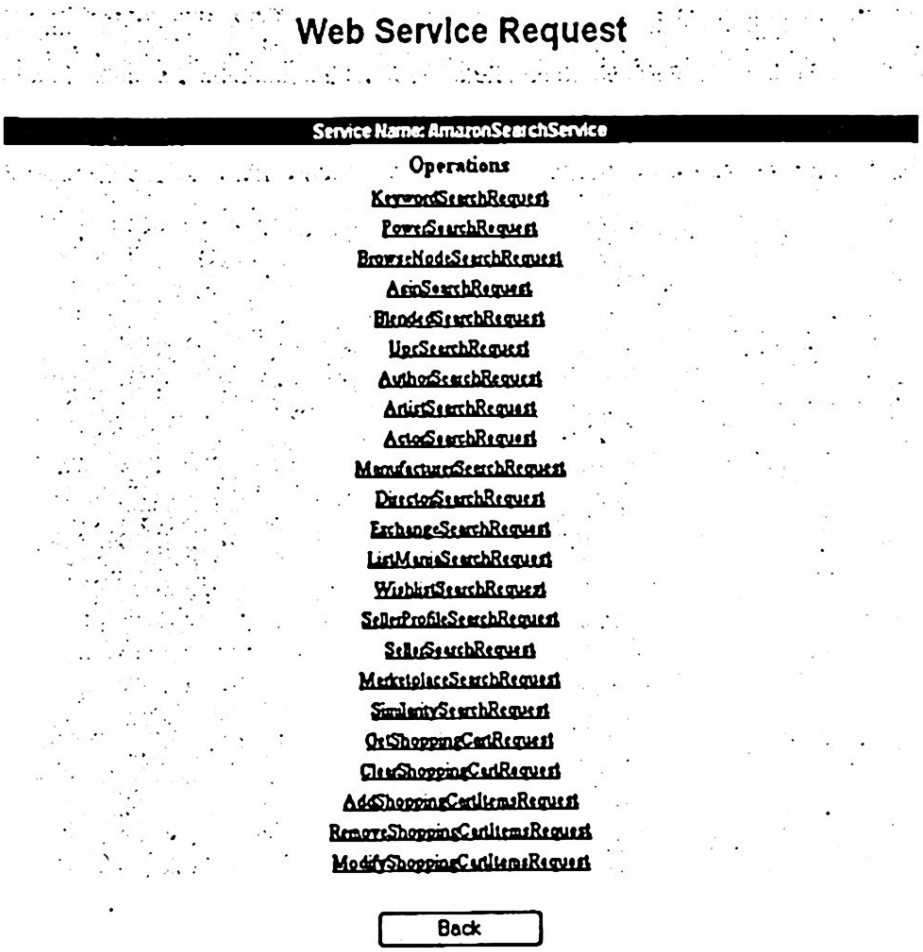


Fig. 6. Operaciones disponibles en el puerto.

En la siguiente sección se presenta la forma en que se lleva a cabo la preparación de la invocación al servicio *Web*.

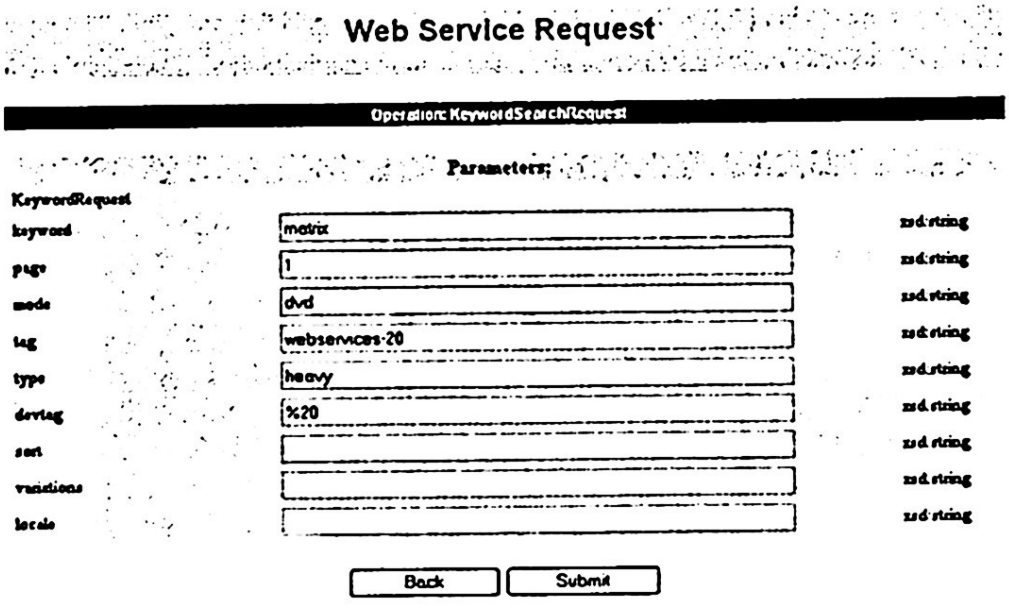


Fig. 7. Parámetros de la operación seleccionada que deben ser llenados por el usuario.

4 Invocación de los servicios Web

A partir de este momento el usuario no necesita ingresar ninguna información adicional. Sin embargo, a pesar de que en este punto se cuenta con gran parte de la información necesaria para hacer la solicitud al servicio *Web* aún faltan datos importantes que el sistema se encargará de determinar automáticamente. Se ha dicho que la invocación de un servicio *Web* se hace a través del envío de mensajes SOAP. Un mensaje SOAP tiene la estructura mostrada en el esquema de la Fig. 8.

Un mensaje SOAP consiste de una cabecera y un cuerpo. En la cabecera se puede agregar información específica de la aplicación como el nombre de espacios, autenticación, entre otros. Mientras que en el cuerpo se agrega el mensaje que se desea enviar. En el caso del portal, se envían los valores de los parámetros asociados con la petición. El portal necesita primeramente determinar el tipo de codificación que se debe hacer en el mensaje SOAP. Existen dos formas de codificar un mensaje SOAP las cuales son *encoded* o literal. En una codificación *encoded* se debe agregar un espacio de nombres al mensaje SOAP. En caso contrario éste no debe ser agregado.

Además se debe determinar si el estilo de la invocación es *document* o *rpc*. En el primer caso se requiere determinar un *SoapAction* el cual es un espacio de nombres para agregarlo a la cabecera de la petición SOAP. Mientras que en el segundo caso no es necesario agregarlo.

Finalmente, se está en condiciones de enviar la solicitud al servicio *Web*. Para realizar dicha solicitud se ha recurrido al modelo tradicional de enviar mensajes a través de las APIs de SOAP debido a que WSIF no permite el envío de mensajes de una manera eficaz en el caso de los tipos complejos para hacer la invocación. La forma que tiene dicha petición es de un documento XML. Se envía entonces la petición y se espera a que el servidor responda, es decir, la petición que se realiza es síncrona.

El proveedor del servicio Web puede responder de muy diferentes maneras. Puede enviar en respuesta uno o más valores de tipo primitivo, tipos complejos o incluso arreglos de tipos complejos. La respuesta que se obtuvo del servicio Web puede ser vista en dos presentaciones: texto plano (documento XML) o con formato. La primera se muestra en la Fig. 9, sin embargo, puede no ser suficientemente legible para el usuario. Sin embargo el portal ofrece la capacidad de dar formato a la respuesta de tal forma que resulte más claro.

En la Fig. 10 se puede ver la respuesta del servicio Web con un formato en tablas. Para conseguir esta presentación, se tuvo que analizar primeramente la respuesta en XML y generar dinámicamente un documento escrito en el eXtensible Stylesheet Language (XSL) a partir del cual se le da un formato en tablas.

Un documento XSL es una hoja de estilos escrita en XML para dar un formato a un documento XML. Este formato puede ser texto plano, html, pdf, entre otros.

Se ha podido entonces constatar que el portal es de gran ayuda para la búsqueda, localización e invocación de los servicios Web publicados en nodos UDDI. Además, el portal ofrece la generación dinámica de GUIs para poder llevar a cabo la invocación de un servicio Web. Finalmente, el portal muestra la respuesta de un servicio Web de una forma clara mediante el uso de hojas de estilo de XSL.

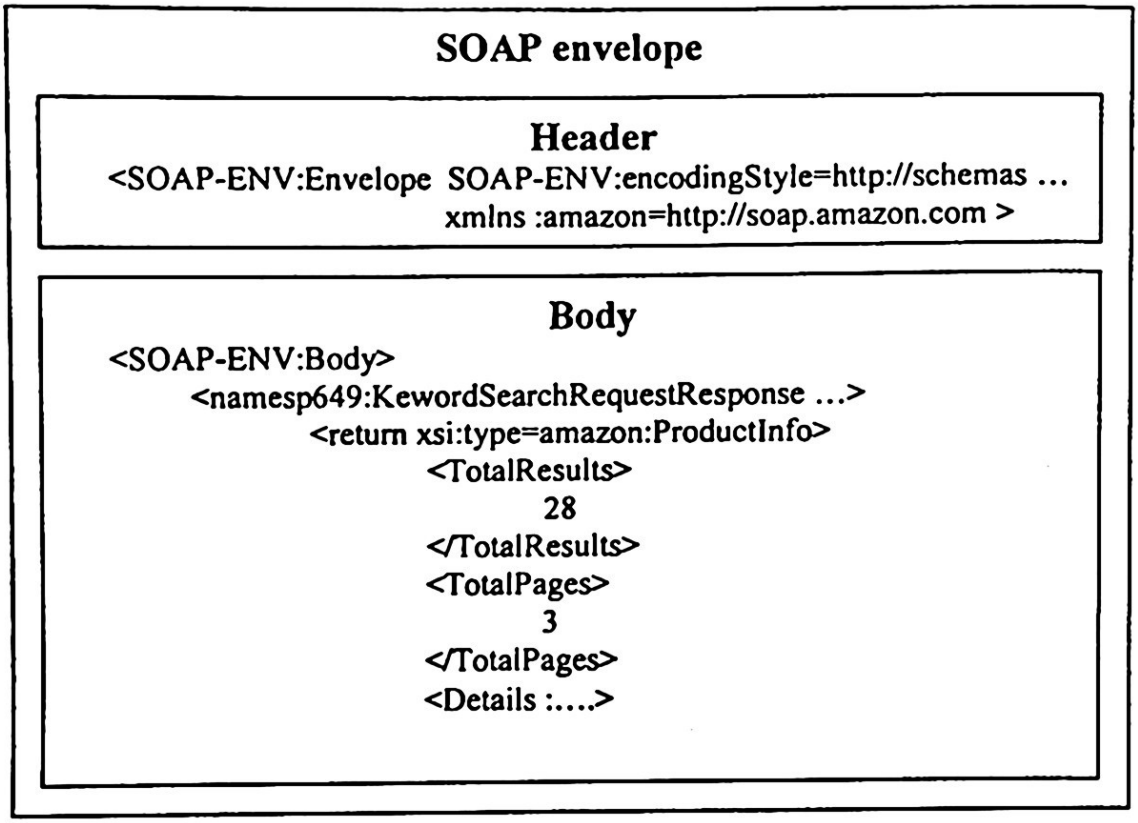


Fig. 8. Estructura de un mensaje SOAP.

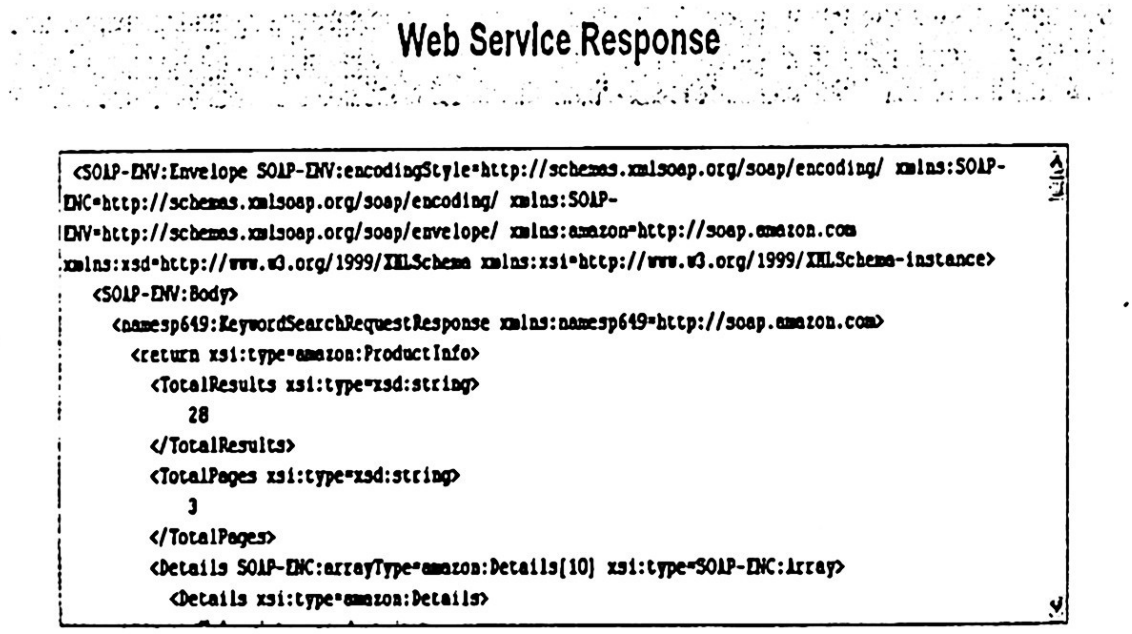


Fig. 9. Respuesta del servidor Web al que se le hizo la petición sin formato

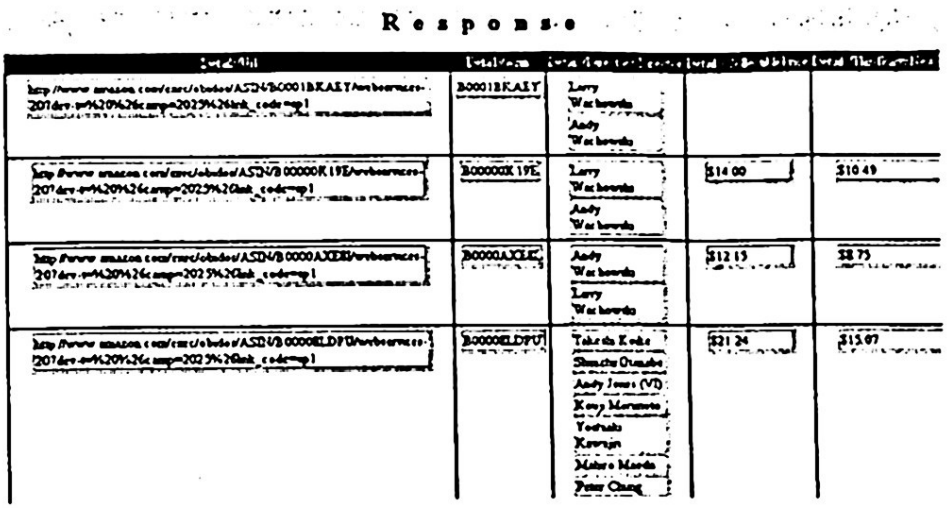


Fig. 10. Respuesta del servidor *Web* al que se le hizo la petición con formato.

5 Trabajos relacionados

En el mercado existen diversos programas que permiten analizar documentos WSDL y la correspondiente ejecución de los servicios disponibles. Uno de los más conocidos es StrikeIron [14] el cual es una herramienta muy útil y fácil de usar debido a que presenta los servicios en estructura de árbol. StrikeIron ha sido desarrollado para encontrar e invocar cualquier tipo de servicios *Web* a partir del WSDL. Permite el manejo de tipos de datos simples y complejos. Sin embargo, requiere de una licencia para su uso y el usuario debe conocer con exactitud la ubicación del documento WSDL. Esto puede resultar inconveniente si no se conoce dicha dirección y lo único que se sabe es el nombre del negocio o servicio.

Otro sistema que también realiza un análisis del documento WSDL es el Mindreef SOAPscope [15] el cual se encuentra parcialmente disponible para su ejecución en Internet. Ha sido desarrollada por Mindreef y XMethods para analizar e invocar los servicios *Web* de XMethods con SOAPscope a partir del documento WSDL. Aunque la versión públicamente disponible en Internet solamente ejecuta servicios que se encuentran en XMethods [16] el cual es un repositorio de UDDI. Si se desea ejecutar otros servicios se debe obtener también una licencia.

En el presente trabajo se ha propuesto un sistema que realiza en esencia las mismas actividades que StrikeIron y Mindreef SOAPscope; es decir, permite la invocación de cualquier tipo de servicio Web. Sin embargo, a diferencia de estos dos programas, el portal ofrece la posibilidad al usuario de hacer búsquedas por nombre de negocio o servicio en diferentes nodos UDDI evitando que el usuario necesite conocer la ubicación del archivo WSDL como en el caso de los programas antes mencionados. Aunque si el usuario conoce la ubicación del documento WSDL también puede ingresarlo y hacer la invocación directamente.

Por otro lado, es necesario precisar que el portal aún tiene que cubrir un aspecto que tanto StrikeIron como Mindreef SOAPscope sí cubren, el cual es el manejo de tipos complejos dentro de otros tipos complejos. En la versión actual del portal, sólo se trabajan con tipos primitivos y tipos complejos de un solo nivel.

6 Conclusiones

En el presente trabajo se ha logrado la construcción de una interfaz de usuario a través de la cual se pueden hacer consultas a un conjunto de nodos UDDI predeterminados para obtener una lista de servicios *Web* que coincidan con ciertos criterios de búsqueda, específicamente palabras clave del nombre del negocio o del servicio. Se puede entonces obtener información detallada del negocio incluyendo la descripción del negocio, su nombre, la ubicación del documento WSDL, entre otras.

El uso de un conjunto de tecnologías relacionadas con los servicios *Web* (UDDI y SOAP), así como de herramientas que permiten su explotación (WSIF, UDDI4J, o WSDL4J) ha permitido la construcción de dicha interfaz mediante la cual se pueden hacer consultas básicas a nodos UDDI sobre los servicios *Web* que ahí residen. Aún más, se han podido hacer invocaciones de estos servicios dados los parámetros introducidos por el usuario. En síntesis se puede decir que el usuario dispone de una herramienta orientada a la *Web* que, dado un nombre de negocio o servicio, puede hacer peticiones al servicio *Web* correspondiente sin disponer de mayor información que la que quiere consultar en dicho servicio *Web*.

Sin embargo, a pesar de que se ha logrado construir un portal capaz de analizar e invocar dinámicamente una gran cantidad de servicios *Web*, aún queda trabajo por realizar. En primer lugar, la invocación de los servicios *Web* podría ser mejorada al plantear una forma eficiente de enviar peticiones de datos híbridos, es decir, complejos y primitivos a la vez y de tipo complejo pero anidados. Esto permitiría hacer más general la aplicación. Por otra parte, también se podrían hacer búsquedas más complejas sirviéndose del *UDDI Search Markup Language* (USML), lo que permitiría por ejemplo, hacer consultas no sólo en un nodo UDDI a la vez sino en varios. Lo anterior serviría para obtener resultados más precisos acerca de lo que se está buscando.

Referencias

1. F. Yergeau et al., Extensible Markup Language (XML) 1.1, W3C Recommendation, World Wide Web Consortium , February 2004, http://www.w3.org/XML/
2. T. Bellwood et al., Universal Description, Discovery and Integration (UDDI) 3.0, Published Specification, Organization for the Advancement of Structured Information Systems (OASIS), July 2002, http://www.uddi.org/specification.html.
3. D. Box et al., Simple Object Access Protocol (SOAP) 1.1, W3C Note 08, World Wide Web Consortium, May 2000, http://www.w3.org/TR/SOAP/
4. IBM UDDI Business Test Registry, https://uddi.ibm.com/testregistry/registry.html.
5. Microsoft UDDI Business Registry Node, http://uddi.microsoft.com/default.aspx.
6. SAP Repository, http://www.sap.com.
7. North American Industry Classification System (NAICS), http://www.naics.com/.
8. United Nations Standard Products and Services Code (UNSPSC), http://www.unspsc.org/.
9. Universal Discovery, Description and Integration for Java (UDDI4J),
10. http://www-124.ibm.com/developerworks/oss/uddi4j/.
11. G. Alor, C. Sandoval y O. Olmedo. "Descubrimiento dinámico de servicios *Web* en nodos UDDI mediante USML." Por publicar.

12. Web Services Description Language Specification (*WSDL*), http://www.w3.org/TR/wsdl12.
13. Web Service Invocation Framework (WSIF), http://ws.apache.org/wsif/.
14. Web Services Description Language for Java (WSDL4J), http://www-124.ibm.com/developerworks/projects/wsdl4j/.
15. StrikeIron Inc., Web Services Business Objects, http://www.strikeiron.com/.
16. Mindreef Web Service Diagnostics, Mindreef SOAPscope, http://www.mindreef.com.
17. XMethods, http://www.xmethods.net.

VCMG: Generador de mapas conceptuales vectoriales

Yuri Vizcarra González, Alberto Pacheco González

Departamento de Estudios de Posgrado,
Instituto Tecnológico de Chihuahua
México
apacheco@itchihuahua.edu.mx,
yavizcar@hotmail.com

Resumen. Se describe una herramienta desarrollada en PHP para generar dinámicamente imágenes vectoriales de mapas conceptuales hipermedia a partir de presentaciones Web de ExpoVision.

1 Introducción

1.1 Presentaciones Web

La herramienta VCMG forma parte de una plataforma para asistir a la generación de presentaciones ultraligeras por Internet (Web Browser based) denominada ExpoVision.

1.2 Mapas conceptuales (MC)

MC en una técnica desarrollada en el departamento de Educación de la Universidad de Cornell en los 70's.

Los MC constituyen una respuesta a la teoría del aprendizaje significativo como parte del programa denominado "Aprender a Aprender". Asimismo, el término "aprendizaje significativo" de David Ausubel involucra asimilar conceptos nuevos mediante su inclusión en estructuras cognitivas ya existentes.

Se han convertido en un elemento muy importante en los planes de perfeccionamiento de los sistemas de enseñanza y han extendido su uso a otras esferas de la actividad humana en las que la gestión del conocimiento ocupa un lugar preponderante.

Un ejemplo de ello, es su uso en educación a distancia o educación semipresencial, donde es fundamental el estudio independiente y el autoaprendizaje por parte de los estudiantes.

Los MC representan, simultáneamente, una estrategia de aprendizaje, un método para *captar lo más significativo de un tema* y un recurso esquemático para *representar un conjunto de significados conceptuales*, incluidos en una estructura de proposiciones.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.80-88.© Centro de Investigación en Computación, IPN, México

1.3 SVG (Scalable Vector Graphics)

SVG es una representación basada en *XML*[3] de objetos geométricos vectoriales en *2D* que pueden ser desplegados y manipulados mediante funciones propias *SVG* o scripts como *Javascript/ECMAScript, PHP* [2], etc. además de poseer las características de ser eficientes, versátiles, portables, ligeros e interactivos en la Web.

Es considerado un estándar abierto [4]-[6] amparado por el Consorcio Web (*W3C*) [7] además de ser el resultado del esfuerzo de compañías tales como: *Adobe, HP, IBM, Microsoft, Netscape, SUN Microsystems y Microsoft Visio* [15].

Fue propuesta como estándar gráfico 2D para la *Web* en Septiembre de 2001 [4], sucediéndole la especificación versión 1.1 en Noviembre de 2002 [5]. Actualmente se encuentra en desarrollo el draft de la especificación 1.2 [6].

SVG posee las siguientes cualidades:

- Código fuente compacto y portable.
- Código fuente editable en cualquier editor de texto (*i.e.* notepad).
- Provee escalamiento de imagen con mucha mayor resolución que archivos *GIF*.
- Permite ampliaciones-reducciones de imágenes *(zooming)* sin pérdida de resolución.
- Permite la generación de imágenes complejas mediante el uso de transformaciones tales como escalamientos y rotaciones.
- Es capaz de crear y manejar elementos gráficos basados en líneas, curvas, imágenes y textos.
- Permite agrupar, animar, filtrar, recortar (*clipping*), enmascarar (*masking*), y trasladar elementos.
- Puede ser generado mediante servlets de *Java, PHP y JSP*.
- Basado en texto estructurado con sintáxis *XML*, soporta *hiperenlaces* de *XLink* [12], hojas de estilos *CSS* [13] *y código embebido* de *JavaScript* [1].
- Soporta compactación *GZIP, zooming y panning* sin necesidad de recargar imágenes.
- Más compacto, rápido de transferir y versátil para transformar y manipular objetos que usando imágenes de *bitmaps*.
- Su independencia del dispositivo permite adaptar imagen a las capacidades del mismo, tales como resolución, tamaño, colores, etc. en computadoras personales, dispositivos móviles e impresoras.
- Hay diversidad de proveedores y supera ciertas deficiencias de tecnologías previas como *SGML* [8], *DrawML, HTML, VML y Flash* entre otros.
- Permite crear dinámicamente imágenes y manipularlas vía *JavaScript*.
- Gran versatilidad al interoperar con otros estándares *W3C* tales como *CSS, DOM* [17], *SMIL* y *JavaScript*.
- Gratuito, solo es necesario *Adobe SVG viewer* [16].
- Requiere conocimientos básicos de geometría y trigonometría.

2 Desarrollo

2.1 Funcionamiento

La herramienta *VCMG* genera mapas conceptuales a partir de presentaciones Web de *ExpoVision*. Los pasos que se siguen para la generación dinámica *Server-based* de los mapas conceptuales, como imágenes vectoriales *SVG*, son los siguientes:

1. Cuando un usuario consulta una presentación Web dentro del servidor de *ExpoVision*, aparece un hiperenlace dentro de la tabla de contenido de la presentación que invoca a *VCMG* para generar un mapa conceptual. Por ejemplo, tenemos la siguiente línea:

```
<embed src="map.php?f=ejemplo" width="900" height="600"
type="image/svg+xml"  />
```

2. VCMG (Script de PHP) verifica la existencia de un archivo auxiliar (z-file), que contiene un resumen de la presentación, generado en una consulta anterior, si no existe lo genera (paso 3). Si existe, verifica si la fecha de la presentación es más reciente que del archivo z-file auxiliar, si está actualizado pasa al paso 4.

3. Genera archivo auxiliar z-file. Se consulta el documento de la presentación y se extraen los elementos de texto necesarios para elaborar posteriormente un mapa conceptual. Para obtener dichos elementos se procede a: limpiar el documento de tags de HTML, espacios innecesarios, símbolos especiales y comandos del intérprete del generador de presentaciones Web de ExpoVision; derivar nombre corto de la presentación (concepto central); obtener títulos cortos de cada lámina (conceptos subordinados); generar resumen de láminas por temas (concatenar y compactar contenidos de láminas bajo un mismo título, además de señalar tablas, listados y figuras); derivar las relaciones entre los conceptos. Cabe mencionar que para realizar el análisis sintáctico anterior se utilizan intensivamente expresiones regulares de PHP. Por último, se graba en el servidor el archivo de texto auxiliar z-file que resume la presentación.

4. Generación dinámica de la imagen SVG: se usa una plantilla SVG como esqueleto de un mapa conceptual, dicha plantilla está embebida en un script de PHP. Se consulta el archivo auxiliar z-file, se analiza y construye un arreglo asociativo para representar los nodos y enlaces del mapa conceptual. Conforme se procesa el archivo z-file se registra e indexa cada elemento de texto, e.g. coordenadas para cada elemento gráfico, texto para cada nodo, relación, hiperenlaces y cuadros pop-up de texto. De acuerdo con el número de nodos resultante se procede a ubicar la información en los elementos gráficos SVG correspondientes.

Finalmente, aparece una nueva ventana en el navegador Web del usuario con una imagen que visualiza el mapa conceptual correspondiente para la presentación Web consultada.

5. El usuario puede interactuar con la imagen *SVG* del *MC* de distintas formas: al ubicar el cursor del ratón sobre un nodo, aparece un cuadro *pop-up* semi-opaco con un

resumen de las láminas asociadas a dicho concepto. Si el usuario se desplaza a otro lugar (fuera del nodo actual), el cuadro de texto desaparece gradualmente. En cambio, si el usuario da *click* sobre el nodo, la ventana del navegador que contiene la presentación *Web* despliega la lámina correspondiente para dicho concepto-nodo. Por otra parte, si selecciona un enlace nodo-a-nodo aparece un texto para describir la relación que existe entre un concepto y otro. Opcionalmente, el usuario puede ampliar/reducir la imagen, grabarla o imprimirla.

2.2 Componentes de la aplicación

A continuación se describe de manera visual cada uno de los principales componentes del gráfico *SVG* para un caso específico de un mapa conceptual que consta de cuatro nodos únicamente. Cada uno de sus elementos será explicado brevemente de manera alfabética.

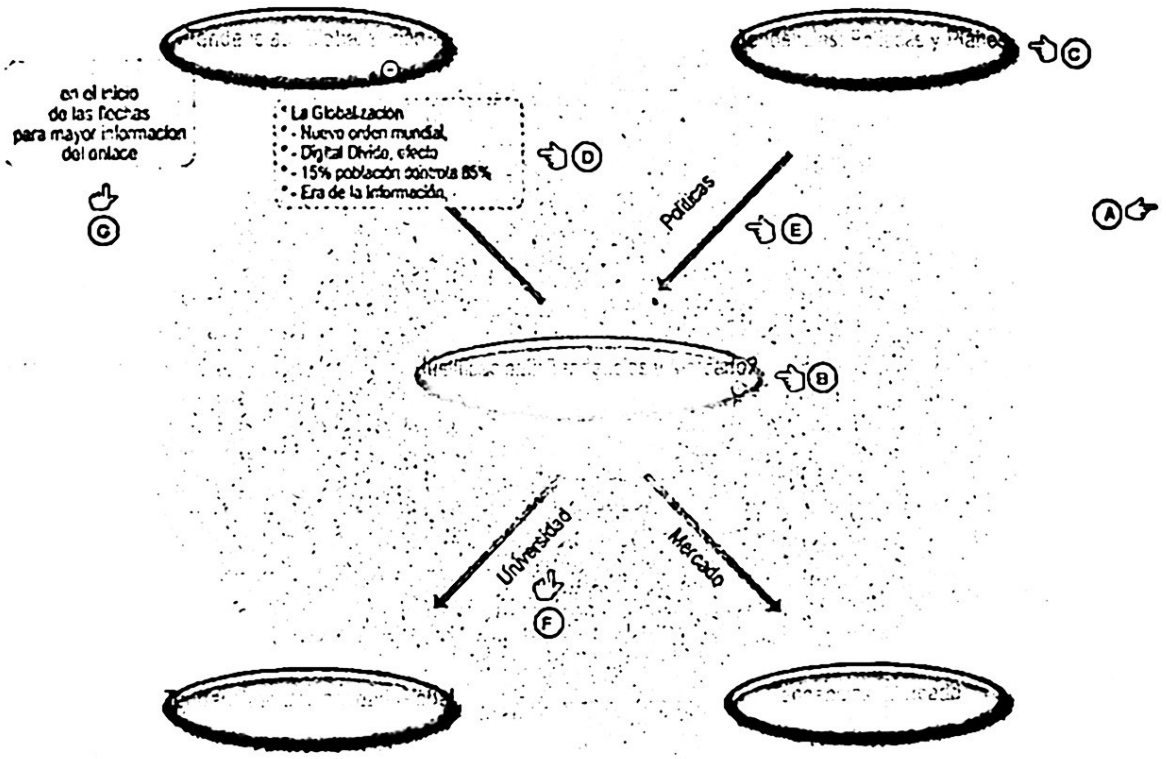


Fig. 1. Mapa Conceptual de Salida en Navegador Web

2.2.1 Gráfico vectorial (*SVG*) de salida

La figura 1 muestra la imagen completa generada tal y como aparece en el navegador Web del cliente.

2.2.2 Tema principal

Ubicado en el centro del gráfico, la figura 2 muestra la porción grafica donde aparece el nodo central (1) y arriba del nodo (2) se muestra el cuadro pop-up con la

información sobre el tema y los autores mostrada a partir de un evento "mouseover" (pose del cursor sobre el tema central).

Implementado mediante la etiqueta:

```
<g onmousedown="go(<?= $ID["#n{$n}pg"] ?>)"
opacity="0.5">
```

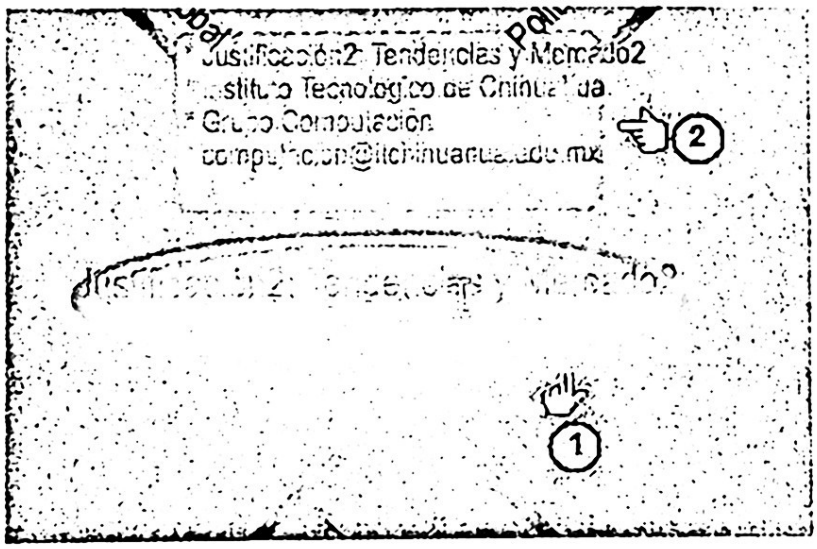


Fig. 2. Tema central e información adicional

2.2.3 Conceptos de tema principal

Los nodos externos, mostrados en la figura 3, representan conceptos del tema principal relacionados entre si mediante palabras enlace (1) y la capacidad de mostrar al usuario texto completo referente al tema mediante un evento de ratón (mousedown) (2).

Implementado mediante:

```
<g onmousedown= "go(0)" opacity="0.5">
```

2.2.4 Información adicional del concepto

La visualización de esta información es activada por el usuario mediante el evento del ratón "mouseover", el puntero es colocado sobre el concepto.

La figura 3 (2) provee un recuadro de fondo de opacidad media (*translúcido*) con información adicional sobre el concepto en cuestión implementado mediante un cambio de un atributo *CSS* para, por ultimo, cambiar a opacidad nula (desvanecerse) en un período de 2 seg. (*Animación*) después de haber retirado el usuario el ratón del concepto (*mouseout*).

Implementado mediante:

```
<set attributeName="opacity" attributeType="CSS" begin=
"n<?=$n?>.mouseover"  to="1" fill="freeze" /> <animate
attributeName="opacity" begin="n<?=$n?>.mouseout"
from="1" to="0" dur="2s" fill="freeze" />
```

En la figura 4 se muestra otra característica importante. El hecho que cuando aparecen recuadros de información, si éste aparece sobre algún otro texto (*e.g. palabras enlace*)

dicho texto se desvanecerá durante el tiempo que esté visible el recuadro de información adicional en pantalla, esto con el fin de proveer mayor legibilidad.

Implementado mediante:

```
<animate attributeName="opacity" begin= "n6.mouseover"
from="<?= $VIS['n6vis'] ?>" to="0" dur="1s"
fill="freeze"  />

<animate attributeName="opacity" begin="n6.mouseout"
from="0" to="<?= $VIS['n6vis'] ?>" dur="1s"
fill="freeze"  />.
```

Fig. 3. Conceptos del tema e información adicional del concepto

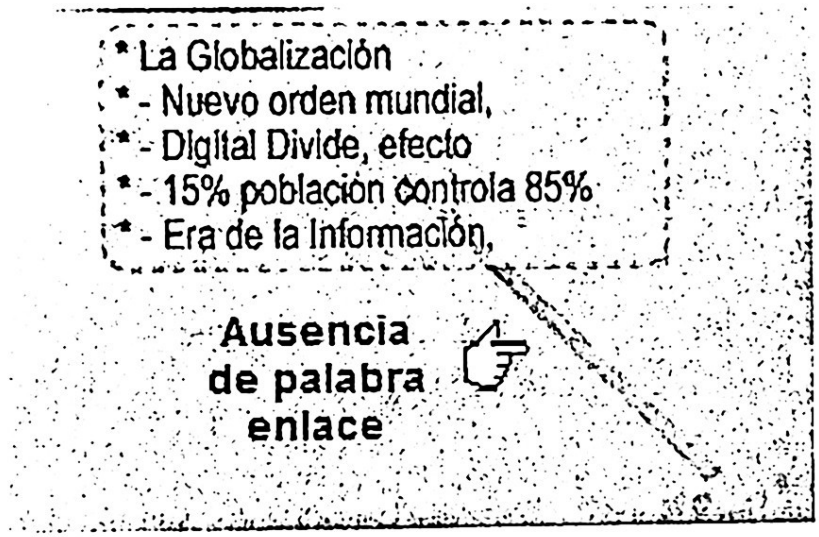


Fig. 4. Ocultamiento de palabras enlace

2.2.5 Relaciones nodo-a-nodo

Define brevemente la relación del concepto con respecto al tema principal mediante la dirección de su flecha.

Es posible conocer la relación mediante la identificación de palabras reservadas en el texto de entrada mediante análisis sintáctico implementado mediante expresiones regulares en PHP como se muestra:

```
If ( preg_match("/^%pg (\S+) (\S+)
([^:]+):(.+)$/S",$x,$t)) {

$t[2]=="to"?"#ArrowLeft":"#ArrowRight";
```

Además posee una manera muy particular de diferenciar su relación mediante diferenciación por color como se muestra en la figura 5, implementada mediante uso de hojas de estilo CSS.

```
<g id="ArrowRight">

    <path d="M 140 0 148 8 140 16 140 11 0 11 0 5 140
5" style="stroke-width:1; opacity:.4; fill:#909" />

 </g>

 <g id="ArrowLeft">

    <path d="M 0 8 8 0 8 5 148 5 148 11 8 11 8 16"
style="stroke-width:1; opacity:.4; fill:#900" />

 </g>
```

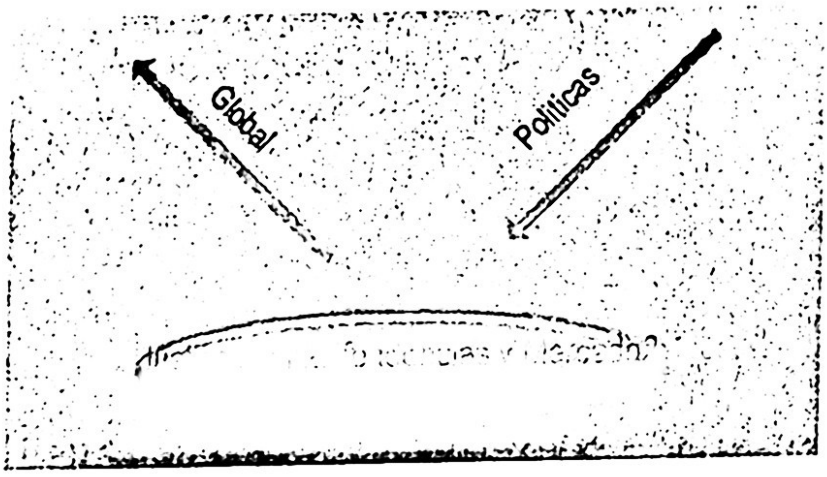


Fig. 5. Diferenciación de color en enlaces

2.2.6 Palabras enlace

Son las palabras que sirven para unir los conceptos y señalar el tipo de relación existente entre ambos [13].

Estas palabras son obtenidas mediante análisis sintáctico del texto de entrada. Implementado mediante:

```
if( preg_match("/^%pg (\S+) (\S+)
([^:]+):(.+)$/S",$x,$t)) {

$ID[$nId."Short"] = $t[3];
```

Las palabras son implementadas en el mapa conceptual mediante breves descripciones de texto alineadas sobre el "enlace nodo-a-nodo", siendo estos permanentemente mostrados.

2.2.7 Información adicional de la palabra enlace

Son obtenidos también mediante análisis sintáctico e implementadas en el mapa conceptual mediante descripciones mas detallas de la palabra enlace.

Implementado mediante:

```
if( preg_match("/^%pg (\S+) (\S+)
([^:]+):(.+)$/S",$x,$t)) $ID[$nId."Long"] = $t[4];
```

Son mostrados únicamente si el usuario realiza un evento de ratón "Mouse down" sobre el "hyperArrow" y desvanecidos cuando el botón del mouse es soltado (mouseup) como se indica a continuación en el siguiente código SVG.

```
<g id="n6linklong" opacity="0">
 <rect x="90" y="385" rx="8px" ry="8px" width="200"
height="32" class="tipce3" />
 <text x="100" y="405" class="d"> <?= $LONG['n6'] ?>
</text>
 <set attributeName="opacity" attributeTpe="CSS"
begin="n6link.mousedown" to="<?=$VIS['n6vis']?>"
fill="freeze" />
<animate attributeName="opacity" begin="n6link.mouseup"
from="<?=$VIS['n6vis'] ?>" to="0" dur="1s"
fill="freeze" /> </g>
```

3 Conclusiones

La herramienta multiplataforma VCMG está completamente implementada en tecnologías basadas en estándares abiertos W3C y GNU además de poder ser utiliza en cualquier navegador Web.

Es capaz de generar imágenes hipermedia interactivas de mapas conceptuales de manera autónoma y dinámica a partir de presentaciones tipo Power Point.

Al aplicar mapas conceptuales como soporte para la generación de presentaciones Web se pretende demostrar su viabilidad y utilidad para:

- Auxiliar en la elaboración de presentaciones más didácticas y efectivas.
- Automatizar y facilitar la generación de resúmenes visuales de presentaciones.
- Visualizar y navegar de una manera más intuitiva en una presentación.
- Favorecer el aprendizaje significativo por su representación visual del conocimiento.

Referencias

1. Javascript : The Definitive Javascript Resource. (2004)
   http://www.javascript.com/
2. PHP: Homepage Oficial de PHP. (2004)
   http://www.php.net
3. XML: Homepage Oficial de XML. (2003)
   http://www.w3.org/XML
4. W3C Recommendation : SVG 1.0 Specification. (Sep. 2001)
   http://www.w3.org/TR/SVG
5. W3C Recommendation : SVG 1.1 Specification . (Jan.2003)
   http://www.w3.org/TR/SVG11
6. W3C Working Draft : SVG 1.2 Specification. (Apr. 2003)
   http://www.w3.org/TR/SVG12
7. W3C : Sitio Oficial de World Wide Web Consortium. (2003)
   http://www.w3.org
8. W3C : Overview of SGML Resources. (2003)
   http://www.w3.org/MarkUp/SGML/
9. Pontificia Universidad Javeriana : Mapas conceptuales de Notoria (1997)
   http://venus.javeriana.edu.co/qualitas/mayo97.htm
10. Febles J., EstradaV.: Aplicaciones de la Inteligencia Artificial. Primera Edición, Parte III, Mapas Conceptuales. Pág. 400. Ed. Univ. de Guadalajara, México (2002)
11. Cuevas A.: Propuesta de Aplicación de los Mapas Conceptuales en un modelo Pedagógico Semipresencial. Instituto Superior Politécnico "José Antonio Echeverría" Cuba.
12. W3C : "XML Linking Language (XLink). (2003)
    http://www.w3.org/XML/Linking
13. W3C : CSS Level 2 Specification (CSS2). (May 1998)
    http://www.w3.org/TR/REC-CSS2.
14. Pacheco A. : Servidor de ExpoVision (2004)
    http://pantera.itch.edu.mx/apacheco/expo
15. Campesato O. : Fundamentals of SVG Programming: Concepts to Source Code. Ch.R. Media, (2004)
16. Adobe : SVG Viewer. (2001)
    http://www.adobe.com.
17. Le Hors A. et al.(2000) "Document Object Model (DOM) Level 2 Core Specification," (2000).
    http://www.w3.org/TR/DOM-Level-2-Core

Un sistema de extracción de información sobre desastres naturales

Alberto Téllez Valero[1], Manuel Montes-y-Gómez[1,2], Luis Villaseñor Pineda[1]

[1]Laboratorio de Tecnologías del Lenguaje
Instituto Nacional de Astrofísica, Óptica y Electrónica, México
[2]Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia, España
{albertotellezv, mmontesg, villasen}@inaoep.mx

Resumen. Hoy día el acceso a vastas cantidades de información dificulta su exploración y análisis de forma manual. Una manera de confrontar el problema es con *extracción de información*, donde la tarea es filtrar y estructurar de manera automática textos en lenguaje natural. En este trabajo se describe el sistema *Topo*, un sistema de extracción de información que toma como dominio de estudio noticias que reportan desastres naturales en español. Se presenta la arquitectura del sistema y los resultados alcanzados actualmente tanto en el filtrado de textos relevantes al dominio, así como en la identificación y extracción de entidades de información. Además, se muestran las conclusiones y el trabajo en proceso.

1 Introducción

El crecimiento explosivo de documentos en lenguaje natural disponibles en computadoras conectadas a la red alrededor del mundo dificulta su exploración y análisis. Por tal motivo, se hace necesario poder filtrar y estructurar información relevante al dominio de interés para contestar a muchas preguntas acerca del mismo. Una forma de lograr esto último de manera automática es con *extracción de información* (EI). La extracción de información es la tarea de localizar piezas específicas de datos desde documentos en lenguaje natural [1], la información extraída es entonces almacenada en una base de datos que puede ser examinada usando lenguajes de consultas estándar para facilitar su análisis.

En este artículo presentamos un sistema que realiza la tarea de extracción de información para el dominio de noticias que tratan sobre desastres naturales. El objetivo principal de nuestro sistema, al que hemos denominado *Topo*, es mostrar el uso de técnicas de clasificación automática de textos en la tarea de extracción de información. Además, se pretende mostrar que para la extracción de ciertas entidades de información es suficiente con emplear un análisis a nivel de palabras, sin tener que lidiar con el hasta hoy incompleto análisis lingüístico, especialmente para el idioma español. Esto último implica el poder extraer información sin tratar de comprender el contenido del texto, que es una postura contraria a lo que generalmente se piensa.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.89-98.© Centro de Investigación en Computación, IPN, México

El resto del documento se encuentra organizado de la siguiente manera. En la sección 2 describimos brevemente el trabajo relacionado. En la sección 3 se presenta el dominio de trabajo y la plantilla de extracción a llenar. La sección 4 exhibe la arquitectura del sistema. Los resultados preliminares se muestran en la sección 5. Finalmente, en la sección 6 se presentan las conclusiones y se describe el trabajo en proceso.

2 Trabajo Relacionado

Un auge importante en el desarrollo de sistemas de extracción de información se dio gracias a la intervención de la Agencia de Defensa de los Estados Unidos (DARPA, por sus siglas), quien fomentó las Conferencias de Entendimiento de Mensajes (MUC, por sus siglas en inglés), las cuales proporcionaron más de una década de experiencia en la definición, diseño y evaluación de este tipo de sistemas [2]. Entre otras cosas, los resultados del MUC demostraron que la extracción de información es una tarea difícil hasta para las personas, donde se reportó que los humanos podemos alcanzar un grado de exactitud entre el 60 y 80% en esta labor [1].

Generalmente, la mayoría de sistemas de extracción de información se basan en arquitecturas como la propuesta por Grishman [3]. En esta arquitectura se tiene una estructura modular, donde la salida de un módulo sirve como entrada del módulo siguiente. Una característica importante de la misma es el amplio uso de recursos lingüísticos para las tareas de análisis léxico, análisis sintáctico parcial y resolución de correferencia. Esto se debe principalmente a que se tiene la hipótesis que para lograr extraer información de un texto se debe entender en el mayor grado posible el mismo.

En contraste a la arquitectura de Grishman, Kushmerick *et al* [4] proponen una arquitectura novedosa, donde la hipótesis no es entender el texto, si no más bien encontrar las combinaciones de palabras (expresiones) que se utilizan para reportar la información que nos interesa extraer. La ventaja de esta arquitectura es que no se tiene que lidiar con un profundo análisis lingüístico, en su lugar se utilizan técnicas de clasificación automática de textos para encontrar las expresiones buscadas. Esta arquitectura fue presentada en el sistema llamado "Cambio de Dirección" (CoA, por sus siglas en inglés), el cual tiene como objetivo filtrar correos electrónicos que reportan un cambio de dirección de e-mail por parte del remitente, y posteriormente actualizar la agenda del destinatario con la información incluida en los mismos. Sin embargo, extraer direcciones de e-mail no refleja completamente los alcances de la arquitectura, por tal motivo nosotros pretendemos llevar a una tarea más compleja este trabajo, con el propósito de tratar de determinar sus límites.

Con respecto a extraer información de noticias que reportan desastres naturales, actualmente no tenemos antecedentes de la existencia de sistemas de extracción de información enfocados al dominio, sólo sabemos que existe una asociación que realiza esta tarea de forma manual (consultar referencia [5]).

3 Dominio

La información a ser extraída se define por medio de la plantilla de extracción, la cual se forma por una serie de atributos que la caracteriza. Los atributos pueden ser opcionales u obligatorios ya que la información puede o no estar presente en los documentos. Su construcción se realiza de antemano y dependen del dominio de trabajo, también llamado escenario, y de la información que se desea obtener.

En nuestro caso, el escenario en el que se decidió trabajar es el de noticias en español que reportan desastres naturales. La razón de su elección es que es un dominio rico en información para ser extraída. No obstante, la principal motivación para esta elección fue nuestra convicción respecto a que la disponibilidad de un inventario con dicha información, en combinación con un conjunto de herramientas para su adecuado análisis, permitirá adquirir un mejor conocimiento sobre los fenómenos naturales desastrosos, y con ello aprender a prevenir y minimizar sus efectos.

Debido a que son muchos los tipos de desastres naturales que se presentan, actualmente sólo nos enfocamos en seis de los más frecuentes en México (ver tabla 1). Las definiciones incluidas en la tabla 1 corresponden con las establecidas en la Guía Metodológica de DesInventar publicada en el 2003 [5], esta guía fue elaborada por la Red de Estudios Sociales en Prevención de Desastres en América Latina (LA RED). Entre otras cosas, DesInventar presenta una metodología de registro de información sobre características y efectos de diversos tipos de eventos. Por tal motivo, los datos a extraer en el presente proyecto, también corresponden con los establecidos por LA RED en su Ficha de Información de Desastres. La plantilla de extracción se muestra en la tabla 2.

Tabla 1. Dominio de estudio

Desastre	Definición
Helada	Disminución de la temperatura hasta el punto de congelación con efectos nocivos en la población, cultivos, bienes y servicios
Huracán	Anomalía atmosférica violenta que gira a modo de torbellino caracterizado por fuertes vientos, acompañados por lluvia
Forestal	Incendio. Incluye todos los incendios en campo abierto en áreas rurales, sobre bosques nativos, bosques cultivados y praderas
Inundación	Subida de aguas que supera la sección del cauce de los ríos o que se relaciona con el taponamiento de alcantarillas
Sequía	Temporada anormalmente seca, sin lluvias, o con déficit de lluvias. En general se trata de períodos prolongados
Sismo	Todo movimiento de la corteza terrestre que haya causado algún tipo de daño o efecto adverso sobre comunidades o bienes

Tabla 2. Plantilla de extracción

Relacionados con el desastre	
Fecha	Fecha de ocurrencia del desastre
Lugar	Nombre del lugar o lugares donde ocurrió el fenómeno
Magnitud	Valores de magnitud internacionalmente usados para sismo y huracán, para otros tipos de eventos variables cuantificadas
Relacionados con las personas	
Muertos	Número de personas fallecidas por causas directas
Heridos/ Enfermos	Número de personas que resultan afectadas en su salud o integridad física, sin ser víctimas mortales, por causa directa del desastre
Desaparecidos	Número de personas cuyo paradero a partir del desastre es desconocido
Damnificados	Número de personas que han sufrido grave daño directamente asociados al evento en sus bienes o servicios
Afectados	Número de personas que sufren efectos secundarios asociados a un desastre
Relacionados con las viviendas	
Destruidas	Número de viviendas arrasadas, sepultadas, colapsadas o deterioradas de tal manera que no son habitables
Afectadas	Número de viviendas con daños menores, no estructurales o arquitectónicos, que pueden seguir siendo habitadas
Relacionados con la infraestructura	
Vías	Longitud de redes viales destruidas o inhabilitadas en metros
Hectáreas	Número de áreas de cultivo, pastizales o bosques destruidas o afectadas
Ganado	Número de unidades perdidas (bovinos, porcinos, caprinos, avícolas)
Centros educativos	Número de guarderías, colegios, universidades, centros de capacitación, etc. destruidas o afectadas directa o indirectamente por el desastre
Centros de salud	Número de centros de salud, clínicas, hospitales destruidos o afectados directa o indirectamente por el desastre
Perdida económica	Monto de las pérdidas directas causadas por el desastre

4 Arquitectura

Básicamente, para realizar extracción de información bajo la arquitectura presentada por Kushmerick *et al* [4], se requiere en primer lugar filtrar los textos relevantes al dominio de estudio, posteriormente detectar las entidades con posibilidad de ser extraídas, y finalmente discriminar entre las entidades identificadas las que proporcionan información útil para llenar la base de datos. El resultado final es una base de datos constituida por la colección de plantillas generadas a partir de los textos filtrados.

En la figura 1 se muestra la interfaz del sistema *Topo*, en esta figura se presenta el ejemplo de una noticia que reporta un "sismo", y con la certeza de que es un texto relevante al dominio, el sistema se ocupa de identificar las entidades con posibilidad de ser extraídas (las frases sombreadas en el cuadro de texto *Noticia*). Para posteriormente, discriminar entre estas entidades aquellas que son útiles para llenar la plantilla de extracción. Para efectuar esto último, a cada una de las entidades se les calcula la probabilidad de que formen parte o no en los registros de la plantilla (cuadro de texto *Distribución*), y se toman como resultado las que tienen una mayor probabilidad. De la información extraída en el ejemplo (panel *Plantilla*), podemos concluir que se reporta un sismo de 7.6 grados que tuvo lugar en Colima, México, dejando un saldo de 28 muertos, 300 heridos o enfermos y 10 mil viviendas afectadas. Lo anterior refleja la utilidad de este tipo de sistemas.

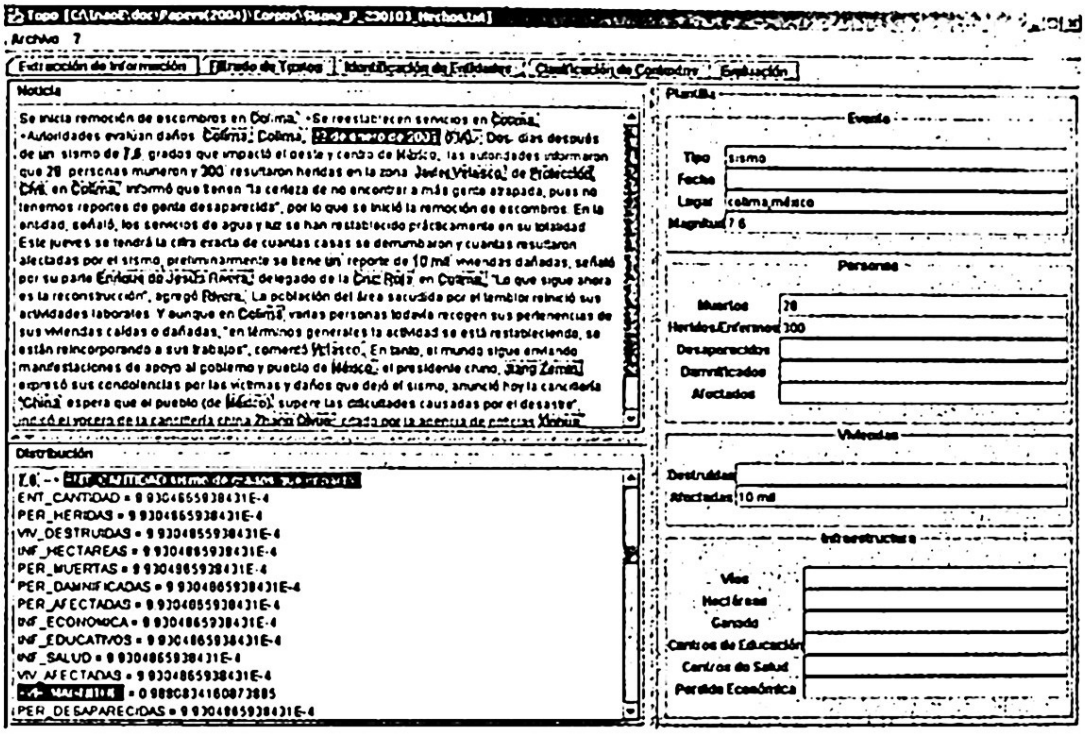


Fig. 1. Interfaz del sistema

El diseño arquitectónico del sistema *Topo* se muestra en la figura 2, este diseño consiste de un modelo estructural compuesto por tres componentes de programa. Donde, la salida del primer componente es un condicionante para la ejecución de los dos siguientes, y la salida del segundo componente sirve como entrada del último. Además, cabe resaltar que el trabajo de extracción recae principalmente en un análisis a nivel de palabras y el uso de clasificadores de textos. Estas características proporcionan flexibilidad al sistema para ser adaptado a nuevos dominios. En las subsecciones siguientes se explica más a detalle cada uno de los componentes de la arquitectura.

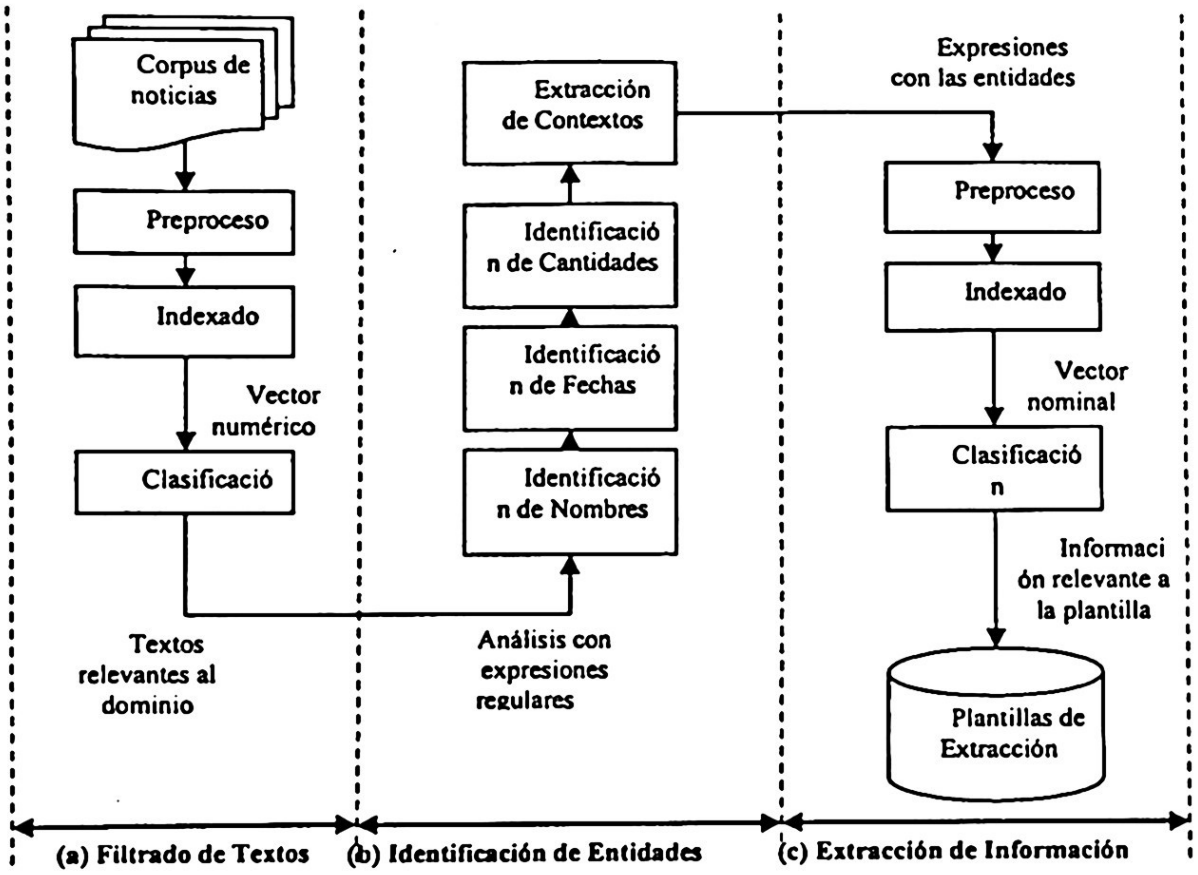


Fig. 2. Arquitectura del Sistema

4.1 Filtrado de texto

El objetivo de este componente es filtrar aquellos documentos que son relevantes al dominio de trabajo. Una forma efectiva de realizar esta tarea es utilizando técnicas de *clasificación automática de textos* (CT). Donde la clasificación de textos se define como la tarea de ubicar correctamente y de manera automática textos en lenguaje natural que contienen información no estructurada en un conjunto de categorías predefinidas [6]. Para la construcción del clasificador se utilizaron técnicas de aprendizaje automático (para más detalle ver referencia [7]). Generalmente, el proceso de clasificar un texto consiste en:

1. Realizar la extracción de características para transformar el texto de su formato inicial a una representación adecuada para la tarea de clasificación.
2. Aplicar el método de clasificación.

En nuestro caso, el modelo utilizado para la representación fue el vectorial junto con un indexado booleano [6]. En otras palabras, el documento está representado por un vector de 0's y 1's que indican la presencia o ausencia de ciertas palabras en el mismo. Para mejorar el indexado se aplicó un preproceso al texto donde se eliminaron símbolos de puntuación (comas, puntos, etc.). Con respecto al método de clasificación, el que se está utilizando actualmente es el simple de Bayes (ver sección

4.3 para más detalle), el cual fue entrenado con un corpus de 471 noticias, de las cuales el 48% es relevante y el 52% restante es irrelevante[1].

Una ventaja de utilizar un clasificador de textos es que el componente puede ser fácilmente adaptado a un nuevo dominio. Para esto, sólo es necesario contar con un conjunto de documentos previamente clasificados del dominio de interés, esto para entrenar nuevamente el método de clasificación.

4.2 Identificación de entidades

Este componente se encarga de detectar aquellas partes del texto con posibilidad de ser incluidas en la plantilla de extracción. En este caso, las entidades probables son: *Nombres* que puedan reflejar el lugar del evento; *Fechas* que puedan reportar la ocurrencia del mismo, y *Cantidades* relacionadas con la magnitud del evento o con efectos sobre personas, viviendas o infraestructura.

Para identificar las entidades se utiliza un análisis con expresiones regulares a partir de los componentes léxicos de la gramática siguiente:

Ent_nombre	→	**nombre | nombre con_nombre** ent_nombre
ent_fecha	→	**mes | mes con_fecha número | número con_fecha** ent_fecha
ent_cantidad	→	**número(. número)? | número(. número)?** ent_cantidad

Donde los terminales generan conjuntos de cadenas dados por las siguientes definiciones regulares:

nombre	→	[A-Z][A-Za-z]*
con_nombre	→	(de | la | ... | se)*
mes	→	enero | ... | diciembre
con_fecha	→	de | - | ... | ϵ
número	→	[0-9]+

Además, a las definiciones regulares *nombre* y *número* se les agrego, respectivamente, un diccionario de expresiones que representan excepciones a la definición. Por ejemplo, las palabras al principio de una oración que inician con letra mayúscula y que no son nombres propios. (*El* sismo ...), y los números reportados con letras, o una combinación de letras y números (... dejó *mil 500* muertos).

Este tipo de análisis nos ha resultado en una baja precisión para identificar nombres, pero en una alta cobertura para todos los casos. La completa cobertura es importante para no dejar fuera del proceso de extracción entidades con alguna probabilidad de formar parte en la plantilla. Finalmente, después de identificar las

[1] Por textos relevantes entendemos todos aquellos que contienen información a ser extraída, mientras que los irrelevantes son los que contienen palabras o frases usadas comúnmente en la descripción de un fenómeno natural, pero que en estos casos se usan en contextos muy diferentes. Por ejemplo, la frase "ojo del huracán" en el contexto de "el presidente está en el ojo del huracán".

entidades, este componente se encarga de extraer los contextos de cada entidad, donde el contexto de la entidad es la expresión donde se encuentra la misma.

También, cabe mencionar que el tipo de entidades identificadas actualmente pueden ser útiles para detectar información en otros dominios, y no exclusivamente del escenario actual, por lo que este componente es completamente adaptable a tareas de extracción similares.

4.3 Extracción de información

Para filtrar entre las entidades identificadas aquellas que son útiles a la plantilla, tomamos a la tarea de extracción de información como una tarea de clasificación de textos, con la variante de que lo que estamos clasificando es el contexto de las entidades identificadas en lugar de los textos completos. Esta clasificación es la que nos permite conocer la probabilidad de que la entidad forme parte en alguno de los registros de la plantilla, o bien que no sea tomada en cuenta.

Para hacer la clasificación de un contexto, la parte de extracción de características es similar a la explicada en la tarea de filtrar un texto (sección 4.1), con la única diferencia que en el indexado el vector resultante tiene como entradas atributos nominales (palabras del contexto) y no atributos numéricos (ponderado Booleano). Finalmente, para discriminar los contextos se usan tres clasificadores de textos, los cuales fueron entrenados con una colección de 2,364 expresiones formadas de seis palabras, las cuales fueron obtenidas de 90 documentos que tratan sobre incendio forestal y sismo. En las expresiones el 26% representan información útil para la plantilla, y el 74% restante son entidades identificadas pero que no deben ser extraídas.

El uso de tres clasificadores se debe a que se especializó cada uno de ellos en el contexto de nombres, fechas y cantidades respectivamente. Los algoritmos utilizados hasta el momento son el simple de Bayes para las fechas y vecinos más cercanos para los otros dos casos.

El método simple de Bayes es del tipo probabilístico y es construido utilizando el conjunto de entrenamiento para estimar la probabilidad de cada clase dadas las características de los textos. Para evaluar dicha probabilidad se utiliza una simplificación del teorema de Bayes:

$$P(c_j \mid d) = P(c_j) \prod_{i=1}^{M} P(d_i \mid c_j) \tag{1}$$

Donde M es el número de términos en el modelo vectorial del texto d_i y $1 \leq j \leq k$, donde k es el número de clases posibles. Las probabilidades $P(c_j)$ y $P(d_i \mid c_j)$ se calculan de la siguiente manera:

$$P(c_j) = \frac{N_j}{N} \qquad P(d_i \mid c_j) = 1 + N_{ij} \Big/ M + \sum_{k=1}^{M} N_{kj} \tag{2}$$

Aquí N es el número de documentos en el conjunto de entrenamiento, N_j es el número de documentos en el conjunto de entrenamiento con clase c_j, y N_{ij} es el número de veces que la palabra i ocurre dentro de los textos con clase c_j en el conjunto de entrenamiento.

Con respecto al método de vecinos más cercanos, este es un método de aprendizaje basado en instancias que consiste en almacenar los datos de entrenamiento para que dado un nuevo texto, se busque en los datos almacenados un caso similar y se clasifique en base a la clase de ese ejemplo similar. Se usa una función de distancia para determinar cual miembro del conjunto de entrenamiento es el más cercano al nuevo caso. La función de distancia más usada es la distancia Euclidiana [6].

Un aspecto importante a resaltar es que el uso de clasificadores de textos permite adaptar el componente a nuevos dominios. Para entrenar nuevamente los clasificadores es necesario contar con una colección de contextos de las entidades nombradas para el nuevo escenario, y además indicar cuales de las expresiones representan información útil a la plantilla y que registro les corresponde.

5 Resultados preliminares

Para el proceso de evaluación utilizamos las medidas de precisión y cobertura, donde el objetivo es valorar la respuesta del sistema basándose en las decisiones del experto. En la evaluación del filtrado de textos y la extracción de información se utilizó el método de validación cruzada con 10 pliegues (10FCV, por sus siglas en inglés) [8]. Los resultados alcanzados hasta el momento se muestran en la tabla 3. Cabe mencionar que los clasificadores incluidos en el componente de extracción de información actualmente han sido entrenados con contextos de tamaño seis (tres palabras a la izquierda y tres palabras a la derecha para cada entidad), y aún no estamos seguros que este tamaño sea el mejor para todos los casos, por tal motivo los resultados se consideran preliminares.

Tabla 3. Resultados Preliminares

Componente	Cobertura	Precisión
Filtrado de textos	96%	96%
Identificación de entidades	99%	88%
Extracción de información	71%	72%

6 Conclusiones

En base a los resultados preliminares podemos concluir que es posible hacer extracción de información de entidades que están de forma explícita en los textos utilizando únicamente un análisis a nivel léxico junto con métodos de clasificación de textos. Esta conclusión es importante porque se muestra que para tareas como la presentada en este trabajo, no es necesario tener un amplio entendimiento del lenguaje para lograr extraer información, que es lo que generalmente se intenta. Además, esto fue probado para un dominio más complejo que el presentado por Kushmerick et al [4].

Hasta el momento, el principal problema que hemos afrontado es extraer el nombre del lugar donde ocurrió el evento. Sin embargo, como se mencionó anteriormente, los resultados que presentamos son preliminares y creemos que podemos mejorar los mismos de la siguiente manera:

1. Experimentando con el tamaño de los contextos, para encontrar el tamaño adecuado para discriminar entre las entidades identificadas.
2. Probar algoritmos de clasificación, para encontrar los que mejor se adapten a la tarea. Para esto hacemos uso del software WEKA [8].

Actualmente, continuamos preparando la colección de contextos para ampliar el componente de extracción de información a los demás eventos naturales propuestos en el dominio de trabajo (ver tabla 1). Como trabajo a futuro pretendemos incorporar a las etapas de preproceso nuevas características, por ejemplo: obtener las partes de la oración. Intentando con esto mejorar los resultados en la tarea de clasificación, y por consiguiente en la extracción.

Finalmente, cabe mencionar que el presente trabajo forma parte del proyecto de investigación *Recolección, Extracción, Búsqueda y Análisis de Información a partir de Textos en Español*, el cual entre sus metas tiene el usar la base de datos sobre desastres en estudios exploratorios y preventivos posteriores.

Agradecimientos

El presente trabajo fue parcialmente financiado por el CONACYT (Proyecto U39957-Y). Asimismo, el primer autor agradece al CONACYT por el apoyo otorgado a través de la Beca para Estudios de Maestría # 171610. Por su parte, el segundo autor agradece a la Secretaría de Estado de Educación y Universidades, España.

Referencias

1. Appelt, D., Israel, D.: Introduction to Information Extraction Technology. A Tutorial Prepared for IJCAI-99 (1999)
2. Chinchor, N.: MUC-7 Test Scores Introduction. In Proceedings of the 7th Message Understanding Conference. Morgan Kaufmann (1997)
3. Grishman, R.: Information Extraction: Techniques and Challenges. Lecture Notes in Artificial Intelligence 1299 (1997)
4. Kushmerick, N., Johnston, E., McGuinness, S.: Information extraction by text classification. Workshop on Adaptive Text Extraction and Mining, Seattle (2001)
5. LA RED: Guía Metodológica de DesInventar. OSSO/ITDG, Lima (2003)
6. Aas, K., Eikvil, L.: Text Categorisation: a Survey. Technical Report, Norwegian Computing Center (1999)
7. Téllez, A., Montes, M., Fuentes, O., Villaseñor, L.: Clasificación Automática de Textos de Desastres Naturales en México. Congreso Internacional en Investigaciones de Ciencias Computacionales, México (2003)
8. Witten, I., Frank, E.: Data Minning: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, Sydney (2000)

Desarrollo de herramienta para lexicógrafo con detección automática de relaciones semánticas implícitas

Wilbert A. Olán Cristobal, Alexander Gelbukh, Grigori Sidorov

Laboratorio de lenguaje natural,
Centro de Investigación en Computación,
Instituto Politécnico Nacional,
Av. Juan de Dios Batiz s/n, esq. Mendizabal, Zacatenco, 07738,
México, D. F.
wolan@correo.cic.ipn.mx, {gelbukh,sidorov}@cic.ipn.mx

Resumen. Uno de los problemas más importantes en lexicografía es la existencia de círculos viciosos en definiciones en los diccionarios. Se presenta una herramienta que ayuda a encontrar los ciclos mencionados, escoger algunas palabras como primitivas o miembros del vocabulario definidor y analizar el impacto de esta selección a los círculos viciosos, es decir, encontrar las relaciones semánticas implícitas y depurarlas.

1 Introducción

A partir de los sistemas de diccionarios explicativos existentes actualmente, existe una parte de ellos la cual no tiene alguna solución práctica. Referimos a un problema en los diccionarios explicativos que es la presencia de los círculos viciosos en definiciones. Por ejemplo,

Gallina: hembra de gallo
Gallo: macho de gallina

Este es un problema en los sistemas de definiciones ya que lo anterior equivale a decir que *gallina es hembra de macho de gallina*, y esto no ayuda a entender lo que es *una gallina* sin saberlo de antemano.

Este es el círculo vicioso de longitud 1, hay círculos más largos. Sin embargo, el verdadero problema es que no se puede evitar los círculos de este tipo porque todo conjunto de palabras se define a través de mismo conjunto. Existen dos posibles soluciones

- Tolerar los círculos y solamente tratar de obtener los círculos de mayor longitud, lo que da la ventaja para un lector humano, porque se aumenta la probabilidad de conocer alguna de las palabras en el círculo, y
- Declarar algunas palabras como las palabras "primitivas" y no dar definiciones algunas para ellas. Es el enfoque más aceptable para las computadoras.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.99-104.© Centro de Investigación en Computación, IPN, México

que se expresa por el verbo *to want* (*querer*), mientras *feel like* (*tener ganas de*), por otra parte, lo debilita.

Así se puede establecer que *to want* (*querer*) no es la palabra semánticamente simple. Además, las acumulaciones semánticas las cuales complican el sentido de propio *to want* son específicas en el idioma inglés.

Se propone la siguiente definición de las palabras primitivas: La palabra se considera como una primitiva si el lenguaje dado no tiene ningún conjunto de las palabras vía las cuales puede ser explicada. Es decir, en este caso tanto *to want*, como *to wish* son las palabras primitivas, y existe una cantidad significativa de las palabras primitivas evaluada en más de 5000 elementos, porque solo las palabras que pueden ser definidas de modo claro se consideran no primitivas.

2.3 Vocabulario definidor

Antes que nada, se representa el diccionario explicativo como un grafo dirigido — cada palabra que tiene definición es un vértice, y las palabras que se encuentra en la definición están conectadas con ella. En su turno, la palabra puede formar definiciones de otras palabras.

Esta idea de representar el diccionario como un grafo es con la finalidad de verlo como una red semántica, y no es nueva, es una idea desarrollada por autores como Evens (1988) y Fellbaum (1990). Kozima y Furugori (1993) también analizan una red semántica, en este caso para saber que palabras "se activan" empezando de alguna palabra determinada.

Obviamente, el diccionario no contiene en sus definiciones las palabras que no son definidas en el mismo diccionario.

En este grafo se puede elegir algunas palabras como las primitivas semánticas. Las primitivas semánticas son las palabras las cuales pertenecen a conjunto de vértices y se marcan de tal modo que existe una ruta en el grafo de cualquier longitud para definir las demás palabras.

A diferencia con (Gelbukh and Sidorov, 2002) vamos a distinguir las primitivas semánticas y el vocabulario definidor.

El vocabulario definidor ese define de la misma forma con la única diferencia que la ruta debe tener longitud uno. Es decir, vocabulario definidor es mucho más interesante que las puras primitivas semánticas. De hecho, vocabulario definidor es un conjunto de las primitivas con cierta propiedad adicional.

Existen algunos diccionarios para inglés donde el vocabulario definidor se forma manualmente, por ejemplo, los de Oxford o de Collins. El número total de los elementos de vocabulario definidor es alrededor de 2000-3000 palabras. El número de primitivas muy parecido se reporta en (Gelbukh and Sidorov, 2002), (Rivera, 2003) para la detección automática de las primitivas semánticas –alrededor de 2000 palabras primitivas.

3 Características de la herramienta

En base del algoritmo descrito en (Gelbukh and Sidorov, 2002), (Rivera, 2003), se desarrolló la herramienta de ayuda a un lexicógrafo a la investigación de las estructuras del diccionario con el fin de detectar y corregir los círculos viciosos cortos a través de búsqueda automática de las primitivas semánticas y del vocabulario definidor.

La herramienta proporciona la siguiente información:

- Muestra un visor del Diccionario (en nuestro caso usamos el diccionario del español del grupo Anaya) de manera electrónica, donde muestra entre otras cosas la palabra del diccionario, su definición, la definición normalizada con las partes de oración de todas las palabras y, como una opción, también con sentidos de las palabras marcados. Nótese que el diccionario fue preprocesado usando la herramienta de análisis morfológico automático (Gelbukh and Sidorov, 2003).
- En este mismo visor nos muestra de las palabras existentes en la base de datos, el número de homónimos y de significados que tienen y cual es la parte de oración que generalmente juega.
- En este mismo apartado cuenta con los servicios de búsqueda, ya sea de manera común insertando la palabra a buscar en un cuadro de texto o bien búsquedas avanzadas en formato SQL, donde se puede dar los criterios de la búsqueda como, por ejemplo, parte de oración, etc.
- La funcionalidad de este modulo consiste en poder añadir más palabras con su significado o corregir la definición existente. En base del analizador morfológico de manera automática se transforman las palabras en definición en la forma normalizada. Existe la posibilidad de corregir manualmente los resultados de análisis automático.
- Para cada palabra se muestra la información si es la palabra primitiva o la palabra de vocabulario definidor. Se puede agregar o quitar la palabra seleccionada en la lista de definidores y observar la estadística de los círculos viciosos para todo el diccionario y para la palabra seleccionada. El algoritmo que se aplica es el algoritmo presentado en (Gelbukh and Sidorov, 2002), (Rivera, 2003).

Ejemplos de los ciclos presentados:

1) acción<117> -> impacto<6563> -> impresión<6601> -> cuerpo<3445> -> objeto<8643> -> ejercicio<4458> -> acción<117>
2) acción<119> -> obra<8666> -> acción<119>
3) aceite<126> -> bacalao<1418> -> aceite<126> etc.

4 Conclusiones

Se presentó el sistema que da la flexibilidad a un lexicógrafo de poder el mismo manualmente establecer a algunas palabras candidatas como primitivas semánticas puras o miembros del vocabulario definidor, esto con un análisis y criterios válidos

para poder considerarla como tal. El sistema permite agregar las palabras con sus definiciones al diccionario explicativo y corregir las definiciones existentes con análisis automático de los círculos viciosos. Sin embargo, las decisiones finales se toman por el lexicógrafo en base de la información proporcionada por el sistema.

Referencias

1. Apresjan, J. (2000) *Semantic Lexicography*, Oxford University Press. New York. 286 pp..
2. García Quesada, M. (2001). *Estudios de lingüística Española*. Universidad de Granada. Volumen 14 (2001) http://elies.rediris.es/elies14/index.html.
3. Gelbukh, Alexander and Grigori Sidorov. (2003) Approach to construction of automatic morphological analysis systems for inflective languages with little effort. In: *Computational Linguistics and Intelligent Text Processing. Proc. CICLing-2003, 4th International Conference on Intelligent Text Processing and Computational Linguistics*, February 15–22, 2003, Mexico City. Lecture Notes in Computer Science N 2588, Springer-Verlag, pp. 215–220.
4. Gelbukh, Alexander and Grigori Sidorov. (2002) Automatic Selection of Defining Vocabulary in an Explanatory Dictionary. *Proc. CICLing-2002, Conference on Intelligent Text Processing and Computational Linguistics*, February 16–23, 2001, Mexico City. Lecture Notes in Computer Science N 2276, Springer-Verlag, pp. 300–303.
5. Evens, M. N. (ed.), (1988). *Relational models of lexicon: Representing knowledge in semantic network.* Cambridge: Cambridge University Press.
6. Kozima, H. and Furugori, T. (1993). Similarity between words computed by spreading activation on an English dictionary. Proc. *6th conf. of the European chapter of ACL*, pp. 232–239.
7. Rivera, G. (2003) *Selección automática de primitivas semánticas para un diccionario explicativo del idioma español*, Tesis de Maestría (director A. Gelbukh, co-director G. Sidorov), CIC, IPN, México D.F.
8. Wierzbicka, A. (1980), *Lingua Mentalis: The semantics of natural language*. New York: Academic Press.
9. Wierzbicka, A. (1990), Prototypes save: on the uses and abuses of the notion of "prototypes" in linguistics and related fields, in S. L. Tsohatzidis (ed.), *Meanings and Prototypes: Studies in Linguistic Categorization*. London: Routledge & Kegan Paul.
10. Wierzbicka, A. (1996), *Semantics: Primes and Universals*. Oxford: Oxford University Press.

Descripción formal de secuencias nominales hispanas

Noé Alejandro Castro Sánchez, José Angel Vera Félix, Igor A. Bolshakov

Centro de Investigación en Computación,
Instituto Politécnico Nacional
México, D. F.
{ncastro, javera}@sagitario.cic.ipn.mx
igor@cic.ipn.mx

Resumen. A las secuencias de nombres y apellidos que forman nombres oficiales para habitantes de los países hispanohablantes, las denominamos Secuencias Nominales (SN) Hispanas. Se propone la descripción de las SN en forma libre y con una gramática generativa. Esta gramática permite desarrollar un algoritmo y un programa de análisis de las SN en textos electrónicos. Se demuestra que el programa distingue la estructura de las SN de manera comparable a como lo realizaría un ser humano.

1 Introducción

A las secuencias de nombres y apellidos que forman nombres oficiales para habitantes de los países hispanohablantes, las denominamos **Secuencias Nominales (SN) Hispanas.**

Es bien conocido que las SN Hispanas son unas de las más complejas entre los lenguajes europeos. Se llegan a encontrar SN con tres nombres seguidas por los apellidos paterno y materno. Es decir, es posible encontrar SN con cinco o más palabras separadas; e.g., *Juan Luis Díaz de León Santiago.*

La estructura de las SN es clara para cada hispanohablante, por lo que puede ser estrictamente formalizada. Las aplicaciones de la formalización incluyen al menos:

- Reconocer los nombres humanos en textos electrónicos arbitrarios, que es un problema bien conocido para varios lenguajes en lingüística computacional [1-4], [6-10];
- Extraer de los títulos de trabajos científicos los apellidos de autores hispanos, solucionando el problema de atribuir trabajos únicamente a autores hispanos específicos, para después recuperarlos correctamente de bases de datos e Internet, e.g., identificar artículos de un determinado autor hispano, aún sin importar las posibles variantes con la que puede aparecer su nombre, i.e., Gregorio Pastor Aldama, ó Pastor Aldama Gregorio, ó G. Pastor A., etc.; ventaja que, por ejemplo, un motor de búsqueda no proporciona.
- Automatizar el control de los nombres de ciudadanos realizado por los organismos oficiales para identificar personas de manera única y no ambigua.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp. 105-116.© Centro de Investigación en Computación, IPN, México

El objetivo de esta investigación es proponer una gramática generativa que estudie todos y cada uno de los elementos que participan en la estructuración de nombres hispanos más comunes y que establezca las reglas necesarias y suficientes que guíen la construcción de SN. De igual forma se considera el desarrollo de un algoritmo capaz de detectar, en textos electrónicos, SN, descomponiendo su estructura en sus elementos constituyentes.

2 Descripción libre de las secuencias nominales

Para analizar y comprender la estructura de una SN hispana, es necesario identificar todo elemento que la constituye.

Consideramos como una **Secuencia Nominal Completa** aquella que se encuentra constituida por una **Secuencia de Nombres** y una **Secuencia de Apellidos.**

2.1 Secuencia de nombres

La Secuencia de Nombres consiste de uno o varios elementos. Tradicionalmente solía constituirse por una larga serie de nombres. Actualmente, por razones prácticas, se limita su número y es más usual encontrar casos de nombres únicos, e.g., *Juan, Ana, Pedro, María*, etc. Dado que es muy raro encontrar casos con más de tres elementos, para efectos de nuestra gramática consideraremos como incorrecta aquella secuencia que rebase este número.

Denominamos **Nombres Solos** a los nombres antes mencionados. Los agrupamos como **Nombres Solos Masculinos** (*Noé, Ángel, Raúl*, etc) y **Nombres Solos Femeninos** (*Josefina, Carlota, Alina*, etc.).

Estructuras más complejas se forman al relacionar dos Nombres Solos a través de una preposición articulada, formación que denominamos **Nombre Compuesto.** Estrictamente hablando, esta estructura la concebimos como la unión de un Nombre Solo y un **Complemento Deado,** el cual está conformado por la preposición (*de*) y un artículo (no obligatorio), concatenados al segundo Nombre Solo. Ejemplos de Complementos Deado son *de la Luz, del Refugio, del Carmen, de Jesús*, etc. Los Nombres Compuestos son del tipo: *María del Refugio, José de Jesús, Antonio de la Caridad*, etc., teniendo la flexibilidad de combinar cualquier Nombre solo, independientemente de su sexo, con cualquier Complemento Deado.

Finalmente, identificamos estructuras que representan concatenaciones de dos Nombres Solos de sexo opuesto. Este tipo de agrupaciones presenta restricciones formativas motivadas seguramente por una estética de tipo fonético, es decir, no es compatible cualquier combinación arbitraria; de tal forma, encontramos un número de combinaciones muy limitado: *José María, María José, José Trinidad*, etc., pero no son aceptadas tuplas de tipo *Antonio Laura, Alejandro Fabiola, Gabriela Humberto*, etc. Este tipo de relaciones las denominamos **Nombres Combinados**, y en términos estrictos constituyen dos Nombres Solos.

2.2 Secuencia de apellidos

Un apellido hispano se conforma por dos secciones de elementos, conocidos como Apellido Paterno y Apellido Materno respectivamente. Esta diferencia presentada obedece únicamente a un orden preestablecido para señalar qué apellido de una persona corresponde al padre y cuál a la madre.

Aunque en un principio el sistema puede parecer confuso para alguien no habituado a él, ofrece muchas ventajas. Esta disposición de los apellidos ha dado origen a la isonomía o isonimia, que no es más que el estudio de la frecuencia y distribución de apellidos en poblaciones humanas, mediante el cual pueden establecerse relaciones de parentesco y origen. De esta forma, los métodos isonímicos que analizan la composición de apellidos en una población, proporcionan una válida información de la estructura genética de la misma, y constituyen un instrumento habitual en estudios de dinámica evolutiva de poblaciones humanas.

Las secuencias de apellidos más sencillas y usuales son las compuestas por un solo elemento, e.g., *Martínez, Castro, Rodríguez*, etc. A estas secuencias las denominamos **Apellidos Solos.**

Un conjunto que podemos señalar como extensión del anterior, es el que contempla Apellidos Solos antecedidos por la preposición *De*, acompañada o no por algún artículo. A diferencia de formaciones recíprocas vistas anteriormente, los **Apellidos Deado,** no son considerados como complementos, i.e., no requieren concatenarse a otro elemento para determinarse como apellidos. Como ejemplos podemos mencionar: *De León, Del Valle, De la Barrera, De los Cobos*, etc.

Una práctica dada por uso social y en determinados ambientes, permite a una mujer casada utilizar la fórmula "De + <apellido del esposo>"; por ejemplo., teniendo las secuencias *Alicia Juárez Moreno* y *Rigoberto Almodóvar Hernández*, la esposa, en actos sociales, puede utilizar la formación *Alicia Juárez De Almodóvar*; y en caso de enviudar, puede aparecer en una nota social de prensa como *Alicia Juárez Viuda de Almodóvar*. Sin embargo, estas fórmulas no se usan con frecuencia, y sobre todo, no pueden aparecer nunca en un documento oficial o legal, censo, acta de Registro Civil, etc.

Una formación más es la que alberga palabras unidas entre sí por la preposición *de*. Una de estas palabras puede existir como una de las variantes de Apellidos ya mencionados, pero no ambas. A esta formación la denominamos **Apellidos Deado Inseparables**. Como ejemplo podemos mencionar *Montes de Oca*, donde encontramos *Montes* como Apellido, pero no *de Oca*.

Formaciones caprichosas encontramos en apellidos muy poco conocidos, y de los cuales no mucho podemos mencionar. Los elementos peculiares de este conjunto pueden ser *La/ Rubia, La Chica, La Moneda*, etc., Catalogamos este tipo de estructuras como **Apellidos Articulados.**

En algunos de nuestros conjuntos existe la intersección de elementos. Por ejemplo, es común utilizar, como Apellidos Solos, Nombres Solos (*Jorge, Santiago, Santos, Félix, Camilo, Miguel*, etc.), y Complementos Deado (*De Miguel, De la Cruz, De los Santos*, etc.). No nos parezca raro en algún momento tratar de precisar, de la secuencia *Jorge Miguel Félix*, qué elementos conforman el nombre, cuáles el apellido y en qué orden se encuentran escritos (¿cómo debemos nombrar a esta persona?

¿como *Jorge* o como *Félix*?). En nuestro modelo, a este conjunto de casos, lo denominamos **Apellidos existentes como Nombres.**

Las Secuencias de Apellidos pueden ser alteradas ajustándose a cánones regionales y/o temporales, o por simple arbitrariedad. De esta manera distinguimos Apellidos cuya estructura escapa de los modelos antes mencionados: *Ponce de León, González-Hermosillo, Molina y Vedia, Barrera-y-Fuentes, De Zárate y Urbina,* etc. Pero si descomponemos estas estructuras en sus elementos atómicos, notamos que a éstos los hemos ya descrito anteriormente. Para conservar la formación completa del apellido introducimos una nueva agrupación, que denominamos **Apellidos Compuestos.** Estos se forman combinando dos segmentos, constituidos cada uno por un elemento de los conjuntos ya mencionados en esta sección, unidos por simple concatenación, por alguna conjunción, guiones o ambos.

Los casos en los cuales nuestra descripción no funciona son bastante raros, y haremos mención de ellos en la conclusión.

3 Gramática generativa

Antes de iniciar con la descripción de nuestra gramática, es necesario anticipar algunas consideraciones.

1. Estos símbolos son terminales:
   a) Los nombres propios (palabras escritas en letra cursiva).
   b) El símbolo de puntuación – (guión).
   c) El conjuntivo *y*.
   d) Las preposiciones articuladas *De*, *De la*, *De las*, *Del* y *De los*.

2. Las denominaciones definidas más adelante, en el apartado 3.1.

3. A menos que se diga otra cosa, el lado izquierdo de la primera producción es el símbolo inicial.

3.1 Denominaciones utilizadas

Es necesario incorporar términos más compactos, a fin de tener referencias rápidas y prepararnos en la sintaxis de nuestra gramática. En la tabla 1 introducimos las expresiones mencionadas. En las columnas impares indicamos las denominaciones a los conceptos tratados y en las columnas pares hacemos mención de su significado correspondiente:

Tabla 1. Expresiones usadas en la gramática

SecNml	Secuencia Nominal	ApPat	Apellido Paterno
		ApMat	Apellido Materno
SecNom	Secuencia de Nombres	ApSolo	Apellido Solo
SecAp	Secuencia de Apellidos	ApDeado	Apellido Deado

		ApDeadoInsprble	Apellido Deado Inseparable
NomSolo	Nombre Solo	ApArtdo	Apellido Articulado
Ncmsto	Nombre Compuesto	ApExsteNom	Apellido existente como Nombre
NCmb	Nombre Combinado	ApCmsto	Apellido Compuesto
CplmtoDeado	Complemento Deado	Sgmto(*j*)	Segmento *j*-ésimo

3.2 Reglas estructurales

En la tabla 2 mostramos las reglas de la gramática que permite definir si una secuencia de palabras corresponde a una SN. Las reglas constan en su mayoría de símbolos no terminales.

Tabla 2. Reglas de la gramática

1	SecNml	→ SecNom(Sex) | SecAp | SecNom(Sex) SecAp
2	SecNom(Sex)	→ NomSolo(1, Sex) | NomSolo(1, Sex) NomSolo(2, Sex) | NomSolo(1, Sex) NomSolo(2, Sex) NomSolo(3, Sex) | NCmsto(1, Sex) | NomSolo(1, Sex) NCmsto(2, Sex) | NomSolo(1, Sex) NCmsto(2, Sex^{-1}) | NCmb(Sex) | NomSolo(1, Sex) NCmb(Sex)
3	NCmsto(*j*,Sex)	→ NomSolo(*j*, Sex) CplmtoDeado
4	SecAp	→ Ap(pat) | Ap(pat) Ap(mat)
5	Ap(Prgtor)	→ ApSolo | ApDeado | ApDeadoInsprble | ApArtdo | ApExsteNom | ApCmsto
6	ApCmsto	→ Sgmto(1) Sgmto(2) | Sgmto(1) *y* Sgmto(2) | Sgmto(1)–Sgmto(2) | Sgmto(1)–*y*–Sgmto(2)
7	Sgmto(*j*)	→ ApSolo | ApDeado | ApExsteNom | *De* ApExsteNom | *De la* ExsteNom | *Del* ExsteNom | *De los* ExsteNom

Nota 1. j es variable con valores 1 o 2.

Nota2. Los elementos NomSolo(1, Sex), NomSolo(2, Sex) y NomSolo(3, Sex) concatenados entre sí son diferentes, es decir, NomSolo(1, Sex) ≠ NomSolo(2, Sex), NomSolo(2, Sex) ≠ NomSolo(3, Sex) y NomSolo(1, Sex) ≠ NomSolo(3, Sex)

Nota 3. Sex es variable de Sexo, con valores *mas* (masculino) o *fem* (femenino). En cada regla el valor de Sex, que puede encontrarse varias veces, es el mismo. Sex^{-1} es complemento (valor opuesto) de Sex.

Nota 4. Prgtor es variable de Progenitor, con valores *pat* (paterno) o *mat* (materno).

Nota 5 En la regla 2 NomSolo(1, Sex) NCmsto(2, Sex^{-1}), el primer elemento del Nombre Compuesto (vea regla 3) guarda relación con NomSolo(1, Sex). Al estar éstos concatenados, siendo de sexo opuesto, se exige que formen un Nombre Combinado válido (vea ejemplo en sección 3.4).

3.3 Instancias a los símbolos no terminales

Definimos como instancia a un símbolo no terminal a aquellos símbolos básicos con los que se forman las SN, conocidos también como terminales –o bien, según los lenguajes de programación, componentes léxicos.

Tabla 3. Símbolos terminales y posibles valores

NomSolo(*i, mas*)	→ *Hugo* | *Gregorio* | *Rodolfo* | *Humberto* |...
NomSolo(*i, fem*)	→ *Rigoberta* | *Dolores* | *Antonieta* | *Andrea* | ...
CplmtoDeado	→ *de la Luz* | *del Carmen* | *del Refugio* | *de Jesús* | ...
NCmb(*mas*)	→ *José María* | *José Trinidad* | ...
NCmb(*fem*)	→ *María José* | *María Jesús* | ...
ApSolo	→ *Sánchez* | *Miranda* | *Vélez* | *Landa* | ...
ApDeado	→ *De la Barca* | *De León* | *De la O* | *De los Cobos* | ...
ApDeadoInsprble	→ *Montes de Oca* | *Cabeza de Vaca* | ...
ApArtdo	→ *La Rubia*| *La Chica* | *La Moneda* | ...
ApExsteNom	→ *Jorge* | *Santiago* | *Félix* | *Camilo* | *Alonso* | ...

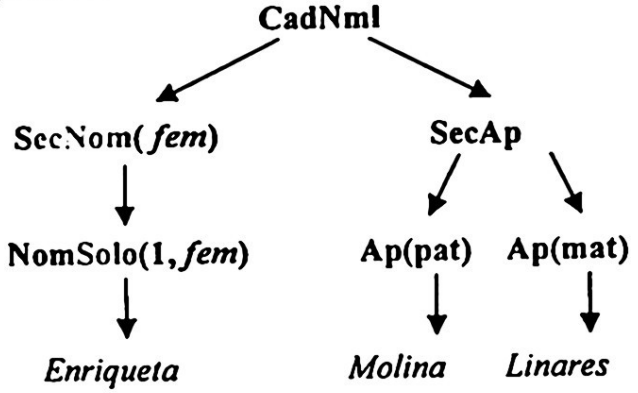
La variable *i* de la fórmula NomSolo(*i*, Sex) tiene valores 1, 2 ó 3.

3.4 Árboles de análisis sintáctico

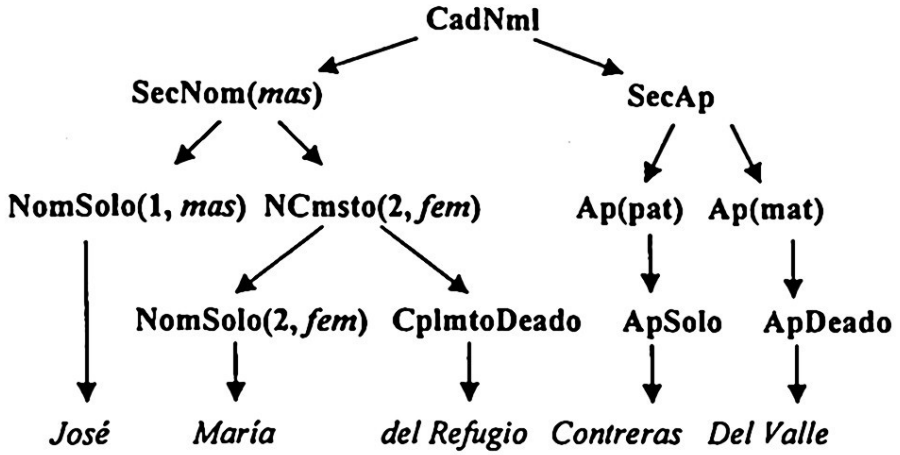
Un árbol de análisis sintáctico se puede considerar como una representación gráfica de una secuencia de sustituciones que no muestra la elección relativa al orden de sustitución.

Mostramos algunos árboles de análisis sintáctico, derivados de las siguientes SN.

- Se deriva de *Enriqueta Molina Linares*:



- El árbol a derivar de *José María del Refugio Contreras Del Valle* es:

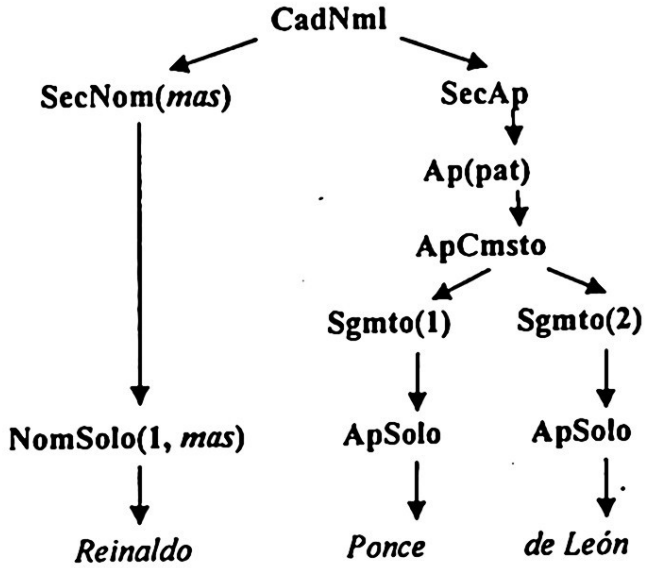


Dada una Secuencia de Nombres formada por un Nombre Solo de sexo *masculino* y un Nombre Compuesto de sexo *femenino*, los Nombres Solos (ver último nivel) deben considerarse como un Nombre Combinado, de lo contrario nuestro modelo aceptaría Secuencias de Nombres extrañas (erróneas) del tipo *Roberto Fabiola, Gerardo Patricia,* etc.

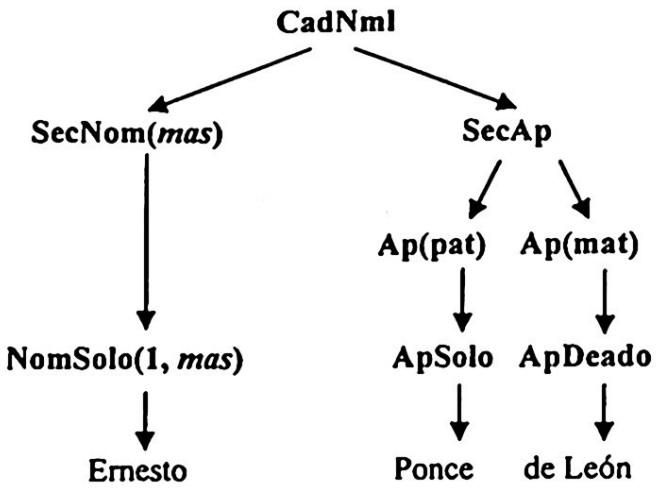
Como integrantes de los Lenguajes Naturales, los Sistemas de Nombres no pueden prescindir de ambigüedades. Específicamente, éstas recaen en la determinación de la estructura de una SN, que puede considerarse en más de una forma distinta. Mostremos la siguiente SN y los dos árboles sintácticos que pueden construirse a partir de ella.

Sea la secuencia *Reinaldo Ponce de León* y los siguientes dos casos:

Caso 1: Es posible determinar la Secuencia de Apellidos como un único Apellido.



Caso 2: Es posible determinar la Secuencia de Apellidos como dos Apellidos.



4 Analizador gramatical

Al construir la gramática que guiará el análisis, se debe construir también el analizador gramatical que determinará si se están cumpliendo o no las reglas que hemos elaborado.

El tipo de analizador que diseñamos es de tipo Ascendente, dado que tiene como objetivo construir el axioma de la gramática a partir de los elementos de léxico que recibe; es decir, es a partir de los elementos más simples que espera conformar el nivel más alto en la gramática: el axioma.

4.1 Denominaciones utilizadas en el programa

Es necesario determinar y precisar algunas expresiones que se utilizan como formato del programa en el resultado del análisis. Éstas se han tratado de ajustar lo más posible a la gramática que hemos formalizado. En la tabla 4 listamos las denominaciones utilizadas por el programa y las definidas en nuestra gramática.

Tabla 4. Correspondencia entre denominaciones de programa y gramática

Gramática	Programa
SecNom(*mas*)	(SecNom,M)
SecNom(*fem*)	(SecNom,F)
NomSolo(*i*, *mas*)	(nom*i*,M)
NomSolo(*i*, *fem*)	(nom*i*,F)
Ap(Prgtor)	(ap,Prgtor)

La forma en que el programa estructura una SN, es por medio de etiquetas que coloca al final de cada elemento encontrado. Las etiquetas quedan definidas en la tabla 5:

Tabla 5. Etiquetas de estructuración de SN

Etiqueta	Significado
(SecTot, x, y)	Etiqueta colocada al final de una SN.
(SecNom, x)	Variante 1. Etiqueta colocada al final de la Secuencia de Nombres
(SecNom, x, z)	Variante 2. Etiqueta colocada al final de la Secuencia de Nombres

El valor y significado de cada variable usada en las etiquetas se dan en la tabla 6:

Tabla 6. Variables para las etiquetas de estructuración de SN

Variable	Significado
SecTot	Secuencia Total.
SecNom	Secuencia de Nombres.
x	Sexo de la SecNom. Toma el valor M o F.
y, z	Consideraciones finales que se hacen al final de cada etiqueta, proporcionando información más específica del análisis.

La variable *y*, presente sólo en la etiqueta colocada al final de una SN, puede llegar a tomar los valores especificados en la tabla 7:

Tabla 7. Valores válidos para la variable *y*

Valor	Significado
Cmpl	[Secuencia] Completa. Valor que indica que la SN analizada cuenta con todos los componentes necesarios.
InCmpl	[Secuencia] Incompleta. En la SN, hace falta ya sea la Secuencia de Nombres o la Secuencia de Apellidos.
inddo	Indeterminado. En la secuencia analizada, los elementos presentes sí se reconocen como nombres pero no forman una estructura coherente.

En la tabla 8 se especifican los valores válidos para *z*:

Tabla 8. Valores válidos para la variable *z*

Valor	Significado
inddo	Indeterminado. Señala que en la secuencia analizada, los elementos presentes sí se reconocen como nombres pero no forman una estructura coherente.
sexoCntrio	Sexo Contrario. Indica que se ha encontrado un par de Nombres de sexo opuesto pero éstos no se encuentran definidos dentro del grupo de Nombres Combinados.
rep	Repetidos. Valor que indica que en la Secuencia de Nombres se han detectado nombres repetidos.
Larg	Larga. La Secuencia de Nombres es muy extensa (más de tres elementos).

4.2 Descripción del programa

En nuestro trabajo, para la identificación de las SN, se emplearon diccionarios (bases de datos) de nombres y de apellidos. El sistema es totalmente dependiente de estas bases, aún cuando se utiliza una pequeña heurística para determinar Nombres Combinados, Apellidos Compuestos, etc. A grandes rasgos, el proceso seguido en el análisis es el siguiente:

En primera instancia, el sistema realiza un parseo por el texto electrónico, determinando si cada palabra encontrada pertenece o no a la base de datos; cuando se ha terminado de extraer una determinada secuencia, se aplica a ésta un segundo proceso de análisis más específico (proceso de etiquetación), que consiste en asignar a cada elemento su denominación correspondiente a la gramática elaborada, según el lugar que ocupe dentro de la SN extraída.

Listamos a continuación una serie de ejemplos tomados del programa para mostrar los resultados arrojados:

Secuencia texto: Angélica María de la Luz Hernández De Alba
Resultado del programa:
[[[[*Angélica*(nom1,F)][*María de la Luz*(nom2,F)]](SecNom,F)]
[*Hernández*(ap,pat)][*De Alba*(ap,mat)]](SecTot,F,Cmpl)

Secuencia en texto: Angélica María de la Lourdes Hernández De Alba
Resultado del programa:
[[[[*Angélica*(nom1,F)][*María de la Lourdes*(nom2,F,inddo)]](SecNom,F)]
[*Hernández*(ap,pat)][*De Alba*(ap,mat)]](SecTot,F,Cmpl)

Secuencia en texto: María Angélica Lorena Patricia De Alba
Resultado del programa:
[[[[*María*(nom1,F)] [*Angélica*(nom2,F)] [*Lorena*(nom3,F)] [*Patricia*(nom4,F)]]
(SecNom,F,larg)][*De Alba*(ap,pat)]] (SecTot,F,Cmpl)

Secuencia en texto: *María Angélica*
Resultado del programa:
[[[[*María*(nom1,F)][*Angélica*(nom2,F)]](SecNom,F)]](SecTot,F,Incmpl)

Secuencia en texto: *De Alba*
Resultado del programa:
[[*De Alba*(ap,pat)]](SecTot,inddo,Incmpl)

5 Conclusiones

Son muchos los factores que dificultan el desarrollo de un modelo que sea capaz de identificar y reconocer Secuencias Nominales Hispanas dentro de textos electrónicos.

Hemos visto como la ambigüedad inherente a sistemas relacionados con el Lenguaje Natural dificultan el proceso de modelación. No todo lo presentado en este capítulo es concluyente, nuestro ámbito de análisis ha tenido que verse restringido por

las variaciones tan extensas que existen en la formación de SN Hispanas. Un ejemplo claro de esto se constata en Secuencias de Apellidos. Una práctica habitual hace tiempo era la concatenación de más de dos apellidos para constituir uno solo, e.g., *Sánchez de Anda y Martínez Salgado, Martí y Zayas-Bazán, Bancés y Fernández-Criado,* etc. Obviamente, el tratar de abarcar todos estos casos (y más) resultaría en un entre marañado de reglas, en una complejidad realmente enorme.

Notamos que es imposible aplicar análisis morfológico [5], porque los apellidos no se encuentran normalmente en las bases de datos de los sistemas de este tipo (por lo tanto usamos las mayúsculas para detectar presencia de los apellidos cuando uno de éstos no se haya en nuestra base de datos).

Estos son sólo algunas características que pueden dificultar alcanzar el objetivo que se planteó en este trabajo. La más clara conclusión de esto es que un sistema de clasificación de nombres no puede ser automatizado en su totalidad, siempre será necesaria la intervención del usuario.

Agradecimientos

Este trabajo fue realizado con el apoyo parcial de CONACyT y PIFI. Deseamos agradecer, asimismo, al Dr. Grigori Sidorov por sus valiosos comentarios y contribuciones.

Referencias

1. Chen, H. H., S. J. Huang, Y. W. Ding, S. C. Tsai. (1998). *Proper Name Translation in Cross-Language Information Retrieval.* Proceedings of the 17th International Conference on Computational Linguistics. pp 232 – 236. http://acl.ldc.upenn.edu/P/P98/P98-1036.pdf
2. Chinchor, N. (1997). MUC-*7 Named Entity Task Definition.* Proceedings of the 7th Message Understanding Conference. http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/ne_task.html
3. Galicia-Haro, S. N., A. Gelbukh, I. A. Bolshakov. *Recognition of Named Entities in Spanish Texts.* Proceedings of International Conference MICAI'2004, Mexico City, Mexico. Lecture Notes in Artificial Intelligence, ISSN 0302-9843, Springer, 2004 (accepted).
4. Galicia-Haro, S. N., A. Gelbukh, I. A. Bolshakov. *Web-based Sources for an Annotated* Corpus Building and Composite Proper Name Identification. Proceedings of International Conference AWIC04, Cancun, Mexico. Lecture Notes in Computer Science, ISSN 0302-9843, Springer, 2004 (accepted).
5. Alexander Gelbukh and Grigori Sidorov. *Approach to construction of automatic morphological analysis systems for inflective languages with little effort.* In: Computational Linguistics and Intelligent Text Processing. Proc. CICLing-2003, 4th International Conference on Intelligent Text Processing and Computational Linguistics, February 15–22, 2003, Mexico City. LNCS, N 2588, Springer-Verlag, pp. 215–220.
6. Huang, F., A. Waibel. *An Adaptive Approach to Named Entity Extraction for Meeting Applications* .http://www.is.cs.cmu.edu/papers/speech/HLT2002/hlt2002_fei.pdf
7. Mani, I., R. McMillian, S. Luperfoy, E. Lusher, S. Laskowiski. (1993). *Identifying Unknown Proper Names in Newswire Text.* Proceedings of the Workshop on Acquisition of Lexical Knowledge from Text. pp 44 – 54. http://acl.ldc.upenn.edu/W/W93/W93-0105.pdf

8. Petasis, G., S. Petridis, G. Paliouras, V. Karkaletsis, S. J. Perantonis, C. D. Spyropoulos. (2000). *Symbolic and Neural Learning for Named-Entity Recognition. Proceedings of the Symposium on Computational Intelligence and Learning*. pp. 58-66. http://iit.demokritos.gr/~paliourg/papers/COIL2000.pdf
9. Thompson, P., C. C. Dozier. (1997). *Name Searching and Information Retrieval.* Proceedings of 2nd Conference on Empirical Methods in Natural Language Processing. Providence, Rhode Island. pp. 134-140. http://acl.eldoc.ub.rug.nl/mirror/W/W97/W97-0315.pdf
10. Wacholder, N., Y. Ravin, M. Choi. (1997). *Disambiguation of Proper Names in Text.* Proceedings of the 5th Conference on Applied Natural Language Processing. pp. 202 – 208 http://www.research.ibm.com/talent/documents/anlp97.pdf

Análisis e implementación de algoritmos para la solución de laberintos de estructura conocida

Omar Rodríguez, Erik Hernández

Facultad de Informática, Universidad Autónoma de Querétaro, México

Resumen. El presente capítulo explica el análisis e implementación de algoritmos para lograr que un robot móvil sea capaz de navegar de forma autónoma dentro de un laberinto, de 3.6x3.6 metros, conocido con anterioridad y que consta de paredes de 25 cm de altura con ángulos rectos y calles de 40 cm de ancho. La principal dificultad, radica en que no se conoce con anterioridad la posición y orientación inicial del robot. En el capítulo se analizan varios algoritmos clásicos aplicables a la solución de laberintos y se escoge uno para desarrollarlo completamente sobre un robot específico con un microcontrolador PIC16F877.

1 Introducción

Los concursos de robótica van cobrando cada vez mayor fuerza, no solo a nivel académico, sino a nivel empresarial, puesto que representan una base muy importante para la creación de proyectos aplicables directamente en los procesos de desarrollo e investigación de la industria.

Entre los objetivos particulares que se persiguen, están analizar los problemas de codificación del algoritmo en un microcontrolador específico, programar las etapas de adquisición de datos y control de motores en el microcontrolador.

La categoría consiste en la navegación autónoma a través de un laberinto de 3.6x3.6 metros desde un punto de partida no conocido hasta encontrar la salida. El laberinto estará constituido por paredes de 25 cm de altura, diseñadas en ángulos rectos y con calles de 40 cm de anchura. Todas las medidas son aproximadas, el robot deberá ser capaz de soportar una incertidumbre de medida del 10%.

2 Desarrollo

2.1 Descripción del robot

Por las características de la categoría, es necesario que el robot tenga la posibilidad de girar sobre su propio eje, y realice giros y paros con rapidez. Las dimensiones máximas del robot están limitadas a 20 cm de ancho, 30 cm de largo y 25 cm de alto, sin restricción de peso.

Por los requerimientos de movilidad mencionados anteriormente, se colocaron dos servomotores en el eje transversal, lo que permite realizar movimientos controlados,

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.117-127.© Centro de Investigación en Computación, IPN, México

como se puede ver en la figura 1, sin perder en momento alguno la posición actual. Para equilibrar el robot este se apoya en una tercera rueda que gira libremente sin representar fricción significativa al sistema, obteniendo un robot que es capaz de alcanzar altas velocidades y máxima maniobrabilidad.

La interacción con el medio se realiza mediante sensores infrarrojos, que detectan la existencia o no de paredes cercanas, estos sensores trabajan con modulación estándar de 40 KHz. En cuanto a los servomotores, estos ya incluyen sus respectivas etapas de amplificación de potencia y se les envía la dirección y velocidad de giro a través de un pulso cuadrado de ancho proporcional a la velocidad deseada. La posición de cada una de las llantas del robot se mide con dos sensores infrarrojos ubicados en cada una de ellas, de tal forma que proporcionan el numero de vueltas que se realiza, por lo que se pueden dar las vueltas de 90°, 180° o 360° con toda precisión, a su vez esto sirve para conocer cuanto ha avanzado el robot.

Como plataforma de programación y control el sistema cuenta con un microcontrolador PIC16F877 [1], que además de controlar sensores y motores, debe contener el programa principal, almacenar la estructura del laberinto en memoria y desarrollar todas las estrategias de control y tareas complementarias que se requieren para el funcionamiento del robot. Para soportar todo esto, se escogió este microcontrolador ya que posee una arquitectura RISC de alto rendimiento que cuenta con 368 bytes de memoria RAM, 256 bytes de EEPROM para datos y 8192 localidades de memoria FLASH de 14 bits por localidad para almacenamiento del programa y datos permanentes, además cuenta con periféricos tales como un ADC de 10 bits con multiplexor de 8 canales, 2 salidas PWM, 3 contadores-temporizadores, comunicación SPI, I^2C, USART, 33 pines de entrada y salida digital de uso general y 13 canales de interrupción. Todos los programas se desarrollan en lenguaje C utilizando el compilador PICC de CCS, el cual tiene como principal limitación, sobre todo para esta tarea, que no pueden manejarse apuntadores a la memoria FLASH.

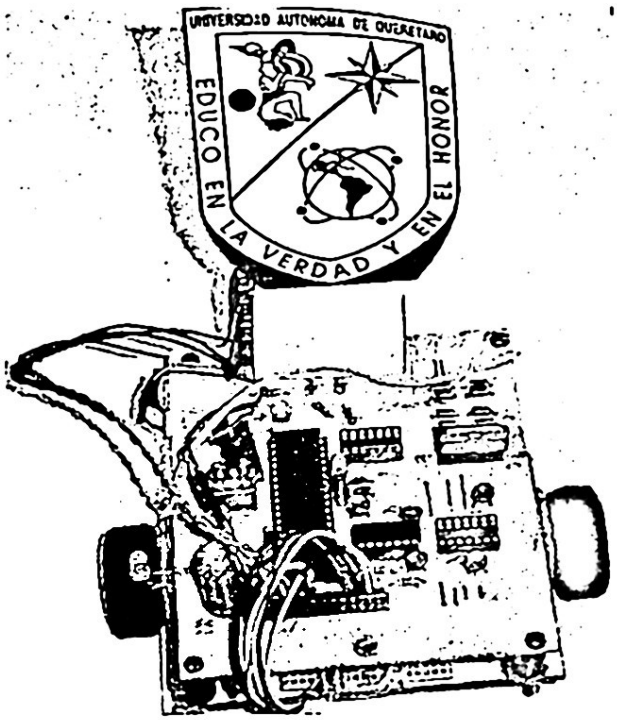


Fig. 1. Imagen del robot

Para equilibrar el robot este se apoya en una tercera rueda que gira libremente sin representar fricción significativa al sistema, obteniendo un robot que es capaz de alcanzar altas velocidades y máxima maniobrabilidad.

La interacción con el medio se realiza mediante sensores infrarrojos, que detectan la existencia o no de paredes cercanas, estos sensores trabajan con modulación estándar de 40 KHz. En cuanto a los servomotores, estos ya incluyen sus respectivas etapas de amplificación de potencia y se les envía la dirección y velocidad de giro a través de un pulso cuadrado de ancho proporcional a la velocidad deseada. La posición de cada una de las llantas del robot se mide con dos sensores infrarrojos ubicados en cada una de ellas, de tal forma que proporcionan el numero de vueltas que se realiza, por lo que se pueden dar las vueltas de 90°, 180° o 360° con toda precisión, a su vez esto sirve para conocer cuanto ha avanzado el robot.

Como plataforma de programación y control el sistema cuenta con un microcontrolador PIC16F877 [1], que además de controlar sensores y motores, debe contener el programa principal, almacenar la estructura del laberinto en memoria y desarrollar todas las estrategias de control y tareas complementarias que se requieren para el funcionamiento del robot. Para soportar todo esto, se escogió este microcontrolador ya que posee una arquitectura RISC de alto rendimiento que cuenta con 368 bytes de memoria RAM, 256 bytes de EEPROM para datos y 8192 localidades de memoria FLASH de 14 bits por localidad para almacenamiento del programa y datos permanentes, además cuenta con periféricos tales como un ADC de 10 bits con multiplexor de 8 canales, 2 salidas PWM, 3 contadores-temporizadores, comunicación SPI, I^2C, USART, 33 pines de entrada y salida digital de uso general y 13 canales de interrupción. Todos los programas se desarrollan en lenguaje C utilizando el compilador PICC de CCS, el cual tiene como principal limitación, sobre todo para esta tarea, que no pueden manejarse apuntadores a la memoria FLASH.

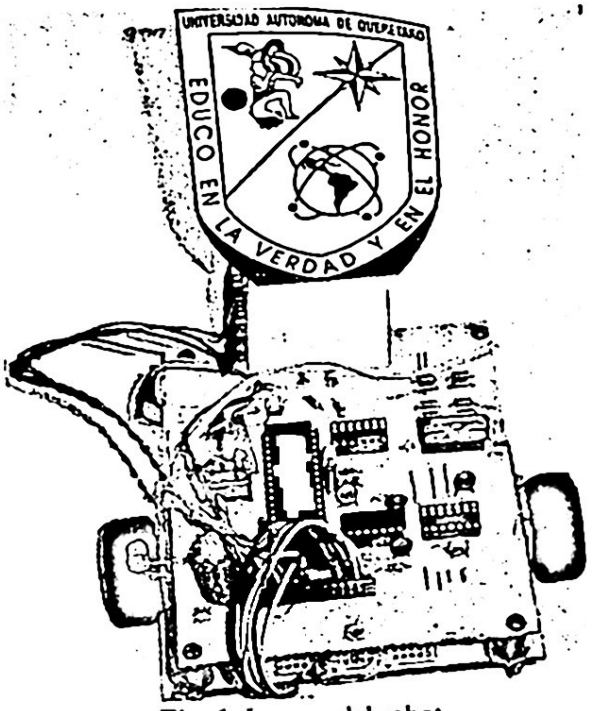


Fig. 1. Imagen del robot

2.2 Descripción y análisis de algoritmos

2.2.1 Algoritmos de la mano derecha y de la mano izquierda

Los primeros algoritmos que tenemos que discutir por su popularidad y sencillez, son los algoritmos de la mano derecha y de la mano izquierda, estos algoritmos se basan en seguir una pared del laberinto (izquierda o derecha según el caso) y resultan efectivos en casi todos los casos, se conozca o no el laberinto. Sus principales desventajas, radican en que no resuelven laberintos no conectados (con islas internas) si no se empieza desde una de las paredes exteriores del laberinto y por otro lado la trayectoria que siguen para encontrar la salida dista mucho de ser la óptima, ya que recorre todas las opciones hasta encontrar la salida.

Dados que el problema que se plantea consiste en un laberinto no conectado y el punto de inicio es desconocido, estos algoritmos no pueden ser aplicados, Aunque modificaciones de este algoritmo, como la planteada por Lucas en 1882 [2] que consiste en ir pintando una línea por las trayectorias recorridas logran resolver satisfactoriamente aún este tipo de laberintos, si utilizamos este tipo de algoritmos perderíamos la ventaja que se tiene por conocer la estructura del laberinto con anterioridad.

2.2.2 Algoritmo de recorrido recursivo

Este algoritmo se basa en el recorrido de un camino, regresando fallo cuando encontramos una pared, y éxito si encontramos una salida, en cualquier otro caso, intentamos recorrer de manera recursiva en cualquiera de las 4 direcciones posibles, cuando intentamos una nueva dirección trazamos una línea, y borramos una línea cuando encontramos fallo. Este método encontrará siempre una salida, si es que existe, pero en la mayoría de los casos, no será el camino mas corto, y requiere recorrer una gran parte del laberinto para salir de él. El hecho de que la competencia sea por tiempo, y la arquitectura que estamos manejando, impide manejar la recursividad de manera sencilla, además del hecho que de las 2 posibles salidas existentes en el laberinto, solo una de ellas es válida y no existe manera de descartar la otra utilizando este algoritmo.

2.2.3 Algoritmo de Tremaux [3]

Esta es la versión del algoritmo anterior pero de manera tal que pueda ser ejecutada por un humano. Se recorre progresivamente el grafo del laberinto. Este asegura que todos los vértices dentro de cierta distancia sean visitados. Es fácil alcanzar los vértices a una distancia 1, y regresar al punto de partida. Cada camino de estos es marcado para llegar de X0 a X1, como el camino de entrada. Para llegar a un vértice a una distancia 2, desde el origen, se selecciona el camino anterior y se vuelve a marcar. Cuando todos los caminos posibles de X1 son visitados, se marca el camino como cerrado, y se toma el siguiente camino desde el origen, hasta que no queden caminos disponibles desde el origen.

2.2.4 Algoritmo de Tarry [3]

Este método publicado por Tarry en 1895, crea un camino cíclico en el laberinto, pasando por cada cruce una y solo una vez en cada dirección. Iniciando en una

posición A, selecciona un camino cualquiera y lo marca de manera especial, cuando llega al siguiente cruce, puede haber sido visitado o no, y lo marca acorde con ello. El algoritmo termina en el lugar de inicio, habiendo recorrido cada cruce, por lo menos una vez en cada dirección.

Una mejora a los dos algoritmos anteriores fue propuesta por Fraenkel [4], [5], la cual asegura que cada cruce será transitado, a lo más, una vez en cada dirección.

Estos tres algoritmos anteriores, fueron descartados porque, al igual que los de mano izquierda y derecha, suponen que el laberinto no se conoce previamente, por lo que requieren recorrer una gran parte del laberinto para encontrar la salida.

2.2.5 Algoritmo de Pledge

Otro algoritmo de muy sencilla implementación, es el de Pledge [6] que consiste en mantener la dirección en que se esta apuntando en cada momento, a pesar de su simplicidad, al igual que los algoritmos de la mano izquierda y derecha, no es aplicable a todo tipo de laberintos, puesto que en ciertos casos, como los encontrados en esta categoría, el robot queda ciclado en un solo camino.

2.2.6 Algoritmo de Lee

El algoritmo planteado por Lee[7], consta de dos partes, primero la creación del mapa del laberinto en memoria, por medio de recorridos sucesivos. Durante estos recorridos, se anotan las paredes que tiene cada celda conforme el robot las visita., así, si la celda solo tuviera pared al norte, se le asigna un 8, pero si además tiene también pared al este, se le asigna un 10, 8 de la pared norte y 2 de la pared este. Esta parte, permite ver que requeriremos una cantidad de memoria, de tamaño suficiente para, por lo menos, almacenar un arreglo con la información de cada celda del laberinto.

Una vez que se tiene un mapa del laberinto, se asignan un número a cada celda, de la siguiente manera:

Iniciamos colocando un 0 en cada una de las celdas del laberinto que se consideren como meta. Deberemos recorrer el laberinto columna por columna en repetidas ocasiones. Para cada celda en la que esté el robot, se busca en nuestro mapa si dicha celda aún no está numerada y si las celdas a las que puedo llegar a partir de la actual ya se encuentran numeradas, si no es así, en esta visita no haremos nada. En caso de que por lo menos alguna de las celdas vecinas ya se encuentre numerada, y la actual no, suponiendo que el número de Lee más pequeño de las celdas vecinas sea n, a la celda actual le asignamos n+1.

Este proceso se repite hasta que ya no pueda realizar cambios en el mapa, lo que supondrá que ya todas las celdas se encuentran numeradas.

Lo anterior supone que ya se tiene en memoria el mapa del laberinto. De no ser así, se puede modificar la generación de números, suponiendo que el laberinto solo tiene paredes en los extremos y calculando los números. Al cambiar de una celda a la siguiente, verificamos si en realidad no existen más paredes, en caso contrario, actualizamos el mapa y recalculamos los números.

Una vez generados los números es fácil encontrar el camino dentro del laberinto, y además se asegura que será siempre el camino más corto, puesto que estando en

cualquier celda del laberinto basta con moverse a la celda vecina que contenga un número menor.

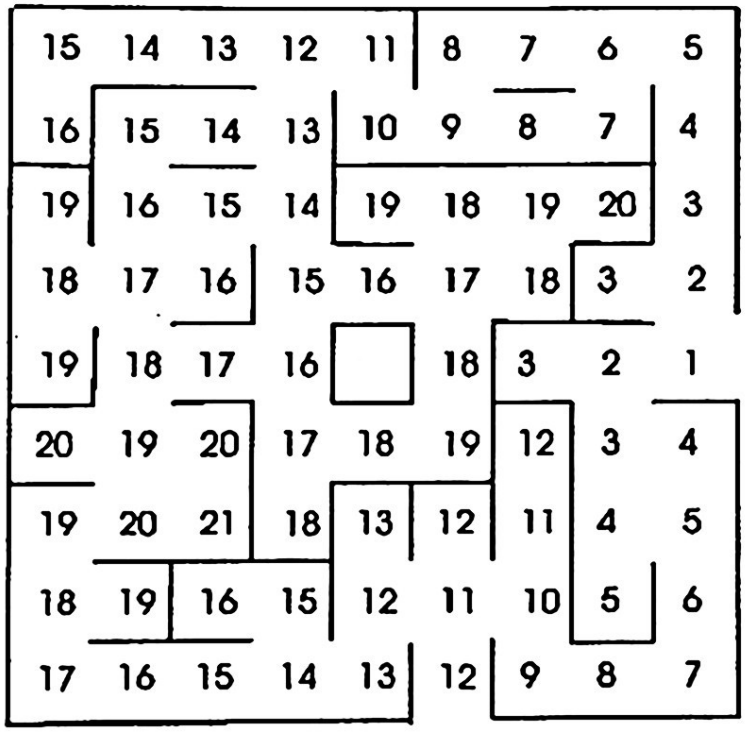


Fig. 2. Estructura del laberinto, con valores de distancia a la salida

Este algoritmo fue escogido como base para la solución de nuestro problema, porque además de resolver todo tipo de laberintos y garantizar la ruta óptima, plantea la solución del problema en dos etapas, primero conocer el laberinto y después recorrerlo, esta característica lo hace muy fácil de adecuar a nuestras necesidades, ya que conocemos de antemano el laberinto y podemos realizar el análisis y la numeración del mismo con antelación.

2.3 Desarrollo e implementación del algoritmo

Debido a las características del problema y a la plataforma sobre la que se va a programar, se escogió desarrollar un algoritmo de bajos requerimientos basado principalmente en el algoritmo de Lee, pero aprovechando el conocimiento previo del laberinto, por lo que el proceso de asignación de números se puede realizar incluso antes de programar el robot. Esto junto con el desarrollo de un método que nos permite identificar en el menor número de movimientos posibles (generalmente entre 3 y 6 movimientos), la ubicación y orientación actual del robot, proporciona la información suficiente para después seguir la segunda parte del algoritmo de Lee hasta llegar a la salida, lo cual se realiza por la ruta óptima.

Primeramente estudiamos el laberinto de prueba de la figura 2 y numeramos cada casilla con la distancia en casillas que se requiere recorrer para llegar desde ahí a la salida. Por lo que para salir de él, sólo es necesario moverse siempre hacia una casilla de menor valor.

Para almacenar el conocimiento previo que se tiene del laberinto, se utilizan 4 arreglos de 9x9 en memoria flash, cada uno de estos arreglos guarda toda la información del laberinto visto desde una de las direcciones posibles (norte, sur, este y oeste) en cada localidad del arreglo (14 bits), se almacena el número de casilla en los 6 bits más significativos y las paredes que tiene esa casilla en los 4 bits menos significativos, quedando como se ilustra en la figura 3.

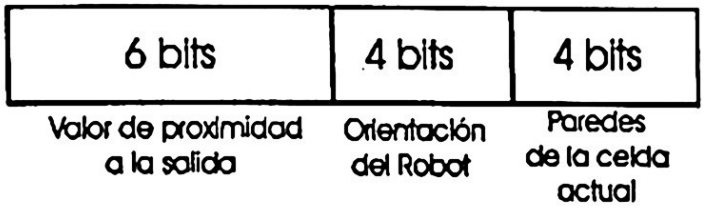


Fig. 3. Forma en que se almacena la información de cada casilla

Además de tener los cuatro arreglos de 9x9 que almacenan el laberinto visto con diferentes orientaciones, se tiene un arreglo más de 9x9 que va generando el laberinto según se mueve el robot, dicho arreglo se almacena en RAM. Un ejemplo de cómo se almacena este laberinto visto desde la orientación de la figura 2, se muestra en la figura 4, en esta, el primer número sigue indicando los 6 bits superiores que corresponden a la distancia a la salida y el segundo número identifica las paredes de cada casilla.

0F/9	0E/C	0D/C	0C/8	08/A	08/9	07/C	06/8	05/A
10/7	0F/9	0E/C	0D/2	0A/5	09/4	08/C	07/6	04/3
13/B	10/1	0E/8	0D/2	13/D	12/8	13/8	14/E	03/3
12/1	11/0	10/6	0F/1	10/C	11/0	12/6	03/D	02/1
13/7	12/1	11/C	10/2		12/3	03/D	02/8	01/4
14/D	13/0	14/A	11/2	12/C	13/6	0C/B	03/1	04/A
13/9	14/4	15/6	12/7	0D/B	0C/B	08/3	04/1	05/2
12/1	13/E	01/D	0F/A	0C/1	08/0	0A/2	05/7	06/3
11/5	10/C	0F/C	0E/4	0D/6	0C/3	09/5	08/C	07/6

Fig. 4. Volcado de memoria de una orientación del laberinto, los números están en hexadecimal

Cuando el recorrido empieza, el robot identifica las paredes que tiene a su alrededor y obtiene un número de identificación de la casilla en cuestión, este número lo va comparando, uno por uno, con la información de las paredes de cada una de las localidades del primer arreglo y supone que se encuentra en la primer casilla compatible del primer arreglo. Después de esto se mueve a la siguiente casilla en

dirección a la salida, si ésta también es compatible con la posición en la que supone que está, continua de la misma forma, si no es compatible, descarta la posición original supuesta y busca la siguiente que sea compatible. Este procedimiento se sigue hasta que la determinación de la posición real del robot sea inequívoca. Dentro de este proceso, si el primer arreglo es agotado sin encontrar las casillas que correspondan, significa que la orientación inicial del robot no era la supuesta y se sigue buscando en los siguientes arreglos.

2.4 Principales problemas de codificación

Como ya se había mencionado, uno de los principales problemas que se tienen es que el compilador usado no maneja apuntadores a la memoria de programa (FLASH), por lo que para poder manejar los arreglos que almacenan la información del laberinto hubo que programar funciones para lectura de la memoria FLASH, estas funciones se desarrollaron de forma tal que ingresan como parámetro las coordenadas de la casilla que se quiere leer y regresan el valor de la misma, una función lee la parte baja que contiene la información de las paredes y otra lee el valor de distancia a la salida.

Otro problema importante es que además de estar ejecutando el algoritmo descrito, el microcontrolador debe estar tomando las lecturas permanentes de los sensores de presencia, lecturas de las posiciones de los motores y mandando de forma continua las señales de control a los motores. Para poder hacer todo esto al mismo tiempo, se utilizó un esquema de programación orientada a eventos, aprovechando muchos de los múltiples canales de interrupción del PIC16F877.

Para el control de los motores se tiene que enviar una señal cuadrada de 40 ms de periodo en donde el tiempo en alto es proporcional a la velocidad del motor, si el tiempo en alto es de 1.5 ms el motor permanecerá en reposo, mientras que se moverá a la derecha con tiempos de encendido mayores y a la izquierda con tiempos menores, la velocidad del motor será igual a la diferencia entre el tiempo de encendido y 1.5 ms.

Para generar dos señales de este tipo para los dos motores, se utilizó el timer 1, con el se hace el calculo del tiempo deseado se configura y posteriormente el timer generará una interrupción cuando el tiempo haya concluido.

Los sensores ópticos (GP2D02 [8]), requirieron ser colocados de manera particular, puesto que, como se puede apreciar en la figura 5, la distancia mínima confiable de medición es de 8 cm, por ello fueron montados en el centro del robot, por debajo de la placa de circuitos, asegurando de esta manera que sea el mismo cuerpo del robot el que garantice esa distancia mínima entre el sensor y las paredes.

Los sensores se leen a través de una interfaz serie síncrona, generando un tren de pulsos como el de la figura 6.

El inicio del ciclo de lectura del sensor, inicia al cambiar la señal de Vin del sensor de alto a bajo, y se activa el canal de interrupción externa. Una vez que el sensor haya realizado la medición y esté listo para enviar el dato, la señal de datos cambia de estado, activando la interrupción externa del microcontrolador. Durante esta interrupción, se habilita la comunicación serie y se habilita la interrupción de la misma. Al activarse esta interrupción, el dato ya se encuentra disponible en el buffer de datos, leemos el dato y habilitamos la interrupción del timer 2, ajustamos el

periodo del mismo al mínimo de 1.5 ms necesarios para que el sensor este listo para realizar una nueva medición. Una vez que se activa la interrupción del timer 2, podemos iniciar nuevamente el ciclo, enviando la señal de inicio de medición, desactivando la interrupción del timer 2 y activando la interrupción externa.

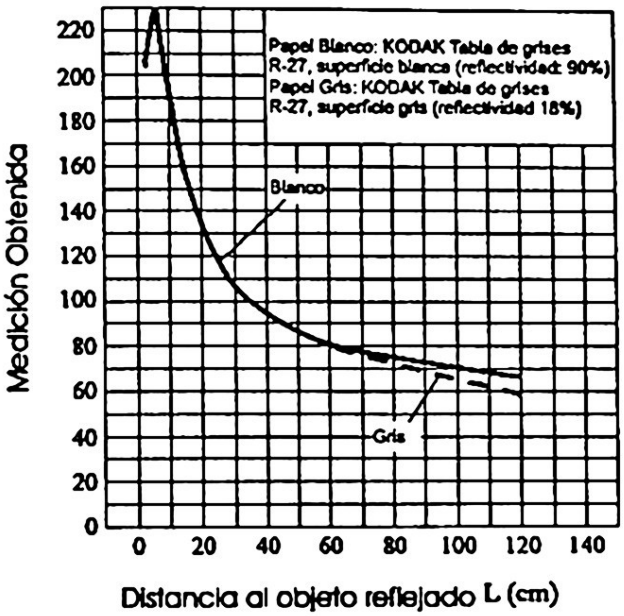


Fig. 5. Gráfica de medición del sensor infrarrojo, el eje vertical representa el número digital que retorna el sensor

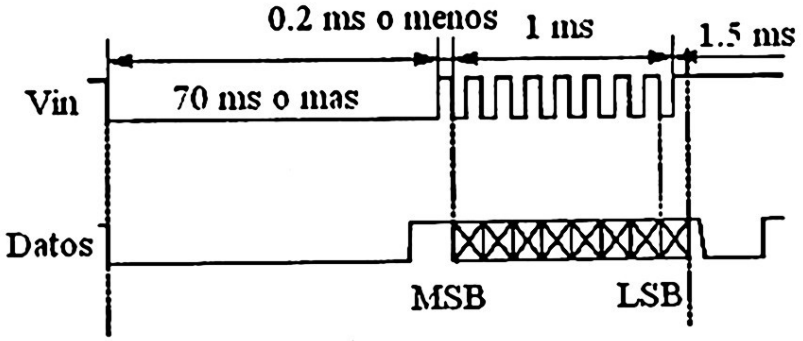


Fig. 6. Gráfica de tiempos del sensor infrarrojo.

Realizar diferentes tareas dentro de un mismo microcontrolador genera ciertos problemas en el momento en que estas se tienen que realizar al mismo tiempo, esto es, si se deseara recorrer una distancia para posicionarse en la casilla siguiente, esto se podría realizar mediante una rutina que llevará cierto tiempo en producirse y con la cual se realizara la tarea; si en el tiempo en que se encuentra trabajando la rutina tienen que realizarse otras tareas para leer los sensores, medir distancia recorrida, etc., esta información se perdería debido a que para ser realizados se tiene que haber terminado la rutina en curso, en cambio con los eventos no importa que ocurrieran

varios eventos dentro de la realización de la tarea porque en el momento que estos se producen el microcontrolador los realiza y regresa a terminar la tarea en curso.

3 Resultados

Las pruebas realizadas, nos permitieron observar que en la mayoría de los casos el robot logra identificar su posición en alrededor de 5 o 6 movimientos, aunque en general podríamos decir que esto depende en gran medida de cual sea su orientación original, ya que cuando se orienta inicialmente al norte determina su posición en 2 o 3 movimientos y cuando su orientación es al oeste llegan a ser hasta 7 u 8 movimientos.

Cabe destacar que la mayor parte de las dificultades que se tuvieron en le implementación del algoritmo, no se dieron en cuanto al manejo de las decisiones, sino en la etapa de control, donde el manejo y calibración de sensores, la medición de distancias, tolerancia de incertidumbres etc. requirieron un trabajo de puesta a punto muy considerable.

4 Conclusiones

Consideramos que la estrategia de programar todas las etapas de control del robot a través de eventos, fue uno de los grandes aciertos en este trabajo, logrando separar este problema, del algoritmo modificado de Lee que se planteó para la elección de la ruta. ya que de lo contrario habría resultado casi imposible controlar cada una de las partes además de hacerlas interactuar con el resto.

Otra posible solución que se pudo haber seguido es la utilización de dos microcontroladores para dividir las tareas, uno de ellos programado para controlar únicamente el movimiento de los motores para no afectar con otras cosas la eficiencia de un recorrido exacto y el otro con la lectura de los sensores y la ejecución del algoritmo.

En cuanto al algoritmo utilizado para la elección de la ruta a seguir, podríamos concluir que es óptimo, dadas las condiciones y la información con que cuenta el robot. Sin embargo, si pudiésemos dotar al robot de mas información, se podrían plantear nuevos algoritmos que reduzcan drásticamente la distancia entre el mejor y peor caso, además de reducir también el número de movimientos necesarios para identificar la posición del robot, logrando que esta identificación sea casi inmediata. En este sentido, la información más relevante y accesible seria la orientación del robot, para lo cual lo tendríamos que incorporarle una brújula electrónica; además sería útil conocer, no solo la existencia o no de cada pared, sino también la distancia en casillas que existe en cada sentido hasta la siguiente pared.

Cabe mencionar, que aunque para el laberinto planteado en este trabajo, el algoritmo presenta un alto grado de eficiencia, si no se modificase y se utilizara en un laberinto de mayores dimensiones, se reduciría considerablemente la eficiencia del peor de los casos, por lo que se plantea como trabajo a futuro la incorporación al robot de los medios necesarios para recabar los datos de orientación y distancia a la pared

mas cercana en cada dirección, lo que nos llevará también a las respectivas modificaciones del algoritmo.

Referencias

[1] Microchip_Technology, *PIC16F87X Data Sheet*: Microchip Technology Inc., 2001.
[2] E. Lucas, *Recréations mathématiques*, vol. 1, 1882.
[3] O. Ore, "Theory of Graphs," *American Math. Soc. Colloquium Publications*, vol. XXXVII, 1962.
[4] A. S. Fraenkel, "Economic traversal of labyrinths," in *Mathematics Magazine*, vol. 43, 1970, pp. 125-130.
[5] A. S. Fraenkel, "Economic traversal of labyrinths," in *Mathematics Magazine*, vol. 44, 1971, pp. 12.
[6] A. d. H. Abelson, *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*: MIT Press, 1980.
[7] C. Y. Lee, "An algorithm for path connections and its applications," *IRE transactions on electronic computers*, vol. 10, pp. 346-365, 1961.
[8] SHARP_CORPORATION, *Databook 1999/2000*, 2000.

Móvil manipulado por la PC

Abigail Flores Cruz, Adriana Martínez Mestas, Leticia E. Torres Ramírez

Escuela Superior de Cómputo, IPN,
Av. Juan de Dios Batiz s/n, Esq. Miguel Othón de Mendizabal
Unidad Profesional "Adolfo López Mateos"
Instituto Politécnico Nacional, 07738,
México, D. F.
abuguis@hotmail.com, adamar@terra.com.mx,
eloburbu@yahoo.com.mx

Resumen. El capítulo describe un sistema capaz de dirigir un móvil hacia un objetivo determinado por la adquisición y procesamiento de una imagen digital tomada de un escenario definido. El procesamiento de la imagen determinará la ubicación de un objeto seleccionado por el usuario generando así la trayectoria a seguir por el móvil.

1 Introducción

La robótica se define como el conjunto de conocimientos teóricos y prácticos que permiten concebir, realizar y automatizar sistemas basados en estructuras mecánicas, dotados de un determinado grado de "inteligencia" y destinados a la producción industrial o al sustitución del hombre en diversas tareas. Un sistema robótico puede describirse, como "Aquel que es capaz de recibir información, de comprender su entorno a través del empleo de modelos, de formular y de ejecutar planes, y de controlar o supervisar su operación.

Los dispositivos y mecanismos que pueden agruparse bajo la denominación genérica del robot, tal como se ha indicado, son muy diversos y es por tanto difícil establecer una clasificación coherente de los mismos que resista un análisis crítico y riguroso. La subdivisión de los robots, con base en su arquitectura, se hace en los siguientes grupos: Poliarticulados, Móviles, Androides, Zoomórficos e Híbridos.

Móvil, en nuestro caso, es el grupo de interés por lo que su definición se presenta a continuación.

Móviles: Son robots con gran capacidad de desplazamiento, basados en carros o plataformas y dotados de un sistema locomotor de tipo rodante. Siguen su camino por telemando o guiándose por la información recibida de su entorno.

2 Descripción general del sistema

La estructura del sistema completo se presenta en la figura 1.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.128-143.© Centro de Investigación en Computación, IPN, México

Se propone que a partir de la adquisición de una imagen digital, se dirija el curso que deberá seguir para llegar al objetivo, dentro de un escenario definido, un móvil.

Una vez adquirida la imagen digital, ésta se almacena en la PC. La imagen digital se procesa con el fin de reconocer el objeto hacia el cual el usuario desea dirigir al móvil.

Una vez que se ha reconocido el objeto se genera la trayectoria que deberá seguir el móvil para llegar al objetivo.

Dicha trayectoria estará contenida en una trama que será enviada al móvil por medio de un módulo de radiofrecuencia (transmisor) cuya matrícula es TXM-315-LC fabricado por LINX Technologies [1]. Dicho módulo estará conectado con el puerto serie de la PC.

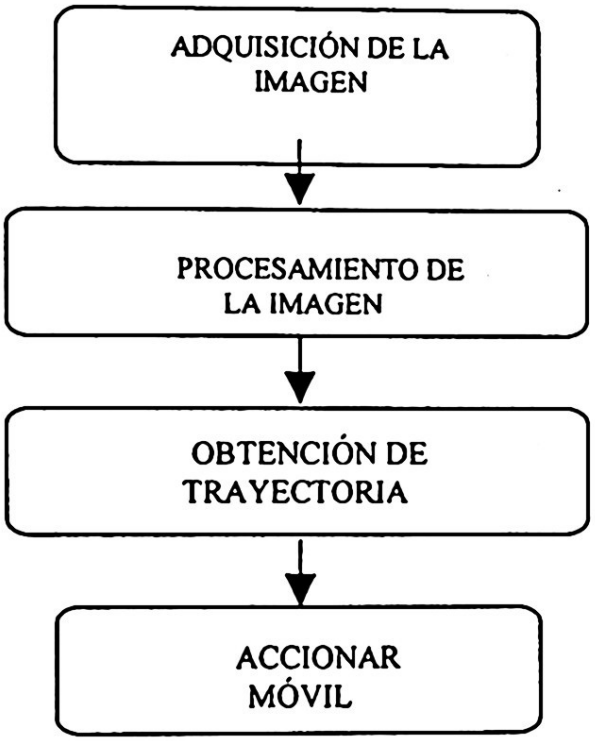


Fig. 1. Descripción general del sistema

En el móvil la trama será recibida por el módulo (receptor) RXM-315-LC-S fabricado también por LINX Technologies [1], el cual estará conectado a la UART del microprocesador con matrícula 90S8535 de la empresa ATMEL, con el fin de que en éste se procese dicha trama para obtener la información que accionará los motores del móvil y la trayectoria que deberá seguir para llegar al objetivo.

3 Descripción del sistema de reconocimiento

El sistema de reconocimiento que se propone se muestra en la figura 2.

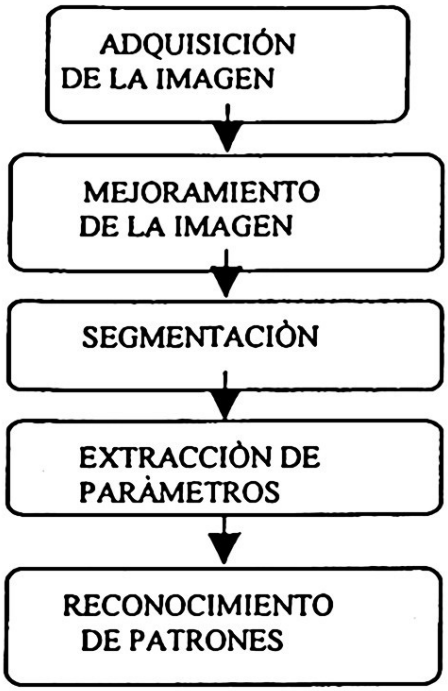


Fig. 2 Diagrama a bloques del sistema de reconocimiento

3.1 Adquisión de la imagen

La imagen se esta adquiriendo con una webcam cuyas características se presentan a continuación. Marca: LOGITECH; Sensor: CMOS; Enfoque manual; Resolución: 320 x 240; Velocidad: 30fps; Puerto: USB; Software: Microsoft NetMeeting.

La siguiente imagen fue capturada con esta webcam.

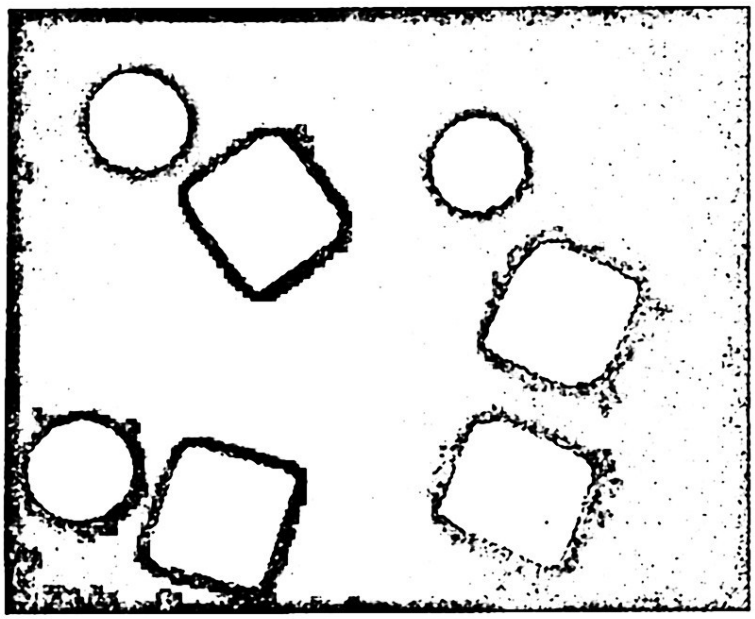


Imagen 1. Imagen de un escenario definido

3.2 Mejoramiento de la imagen

A partir de la imagen adquirida obtenemos una imagen a escala de grises [2].

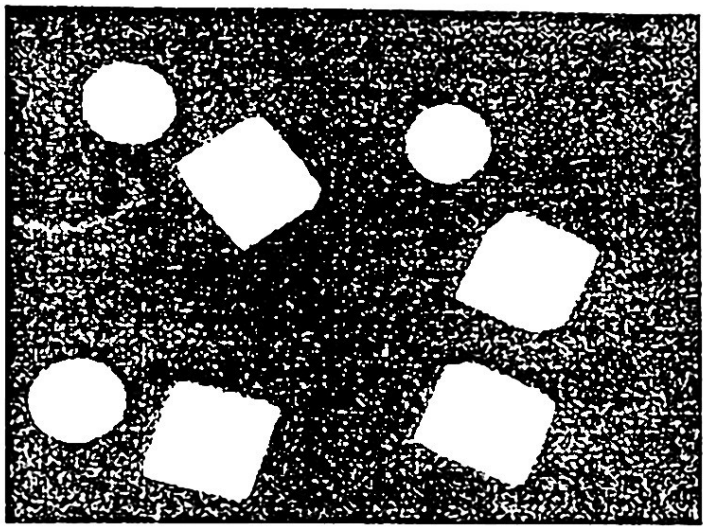


Imagen 2. Imagen a escala de grises

Posteriormente, aplicamos el filtro de la mediana con el fin de eliminar ruido, como el que se ve en la Imagen 1.

Filtrado de la mediana: la mediana m de un conjunto de valores es tal que la mitad de los valores del conjunto quedan por debajo de m y la otra mitad por encima [2].

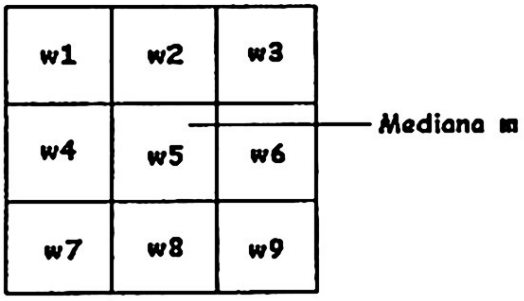


Fig. 3. Filtrado especial por la mediana

El filtrado de un píxel se lleva a cabo de la siguiente forma: primero se deben extraer los valores del píxel y de su entrono, determinar la mediana y asignar este valor al píxel. Si hay un valor más pequeño que otro en el centro de un entrono n*n entonces se agrupan los valores de tal forma que en el centro quede el mayor. Y así hay más parecidos de intensidades entre vecinos.

La imagen que se obtiene con este filtro es la siguiente.

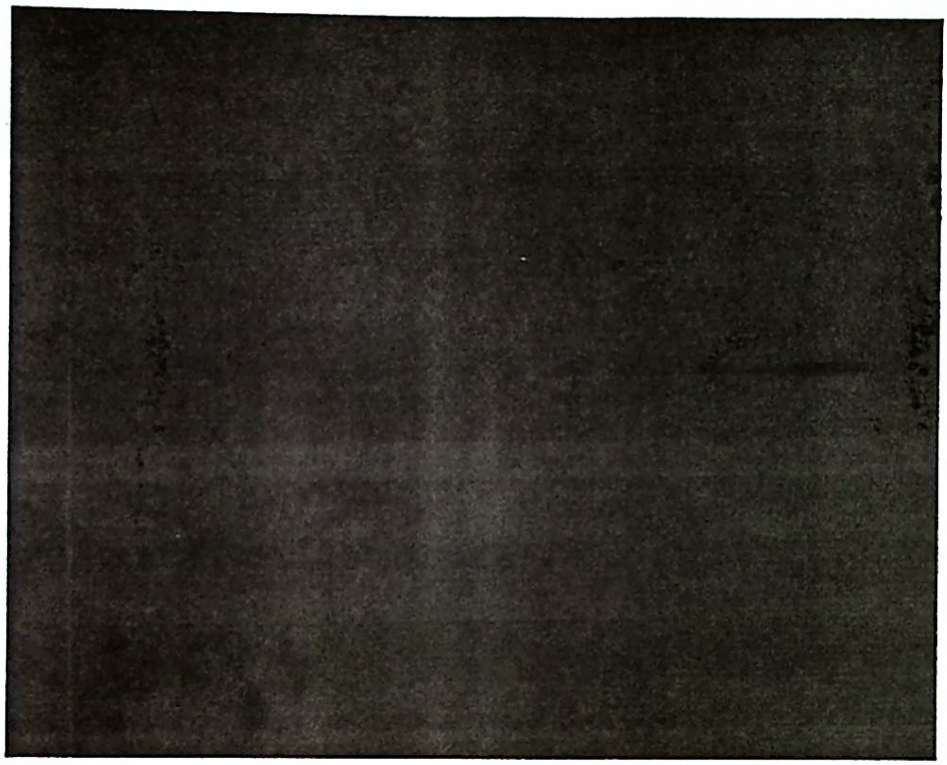


Imagen 3. Imagen después de aplicar la mediana.

3.3 Segmentación

El segmentado de la imagen se realiza con el método de umbralización.
Una imagen umbralizada [2] $g(x, y)$ se define como:

$$g(x, y) = \begin{cases} 1 : \text{Sí } f(x,y) > T \text{ -> Objetos} \\ 0 : \text{Sí } f(x,y) <= T \text{ -> Fondo} \end{cases}$$

0 -> negro
1 -> blanco

A la anterior asignación de pixeles se le conoce como binarización.
El umbral T se obtiene a partir del histograma de la imagen.
La siguiente imagen es una imagen umbralizada.

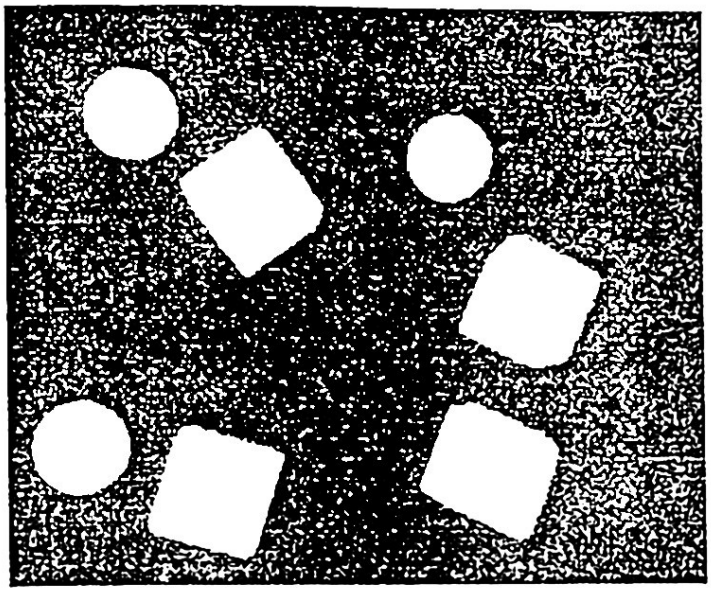


Imagen 4. Figura umbralizada

3.4 Extracción de parámetros

A partir de la imagen umbralizada se etiquetan los objetos con el fin de diferenciar cubos de círculos.

3.4.1 Etiquetado de componentes 8 conexas [2]

El barrido de una imagen pixel a pixel, de izquierda a derecha y de arriba a abajo. Sea p el pixel en cada paso de proceso del barrido y sean q, r, s y t, los vecinos a la izquierda, el superior izquierdo, el superior y el superior derecho, respectivamente, ver Figura 4. La naturaleza de la secuencia de barrido asegura que cuando se llega a p, los puntos q, r, s y t ya han sido hallados (y etiquetados si fueron unos).

r	s	t
q	p	

Fig. 4. Distribución de los pixeles en el recorrido del etiquetado.

Con los conceptos anteriormente establecidos considérese el procedimiento siguiente: si el valor de p es cero, simplemente se continua hasta la siguiente posición de barrido. Si el valor de p es uno, se examinan q, r, s y t, si todos son cero, se asigna una nueva etiqueta a p. Si uno o más vecinos son uno y tienen la misma etiqueta, se asigna esta etiqueta a p. Si más de un vecino son uno con etiquetas distintas, se asigna alguna de estas etiquetas a p y una nota de que estas etiquetas son equivalentes. Al término del barrido todos los pixeles que tenían uno, han sido etiquetados. Lo único

que resta es asignar una misma etiqueta a cada conjunto de etiquetas que hayan sido anotadas como equivalentes.

3.4.2 Prueba

Para diferenciar las distintas etiquetas se le asigna un color diferente a cada una de ellas, como se muestra en la siguiente imagen.

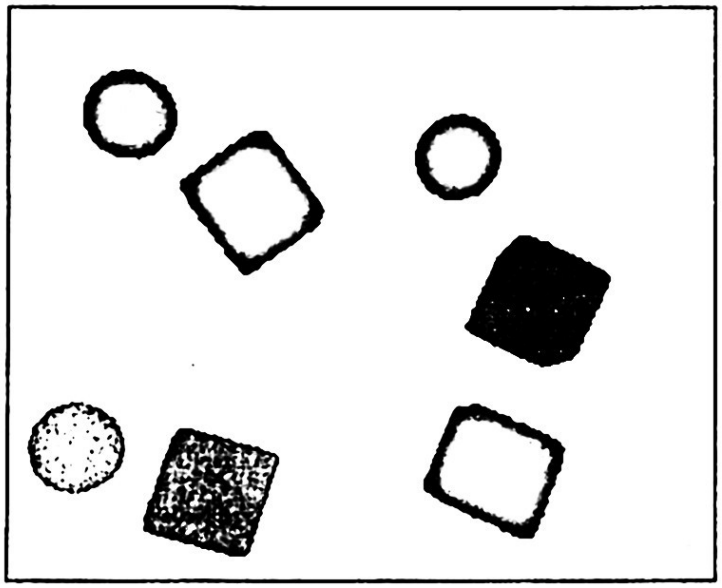


Imagen 5. Imagen etiquetada

El proceso de extracción de parámetros partirá de las etiquetas que se obtuvieron.

Existen dos tipos de descriptores, por contorno y por región [2]. Los descriptores por contorno se obtienen exclusivamente a partir de los pixeles que conforman el contorno de los objetos. A diferencia de estos, los descriptores por región utilizan a todo el conjunto de pixeles que conforman a un objeto, ejemplo de estos descriptores son los momentos de Hu y los de Flusser Suk que se detallan a continuación.

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y)$$

para p, q = 0, 1, 2, ...

Los momentos centrales se pueden expresar como:

$$m_{pq} = \sum_x \sum_y (x-\bar{x})^p (y-\bar{y})^q f(x,y)$$

donde:

$$\bar{y} = \frac{m_{01}}{m_{00}}; \bar{x} = \frac{m_{10}}{m_{00}}$$

Los momentos centrales normalizados, representados por η_{pq}, se definen como:

$$n_{pq} = \frac{\mu_{pq}}{\mu_{00} g}, \quad g = \frac{p+q}{2} + 1, \quad p+q = "2,3,. \; . \; .\{ \}\{ \}$$

De los momentos segundo y tercero se puede derivar un conjunto de siete momentos invariantes o momentos de Hu:

$$\varphi_1 = n_{20} + n_{02}$$

$$\varphi_2 = (n_{20} - n_{02})^2 + 4n_{11}$$

$$\varphi_3 = (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2$$

$$\varphi_4 = (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2$$

$$\varphi_5 = (n_{30} - n_{12})(n_{30} + n_{12})\left[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2\right] + (3n_{21} - n_{03})(n_{21} + n_{03})\left[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2\right]$$

$$\varphi_6 = (n_{20} - n_{02})\left[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2\right] + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03})$$

$$\varphi_7 = (3n_{21} - n_{30})(n_{30} + n_{12})\left[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2\right] + (3n_{12} - n_{30})(n_{21} + n_{03})\left[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2\right]$$

Este conjunto de momentos es invariante a traslaciones, rotaciones y cambios de escala.

Momentos de Flusser y Suk. Se considera una representación regional, y se utilizan como rasgos de interés para la descripción de cada una de estas regiones los 6 primeros momentos invariantes afines de Flusser y Suk.

3.5 Reconocimiento de patrones

Los momentos de Hu nos describen las características del objeto por lo que estos serán utilizados para la entrada de la red neuronal.

Las redes neuronales resuelven problemas de clases linealmente separables como problemas de clases no linealmente separables.

Utilización de una multitud de elementos de cálculo no lineales y elementales (denominados neuronas), organizadas como redes que se asemejan a la forma en que se piensa que están interconectadas las neuronas en el cerebro. Usaremos estas redes como vehículos para el desarrollo de los coeficientes de las funciones de decisión mediante la presentación sucesiva de conjunto de patrones de entrenamiento.

Los modelos de redes neuronales se muestran a continuación:

Perceptrón simple: Aprende una función de decisión lineal que dicotomiza dos conjuntos de entrenamiento linealmente separables.

Perceptrón Multicapa: Resuelve problemas linealmente y no linealmente separables

Algoritmo de Entrenamiento: BackPropagation

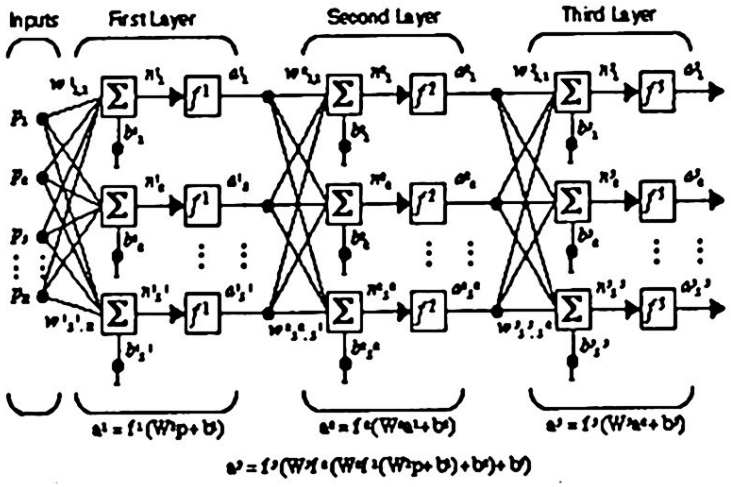
La red de retropropagación adapta sus pesos, para adquirir un entrenamiento a partir de un conjunto de pares de patrones entrada/salida. Después que la red ha aprendido ha esta se le puede aplicar un conjunto de patrones de prueba, para ver como esta generaliza a patrones no proporcionados.

El flujo de información fluye de la capa de entrada a la de salida a través de la capa oculta.

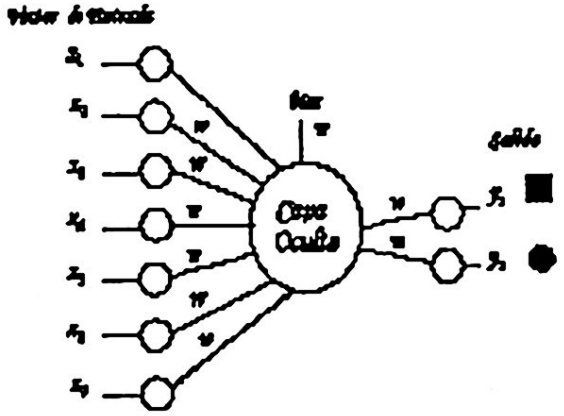
Cada unidad de procesamiento en la capa se conecta a todas las de la siguiente capa. El nivel de activación en la capa de salida determina la salida de la red.

Capa de Salida

$$\delta_j(n) = e_j + (V_j(n))$$



Capa oculta



$$\delta_j(n) = (V_j(n))\sum_k \delta_k(n) w_{jk}(n)$$

Actualizar los pesos

$$w_{ij}(n+1) = w_{ij}(n) + n\delta_j(n)y_j(n)$$

4 Descripción del sistema de navegación

Navegación.- Metodología que permite guiar el curso de un robot móvil a través de un entorno con obstáculos. Existen diversos esquemas, pero todos ellos poseen en común el afán por llevar el vehículo a su destino de forma segura.

Las tareas involucradas en la navegación de un robot móvil son:

- Percepción del entorno a través de sus sensores, de modo que le permita crear una abstracción del mundo.
- Planificación de una trayectoria libre de obstáculos, para alcanzar el punto destino seleccionado.
- Generación del camino.- Define una función continua que interpola la secuencia de objetos construida por el planificador. Posteriormente procede al a discretización de la misma a fin de generar el camino.
- Seguimiento del camino.- Efectúa el desplazamiento del vehículo, según el camino generado mediante el adecuado control de los actuadores del vehiculo.

La planificación de tareas y movimientos de un robot admite una clasificación según el nivel de abstracción al que se realice el trabajo. Se considera un modulo al que se le debe dar un objetivo y una situación inicial y que en función de ellos devuelve un plan de acción (Torras, 1989). El módulo debe poseer (o ser capaz de obtener), por ejemplo, una descripción del entorno, las acciones posibles, estrategias predefinidas, restricciones especificas, entre otra información.

El enfoque más generalizado de funcionamiento es mediante el fraccionamiento de la tarea en un número finito de estados diferentes y hacer un plan consistente en una secuencia de estados que lleve del inicial al deseado acompañada de los operadores para realizar las transiciones de un estado a otro. La forma más usual de representación del problema consiste en el uso de grafos de estados, en los que los nodos representan los estados de la tarea y los arcos las distintas transiciones posibles, puede que con algún índice de coste de la transición asociado a los mismos. De esta manera, el problema de planificación se convierte en el problema de búsquedas de caminos en grafos, donde se pueden aplicar diversas estrategias.

Dentro del ámbito de la robótica, una vez resuelto el problema de las distintas fases o estado de la tarea se debe pasar a la planificación de las trayectorias del robot que pueden ser en el espacio libre evitando colisiones o en contacto cuando este sea inevitable.

La planificación de movimientos libres de colisión es un campo que ha tenido un gran desarrollo, en particular debido a los denominados robots móviles, que, en la mayoría de los casos, de forma natural reducen el problema a tres grados de libertad, haciéndolo accesible a aplicaciones reales. La herramienta más popular en este campo es el Espacio de Configuraciones, es decir el espacio determinado por el conjunto de variables necesarias para determinar unívocamente la posición de los objetos móviles

involucrados en la tarea. El Espacio de Configuraciones puede representar de manera muy directa las posiciones accesibles e inaccesibles del robot, y por tanto realizar un plan de movimientos se reduce a encontrar un camino continuo en dicho espacio cuyos puntos sean todos accesibles. Para ello existen diferentes métodos, como por ejemplo, los basados en: grafos de visibilidad, diagramas de Voronoi, particiones del espacio libre y búsqueda de caminos entre cada par de subconjuntos vecinos, y funciones potenciales.A continuación se muestra el diagrama a bloques de la navegación implementada en este proyecto:

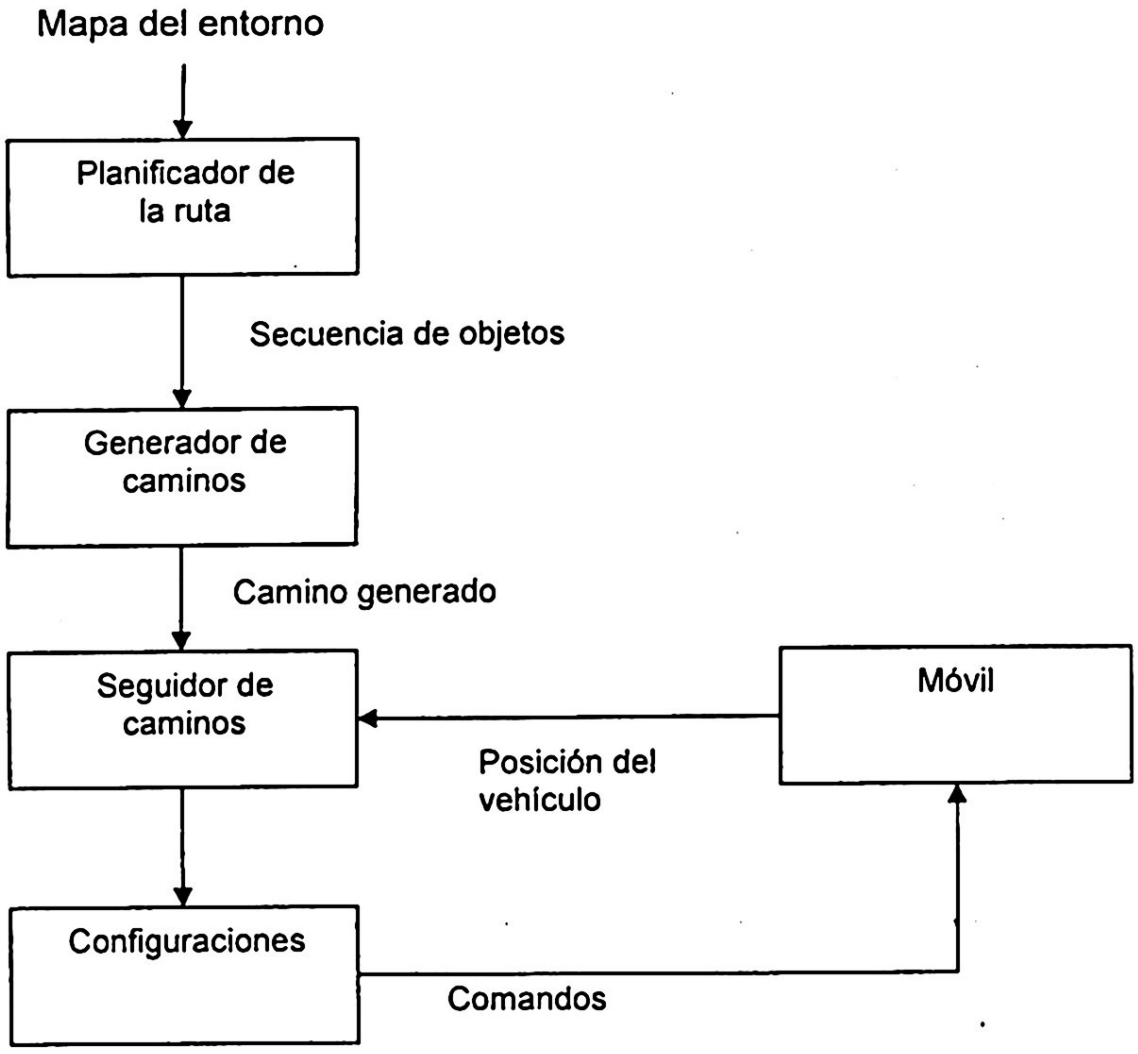


Fig. 5.. Planificador de la ruta

Mediante lo explicado anteriormente se descompone la imagen en celdas (Figura 6) para minimizar los datos con los que se va a obtener la ruta óptima.

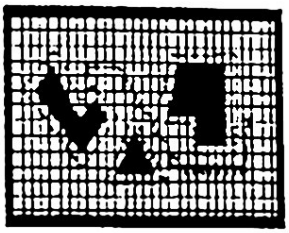
Esta celdas pueden pertenecer alguno de los siguientes espacios:

Espacio de configuración prohibido

Cforb(R,S)

Espacio de configuración libre

Cfree(R,S)
Espacio de configuración del obstáculo
C-obstaculo



Descomposición en celdas - Entrejardín

Fig. 6.

Para la obtención de la ruta optima se usara un método heurística el cual consiste en verificar los vecinos cercanos y lejanos con el fin de encontrar al que se encuentre mas cerca de la posición final.

5 Descripción del sistema de comunicación

Una vez que se tiene la trayectoria se forma la trama que saldrá por el puerto serie(UART- Universal Asynchronous Receiver Transmitter) de la PC[4].

5.1 Transmisión de la trama

CARACTERÌSTICAS DEL MAX232
El circuito integrado MAX232 cambia los niveles TTL a los del estándar RS-232 cuando se hace una transmisión, y cambia los niveles RS-232 a TTL cuando se tiene una recepción. El circuito típico se muestra en la siguiente figura:

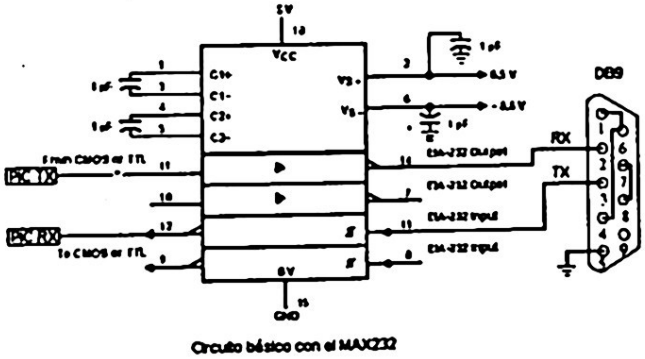


Fig. 7. Circuito bàsico de conexión del MAX232

CARACTERÌSTICAS DEL TRANSMISOR TXM-315-LC
Es ideal para aplicaciones donde se utilice control remoto, seguridad, identificación y transferencia de datos periódica, no utiliza componentes RF a excepción de una antena que es necesaria para su uso.
Características:

- Bajo costo.
- Muy bajo consumo de energía.
- Basado en arquitectura SAW.
- Transfiere información a 5Kbps.
- El rango de voltaje es de 2.7-5.2VDC.
- Presenta armónicos pequeños.

5.2 Recepción de trama

El siguiente módulo estará sobre el chasis del móvil.
CARACTERÌSTICAS DEL RECEPTOR RXM-315-LC

- Es eficiente, de bajo costo.
- Consume poca energía.
- Basado en arquitectura SAW.
- Notable sensibilidad.
- Soporta transferencias de 5Kbps.
- Cuenta con interfaz serial.
- Recibe información a una distancia máxima de 90 metros.

Modo de operación.

El tipo de modulación se presenta con un '0' lógico cuando no se presenta un mensaje, '1' lógico cuando hay mensaje.

Este método de modulación aporta numerosos beneficios, dos de los más importantes son:

1.- Eficacia contemplando el costo debido al diseño.
2.- La salida se produce con mucha potencia.

La SAW provee de una precisión muy alta de la frecuencia original con excelente inmunidad al ruido, permite incrementar la sensibilidad y reduce la interferencia por bandas cercanas.

La calidad de los componentes y arquitectura utilizada en la serie LC es inusual en un producto de bajo costo y esto hace que sean capaces de igualar la salida de algunos componentes más caros.

CARACTERÌSTICAS DEL MICROPROCESADOR ATM90S8535[3]

En este micro se va a decodificar la trama.

- Arquitectura RISC
- 32 registros de propósito general de 8bits
- 118 instrucciones
- Memoria de programa no volátil y datos
- Flash de 8Kbytes
- 512 bytes de EEPROM
- 512 bytes de memoria SRAM interna

Características de los periféricos

- 8 canales , convertidos AD de 10 bits
- UART programables

- 2 Timer/Counter de 8 bits con preescala y modo de comparación separado
- 1 Timer/Counter de 16 bits con preescala y modo de comparación separado y de 8,9 ó 10 bits para PWM (Modulación por ancho de pulso)
- Watchdog programable con oscilador interno
- Comparador analógico

Características Especiales del Microprocesador

- Circuito de reset
- Reloj de tiempo real con oscilador separado y modo contador
- Fuentes de interrupción internas y externas
- Tres modos de dormido, power save,power down e Idle
- Interfaz serial
- Consumo de corriente a 4MHz , 3V y 20°C
- I/O y paquetes
- 32 líneas programables de I/O
- Voltaje de operación
- Vcc 4.0 - 6V
- Grado de velocidad 0 - 8 MHz

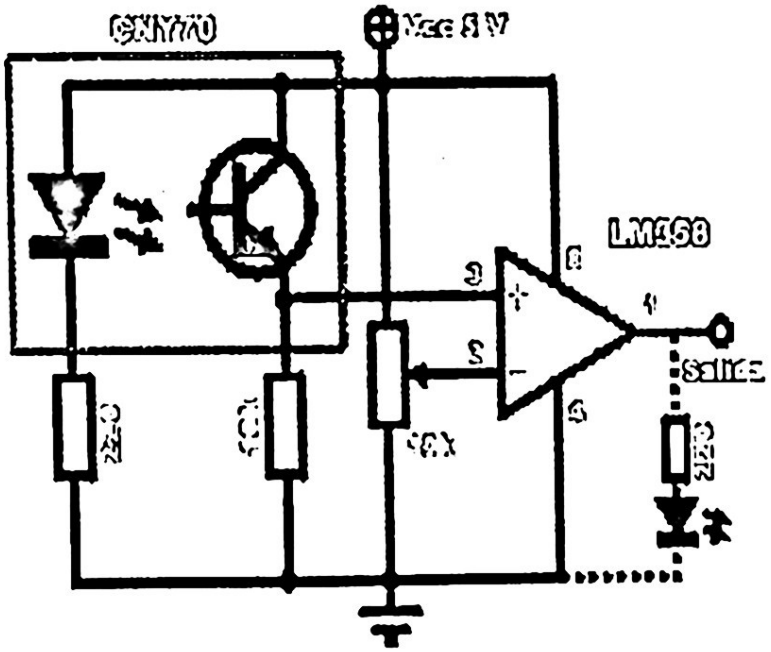


Fig. 8. Configuración del CNY70

CARACTERÌSTICAS DEL INTEGRADO L293D

El control de los motores se llevará a cabo mediante este circuito., cuya característica principal es que cuenta con un puente H, lo que elimina hardware al no requerir de una etapa de potencia externa

CARACTERÌSTICAS DEL SENSOR CNY70

Este sensor a través de un encoder como el de la figura inferior, nos permitirá conocer la distancia específica que recorrió el móvil, esto será gracias a un disco con

"n"zonas diferenciadas de este tipo acopladas a cada rueda. Esto proporciona una resolución de "n" pulsos por vuelta. Permitiendo así tener una referencia del movimiento de cada motor.

6 Móvil

Espacio de configuración.- Es el principal elemento que se considera en el análisis de un problema. En él se describen todas la posibles configuraciones del entorno y es el espacio n-dimensional donde se ubica cada grado de libertad del robot móvil.

El espacio de configuraciones sirve para saber si existe una ruta que pueda utilizar el móvil para desplazarse de un punto a otro.

Por lo anterior, se decidió que el arreglo de ruedas a utilizar sea un arreglo de ruedas de triciclo, cuya definición y configuración se muestra a continuación.

Tipo triciclo.- Cuenta con dos ruedas fijas con el mismo eje, que dan tracción y una rueda independiente para la dirección.

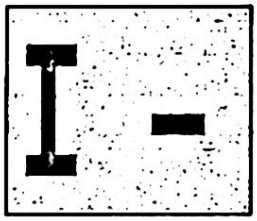


Fig. 9 Arreglo de ruedas triciclo

La restricción cinemática con la que cuenta es holonómica. Los grados de libertad están desacoplados, es decir se cuenta con un arreglo de ruedas diferencial o sincrono. Debido a esta situación es posible desacoplar la posición de orientación, ya que el móvil tiene la capacidad de girar sobre su propio eje.

El diseño del móvil es el siguiente:

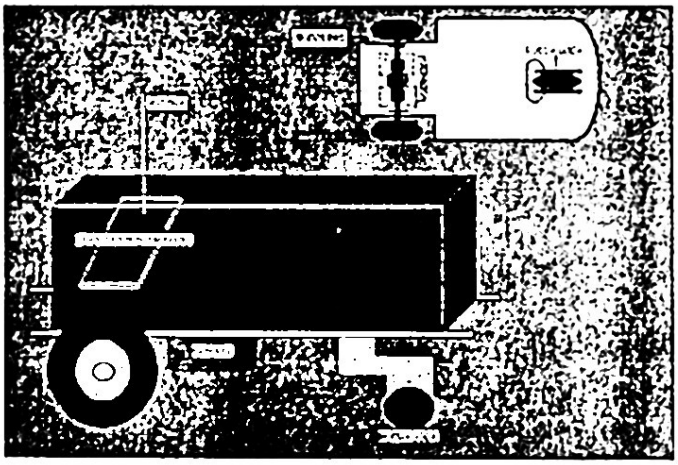


Fig. 10 Diseño físico del móvil

La forma del robot móvil tiene un fuerte impacto en su facilidad de navegación, en particular si el escenario que se va a recorrer cuenta con obstáculos y pasillos angostos. La forma del mòvil es cuadrado. La navegación depende de la orientación del robot, aumentando así el espacio de configuraciones a 3D.

7 Conclusiones

Existen numerosas técnicas para el procesamiento de imágenes en este trabajo se utilizan algunas de ellas para el reconocimiento de objetos, y también se buscaron aquellas que no complicaran tanto el proceso de búsqueda de la ruta óptima que el móvil deberá seguir para llegar al objetivo. Actualmente se están realizando pruebas del sistema de reconocimiento y generador de trayectorias en cada una de las etapas que lo conforman.

El sistema de hardware que se presenta pretende realizar la comunicación por radiofrecuencia y el control del móvil en los módulos de transmisión y recepción respectivamente. Este sistema se encuentra en su etapa de construcción teniendo el diseño del mismo en un 100%.

Al móvil se le están haciendo pequeños ajustes.

De esta manera se pretende tener un sistema completo, con un software de reconocimiento y generador de trayectorias en conjunción con un hardware específico para la aplicación.

Referencias

[1] http://linxtechnologics.com
[2] R. C. González and E. C. Woods, "Digital Image Processing", Ed. Addison Wesley, 1992.
[3] www.atmel.com
[4] Norma RS232

A Comprehensive Description of Kilo-Instruction Processors

Adrian Cristal, Oliverio J. Santana, and Mateo Valero

Departament d'Arquitectura de Computadors
Universitat Politècnica de Catalunya
Barcelona, Spain
{adrian,osantana,mateo}@ac.upc.es

Abstract. Although the memory access latency can be tolerated by maintaining a high number of in-flight instructions, the continuous increase in the gap between processor and memory speed increases the number of in-flight instructions required, causing scalability problems in the design of the critical structures of the processor. Our approach to support thousands of in-flight instructions, while avoiding scalability problems, is the kilo-instruction processor. This affordable architecture relies on an intelligent use of the available resources instead of simply up-sizing the processor structures. The high number of in-flight instructions maintained by our architecture allows it to achieve a high performance, even in the presence of large memory latencies, which makes the kilo-instruction processor an efficient architecture for dealing with future memory latencies.

1 Introduction

A lot of research effort is devoted to design new architectural techniques able to take advantage of the continuous improvement in microprocessor technology. The current trend leads to processors with longer pipelines, which combines with the faster technology to allow an important increase in the processor clock frequency every year.

However, the main memory access latency has become an important limiting factor for the performance of high-frequency microprocessors. The DRAM technology improves at a speed much lower than the microprocessor technology. Due to this fact, each increase in the processor clock frequency causes that a higher number of processor cycles are required to access the main memory, degrading the potential performance achievable with the clock frequency improvement.

If the main memory access latency increase continues, it will be a harmful problem for future microprocessor technologies. Therefore, dealing with the gap between the processor and the memory speed is vital in order to allow high-frequency microprocessors to achieve all their potential performance. A plethora of well-known techniques has been proposed to overcome the main memory latency, like cache hierarchies or data prefetching, but they do not completely

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.144-154.© Centro de Investigación Computación, IPN, México

solve the problem. A different approach to tolerate the main memory access latency is to dramatically increase the number of in-flight instructions that can be maintained by the processor.

2 Increasing the Number of In-Flight Instructions

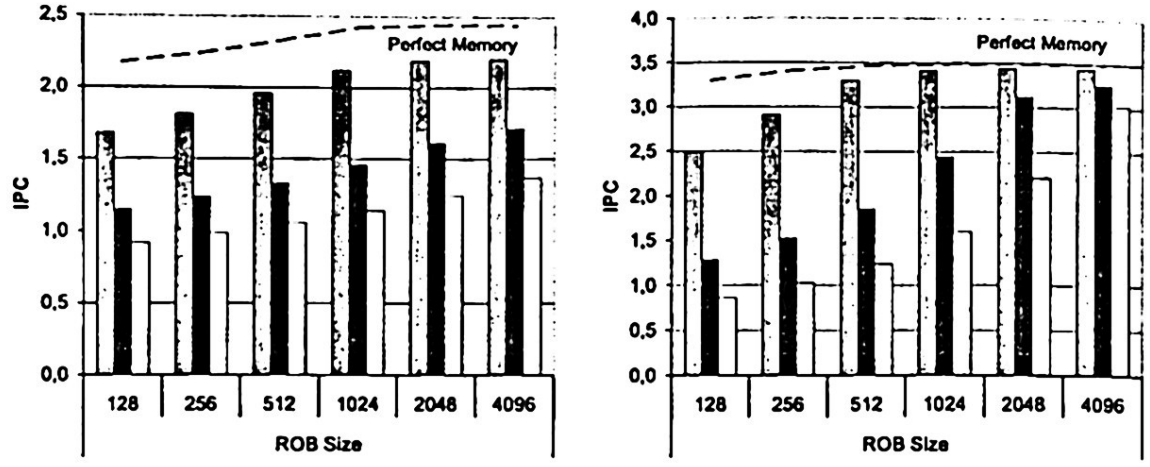
If the processor is able to maintain many in-flight instructions, the latency of a load instruction that access to the main memory can be overlapped with the execution of subsequent independent instructions, that is, the processor can hide the main memory access latency by executing useful work. Figure 1 shows an example of the impact of increasing the maximum number of in-flight instructions supported by an eight instruction wide out-of-order superscalar processor. The main memory access latency is varied from 100 to 1000 cycles. Data is provided for both the SPECint2000 integer applications and the SPECfp2000 floating point applications.

A first observation from this figure is that increasing the main memory latency from 100 to 1000 cycles causes enormous performance degradation. In a processor able to support 128 in-flight instructions, the integer applications suffer from an average 45% performance reduction. The degradation is even higher for floating point applications, whose average performance is reduced by 65%.

It is also clear that a higher number of in-flight instructions improves the processor performance. Increasing the number of in-flight instructions from 128 to 4096 in a processor having 100-cycle memory latency, the integer programs achieve an average 30% performance speedup and the floating point programs achieve 40% speedup. Nevertheless, it is important to note that the improvement is higher for larger memory access latencies. Increasing the number of in-flight instructions in a processor having 1000-cycle memory latency causes an average 50% performance improvement for the integer programs, while the floating point programs achieve a much higher 250% improvement. Such a high speedup is due to the larger amount of instruction-level parallelism available in floating point applications. This fact, along with a better branch prediction accuracy, allows floating point programs to take more advantage of a higher number of in-flight instructions.

These results show that increasing the number of in-flight instructions is an effective way of tolerating large memory access latencies. Although increasing the main memory latency from 100 to 1000 cycles causes big performance degradation for a processor able to maintain up to 128 in-flight instructions, a higher number of in-flight instructions alleviates this degradation, especially for floating point applications. If a processor is able to maintain up to 4096 in-flight instructions, the performance degradation caused by increasing the memory access latency from 100 to 1000 cycles is reduced by 10% in integer programs, and by 50% in floating point programs.

On average, executing integer programs, a processor able to maintain up to 4096 in-flight instructions having 1000-cycle memory latency is only 18% slower than a processor having 100-cycle latency but only being able to maintain up to



(a) SPECint2000 (b) SPECfp2000

Fig. 1. Average performance of an 8-wide out-of-order superscalar processor executing both the SPEC2000 integer and floating point programs. The maximum number of in-flight instructions supported is varied from 128 to 4096, and the main memory access latency is varied from 100 to 1000 cycles.

128 in-flight instructions. Moreover, when executing floating point applications, the processor supporting 4096 in-flight instructions with a 1000-cycle memory latency performs 22% better than the processor supporting 128 in-flight instructions, even when this processor has a much lower 100-cycle memory latency.

Therefore, future microprocessors will be able to tolerate large memory access latencies by maintaining thousands of in-flight instructions. The simplest way of supporting so much in-flight instructions is to scale all the processor resources involved, that is, the reorder buffer, the physical register file, the general purpose instruction queues (integer and floating point ones), and the load/store queue. However, scaling-up the number of entries in these structures is impractical, not only due to area and power consumption constraints, but also because these structures often determine the processor cycle time [14].

This is an exciting challenge. On the one hand, a higher number of in-flight instructions allows to tolerate large memory access latencies and thus provide a high performance. On the other hand, supporting such a high number of in-flight instructions involves a difficult scalability problem for the processor design. Our approach to overcome this scalability problem, while supporting thousands of in-flight instructions, is the kilo-instruction processor.

3 The Kilo-Instruction Processor

In essence, the kilo-instruction processor [5] is an out-of-order processor that keeps thousands of in-flight instructions. The main feature of our architecture is that its implementation is affordable. In order to support thousands of in-flight instructions, the kilo-instruction architecture relies on an intelligent use of the processor resources, avoiding the scalability problems caused by an ex-

cessive increase in the size of the main processor structures. Our design deals with the problems of each of these structures in an orthogonal way, that is, we apply particular solutions for each structure. These solutions are described in the following sections.

3.1 Multi-Checkpointing the Reorder Buffer

In a superscalar out-of-order processor, all instructions are inserted in the reorder buffer (ROB) after they are fetched and decoded. Therefore, the ROB is a microarchitectural mechanism that keeps a history window of all in-flight instructions, allowing for the precise recovery of the program state at any of those instructions. Instructions are removed from the ROB when they commit, that is, when they finish executing and update the architectural state of the processor.

However, for implementing precise recovery, instructions should be committed in-order, which is a serious problem in the presence of large memory access latencies. Let us suppose that a processor has a 128-entry reorder buffer and 500-cycle memory access latency. If a load instruction does not find a data in the cache hierarchy, it accesses the main memory, and thus it cannot be committed until its execution finishes 500 cycles later. When the load arrives to the head of the ROB, it blocks the in-order commit, and no later instruction will commit until the load finishes. Part of these cycles can be devoted to do useful work, but the ROB will become full soon, stalling the processor during several hundreds cycles.

To avoid this, a larger ROB is required, that is, the processor requires a higher number of in-flight instructions to overlap the load access latency with the execution of following instructions. Since each in-flight instruction requires an entry in the ROB, it should contain a high number of entries. However, scaling-up the number of ROB entries is impractical, mainly due to cycle time limitations.

The problem here is the presence of a centralized ROB structure devoted to provide precise recovery of the processor state. The kilo-instruction architecture solves this problem by replacing the ROB with a multi-checkpointing mechanism which also allows precise state recovery. Checkpointing is a well established and used technique [7]. The main idea is to create a checkpoint at specific instructions of the program being executed. This checkpoint can be though of as a snapshot of the state of the processor at that point, which contains all the information required to recover the architectural state and restart execution at that point.

The novelty of our mechanism is that the kilo-instruction architecture uses checkpointing to allow an early release of resources. Figure 2 shows an example of our checkpointing process [5]. First of all, it is important to state that there always exists at least one checkpoint in the processor (timeline A). The processor will fetch and issue instructions, taking new checkpoints at particular ones. If an instruction is miss-speculated or an exception occurs (timeline B), the processor rolls back to the previous checkpoint and resumes execution from there. When all instructions between two checkpoints are executed (timeline C), the last checkpoint is removed and its resources are freed (timeline D).

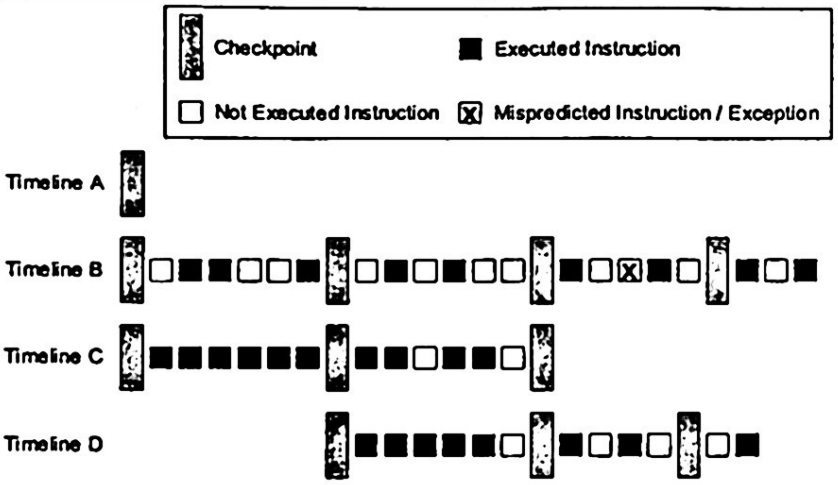


Fig. 2. The checkpointing process performed by the kilo-instruction processor.

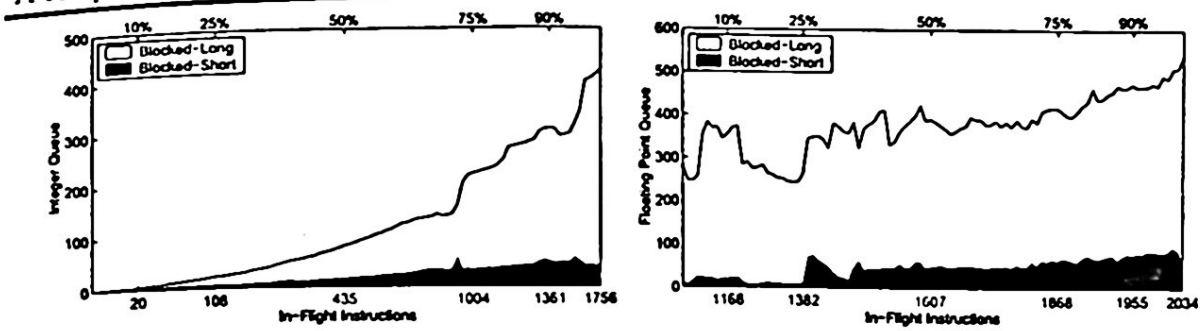
In case of a long-latency load, which accesses the main memory because its data is not in the cache hierarchy, the presence of a previous checkpoint allows that all the following instructions independent of the load result can finalize their execution and commit out-of-order, that is, they release their associated resources without having to wait until the load commits several hundreds cycles later. As a consequence, the multi-checkpointing mechanism makes it possible for the kilo-instruction processor to overlap large memory access latencies with the execution of thousands of following independent instructions without requiring an unimplementable centralized ROB structure with thousands of entries.

3.2 Instruction Queues

At the same time that instructions are inserted in the ROB, they are also inserted in their corresponding instruction queues. Each instruction should wait in an instruction queue until its execution finishes. All the instructions following a long-latency load can finalize their execution and be removed from the instruction queues due to the presence of a previous checkpoint. However, all the dependent instructions should be kept in the instruction queues until they finish their execution several hundreds cycles later.

This means that, in order to hide the load latency with the execution of thousands of following instructions, a typical instruction queue design should contain a high amount of entries, which makes it unpractical. The kilo-instruction processor solves this problem by taking advantage of the different waiting times of the instructions in the queues. These instructions can be divided in two types: blocked-short instructions when they are waiting for a functional unit or for results from short-latency operations, and blocked-long instructions when they are waiting for some long-latency instruction to complete.

Figure 3 shows the accumulative distribution of allocated entries in the integer queue (for SPECint2000 programs) and in the floating point queue (for SPECfp2000 programs) with respect to the amount of total in-flight instructions.



(a) Integer Queue (SPECint2000) (b) Floating Point Queue (SPECfp2000)

Fig. 3. Accumulative distribution of allocated entries in the integer queue (using the SPECint2000 programs) and in the floating point queue (using the SPECfp2000 programs) with respect to the amount of total in-flight instructions for a processor able to maintain up to 2048 in-flight instructions.

This data corresponds to a processor able to maintain up to 2048 in-flight instructions and having 500-cycle memory access latency. For example, in floating point applications, 50% of the time there are 1600 or less in-flight instructions, requiring 400 floating point queue entries. On average, the amount of entries allocated in the instruction queues is much smaller than the amount of in-flight instructions. However, to cope with over 90% of the scenarios the processor is going to face, the integer queue requires 300 entries and the floating point queue requires 500 entries, which is definitely going to affect the cycle time [14].

Fortunately, not all instructions behave in the same way. Blocked-long instructions represent by far the largest fraction of entries allocated in the instruction queues. These instructions are dependent on long-latency loads or on their dependents. Since these instructions take a very long time to even get issued for execution, maintaining them in the instruction queues just takes away issue slots from other instructions that will be executed more quickly. Multilevel queues can be used to track this type of instructions, delegating their handling to slower, but larger and less complex structures. Some previous studies have proposed such multilevel queues [8, 2], but they require a wake-up and select logic which might be on the critical path, thus potentially affecting the cycle time.

The kilo-instruction processor deals with this problem by using a simple secondary buffer called Slow Lane Instruction Queue (SLIQ). This queue is a FIFO-like structure that enables a simple but efficient wakeup and select process [5]. All the instructions dependent on a long-latency load are removed from the general purpose instruction queues and stored in-order in the SLIQ, freeing entries from the instruction queues that can be used by short-latency operations. Once the long-latency load finishes its execution, the dependent instructions are removed from the SLIQ and inserted back into their corresponding instruction queue, where they can start their execution. This mechanism allows to effectively implement the functionality of a large instruction queue while requiring a reduced number of entries, and thus it makes it possible to support a high number of in-flight instructions without scaling-up the instruction queues.

3.3 Load/Store Queue

Load and store instructions are inserted in the load/store queue at the same time they are inserted in the ROB. This queue takes care of memory disambiguation, that is, it guarantees that load and stores arrive to the memory in the correct order. Increasing the number of in-flight instructions also increases the number of loads and stores that should be taken into account, which can make the memory disambiguation logic a true bottleneck both in latency and power.

As for the instruction queues, the solution for this problem is using multilevel structures. Some recent works [1, 15, 16] describe such multilevel structures for performing memory disambiguation in a load/store queue containing a great amount of instructions. These works propose different filtering schemes that use two-level structures for storing most or all instructions in a big structure, while a smaller structure is used to easily check the dependencies.

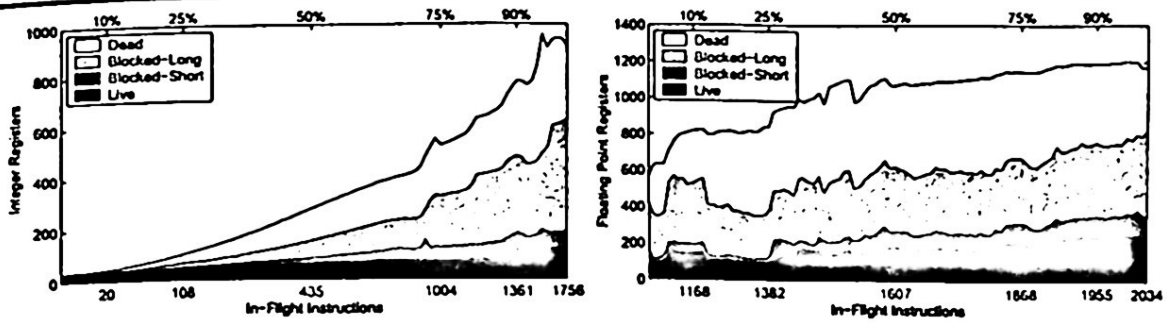
3.4 Physical Register File

Each instruction that generates a result uses a physical register to store it. Therefore, maintaining thousands of in-flight instructions involves that a high amount of physical registers is required. This high amount of registers increases the register file access time, especially taking into account the large number of access ports needed by this structure to implement an efficient issue mechanism. Nevertheless, since the physical register file is a critical component of superscalar processors, increasing its access time will surely involve an increase in the processor cycle time.

In order to reduce the number of physical register needed, the kilo-instruction processor relies on early register release and late register allocation. Figure 4 shows the accumulative distribution of allocated integer registers (SPECint2000 programs) and floating point registers (SPECfp2000 programs) with respect to the amount of total in-flight instructions. This data is provided for a machine able to maintain up to 2048 in-flight instructions and having 500-cycle memory access latency.

Registers are classified in four categories. Live registers contain values currently in use. Blocked-short and blocked-long registers have been allocated during rename, but are blocked because the corresponding instructions are waiting for the execution of predecessor instructions. Blocked-short registers are waiting for instructions that will issue shortly, while blocked-long registers are waiting for long-latency instructions. Finally, dead registers are no longer in use, but they are still allocated because the corresponding instructions have not yet committed.

It is clear that blocked-long and dead registers constitute the largest fraction of allocated registers. Some previous proposals describe how to made these registers available to other instructions, reducing the total amount of physical registers needed. In order to avoid blocked-long registers, the assignment of physical registers can be delayed using virtual tags [11]. These virtual register mapping keeps track of the rename dependencies, making unnecessary the assignment of



(a) Integer Registers (SPECint2000) (b) Floating Point Registers (SPECfp2000)

Fig. 4. Accumulative distribution of allocated integer registers (using the SPECint2000 programs) and floating point registers (using the SPECfp2000 programs) with respect to the amount of total in-flight instructions for a processor able to maintain up to 2048 in-flight instructions.

a physical register to an instruction until it starts execution. Dead registers can also be eliminated by using mechanisms for early register recycling [12]. These mechanisms release a physical register when it is possible to guarantee that it will not be used again, regardless the corresponding instruction has committed or not.

The kilo-instruction architecture combines these two techniques with checkpointing, leading to an aggressive register recycling mechanism that we call ephemeral registers [4, 10]. This is the first proposal that integrates both a mechanism for delayed register allocation and early register release and analyzes the synergy between them. The combination of these two techniques with checkpointing allows the processor to non-conservatively deallocate registers, making it possible to support thousands of in-flight instructions without requiring an excessive number of registers.

4 Real Performance

Figure 5 provides some insight about the performance achievable by the kilo-instruction processor. It shows the average performance of a kilo-instruction processor executing the SPECint2000 floating point applications. The kilo-instruction processor modeled is able to support up to 2048 in-flight instructions, but it uses just 128-entry instruction queues. It also uses 32KB separate instruction and data caches as well as an unified 1MB second level cache. The figure is divided into three zones, each of them comprising the results for 100, 500, and 1000 cycles of main memory access latency. Each zone is composed of three groups of two bars, corresponding to 512, 1024, and 2048 virtual registers or tags [11]. The two bars of each group represent the performance using 256 or 512 physical registers.

In addition, each zone of the figure has two lines which represent the performance obtained by a baseline superscalar processor, able to support up to 128 in-flight instructions, and a limit unfeasible microarchitecture where all

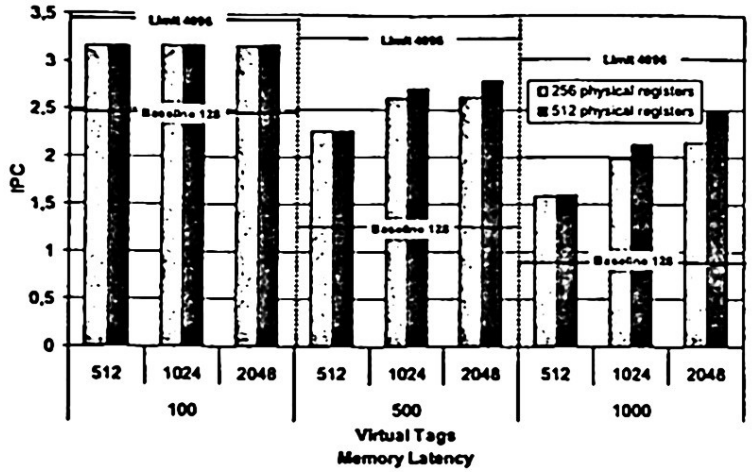


Fig. 5. Average performance results of the kilo-instruction processor executing the SPECfp2000 programs with respect to the amount of virtual registers, the memory latency, and the amount of physical registers.

the resources have been up-sized to allow up to 4096 in-flight instructions. The main observation is that the kilo-instruction processor provides important performance improvements over the baseline superscalar processor. Using 2048 virtual tags, the kilo-instruction processor is more than twice faster than the baseline when the memory access latency is 500 cycles or higher. Moreover, a kilo-instruction processor having 1000 cycles memory access latency is only a 5% slower than the baseline processor having a memory access latency 10 times lower.

These results show that the kilo-instruction processor is an effective way of approaching the unimplementable limit machine in an affordable way. However, there is still room for improvement. The distance between the kilo-instruction processor performance and the limit machine is higher for larger memory access latencies. This causes that, although the performance results for more aggressive setups nearly saturate for a memory access latency of 100 or 500 cycles, the growing trend is far from saturating when the memory access latency is 1000 cycles. This makes us believe that a more aggressive machine, able to support a higher number of in-flight instructions, will still provide a better performance.

5 Related Work

The first step in the design of our kilo-instruction architecture [3] was using checkpointing as an efficient way to control and manage the use of critical resources inside the processor. We propose to checkpoint critical long-latency instructions, which allows to create a very large virtual ROB, while actually using a small physical one. This multi-checkpointing mechanisms allows to release physical registers early and to remove load instructions early from the load/store queue. In addition, the multi-checkpointing mechanism is used to release instructions from the ROB early, which leads to an architecture where the classical ROB is essentially unnecessary [5].

Cherry [9] is another checkpointing scheme that was developed in parallel with the kilo-instruction processor. Instead of using a multi-checkpointing mechanism, Cherry is based on a single checkpoint outside the ROB. The ROB is divided in two regions: the region occupied by speculative instructions and the region occupied by non-speculative instructions. Cherry is able to release registers and load/store queue entries early in the ROB area not subject to misspeculation, providing precise exception handling using the checkpoint. On the other hand, the instructions belonging to the region subject to misspeculation (like speculative instructions after a non-resolved branch prediction) still depend on the reorder buffer to recover the correct state in case of misspecualtion, and so they are not able to release their corresponding resources.

A later proposal based in checkpointing is runahead execution [13], which follows the conceptual path of [9]. This technique creates a checkpoint of the architectural state when the head of the reorder buffer is reached by a load that has missed in the second level cache. In addition, the processor start executing instructions in a special mode using a bogus result for the load. When the load instruction actually completes, the processor returns to the normal mode, restoring the checkpoint. The first execution provides useful knowledge, like accurate data and instruction prefetches, that improves the performance during the second execution.

6 Conclusions

Tolerating large memory access latencies is a key topic in the design of future processors. Maintaining a high amount of in-flight instructions is an effective mean for overcoming this problem. However, increasing the number of in-flight instructions requires up-sizing several processor structures, which is is impractical due to power consumption, area, and cycle time limitations. The kilo-instruction processor is an affordable architecture able to support thousands of in-flight instructions. Our architecture relies on an intelligent use of the processor resources, avoiding the scalability problems caused by an excessive increase in the size of the critical processor structures. The ability of maintaining a high number of in-flight instructions makes the kilo-instruction processor an efficient architecture for dealing with future memory latencies, being able to achieve a high performance even in the presence of large memory access latencies.

Acknowledgements

This research has been supported by CICYT grant TIC-2001-0995-C02-01, the European Network of Excellence on High-Performance Embedded Architecture and Compilation (HIPEAC), and CEPBA. O. J. Santana is also supported by Generalitat de Catalunya grant 2001FI-00724-APTIND. Special thanks go to Francisco Cazorla, Ayose Falcón, Marco Galluzzi, Josep Llosa, José F. Martínez, Daniel Ortega, and Tanausú Ramírez for their contribution to the kilo-instruction processors.

References

1. H. Akkary, R. Rajwar, and S. T. Srinivasan. Checkpoint processing and recovery: towards scalable large instruction window processors. *Proceedings of the 36th International Symposium on Microarchitecture*, 2003.
2. E. Brekelbaum, J. Rupley, C. Wilkerson, and B. Black. Hierarchical scheduling windows. *Proceedings of the 35th International Symposium on Microarchitecture*, 2002.
3. A. Cristal, M. Valero, A. Gonzalez, and J. Llosa. Large virtual ROBs by processor checkpointing. *Technical Report UPC-DAC-2002-39*, Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, 2002.
4. A. Cristal, J. F. Martinez, J. Llosa, and M. Valero. Ephemeral registers with multicheckpointing. *Technical Report UPC-DAC-2003-51*, Departament d'Arquitectura de Computadors, Universitat Politècnica de Catalunya, 2003.
5. A. Cristal, D. Ortega, J. Llosa, and M. Valero. Out-of-order commit processors. *Proceedings of the 10th International Symposium on High-Performance Computer Architecture*, 2004.
6. M. Galluzzi, V. Puente, A. Cristal, R. Beivide, J. A. Gregorio, and M. Valero. A first glance at kilo-instruction based multiprocessors. *Proceedings of Computing Frontiers*, 2004.
7. W. M. Hwu and Y. N. Patt. Checkpoint repair for out-of-order execution machines. *Proceedings of the 14th International Symposium on Computer Architecture*, 1987.
8. A. Lebeck, J. Koppanalil, T. Li, J. Patwardhan, and E. Rotenberg. A large, fast instruction window for tolerating cache misses. *Proceedings of the 29th International Symposium on Computer Architecture*, 2002.
9. J. F. Martinez, J. Renau, M. Huang, M. Prvulovic, and J. Torrellas. Cherry: checkpointed early resource recycling in out-of-order microprocessors. *Proceedings of the 35th International Symposium on Microarchitecture*, 2002.
10. J. F. Martinez, A. Cristal, M. Valero, and J. Llosa. Ephemeral registers. *Technical Report CSL-TR-2003-1035*, Cornell Computer Systems Lab, 2003.
11. T. Monreal, A. Gonzalez, M. Valero, J. Gonzalez, and V. Viñals. Delaying physical register allocation through virtual-physical registers. *Proceedings of the 32nd International Symposium on Microarchitecture*, 1999.
12. M. Moudgill, K. Pingali, and S. Vassiliadis. Register renaming and dynamic speculation: an alternative approach. *Proceedings of the 26th International Symposium on Microarchitecture*, 1993.
13. O. Mutlu, J. Stark, C. Wilkerson, and Y. N. Patt. Runahead execution: an alternative to very large instruction windows for out-of-order processors. *Proceedings of the 9th International Symposium on High-Performance Computer Architecture*, 2003.
14. S. Palacharla, N. P. Jouppi, and J. E. Smith. Complexity-effective superscalar processors. *Proceedings of the 24th International Symposium on Computer Architecture*, 1997.
15. I. Park, C. Ooi, and T. Vijaykumar. Reducing design complexity of the load/store queue. *Proceedings of the 36th International Symposium on Microarchitecture*, 2003.
16. S. Sethumadhavan, R. Desikan, D. Burger, C. Moore, and S. Keckler. Scalable hardware memory disambiguation for high ILP processors. *Proceedings of the 36th International Symposium on Microarchitecture*, 2003.

Evaluación de una versión paralela para el Codec H.264/AVC

Carlos Genis Triana, Abelardo Rodríguez León

Departamento de Sistemas y Computación, Instituto Tecnológico de Veracruz,
Calzada Miguel Ángel de Quevedo 2779, Veracruz, México
carlosgenis@yahoo.com.mx, arleon@itver.edu.mx

Resumen. En este capítulo se presenta una evaluación del codificador de video basado en el estándar H.264/AVC propuesta por JVT contra una versión paralela implementada por GAP de DISCA-UPV. El primero realiza la tarea de manera secuencial mientras tanto el segundo segmenta y distribuye paralelamente una secuencia de video (GOPs) entre los nodos de un cluster. Para la distribución de la carga utilizaron el estándar MPI, desarrollada por el grupo Forum de MPI. Se observó un patrón de codificación no equivalente entre ambas versiones y una codificación no cíclica en la versión paralela, con lo cual el número de frames es una limitante para realizar pruebas de codificación con mayor número de GOPs. Las pruebas realizadas y analizadas en este capítulo, son el punto de partida que determinará el rumbo que seguirá la presente investigación a corto plazo. En dicha investigación se buscará implementar y evaluar un algoritmo paralelo con balanceo de carga para la compresión de secuencias de video, usando el codificador H.264, buscando tiempo real en la codificación, así como determinar el tipo de video más adecuado para esta versión.

1 Introducción

Existen muchas aplicaciones para la transmisión de imagen digital en movimiento tales como la videoconferencia, el control y monitorización de sistemas robotizados, la telemedicina, los canales de difusión en Internet o los sistemas de video por demanda. Sin embargo, la transmisión de imagen digital en movimiento se encuentra con el problema de que requiere el almacenamiento y procesamiento de datos multimedia. Debido a la naturaleza de estos datos, se necesita un ancho de banda excesivo y que hoy en día no está disponible para la mayoría de los usuarios. Por ejemplo, para transmitir video crudo (sin codificar) en una resolución de 176x144, utilizando 24 bits de profundidad y una tasa de 10 frames/seg, se necesitaría un ancho de banda de casi 6 Mbps. Además del problema del ancho de banda también existe el inconveniente del espacio utilizado para almacenar dicho video. Así, un segundo de un video de color con el estándar NTSC requiere casi 23 MBytes y un video de 90 minutos, en este mismo formato, requeriría aproximadamente 120 GBytes. Por lo tanto, se tiene que reducir la cantidad de información que generan las imágenes en movimiento a través del uso de técnicas de compresión de datos respetando, en la medida de lo posible, su calidad.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación México, 155-166.© Centro de Investigación en Computación, IPN, México

Para comprimir las imágenes digitales lo que se hace es eliminar la información redundante. Para ello, los sistemas de compresión se aprovechan de los tres tipos de redundancia existentes en los videos digitales:

- *Espacial*. Normalmente, los valores entre píxeles vecinos en una imagen son similares.
- *Espectral.* En imágenes compuestas por más de una banda espectral, los valores espectrales de un mismo píxel están fuertemente correlacionados.
- *Temporal*. Frames consecutivos en una secuencia de video presentan frecuentemente pocos cambios.

La redundancia espacial y la redundancia espectral se tratan a nivel de imágenes, ya que explotan la redundancia implícita en las imágenes, mientras que la redundancia temporal es explotada por técnicas (como predicción y compensación de movimiento) que sólo codifican las diferencias entre frames adyacentes pertenecientes a una secuencia de video.

En los últimos años se han propuesto y se han desarrollado una variedad de estándares de compresión de video e imagen (H.26X, MPEG-X, JPEG200,...). Sin embargo, no todos los estándares proporcionan las mismas características. Unos codificadores obtienen mayor calidad de imagen, a costa, sobre todo, de un mayor tiempo de procesamiento, por lo que se ha tenido que buscar un equilibrio entre el tiempo de procesamiento (tiempo de codificación) y la calidad que se obtiene de las imágenes decodificadas. Por tanto, si se aumentara la velocidad de procesamiento se podrían utilizar codificadores que ofrecen una calidad mayor de las imágenes y de las secuencias de video. Para conseguir una capacidad de procesamiento muy alta se tienen varias alternativas:

- *Sistemas fuertemente acoplados.* Como el proceso de codificación suele requerir mucho poder de cómputo generalmente se logra, usando computadoras de altas prestaciones que ayuden con sus sistemas multiprocesadores integrados en un solo gabinete, llevar a cabo este proceso con tiempos de retardo aceptables. El inconveniente es que son equipos muy costosos.
- *Tarjeta codificadora.* Desafortunadamente son pocas las tarjetas codificadoras que pueden dar un buen rendimiento en tiempo real (codificación menor o igual a la duración de la secuencia de video). Las que lo logran lo hacen con estándares como MPEG1 o MPEG2 como la Broadway de DataTraslation. Existen tarjetas que codifican también en MPEG4 aunque no lo hacen en tiempo real y con configuración limitada.
- *Sistemas débilmente acoplados.* Una alternativa que no requerirá usar costosos equipos es usar cluster de computadoras personales conectadas por una red, las cuales se han hecho muy populares ya que ofrecen muy buenas prestaciones a un precio más económico y con mayor escalabilidad que los sistemas fuertemente acoplados. Para lograr esto hay que implementar los algoritmos de compresión usando técnicas de programación paralela.

En este trabajo de investigación se presenta una evaluación del codificador de video basado en el estándar H.264/AVC propuesta por el Joint Video Team (JVT)

contra una versión paralela implementada por el Grupo de Arquitecturas Paralelas (GAP) del Departamento de Informática, Sistemas de Cómputo y Automática de la Universidad Politécnica de Valencia (DISCA-UPV) [1]. Cabe señalar que la paralelización de la carga la hicieron utilizando la técnica estándar de programación paralela para el paso de mensajes (MPI), desarrollada por el grupo Forum de MPI

En el punto 1 se hace una introducción a la codificación de video. En los puntos 2 y 3 se hacen descripciones generales de los estándares MPI y H.264/AVC, respectivamente. En el punto 4 se enuncian las características de la versión paralela. En el punto 5 se muestran algunos resultados obtenidos de la evaluación del codificador secuencial contra el paralelo y finalmente en el punto 6 se hace mención sobre el trabajo futuro que se tiene contemplado realizar tomando como punto de referencia las pruebas citadas en este documento.

2 MPI

MPI fue desarrollado por el Forum MPI, que es un grupo abierto en representación de una amplia sección de la industria y de intereses académicos. MPI es un conjunto de funciones (API) que permite a los programadores escribir aplicaciones paralelas. Estas aplicaciones están formadas por una serie de procesos que intercambian mensajes (operaciones cooperativas) a través de MPI para llevar a cabo un trabajo común. Esta librería de paso de mensajes está especificada para C, C++ y Fortran.

MPI se puede utilizar tanto en sistemas fuertemente o débilmente acoplados. Sin embargo, hoy en día, se suele utilizar fundamentalmente con cluster de computadores, ya que permiten obtener una gran potencia de computación a un bajo costo. Debido a la gran variabilidad de entornos donde se utiliza MPI, se puede deducir que el estándar es portable e independiente de la plataforma en la que se ejecuta.

La especificación está dividida en dos partes: MPI-1 (versión 1.2) y MPI-2 (versión 2.0). En un principio fue diseñada para permitir el desarrollo de librerías paralelas, pero actualmente también proporciona un acceso paralelo al hardware tanto a usuarios finales, como a desarrolladores de librerías y de aplicaciones.

MPI utiliza el paso de mensajes porque es un paradigma de programación fácilmente entendible, eficiente y que emplean múltiples aplicaciones. Además es ideal para desarrollar programas paralelos que sean portables o para aumentar la productividad de un sistema secuencial. Además las características de MPI le hacen ser un sistema muy completo: modular (ideal para las librerías), portable, dispone de comunicaciones seguras, permite crear subgrupos y existen herramientas para medir el rendimiento que ofrece.

3 H.264

H.264 es uno de los estándares de codificación de video más nuevos desarrollado por el VCEG (*Video Coding Experts Group*) y por el MPEG (*Moving Picture Experts Group*). El VCEG es un grupo que pertenece a la ITU-T (*International Communications Union*), mientras que el MPEG pertenece a la ISO/IEC

(*International Standarization Organization/International Electrotechnical Commission*).

El objetivo principal por el cual surgió el estándar H.264 fue el de aumentar la tasa de compresión de video. Antes de salir este estándar, existían otros codificadores de video como el MPEG-2. Este codificador se utilizaba básicamente para la transmisión de señales de alta definición (HD) y de definición estándar (SD) sobre satélite o cable y para el almacenamiento de señales de SD en DVDs. Pero, debido al crecimiento de la popularidad de la televisión de alta definición, se hace necesario utilizar técnicas de codificación más eficientes que MPEG-2. Además, existen otros medios de transmisión (xDSL, UMTS) que ofrecen tasas de transferencia mucho menores que los canales broadcast y, por lo tanto, necesitan de una mayor tasa de compresión.

En 1998 el VCEG publicó un proyecto (H.26L) que tenía el propósito de doblar la eficiencia de la codificación que obtenían los estándares de codificación que había en ese momento. En octubre de 1999 salió la primera versión de este proyecto. En diciembre del 2001, el VCEG y el MPEG se unieron y formaron el JVT (Joint Video Team) con el objetivo de desarrollar un estándar de codificación de video nuevo. Fruto de esta unión, en marzo del 2003, se publicó el H.264 /AVC.

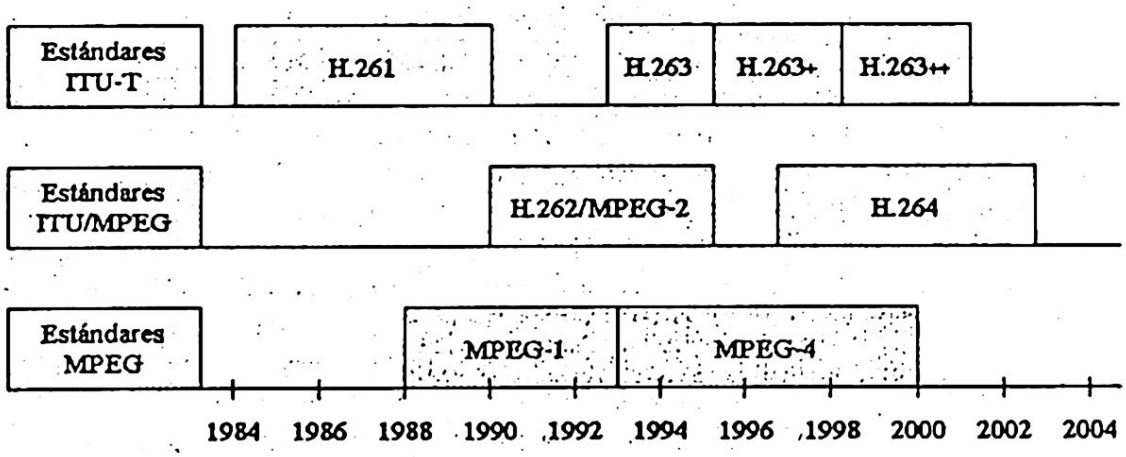


Fig. 1. Evolución de las recomendaciones ITU-T y estándares MPEG

En la figura 1 se muestra la evolución de las recomendaciones propuestas por la ITU-T y de los estándares de la ISO/IEC. Las recomendaciones de la ITU-T (denotadas como H.26x) fueron diseñadas principalmente para aplicaciones de video en tiempo real. Mientras que los estándares de la ISO/IEC (denotados como MPEG-x) están más enfocados a cubrir las necesidades de almacenamiento de video (DVD), multidifusión de video (Cable, DSL, TV por satélite) y streaming de video (video en Internet o sobre redes inalámbricas). La mayor parte de los estándares y recomendaciones han sido desarrollados por estos dos comités de forma independiente, exceptuando los estándares H.262/MPEG-2 y H264 que fueron desarrollados conjuntamente.

Algunas de las ventajas que presenta el estándar H.264 frente a los estándares anteriores son:

- *Ahorro de tasa de bits superior al 50%.* Comparado con el H.263+ o con el MPEG-4, el estándar H.264 permite reducir la tasa de bits más de un 50%.

- *Vídeo de alta calidad.* Ofrece video de buena calidad tanto con tasas de bits altas como bajas.
- *Tolerancia a fallos.* El H.264 ofrece herramientas para trabajar correctamente cuando se pierden paquetes en la red o se producen errores a nivel de bits en redes inalámbricas.
- *Entorno de red amigable.* Debido a la capa NAL, se pueden transportar flujos de bits de forma sencilla.

4 Versión paralela del H.264

En este apartado se hace una descripción de una implementación paralela desarrollada por el GAP. El objetivo de esta versión fue conseguir un codificador de video con mayor rendimiento que el propuesto por el JVT, mediante el reparto de la carga de trabajo entre varios procesadores. Estos últimos cooperarán en la codificación consiguiendo así una significativa reducción del tiempo de ejecución que permitirá la codificación de video en tiempo real. Eligieron este estándar porque hasta el momento es la última versión propuesta por el JVT; además proporciona video codificado de alta calidad y el costo computacional de este codificador es bastante alto.

Como ya se sabe, los flujos de video están formados por una secuencia de cuadros o *frames* linealmente distribuidos en el tiempo. La cantidad de trabajo que conlleva la codificación de cada cuadro es variable y en la mayor parte de los casos impredecible.

Dada la naturaleza del procesamiento paralelo, y más concretamente el procesamiento paralelo mediante paso de mensajes, los mayores beneficios se obtienen balanceando lo máximo posible la carga de cómputo entre los distintos procesadores. Intentando conseguir un reparto lo más equitativo posible. Y, al mismo tiempo, intentar reducir las comunicaciones necesarias entre ellos. Esta característica, aunque deseable, no es siempre posible. Como límite adicional al aumento de las prestaciones, la parte paralelizable de un algoritmo casi nunca supone el 100% de éste, lo que limita el *speed-up*, o reducción del tiempo de ejecución posible. Adicionalmente, las comunicaciones entre procesadores conllevan siempre una sobrecarga muchas veces considerable.

4.1 Segmentación de la carga basada en GOPs

El reparto de la carga es la distribución de la secuencia de *frames* entre los distintos procesadores. Para que cada procesador pueda llevar a cabo su tarea de la forma más independiente posible, no debería existir interdependencia entre la codificación de una imagen y otra, o bien, debería de ser mínima. Evidentemente, esta premisa no se cumple, ya que el estándar de codificación de video trata de aprovechar la redundancia temporal existente en la secuencia. Con ello consigue mayor eficiencia en la codificación. Por esta razón, en una primera implementación paralela, definieron como la unidad mínima de carga a repartir entre los procesadores la secuencia de imágenes que constituyen un GOP (*Group of pictures*) en lugar de un único *frame*. Cada uno de los GOPS a codificar se puede tratar de forma independiente, es decir, no

existe ninguna interdependencia entre un GOP y otro a la hora de su codificación. Esto se debe principalmente a que cada GOP comienza con un cuadro de tipo I. Este tipo de *frame* se codifica de forma independiente. El resto de cuadros se codificará con información recogida a partir del cuadro I o bien de alguno derivado de éste. Por tanto el GOP es una unidad independiente, en cuanto a la codificación concierne.

Ya que en principio el costo computacional de la codificación de los cuadros que componen un GOP es variable e impredecible, no se puede hacer ninguna distinción entre un GOP y otro. Lo que dificulta la posibilidad de hacer un reparto equitativo del trabajo. Es por esto que simplificaron en este algoritmo, asumiendo que la tarea tiene un costo idéntico de cómputo para cada GOP. Bajo esta premisa utilizaron el siguiente esquema de reparto: la secuencia de video se divide en GOPs, cada uno de los cuales contendrá una secuencia de cuadros. A priori esta secuencia puede ser arbitraria con tal de que se cumpla la premisa de que el primer cuadro a codificar sea siempre un cuadro de tipo I. Por tanto la secuencia de video estará formada por un número determinado de GOPs.

Los procesadores serán etiquetados con un índice que se moverá en el conjunto de los enteros entre el 0 hasta el número de procesadores disponibles menos 1. Se divide aritméticamente el número de GOPs entre el número de procesadores, y se asigna a cada procesador el número de GOPs consecutivos resultante de manera secuencial. El número de GOPs restante (n) de la división entera entre los procesadores se asignarán a los n primeros procesadores, a razón de un GOP por procesador.

Esta división de la carga es de grano grueso, es decir, la unidad mínima de carga que corresponde a cada procesador es considerablemente grande. Este tipo de paralelización tiene ciertos inconvenientes y también algunas ventajas. La principal ventaja de este tipo de reparto es que minimiza la frecuencia con la que hay que realizar comunicaciones entre los distintos procesos. Y por tanto la sobrecarga que introducen las comunicaciones es menor.

En este caso, cada procesador realiza todo su trabajo asignado sin intercambiar información. Únicamente en una última fase de recomposición de las diferentes partes del video codificándose intercambiarán información. El gran inconveniente es que, si los datos de entrada se generan de forma secuencial en tiempo de codificación, es posible que los procesadores tengan que esperar un tiempo considerable a la recepción de los mismos. Esta técnica puede servir para acelerar la codificación de secuencias de video previamente almacenadas. Sin embargo, carece de utilidad para secuencias de video generadas en tiempo real, como en una videoconferencia.

5 Resultados obtenidos

La versión paralela anteriormente citada ha sido sometida a pruebas de evaluación a fin es establecer un punto de referencia para una futura implementación mejorada de dicho codec.

El *banco de pruebas* consiste en un cluster Aldebaran.upv.es, el SGI Altix 3000 de la UPV (Universidad Politécnica de Valencia). Este es un sistema de 48 procesadores Itanium II con memoria distribuida NUMA, con conexión directa a la SAN (red de almacenamiento) del CPD (centro de proceso de datos de la UPV) y sistema operativo

Linux RedHat. La *carga de entrada* consiste en 6 secuencias de video de dominio público en formato YUV 4.2.0. Sus características se enuncian en la tabla 1.

Tabla 1. Atributos de las secuencias de video empleadas para la evaluación

Secuencia (Sec)	Formato	Frames	Tamaño	Fondo	Fig. Central	Objs. Mov
1. foreman	qcif (176x144)	300	11,404,800	1	1	1
2. students	qcif (176x144)	1007	38,282,112	0	1	2
3. foreman	cif (352x288)	300	45,619,200	1	1	1
4. students	cif (352x288)	1007	153,128,448	0	1	2
5. martin	720x480	240	124,416,000	1	1	1
6. ayersroc	720x480	240	124,416,000	1	2	3

El *patrón de codificación* utilizado para todas las pruebas secuenciales se muestra en la tabla 2; por otro lado, el patrón de codificación utilizada para las pruebas paralelas se proporciona en la tabla 3.

Tabla 2. Patrón de codificación secuencial: P0 – Codifica 31 Frames

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I	B	B	P	B	B	P	B	B	P	B	B	P	B	B	I
0			**1**			**2**			**3**			**4**			**5**
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	
B	B	P	B	B	P	B	B	P	B	B	P	B	B	I	
		6			**7**			**8**			**9**			**10**	

El patrón de codificación secuencial es exactamente la secuencia de compresión deseada; sin embargo, en la versión implementada por GAP se tiene esa limitante, ya que no es posible obtener el mismo patrón que el codec secuencial. El último Frame del GOP 0 deberían ser tipo I (Frame 15) y ser codificado como primer Frame del GOP 1. El resto de los Frames del GOP 1 debería seguir la misma secuencia que el GOP 0 de tal forma que se tenga 15 frames por cada GOP. Lo anterior se observa en las tablas 3 y 4.

Cabe mencionar que la obtención de dicho patrón es importante por dos razones: para tener resultados equivalentes (comparables) se requiere el mismo patrón de codificación tanto en el codec secuencial como en el paralelo y además este patrón es el idóneo para obtener la compresión de video manteniendo un equilibrio entre *tiempo y calidad.*

Tabla 3. Patrón de codificación paralelo

P0 – Codifica GOP 0 (16 Frames)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I	B	B	P	B	B	P	B	B	P	B	B	P	B	B	P
0			**1**			**2**			**3**			**4**			**5**

P1 – Codifica GOP 1 (16 Frames)

16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
I	B	B	P	B	B	P	B	B	P	B	B	P	B	B	P
6			**7**			**8**			**9**			**10**			**11**

Tabla 4. Patrón de codificación paralelo deseado

P0 – Codifica GOP 0 (15 Frames)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
I	B	B	P	B	B	P	B	B	P	B	B	P	B	B
0			**1**			**2**			**3**			**4**		

P1 – Codifica GOP 1 (15 Frames)

15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
I	B	B	P	B	B	P	B	B	P	B	B	P	B	B
6			**7**			**8**			**9**			**10**		

Para evaluar el aprovechamiento del algoritmo se ha obtenido el tiempo de ejecución secuencial (T_s) y paralelo (T_p). Los tiempos tomados comprenden: *carga inicial del proceso, lectura de disco, codificación, escritura de disco y las comunicaciones.*

Tabla 5. Tiempos de codificación secuencial y paralelo en segundos

Sec.	Ts	Tp-2P	Tp-4P	TP-6P	Tp-8P	Tp-10P	Tp-12P	Tp-14P
1	306.12	147.45	84.37	63.25	42.27	42.35	42.03	21.08
2	306.94	148.05	83.94	63.01	41.95	41.99	41.80	21.05
3	1258.00	598.60	341.70	257.33	171.59	171.39	172.03	85.78
4	1218.12	591.63	336.17	254.74	168.84	169.43	168.22	85.06
5	4216.10	2060.01	1196.77	895.68	605.17	600.16	595.81	300.15
6	4268.58	2087.13	1196.08	896.81	99.18	599.30	592.46	297.24

La tabla 5 muestra los resultados experimentales obtenidos bajo las condiciones anteriormente descritas. Cabe señalar que se codificaron 226 Frames para el codec secuencial y 14 Gops para el codec paralelo, donde cada GOP contiene 16 Frames, por lo que se codificaron 224 Frames (No. de frames = Fr). La diferencia del número

de Frames tiene que ver con la limitante en el patrón de codificación antes citada del algoritmo paralelo. Dicha versión tiene otra limitante, se tuvieron que establecer las pruebas con 14 GOPs, debido a que no se hace una codificación cíclica con lo cual el número GOPs depende del *tamaño* de la secuencia. En cuanto a los tiempos obtenidos y mostrados en la tabla 5 cabe subrayar que conforme se tiene mayor número de procesadores es menor el tiempo de codificación; sin embargo, esto no siempre se cumple entre 8 , 10, y 12 procesadores. Esto se debe a la forma en que se realizó la distribución de la carga para esta versión paralela.

En las figuras 2 y 3, se muestra la distribución de GOPs para 8 y 10 procesadores respectivamente, las cuales ilustran el *balance de la carga*. Se puede observar que con 10 procesadores se tienen más *tiempos muertos* que con 8. Si se hiciera el reparto de GOPs para 12 se harían evidentes aun más *tiempos ociosos* de los procesadores. Debido a este esquema de paralelización, se puede deducir que el tiempo de codificación no siempre se decrementa a pesar de ir incrementando el número de procesadores, ya que la distribución de GOPs esta predeterminada desde el inicio, con lo cual no se aprovechan los procesadores que vayan siendo liberados; retrasando así el *tiempo de codificación final*. Por esa misma razón el tiempo de codificación para 14 procesadores decrementa considerablemente, debido a que se tiene el mismo número GOPs con lo cual no se tiene ningún *tiempo muerto*. Cabe señalar que el tiempo de codificación que se toma es el del último procesador que termina el trabajo.

GOPS	1	3	5	7	9	11		
	0	2	4	6	8	10	12	13
P	1	2	3	4	5	6	7	8

Fig. 2: Distribución de 14 GOPs en 8 procesadores

GOPS	1	3	5	7						
	0	2	4	6	8	9	10	11	12	13
P	1	2	3	4	5	6	7	8	9	10

Fig. 3: Distribución de 14 GOPs en 10 procesadores

Tabla 6. FrameRate con 14 Gops calculado con la siguiente fórmula: Fr = Nf / Tp

Secuencia	2P	4P	4P	8P	10P	12P	14P
foreman_qcif	1.52	2.66	3.54	5.30	5.29	5.33	10.63
students_qcif	1.51	2.67	3.55	5.34	5.34	5.36	10.64
foreman_cif	0.37	0.66	0.87	1.31	1.31	1.30	2.61
students_cif	0.38	0.67	0.88	1.33	1.32	1.33	2.63
martin_720x480	0.11	0.19	0.25	0.37	0.37	0.38	0.75
ayersroc_720x480	0.11	0.19	0.25	0.37	0.37	0.38	0.75

Como se puede observar en la tabla 6, cada vez que se incrementa el número de procesadores el FramaRate incrementa más o menos en la misma proporción, no así

cuando se tienen 8, 10 y 12 procesadores. Esto es por lo mismo que se mencionaba con respecto a los tiempos.

El *FrameRate natural* o *tiempo real* de una secuencia de video es de 30 fps. Se puede observar en la figura 4 que con 14 procesadores se esta muy lejos de poder alcanzar ese número de frames codificados por segundo y que conforme se tiene mayor resolución más lejos se está de tener tiempo real. Dicho tiempo para codificar 14 GOPs es de 7 segundos, lo cual quiere decir que para comprimir 14 GOPs de la secuencia *foreman* en su formato *cif* se necesitan 37.5 procesadores y para *ayersroc_720x480* se requieren 130.6 procesadores.

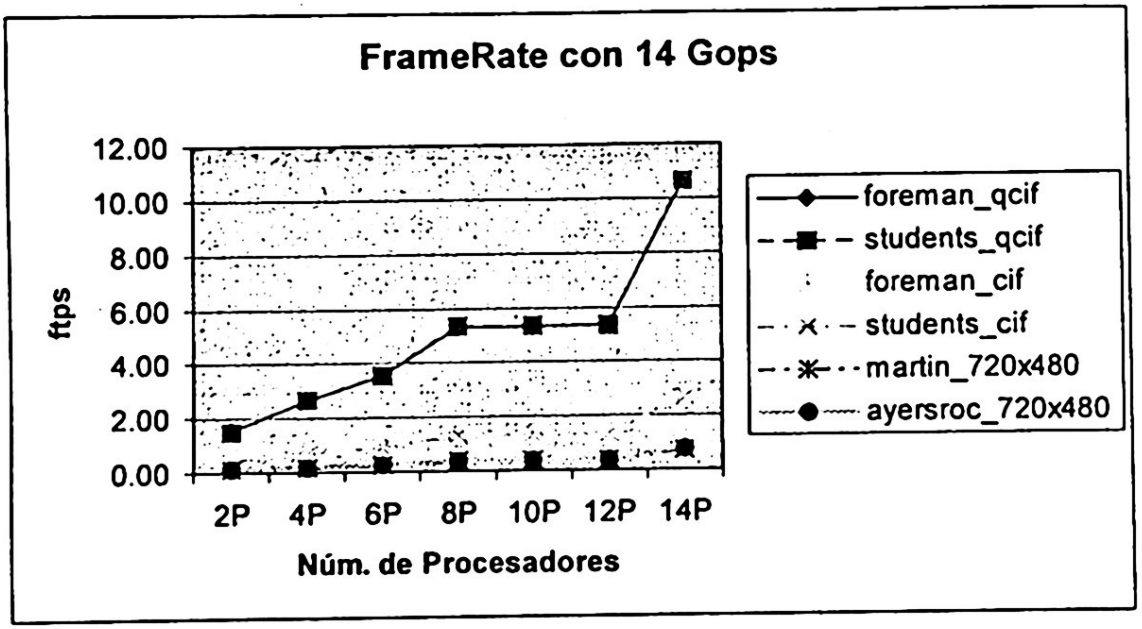


Fig. 4: FrameRate con 14 Gops

En la tabla 7 se muestra el SpeedUp (*aceleración*) de cada una de las secuencias para los números de procesadores empleados. Al igual que en los tiempos de codificación mostrados en la tabla 5 y su FrameRate expuesto en la tabla 6, con los procesadores 8, 10 y 12, el SpeedUp en ocasiones, en vez de aumentar, disminuye. Esto se debe a la misma causa, los tiempos muertos de los procesadores es mayor para 12 que para 10 y de éste que para 8.

Tabla 7. SpeedUp con 14 Gops calculado con la siguiente fórmula: $Sp = Ts / Tp$

Secuencia	Ts	2P	4P	6P	8P	10P	12P	14P
foreman_qcif	306.12	2.08	3.63	4.84	7.24	7.23	7.28	14.52
students_qcif	306.94	2.07	3.66	4.87	7.32	7.31	7.34	14.58
foreman_cif	1258.00	2.10	3.68	4.89	7.33	7.34	7.31	14.66
students_cif	1218.12	2.06	3.62	4.78	7.21	7.19	7.24	14.32
martin_720x480	4216.10	2.05	3.52	4.71	6.97	7.02	7.08	14.05
ayersroc_720x480	4268.58	2.05	3.57	4.76	7.12	7.12	7.20	14.36

codificar, no existen tiempos ociosos para ninguno de lo procesadores en ningún momento.

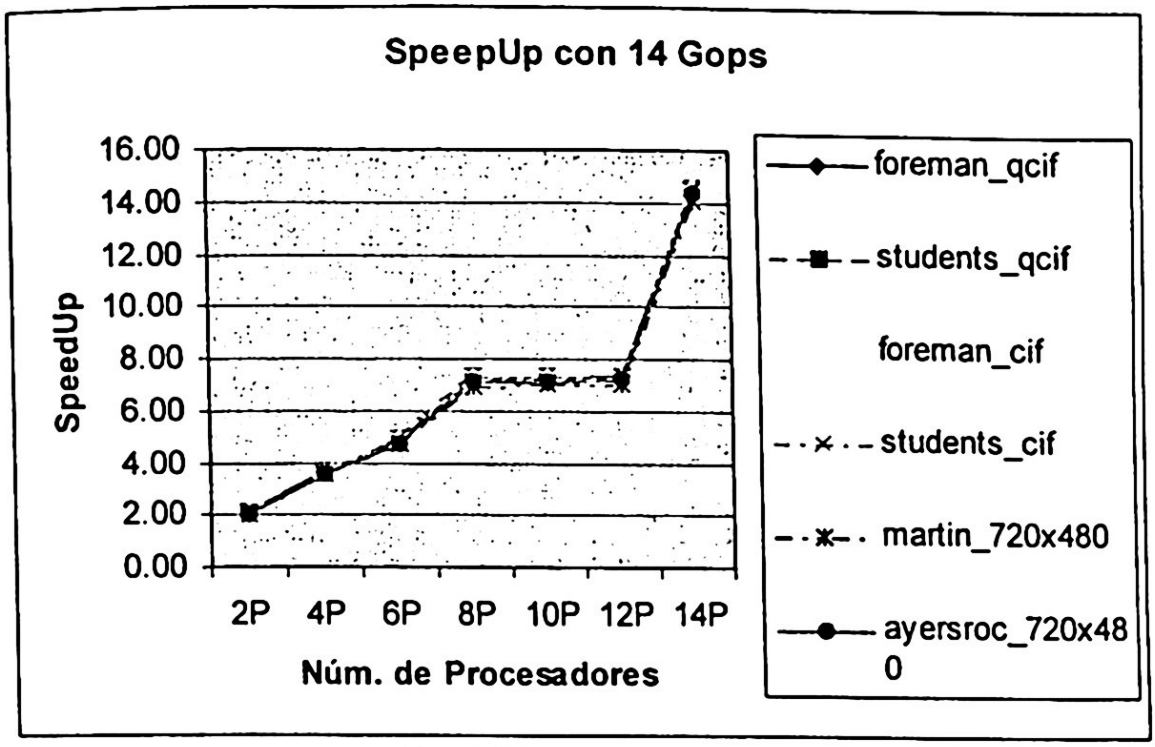


Fig. 5. SpeedUp con 14 Gops

6 Trabajo futuro

Como ya se había mencionado anteriormente, las pruebas mostradas en este capítulo establecen la base de una investigación con la finalidad de realizar una versión paralela que permita mejorar lo hecho. El objetivo es *implementar y evaluar un algoritmo paralelo con balanceo de carga* para la compresión de secuencias de video utilizando el codificador H.264, buscando reducir los tiempos de codificación bajo el *patrón idóneo* ya antes referido. Además se busca poder establecer una *codificación cíclica* a fin de que el número de frames no sea una limitante para realizar pruebas de codificación con mayor número de GOPs. También se espera determinar el tipo de video más adecuado para el codec a implementar.

El balanceo de la carga será por demanda, similar al planteado en el codec A3-frontera [4]. Su esquema de distribución consiste en que el proceso 0 (maestro) es el encargado de distribuir los identificadores de Gops a los procesadores conforme van desocupándose. El proceso maestro sabe que se ha desocupado un procesador porque cada uno de los procesos codificadores le informa que ha terminado su tarea.

Referencias

1. Cuesta, S. B., Gonzáles, I. F.: Paralelización del estándar de codificación de video H264.

2. Effelsberg, W., Steinmetz, R.: Video Compression Techniques (heidelberg: dpunkt.verlag). 1998
3. Hernández, S. J.: Análisis Comparativo de los Modelos de Programación Distribuida, Utilizando Cluster de Estaciones de Trabajo. Tesis de Maestría, Universidad Mexicana Plantel Veracruz. Agosto del 2001
4. Rodríguez, L. A.: Diseño e implementación de algoritmos paralelos para la compresión de secuencias de video MPEG4. Reporte Técnico. Universidad Politécnica de Valencia. Noviembre del 2002.

Audiómetro controlado por computadora, implementación mediante programación visual con observancia de normas internacionales

Ma. Julia Calderón Sambarino[1], Sergio Suárez Guerra[1,2]

[1] Centro de Investigación en Computación,
Instituto Politécnico Nacional,
México, D. F.
[2] ICIMAF, Cuba.
msambarino@sagitario.cic.ipn.mx, ssuarez@cic.ipn.mx

Resumen. El examen audiométrico es una evaluación que nos permite conocer la capacidad de un individuo para escuchar diferentes sonidos y así determinar su umbral auditivo. Un audiómetro es el equipo que permite llevar a cabo un examen audiométrico. Se presentan los resultados del proyecto que incluye el desarrollo de un audiómetro de conducción aérea controlado por computadora que realiza un examen audiométrico mediante una prueba de tonos puros (Audiometría Tonal Liminar) generando resultados gráficos (audiogramas) sobre la base de normas internacionales para audiómetros. El audiómetro permite la generación de pruebas audiométricas adicionales de acuerdo a las necesidades del especialista para facilitar el diagnóstico, así como también el almacenamiento de las audiometrías realizadas junto al historial sustentado en la Norma Oficial Mexicana del Expediente clínico y garantizando la confidencialidad del mismo entre los especialistas usuarios del sistema. La implementación del audiómetro comprende la calibración relativa de los dispositivos requeridos para las evaluaciones audiométricas.

1 Antecedentes

La audición constituye la vía habitual para adquirir el lenguaje, el cual nos permite la comunicación a distancia y a través del tiempo, además es el responsable del desarrollo de la sociedad y de la conservación de la cultura.

La sordera es un grave impedimento cuyos efectos trascienden la posibilidad de comunicación a través del habla, repercutiendo en la armonía familiar, el desempeño académico, el retraso en el lenguaje y las expectativas laborales y profesionales de las personas afectadas.

Desde la hipoacusia leve hasta la sordera profunda, son entidades tratables, y los resultados del tratamiento en término de incorporación de los niños afectados, dependen de lo precoz y fiable que resulte el diagnóstico. Se considera una audición normal cuando los límites del sonido son captados entre los 0 y 20 decibeles.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp. 167-175.© Centro de Investigación en Computación, IPN, México

1.1 Introducción

La Audiometría es un conjunto de pruebas para medir la audición. Dentro de este conjunto de pruebas, vamos a referirnos a la más frecuente cuya denominación médica es Audiometría Tonal Liminar, porque en ella se emplean tonos puros y se busca el umbral de audición o mínima intensidad de sonido capaz de ser percibida. La audiometría se realiza mediante la utilización de un aparato electrónico denominado audiómetro. Dicho aparato se compone de un generador de estímulos que son transmitidos al individuo a través de unos auriculares aplicados estrechamente al oído.

Existen varios tipos de audiometrías, pero la mas usada en el Sector Salud es la audiometría tonal o de tonos puros, la cual consiste en la estimulación auditiva por medio de la vía aérea y de la vía ósea. La transmisión sonora por la vía aérea se realiza a través del aire colocando unos auriculares en el pabellón de la oreja, y la estimulación sonora por la vía ósea se realiza colocando un vibrador en la apófisis mastoides. Las frecuencias exploradas por medio de esta técnica son las de 250, 500, 1000, 2000, 4000 y 8000 Hz y la intensidad del estímulo varía desde 10 a 110 Dbs[CIFT97]. Las frecuencias medias, es decir, las comprendidas entre 500 y 2000 Hz, son las más importantes para la audición de las palabras ya que en dicho rango se sitúan la mayoría de los fonemas que componen la voz humana.

Para la realización de una audiometría tonal pura es necesario seguir las guías propuestas por los estándares internacionales sobre las frecuencias de inicio, así como la intensidad incremental. El ruido debe transmitirse a través del conducto auditivo, a través del oído y del oído interno, más conocido como caracol, hasta llegar al nervio auditivo y al cerebro. En el caso de la estimulación por vía ósea se prescinde tanto del oído externo como del medio, produciéndose una estimulación directa del oído interno a través de los huesos del cráneo.

La audiometría se representa en una gráfica llamada audiograma la cual se define por un eje de ordenadas dividido en intervalos de 10 dBs. y un eje de abscisas donde se sitúan los diferentes tonos que utilizamos para estimular, desde los más graves (125 y 500 Hz) hasta los más agudos (4000 y 8000 Hz).

La simbología a emplear en el audiograma se define en los estándares ISO 389-2 y en el IEC 60645-1. El cero audiométrico (0 dBs.) del audiograma viene determinado por una normativa internacional y se corresponde con la intensidad media mínima en cada una de las frecuencias a la cuál poblaciones de personas sin antecedentes de daño audiológico comienzan a percibir un estímulo tonal determinado.

2 Propuesta y consideraciones

El sistema consiste en un equipo formado por una computadora, con entradas y salidas estándar: audífonos para la emisión de sonidos, impresora para la graficación de resultados, y un sistema de software diseñado para llevar a cabo estas emisiones.
La aplicación desarrollada realiza una Evaluación Audiométrica Tonal Liminar por conducción aérea. Esta prueba se considera la evaluación básica necesaria en los estudios audiométricos y se incluye en los sistemas de diagnóstico audiométrico comerciales.

Una consideración indispensable es la integración de bases de datos para el almacenamiento del perfil del paciente y su expediente clínico, de los resultados obtenidos en cada una de las pruebas audiométricas que realice, la configuración de las pruebas audiométricas generadas así como los expedientes laborales de los usuarios del sistema (especialistas).

La siguiente consideración se refiere a los costos. Los equipos biomédicos convencionales conllevan altos costos no solo de adquisición, sino de mantenimiento, accesorios y garantías, mismos que se incrementan por ser tecnologías de importación, por lo que el Audiómetro desarrollado tiene como premisa ser una solución de bajo costo proporcionando resultados confiables que permitan un diagnóstico certero.

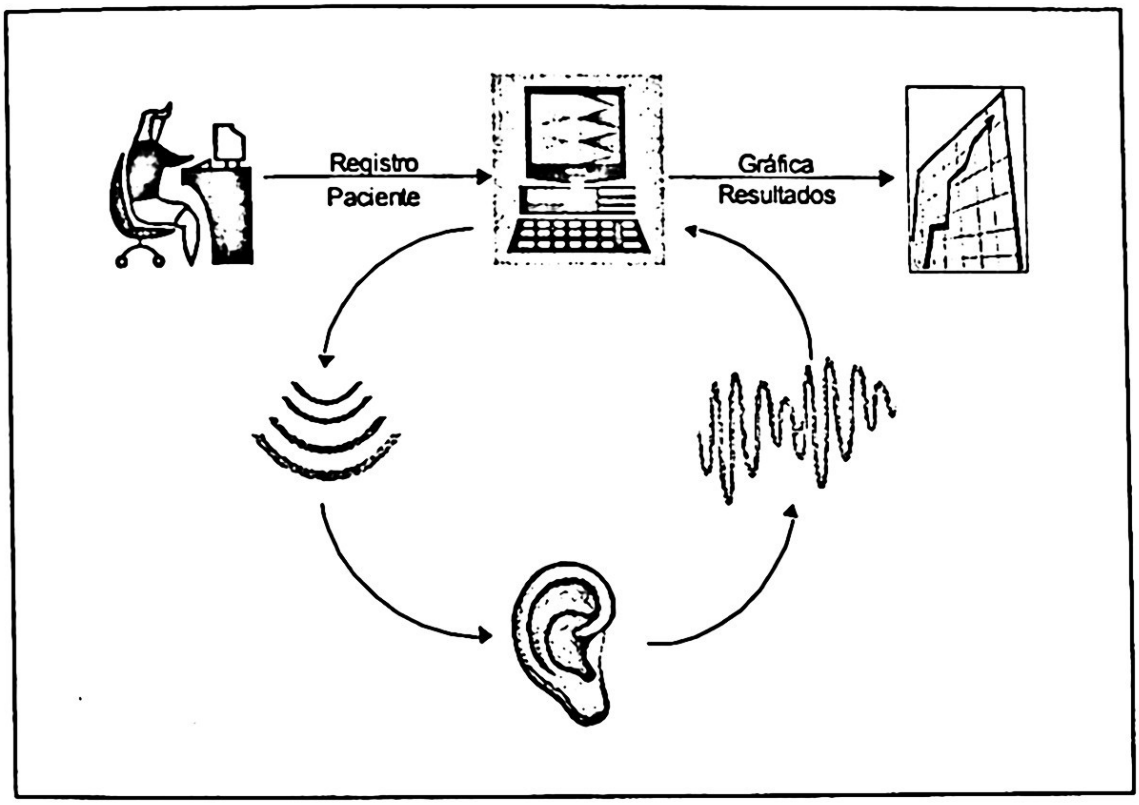


Fig. 1. Esquema general del sistema desarrollado

La interfaz intuitiva se considera un beneficio adicional del sistema, ya que resulta innegable el aspecto interdisciplinario que se refleja en el uso del producto final por parte de especialistas médicos (audiólogos) que requieren sencillez en el uso de las herramientas computacionales.

3 Metodología de desarrollo

Para el análisis y desarrollo del Sistema de Evaluación Audiométrico se consideró el marco teórico integrado por cinco aspectos fundamentales:

3.1 Ingeniería de software: Paradigma orientado a objetos

Como parte del desarrollo interno del sistema y materia fundamental de la ingeniería de software, debe existir una metodología de desarrollo, en este caso, empleamos el Paradigma Orientado a Objetos donde el ciclo de vida o desarrollo del proceso orientado a objetos se mueve a través de una espiral evolutiva que comienza con la comunicación del usuario, la definición del dominio del problema y la identificación de las clases básicas del problema [BOHB85].

3.2 Persistencia de la información: Uso de bases de datos

Las aplicaciones de bases de datos permiten a los usuarios interactuar con información almacenada. Las bases de datos proveen la estructura para esta información y permite compartirlas entre varias aplicaciones.

A partir de esto se trabajó en un diseño de bases de datos que involucra a tres actores principales: usuarios (doctores que llevan a cabo una evaluación audiométrica), pacientes (sujetos con expediente clínico y posibles candidatos para evaluaciones auditivas) y la evaluación audiométrica en sí misma, ya que una de las principales ventajas del sistema propuesto, es que cada uno de los usuarios puede configurar pruebas audiométricas particulares a cada paciente o caso de estudio, y no conformarse con la prueba estándar indicada por las Normas Internacionales e incluida como prueba base o única, en los equipos comerciales de evaluación audiométrica.

3.3 Pautas de desarrollo: Observancia de las normas

Los estándares son documentos o acuerdos que contienen especificaciones técnicas o criterios precisos para usarse como reglas, pautas o definiciones para asegurar que los materiales, productos, procesos y servicios son los adecuados a nuestros propósitos [ISO02]. La Organización Internacional para la Estandarización ISO es un organismo internacional con aceptación en 140 países entre ellos México, que tiene como propósito promover el desarrollo de estándares en el ámbito científico, tecnológico y económico.

Los estándares nos brindan una garantía de funcionalidad en los productos o servicios sobre los cuales se apliquen. Los desarrollos médicos y tecnológicos que se guíen por los estándares indicados por la ISO podrán participar en un ámbito internacional y permiten la interacción con desarrollos previos existentes. Nuestro trabajo se ajusta a estándares internacionales, tanto de la ISO como de la IEC y la ANSI, así como en Normas Oficiales Mexicanas, comentadas en la siguiente tabla:

Tabla 1. Descripción de estándares considerados en la investigación

Organismo	Norma/ Estándar	Descripción
ISO	389-1	Referencia cero para calibración de equipo audiométrico: Niveles de referencia equivalentes de presión acústica

		para tonos puros para audífonos supra-aurales
ISO	389-2	Referencia cero para calibración de equipo audiométrico: Niveles de referencia equivalentes de presión acústica para tonos puros para audífonos insertados
ISO	389-3	Referencia cero para calibración de equipo audiométrico: Niveles de referencia equivalentes de nivel de fuerza para tonos puros y para vibradores óseos.
ISO	389-4	Referencia cero para calibración de equipo audiométrico: Niveles de referencia para ruido enmascarado en banda ancha.
IEC	60645-1	Equipo audiológico: Audiómetros de tonos puros
IEC	60645-4	Audiómetros: Equipos para la audiometría extendida al dominio de la alta frecuencia
ANSI	S3.6-1996	Especificaciones para audiómetros
DGN	NOM-168-SSA1-1998	Del expediente clínico

3.4 Paradigma de detección simple

La situación en los paradigmas de detección es simple: una señal es presentada al sujeto y éste debe decir si la detecta o no. Este es el esquema básico que siguen los distintos procedimientos audiométricos para la determinación del umbral de audición del sujeto.

Desde una perspectiva metodológica es claro que las respuestas dadas por el sujeto no sólo dependen de su sensibilidad, sino también de un conjunto amplio de variables no sensoriales tales como la motivación, los intereses o lo que los sujetos esperan ganar o perder en función de su respuesta.

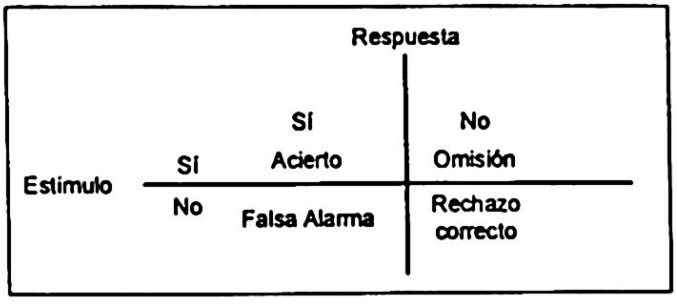


Fig. 2. Resultados posibles tras realizar un experimento de detección de señales siguiendo el procedimiento de detección simple: si / no

En el procedimiento sí/no el sujeto se enfrenta a dos situaciones posibles: la captación (escucha) de la señal o la no-presentación de la señal (silencio). Durante el experimento el sujeto ha de decidir, en cada ensayo, cuál de las dos situaciones le fue presentada. Sin embargo, la intensidad de la señal es un factor modificable, por lo tanto, en caso de una respuesta no-presentación, se debe incrementar la intensidad de la señal hasta lograr un resultado de captación de la señal.

En ambos casos, se debe registrar la intensidad con la cual la señal fue captada o no. Por lo tanto, el procedimiento sí/no genera cuatro clases de respuestas diferentes que se recogen en la siguiente figura.

3.5 Calibración relativa: Manteniendo la solución sencilla

Las calibraciones requieren de dispositivos especializados para ello: oidos artificiales con micrófonos calibrados, ambos con un alto costo, por lo que para la puesta a punto de los auriculares se optó por una calibración relativa, eficiente pero con un costo considerablemente menor.

La calibración relativa consiste en calibrar los audífonos a partir de la mínima audición captada por el usuario para cada una de las frecuencias. Se selecciona un usuario de audición normal y se le presenta cada una de las frecuencias, primero en el oído izquierdo, después el oído derecho; el usuario ajusta el volumen a su mínima intensidad audible y los resultados son almacenados en un archivo de texto, relacionados en una base de datos.

Esta calibración relativa permite que el sistema funcione con distintos dispositivos emisores, es decir, los auriculares pueden ser sustituidos y únicamente requerirá repetir el proceso de calibración, siempre y cuando la persona que lo lleve a cabo cuente con una tabla de calibración previa almacenada.

Si el sistema es instalado en otro equipo de cómputo, aún usando los mismos auriculares, será necesario almacenar una nueva calibración, ya que la tarjeta de sonido encargada de la emisión de la señal varía de una computadora a otra, aún cuando las características, como procesador y marca sean similares.

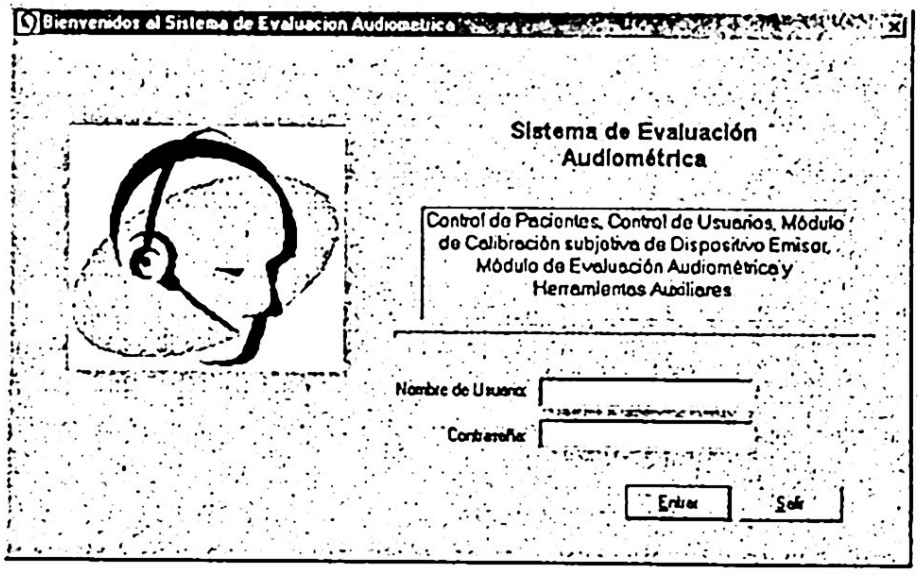


Fig. 3. Pantalla de validación de usuario y acceso al sistema

Cuando se realiza una evaluación audiométrica, se relaciona en la base de datos con la calibración perteneciente a los dispositivos que se emplean para dicha evaluación, lo cual permite a futuro conocer y relacionar el nivel auditivo con la calibración almacenada, aún cuando los dispositivos hayan cambiado.

4 Sistema de evaluación audiométrica: Nuestra propuesta

La primera pantalla corresponde a la validación que deben realizar los usuarios del sistema para su reconocimiento y acceso. Esto garantiza que los expedientes de los pacientes sólo serán conocidos y manipulados por el especialista que los generó.

Una vez que se ha validado al usuario, se presenta la pantalla principal donde podemos dar de alta nuevos usuarios y pacientes, así como configurar evaluaciones audiométricas, realizar calibraciones, o llevar a cabo una evaluación audiométrica con un paciente ya registrado.

El siguiente paso es el registro del paciente, si el sujeto ya se encuentra dado de alta, es posible acceder directamente a su expediente para revisar su historial de evaluaciones y modificar datos relativos al diagnóstico o patología, o bien proceder a una evaluación audiométrica. Un sujeto o paciente puede tener tantas evaluaciones como el usuario considere necesario. Por ello el módulo de configuración de evaluaciones resulta no sólo necesario, sino una característica particular de la aplicación propuesta. En el módulo de configuración de la evaluación, el usuario puede crear nuevas evaluaciones audiométricas decidiendo entre las 12 frecuencias manipuladas por el sistema y ajustándolo al perfil de un sujeto en particular.

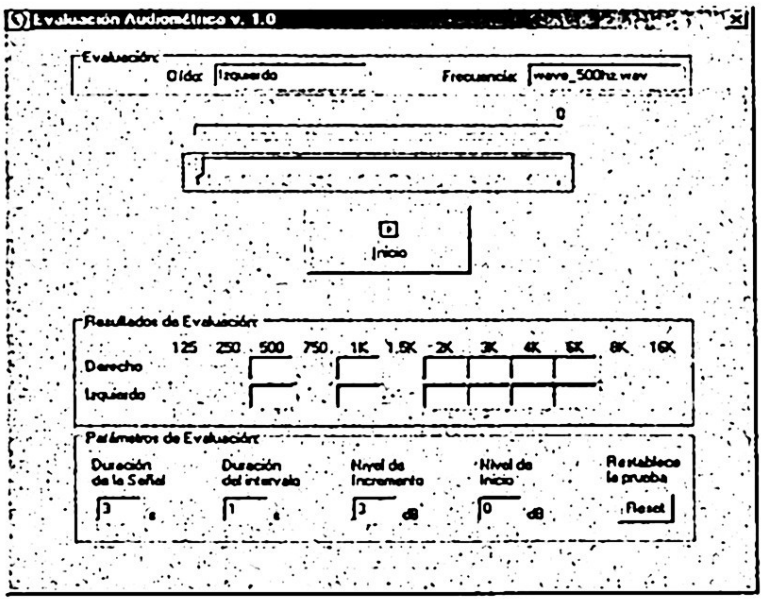


Fig. 4. Generación de una evaluación audiométrica estándar

Los resultados pueden observarse en el audiograma, que presenta además de las gráficas estándar, la opción de graficar la calibración relativa asociada y sus rangos de aceptación normal.

Para el usuario del sistema, es posible dar de alta a pacientes o bien, a nuevos usuarios, y atribuirle a cada uno de ellos, un estado: Activo, Suspendido, Baja. Esto surge por las características observadas en las clínicas y centros de salud, donde los doctores pueden comisionarse a otros centros o concluir su período de servicio, mientras que los usuarios permanecen. De aquí surge otra necesidad, la transferencia de pacientes de un doctor a otro para continuar el seguimiento o distribuir la carga de

trabajo. La transferencia de pacientes se ofrece como parte de las herramientas útiles al especialista.

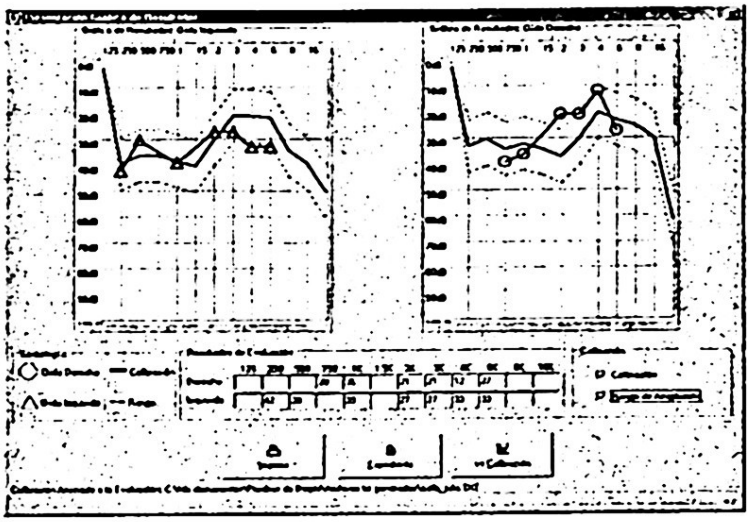


Fig. 5. Audiograma estándar con líneas guía de calibración y rango de audición normal

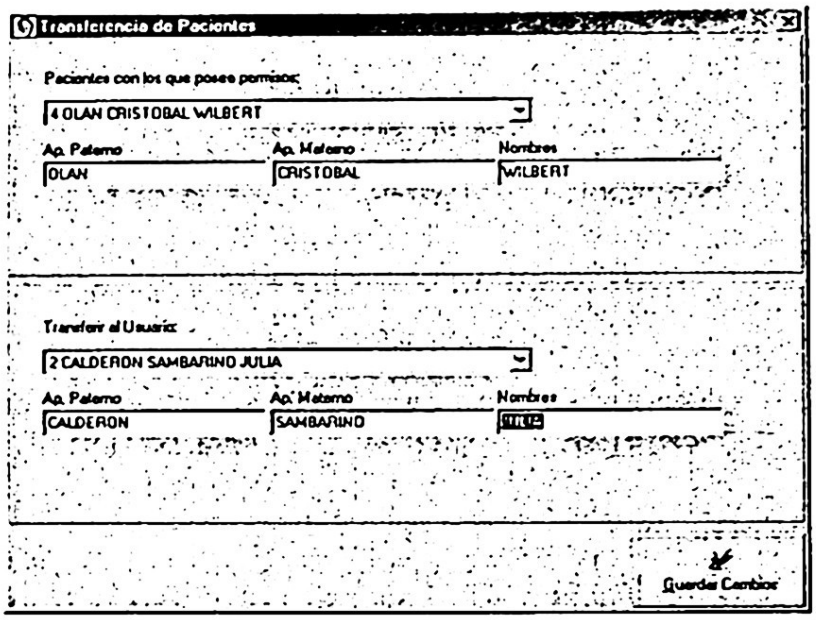


Fig. 6. Pantalla de transferencia de pacientes entre usuarios del sistema

5 Conclusiones

El trabajo realizado, así como los similares que se han desarrollado en México, manifiestan la creciente necesidad de dotar de nuevas posibilidades el mercado actual de equipos biomédicos, particularmente Audiométricos en su mayoría con tecnología importada, de soluciones confiables y de bajo costo.

El tema de diagnóstico audiométrico es amplio y este trabajo alcanza a satisfacer únicamente uno de sus requisitos, tal vez el básico, la audiometría tonal liminar por conducción aérea. Se muestra un largo camino a futuro respecto a las evaluaciones audiométricas basadas en computadora: pruebas objetivas, timpanometría, audiometría por conducción ósea, entre otros.

El análisis y desarrollo de dichas pruebas basándonos en normas aceptadas internacionalmente brindaría al país opciones de mejorar la calidad de vida de los sectores marginados y lo colocaría como desarrollador de tecnología biomédica, trabajo asignado comúnmente a los países más desarrollados.

Referencias

1. [BOHB85]. Bohem, B. The Basic Principles of Software Engineering, Publishers Inc. 1985.
2. [CIFT97]. Cifuentes T. En Med Mar: La pérdida auditiva. Su exploración a través de las pruebas funcionales liminares. Medicina Marítima. 1997
3. [IEC60645-1]. IEC. Electroacoustics- Audiological equipment Part 1. IEC 2001
4. [IEC60645-4].IEC. Electroacoustics- Audiological equipment Part 4. IEC 1998
5. [IEC02]. IEC. Mission and Objectives. http://www.iec.ch. 2002
6. [ISO389-1]. ISO. Acoustics- Reference zero for the calibration of audiometric equipment. Part 1 ISO 1998
7. [ISO389-3]. ISO. Acoustics- Reference zero for the calibration of autiometric equipment Part 3. ISO 1994
8. [ISO389-4]. ISO. Acoustics- Reference zero for the calibration of autiometric equipment Part 4. ISO 199
9. [ISO02]. ISO. ¿What are standars? Http://www.iso.ch/iso. 2002
10. [OTORE99]. Otorrino, El. Audiometría. http://www.elotorrino.com. 1999

Análisis de desempeño de un filtro para el mejoramiento de imágenes de huellas dactilares

Victor Hugo García Ortega, Osvaldo Espinosa Sosa

Centro de Investigación en Computación, IPN,
Av. Juan de Dios Batíz, esq. M. Otón de Mendizábal, 07738
México, D.F
v.garciaortega@lycos.com

Resumen. Este trabajo presenta los resultados de un algoritmo basado en la Transformada de Fourier para el filtrado y mejoramiento de imágenes de huellas dactilares, dicho algoritmo es parte de un sistema AFAS (Automatic Fingerprint Authentification System), el cual esta basado en la detección de minucias en huellas dactilares para el control de acceso a recintos. Este algoritmo fue probado en imágenes de buena y mala calidad obteniendo resultados muy satisfactorios. El algoritmo esta sustentado en los trabajos conocidos, sin embargo, presenta variantes con respecto a éste, además, fue probado en una computadora personal Pentium III y un Procesador Digital de Señales (DSP) a fin de poder medir y comparar el tiempo de ejecución del algoritmo en este procesador de arquitectura específica para la ejecución de algoritmos como el que aquí se plantea

1 Introducción

Básicamente los sistemas biométricos [12] basados en huellas dactilares son de dos tipos [8]:

- Automatic Fingerprint Authentification System (AFAS).
- Automatic Fingerprint Identification System (AFIS).

En un AFAS la entrada es la identidad de la persona y la imagen de la huella dactilar de esa persona; y la salida es una respuesta de SI ó NO, indicando si la imagen de entrada pertenece a la persona cuya identidad es proporcionada.

En un AFIS la entrada es solo la imagen de la huella dactilar y la salida es una lista de identidades de personas que pueden tener la huella dada, además de una puntuación de cada identidad indicando el grado de similitud entre ésta y la huella dada.

Ambos sistemas utilizan los detalles formados en las huellas dactilares. Estos detalles llamados "*ridges*" son definidos como un segmento de curva simple. La combinación de varios *ridges* forman un patrón de huella dactilar. Las pequeñas características formadas por el cruce y terminación de *ridges* son llamadas *minucias*. La finali-

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.176-183.© Centro de Investigación en Computación, IPN, México

dad del filtro propuesto es el poder eliminar todo el ruido posible de la imagen a fin de poder extraer las minucias de la imagen [5,6, 2].

Cabe mencionar que para la adquisición de la imagen se uso el *5th Sense Parallel Evaluation Kit*, el cuál es el Kit de desarrollo del sensor de huella dactilar FPS110. Dicho sensor es fabricado por la compañía Veridicom. El sensor es del tipo capacitivo, el cuál cuenta con una resolución de 500 dpi y con un tamaño de arreglo de píxeles de 300x300 [4].

Existen muchas técnicas específicas para realizar un mejoramiento de huellas dactilares. Las técnicas descritas en [13] y [11] son de las más efectivas, pero estas técnicas están diseñadas para usarse en los sistemas del tipo AFIS(como en sistemas policíacos del FBI); y por lo tanto requieren demasiado tiempo de procesamiento en comparación con sistemas mas pequeños como son los sistemas del tipo AFAS.

El principio básico de operación en estos métodos es dividir a la imagen en bloques, creando un filtro específico para cada bloque basado en la dirección promedio del bloque (imagen direccional) [5, 6, 9] y entonces filtrar cada bloque. En [13] este proceso se hace en el dominio del tiempo y en [11] se hace en el dominio de la frecuencia.

Por lo tanto, se requiere un sistema el cual cree un filtro adaptado a la dirección exacta del bloque. La información direccional de un bloque específico de la imagen esta contenido en la magnitud de la Tranformada de Fourier del bloque, por lo que sí se considera un bloque el cual contenga dos o tres ridges aproximadamente paralelos y se toma su Transformada de Fourier, entonces se supone que las frecuencias dominantes de ese bloque corresponden a los ridges en el bloque.

2 Algoritmo

Es posible mejorar un bloque específico por sus frecuencias dominantes multiplicando la Transformada de Fourier del bloque por su magnitud un cierto número de veces, entonces cuando se calcula la Transformada Inversa de Fourier se observa que el bloque ha sido mejorado.

Si multiplicamos por el cuadrado o el cubo de la magnitud de la transformada de Fourier se obtendrían mejores resultados, sin embargo las regiones donde se localizan las minucias suelen ser alteradas, por lo que en vez de multiplicar por factores enteros a la magnitud de la transformada de Fourier se debe utilizar una fracción[7].

Para el cálculo de la transformada de Fourier se implementó el algoritmo de la transformada rápida de Fourier(FFT– Fast Fourier Transform) descomposición en tiempo(Decimation in Time) en la PC y en frecuencia(Decimation in Frequency) en el DSP, ambos base 2 (radix 2).

El algoritmo propuesto consiste básicamente en realizar dos filtrados a la imagen (ver figura 1), donde el segundo filtrado solo se hace a las regiones que lo necesiten (las mas ruidosas). En este trabajo se llegó a la conclusión de que este filtro basado en la Transformada de Fourier trabaja mucho mejor sobre la imagen binaria que sobre la imagen en niveles de gris como se describe en [7]. Por esta razón primero se binariza la imagen con un umbral dinámico y después se filtra. Después de realizar el primer filtrado, en algunas ocasiones suelen haber regiones todavía con demasiado ruido, por

lo que un segundo filtrado es necesario a fin de poder eliminar dicho ruido. El umbral compuesto determina las regiones ruidosas que necesitan el segundo filtrado, las cuales se les aplica el filtro nuevamente.

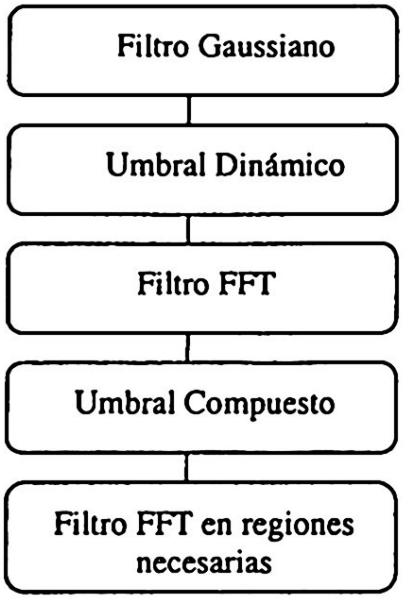


Fig. 1. Algoritm

2.1 Filtro Gaussiano

Este filtro es un filtro pasabajas, cuyo efecto es el de realizar un suavizado en la imagen, en el cuál el núcleo es una Gaussiana 2-D, de media cero y desviación estándar σ, de la forma:

$$G(x,y) = e^{-(x^2+y^2)/2\sigma^2} \tag{1}$$

La transformada de Fourier de la Gaussiana es todavía Gaussiana y prácticamente no posee lóbulos laterales, por lo que su respuesta en frecuencia es muy buena.

2.2 Umbral dinámico

Para poder separar claramente las regiones que pertenecen al fondo de la imagen, de los ridges, se requiere aplicar alguna técnica de segmentación, esto se hace binarizando la imagen a través de un proceso de umbralización [10].

Existen diferentes tipos de umbrales, global, local y dinámico [10], este último consiste en dividir la imagen en bloques B_{KL} de 15x15 píxeles, de los cuales se obtiene su media μ_{KL}. Entonces la media μ_{KL} de un bloque B_{KL}, esta definida por:

$$\mu_{KL} = \left(\sum_{j=LB}^{(L+1)(B-1)} \sum_{i=KA}^{(K+1)(A-1)} I(i,j) \right) / (AB) \tag{2}$$

Donde A, B es el tamaño del bloque y K, L es el número de bloques obtenidos de la imagen. Entonces μ_{KL} es el umbral del bloque B_{KL}, de tal forma que cada bloque tendrá un umbral diferente. Este método funciona muy bien sobre imágenes de mala calidad, a diferencia del umbral global que funciona bien en imágenes de buena calidad [1].

2.3 Filtro FFT

Para mejorar a la imagen a través de la magnitud de su transformada de Fourier, ésta es dividida en bloques de 32x32 píxeles, a los cuales su transformada de Fourier F(u,v) es calculada de acuerdo a:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \exp\left\{ -j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \tag{3}$$

La imagen mejorada g(x,y) en cada bloque es obtenida por:

$$g(x,y) = F^{-1}\left\{ F(u,v) x \left| F(u,v) \right|^{k} \right\} \tag{4}$$

Donde F(u,v) es la transformada de Fourier del bloque y k es la fracción por la cual se multiplica a la magnitud de la transformada de Fourier, y F^{-1} es la transformada Inversa de Fourier obtenida de acuerdo a:

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) \exp\left\{ j2\pi \left(\frac{ux}{M} + \frac{vy}{N} \right) \right\} \tag{5}$$

2.4 Umbral compuesto

Puesto que la umbralización global funciona bien en una imagen de buena calidad, y el umbral dinámico en imágenes de mala calidad, en este trabajo se combinaron ambos métodos para obtener un algoritmo al cual se le denominó *umbralización compuesta*. Este algoritmo se muestra en la figura 2. y se utiliza después de realizar el primer filtrado basado en la FFT, en donde a veces, se tienen todavía regiones de mala calidad. La determinación de estas regiones de mala calidad se hace usando el

método de segmentación por varianza [14], en este método la imagen se divide en bloques B_{KL} de 15x15 píxeles de los cuales se extrae su varianza v_{KL} dada por:

$$v_{KL} = \left(\sum_{j=LB}^{(L+1)(B-1)} \sum_{i=KA}^{(K+1)(A-1)} \left[I(i,j) - \mu_{KL} \right]^2 \right) / (AB) \tag{6}$$

Donde A, B, K, L tienen el mismo significado que en la ecuación 2. La varianza obtenida de los niveles de gris correspondientes al fondo y/o a regiones uniformes es muy pequeña (cero en regiones perfectamente uniformes), y la varianza obtenida en regiones donde se encuentra la imagen de la huella dactilar en buenas condiciones es alta, con este método los bloques de mala calidad se marcan para que posteriormente estos sean nuevamente filtrados.

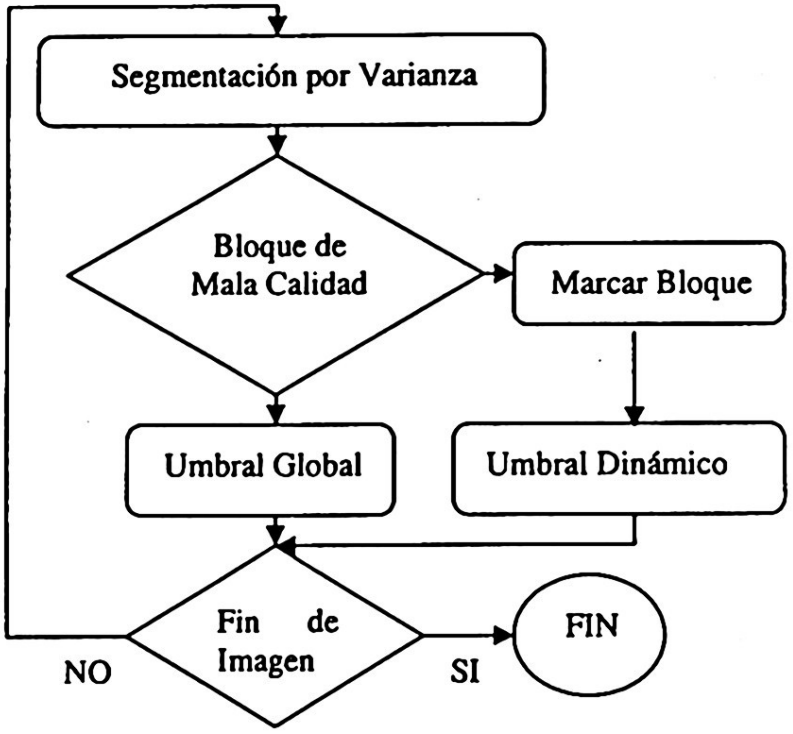


Fig. 2. Algoritmo propuesto denominado umbralización compuesta

3 Plataformas usadas para la ejecución del algoritmo

3.1 Computadora personal

Este algoritmo se programó en una computadora personal (PC) en lenguaje "C" usando el ambiente de programación visual Builder C++ versión 5.0 y en Matlab versión 5.0. Los requerimientos de la PC donde se hicieron las pruebas es el siguiente:

- Computadora personal Pentium III a 550Mhz,
- 128Mb de memoria RAM.

3.2 DSP

El DSP usado pertenece a la famlia TMS320C3x, de Texas Instruments. La arquitectura de la familia 'C3x' tiene un alto desempeño debido a su amplio y preciso rango dinámico de unidades de punto flotante, gran capacidad de memoria, alto grado de paralelismo(al tener una arquitectura Harvard) y el controlador de DMA[3].

Específicamente se uso el DSP TMS320C31-50, el cuál realiza el procesamiento de la imagen. Este DSP es un procesador con 32 bits en el bus de datos y 24 bits en el bus de direcciones, así mismo trabaja con una frecuencia de reloj de 25 MHz, al dividir internamente su frecuencia de oscilador de 50 MHz en un factor de dos. Por lo tanto su ciclo de instrucción es 40ns y realiza 275 Millones de operaciones de punto fijo por segundo(MOPS), 50 Millones de operaciones de punto flotante por segundo(MFLOPS), 25 Millones de instrucciones por segundo(MIPS).

Además estos procesadores tienen un modo de direccionamiento especializado en realizar el proceso conocido como reversión de bits para el cálculo del algoritmo de la FFT. Con esto la ejecución de dicho algoritmo es muy rápida.

4 Resultados y conclusiones

Experimentalmente se encontró que un valor de k=1.1 en el filtro fue el óptimo para el mejoramiento en imágenes de buena calidad, y para imágenes de mala calidad k=1.3 presenta mejores resultados [5, 6, 2]. Este método presenta un problema en los bordes de cada bloque por lo se necesito un traslape de 26 píxeles entre cada bloque para eliminar este efecto [5, 6].

El filtro basado en la FFT se probó con 5 imágenes de diferentes personas, las cuales tienen diferentes tamaños y diferentes tipos de calidad. Los resultados del algoritmo en 4 de esas imágenes se muestran en la figura 3.

De los resultados mostrados se puede concluir que la eficiencia de la metodología propuesta para el filtrado y mejora de imágenes de huellas dactilares es grande. Se puede observar que para las imágenes de mala calidad los resultados de la metodología propuesta son muy buenos, en cambio con las imágenes de buena calidad los resultados son muy similares a los descritos en [7].

Con estos resultados el porcentaje de eficiencia obtenido en el AFAS es muy grande [2, 1].

Los resultados de los diferentes tiempos de ejecución de este algoritmo para las 5 personas en los diferentes lenguajes de programación y plataformas, se muestran en la tabla 1.

Cabe mencionar que para realizar la implementación del filtro en el ensamblador del DSP se implementaron las funciones de logaritmo natural y exponencial en el DSP, para esto se utilizó el código de las funciones que proporciona Texas Instruments [3].

En cuanto a los tiempos de ejecución del algoritmo en la PC se tiene la ejecución más rápida al programarse el algoritmo en lenguaje "C", en esta misma plataforma al programarse el algoritmo sobre Matlab los tiempos se incrementan considerablemente.

Imagen Original – Usuario 1

Filtro Aplicado sobre Niveles de Gris

Filtro con la Metodología Propuesta

Imagen Original – Usuario 2

Filtro Aplicado sobre Niveles de Gris

Filtro con la Metodología Propuesta

Imagen Original – Usuario 3

Filtro Aplicado sobre Niveles de Gris

Filtro con la Metodología Propuesta

Imagen Original – Usuario 5

Filtro Aplicado sobre Niveles de Gris

Filtro con la Metodología Propuesta

Fig. 3. Imágenes de prueba

Tabla 1. Tiempos de ejecución del filtro basado en la FFT

Imagen	Algoritmo en "C"	Algoritmo en el DSP	Algoritmo en Matlab
Usuario1	10 seg.	42 seg.	54 seg.
Usuario2	4 seg.	14 seg.	26 seg.
Usuario3	10 seg.	39 seg.	51 seg.
Usuario4	5 seg.	21 seg.	33 seg.
Usuario5	7 seg.	31 seg.	38 seg.

En el DSP los tiempos son mayores que en la PC, a pesar de la arquitectura de este procesador, lo posibles factores que ocasionan esto, son sin duda primeramente la velocidad de 25Mhz del DSP contra 550 Mhz de la PC y el uso de métodos numéricos para el cálculo y evaluación del logaritmo natural y la exponencial en el ensamblador del DSP, en la PC se hace uso del coprocesador matemático para el cálculo de estas funciones.

Referencias

[1] V. H. Garcia, "Tesis de Maestría: Sistema de Reconocimiento de Huellas Dactilares para el Control de Acceso a Recintos",CIC-IPN, México D.F., 2004.

[2] V. H. Garcia, O. Espinosa, "Estrategias para la Autentificación de Huellas Dactilares",Congreso Internacional ANIEI, Zacatecas, Zacatecas, 2003.

[3] Texas Instruments, TMS320C3X User's Guide. 2003.

[4] Veridicom, FPS110 Data Sheet. 2003

[5] V. H. Garcia, O. Espinosa, "Sistema de Reconocimiento de Huellas Dactilares para el Control de Acceso a Recintos",Congreso Internacional CIC-INDI, México DF, 2001.

[6] V. H. Garcia, "Sistema de Reconocimiento de Huellas Dactilares para el Control de Acceso a Recintos Basado en el DSP TMS320C31", Congreso Internacional ELECTRO-2001, Chihuahua, Chihuahua, 2001

[7] A.J. Willis and L. Myers, "A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingertips", Pattern Recognition, vol. 34, No 2, pp. 255-270, February 2001.

[8] L. C. Jain, U. Halici, I. Hayashi, S. B. Lee, Intelligent biometric techniques in fingerprint and face recognition, 1999.

[9] Mardia, K.V., Baczkowski, A. J., Feng, "Statistical methods for automatic interpretation of digitally scanned finger prints", Pattern Recognition Letters. pp 1197-1203,1997.

[10] R. González, R. Woods, Digital Image Processing, 1996.

[11] Sherlock, B.G., Monro, D.M., and Millard, K., "Fingerprint Enhancement by Directional Fourier Filtering", IEEE Proceedings Vision, Image and Signal Processing, Vol. 141, No 2. pp 87-94. 1994.

[12] B. Miller, "Vital signs of identity", IEEE Spectrum, 31(2), pp. 22-30, 1994.

[13] O' Gorman, L. And Nickerson, J.V., "An approach to fingerprint filter design", Pattern Recognition, Vol. 22, No. 1, pp 29-38. 1989.

[14] B.M. Mehtre and B. Chatterjee, "Segmentation of fingerprint images A composite method", Pattern Recognition, vol. 22, pp. 381-385, 1989.

Control y monitoreo de variables físicas vía remota usando la plataforma TINI

Osiris Carreño Bernal[1] , Brenda García Rojas[1], Karen de Jesús González Solis[1], Paola Noriega Moreno[1], Victor Hugo García Ortega[2]

[1] Escuela Superior de Cómputo
Instituto Politécnico Nacional
Av. Juan de Dios Batíz, s/n, esq. M. Othón de Mendizábal
Col. Nueva Industrial Vallejo, 07738,
México, D. F.
[2]Centro de Investigación en Computación
Instituto Politécnico Nacional
Av. Juan de Dios Batíz, s/n, esq. M. Othón de Mendizábal
Col. Nueva Industrial Vallejo, 07738,
Unidad Profesional "Adolfo López Mateoz"
México, D. F.
{osiris_bernal, brestarhi, chuchita_gs_,
noriega_paola}@hotmail.com, vgarcia@cic.edu.mx

Resumen. Este trabajo presenta el desarrollo de un sistema de control y monitoreo de variables físicas tales como: iluminación y temperatura de una habitación vía remota mediante la plataforma TINI –Tiny InterNet Interface (Pequeña Interfaz de Red)– desarrollada por Dallas Semiconductor.

1 Introducción

Sabemos que hasta ahora se ha llevado a cabo el control y monitoreo de diferentes variables a través de una PC, ocasionando que el sistema sea costoso y complejo. Dichos sistemas permiten el acceso a recursos de cómputo externo utilizando una arquitectura Cliente-Servidor, de forma que el proceso de simulación es controlada por medio de Internet desde una computadora cliente. Como sistemas desarrollados con esta arquitectura se tienen [1] y [2].

El sistema que estamos desarrollando utiliza la arquitectura Cliente-Servidor. Del lado del servidor se está manejando la plataforma TINI –*Tiny InterNet Interface* (Pequeña Interfaz de Red)– y del lado del cliente se tiene una PC (Personal Computer). TINI es una plataforma que puede ser usada para tareas como monitoreo y control de un dispositivo local o todo un sistema, pero en la mayoría de sus aplicaciones se utiliza por su capacidad de red.

Provee diseños de sistemas y desarrollos de software de manera flexible y de bajo costo para diseñar una amplia variedad de hardware que permita ser conectado directamente en PC's. La plataforma es una combinación de un pequeño pero poderoso conjunto de chips y un ambiente programable en JAVA. El conjunto de

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.184-193.© Centro de Investigación en Computación, IPN, México

chips provee procesamiento, control y la capacidad de trabajo en red. Las características fundamentales del hardware son expuestas al desarrollador del software a través de un conjunto de interfaces programables en aplicaciones JAVA.

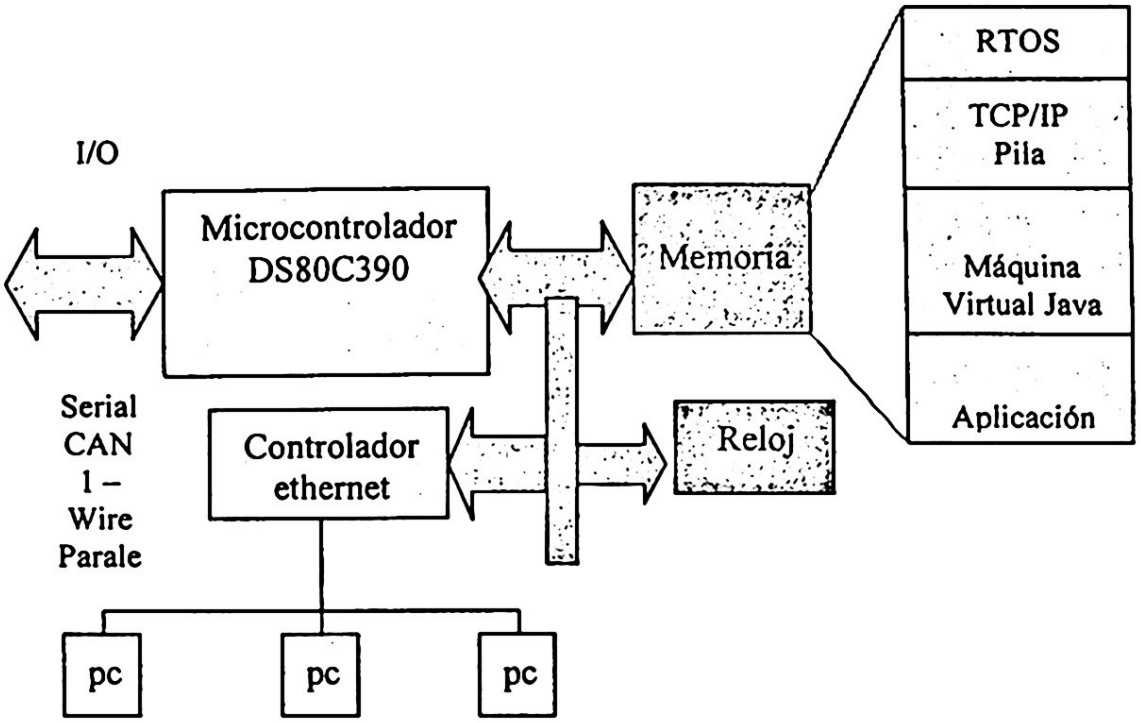


Fig. 1. Diagrama de la plataforma TINI

2 Descripción general del sistema

El sistema completo se puede apreciar en la figura 2.

TINI a través de la combinación de la amplia capacidad de I/O (puertos), una red basada en el protocolo TCP/IP y un ambiente de programación orientada a objetos, forman una plataforma para crear aplicaciones que provean no solo control local, sino también remoto. La capacidad de red de TINI se extiende a la conectividad de cualquier dispositivo permitiendo la interacción con sistemas remotos y usuarios a través de una red de aplicaciones estándar como los Web browsers [3] y [4].

El sistema que se encuentra en desarrollo pretende controlar y monitorear iluminación y temperatura de una habitación vía remota. Dicho sistema se basa en la arquitectura Cliente / Servidor que esta siendo programada en lenguaje Java, el cliente interactuará con el servidor proporcionando y adquiriendo información acerca del estado de las variables de iluminación y temperatura dentro de la habitación. El

acceso al sistema será mediante una aplicación independiente desarrollada en Java mediante Internet y el procesamiento de la información la llevará a cabo el servidor.

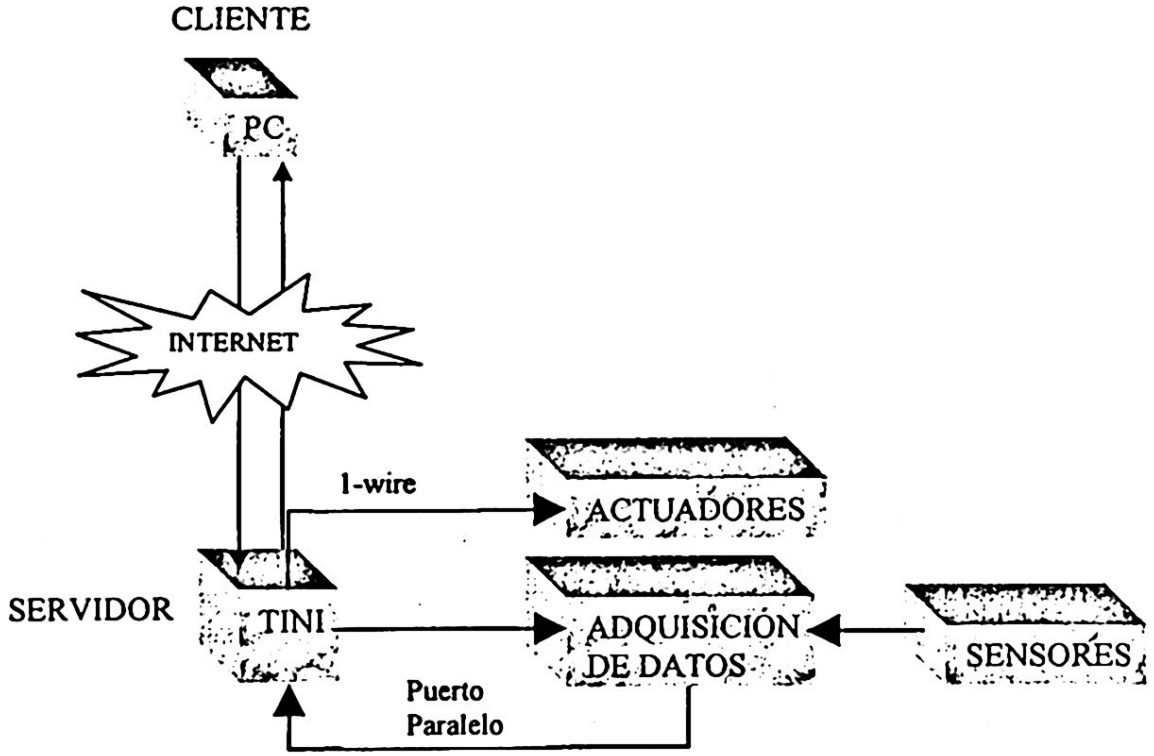


Fig. 2. Descripción General del Sistema

De los bloques mostrados en la figura 2. Describiremos el bloque de Adquisición de Datos. Este bloque se encarga de obtener los datos de temperatura a través del bloque de Sensores. El sensor utilizado para la adquisición de temperatura es el LM35, el cual proporciona 10 mV/°C, salida lineal y rangos de temperatura de -40 a 100°C [5].

2.1 Electrónica del sistema

La tarjeta de adquisición consta de los componentes presentes en la figura 3.

Convertidor Analógico – Digital (ADC0804) con una resolución de 8bits y tiempo de conversión de 100 µseg[5]. Multiplexor (ADG509), Buffer (74LS541) y Amplificador Operacional (LF351).

Los actuadores que se utilizan se catalogan conforme a las variables monitoreadas / controladas:

Variables	Actuador
Temperatura Ambiente	Aire acondicionado
Luz	Foco

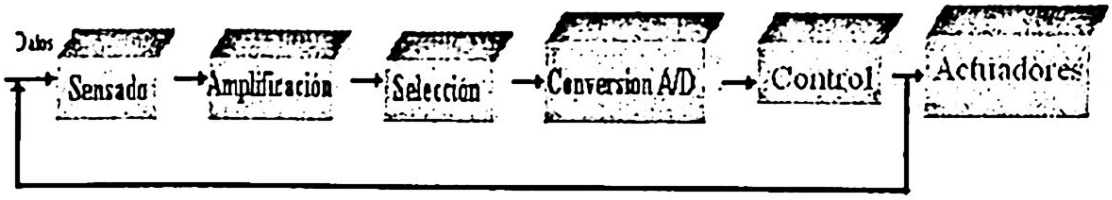


Fig. 3. Diagrama a bloques del Sistema de Control

El Aire acondicionado es empleado para aumentar / disminuir la temperatura dentro de la habitación, el cual consume 1500 Watts y 127 V.

Otro actuador empleado en este sistema es un foco, el encendido / apagado de dicho actuador será manipulado mediante control ON/OFF.

3 Controlador difuso

Un control difuso se refiere primordialmente al control de procesos a través de descripciones lingüísticas difusas. Es una de las técnicas empleadas para materializar sistemas de control, con controladores que toman decisiones similares al razonamiento humano. Puede ser diseñado de manera que imite el proceso deductivo humano.

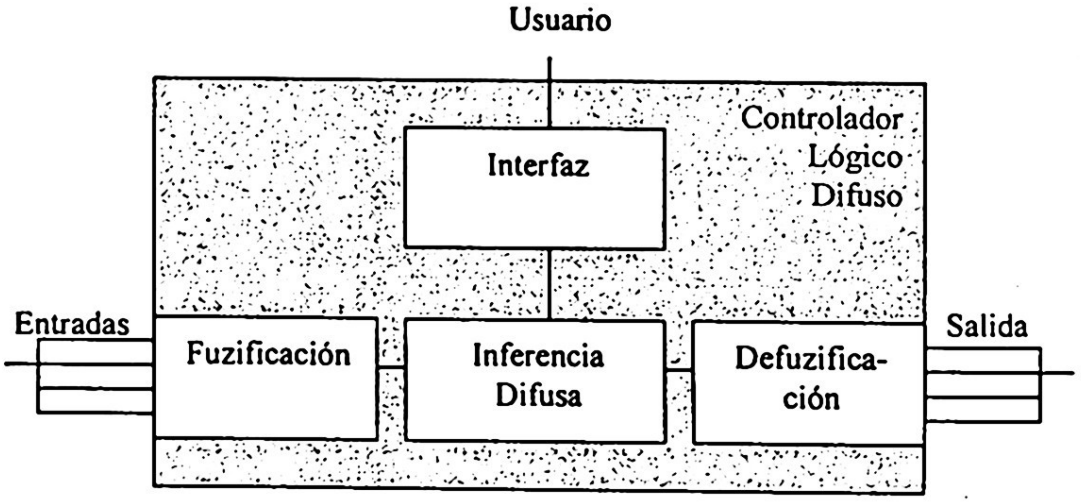


Fig. 4. Controlador Difuso

Un conjunto difuso contiene elementos, los cuales varían su grado de pertenencia en el conjunto. Dentro de la Lógica Difusa un conjunto es una clase de objeto con un continuo de grados de membresía, tal que es caracterizado por una función de membresía.

Los conjuntos difusos describen conceptos vagos (corredor rápido, tiempo caliente, días de la semana). Un conjunto difuso admite la posibilidad de pertenencia parcial en él (viernes es más o menos un día del fin de semana).

Funciones de Membresía. Es una medida de la pertenencia graduada de un elemento en un conjunto difuso. Una *función de membresía* asigna a cada objeto un intervalo de *grado de membresía* entre cero y uno. Las funciones de membresía deben definirse a partir de la experiencia o la intuición o simplemente utilizando el sentido común.

El grado de pertenencia de un objeto a un conjunto difuso es denotado por un valor de pertenencia entre 0 y 1.

Regla Difusa. Una regla difusa es una condición que expresa la dependencia entre las variables. Estas tienen la siguiente forma:

```
If <condición> then <consecuencia>
```

La condición esta compuesta de antecedentes, donde un antecedente es de la forma "x esta en A", donde x es un elemento en el espacio X y A es un conjunto difuso. Una condición puede tener uno o más antecedentes:

```
If < <antecedente1>  and  <antecedente2> > then
<consecuencia>
```

Similarmente, la consecuencia esta compuesta de consecuentes, donde un consecuente es de la forma "y esta en B", donde y es un elemento en el espacio Y, y B es un conjunto difuso. Una consecuencia puede tener un o más consecuentes:

```
If <condición> then < <consecuente1> and
<consecuente2>>
```

Proceso de Inferencia. El proceso de inferencia es un mecanismo interno que produce un valor de salida clásico (no difuso) de un conjunto de reglas difusas.

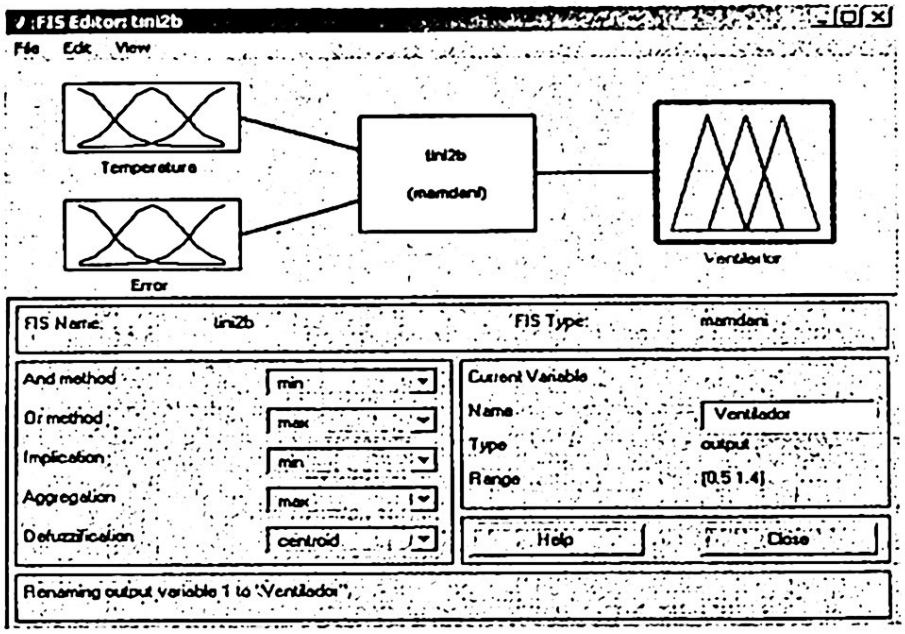


Fig. 5. Variables de entrada y salida utilizando Matlab

El proceso de inferencia difusa involucra tres pasos: Fuzificación, evaluación de reglas o inferencia difusa y defuzificación.

La variable de entrada Temperatura que es la temperatura que actualmente esta midiendo el sistema, consta de los siguientes conjuntos difusos: Frío, Templado, Ambiente, Tibio y Caliente.

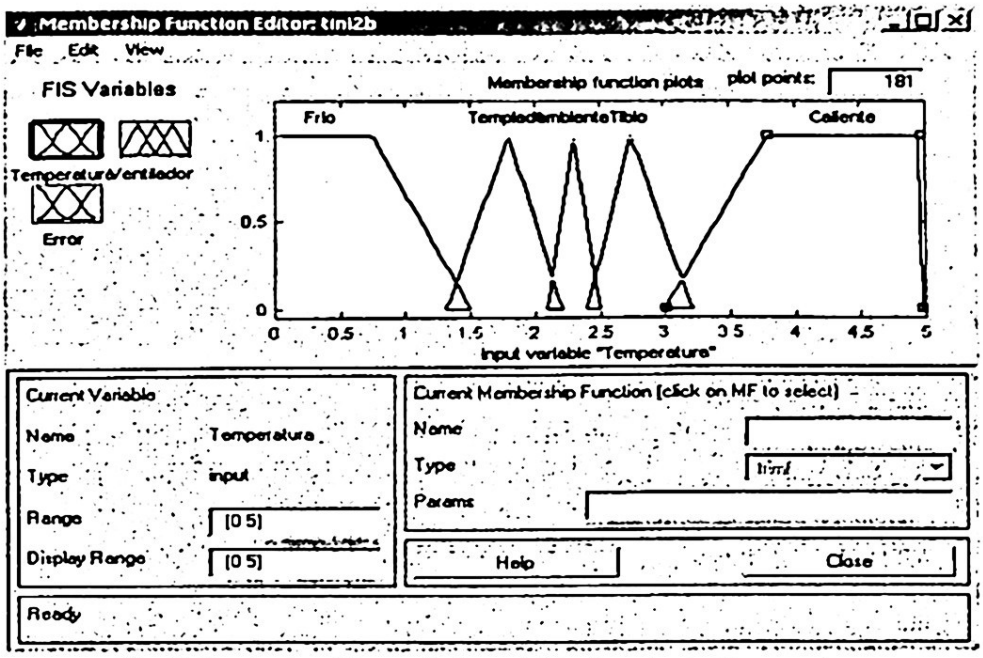


Fig. 6. Conjuntos difusos de la variable de entrada *Temperatura*

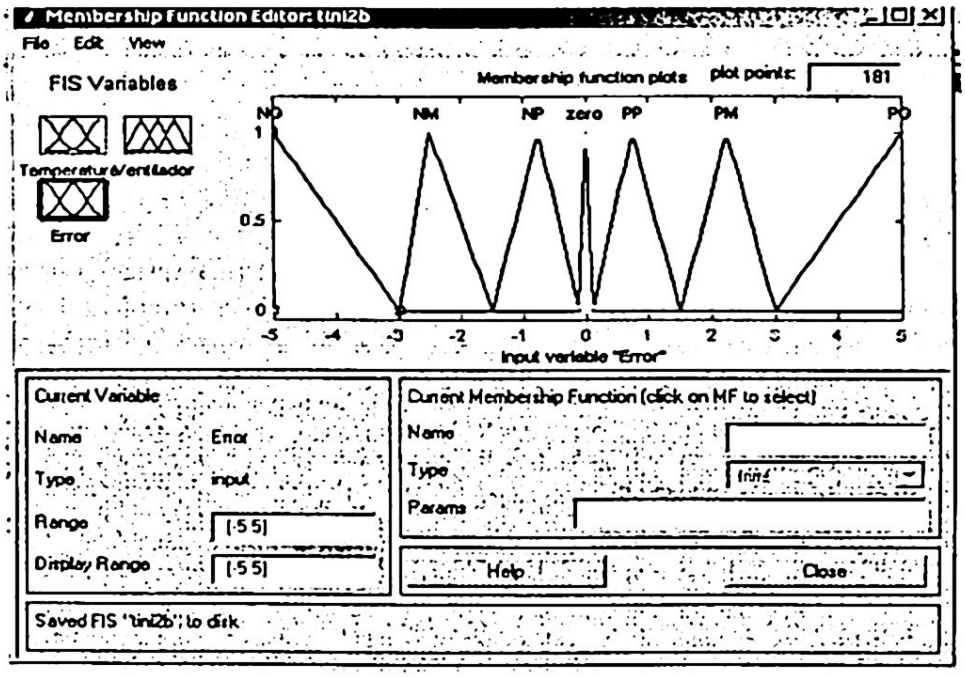


Fig. 7. Conjuntos difusos de la variable de entrada *Error*

Fuzificación. Proceso de convertir los valores de entrada a grados de pertenencia de uno o varios conjuntos difusos. Cada grado de pertenencia es asociado con una función de membresía particular.

Para nuestro sistema tenemos dos variables de entrada (Temperatura y Error) y una variable de salida (Ventilador).

La otra variable de entrada que es el Error (Error = temperatura deseada (setpoint) – temperatura actual). Consta de los siguientes conjuntos difusos: NG (negativo grande), NM (negativo mediano), NP (negativo pequeño), PP (positivo pequeño), PM (positivo mediano) y PG (positivo grande).

Y como salida de nuestro controlador difuso tenemos la variable Ventilador que contiene los siguientes conjuntos difusos: vfrío, vcaliente y vmuycaliente.

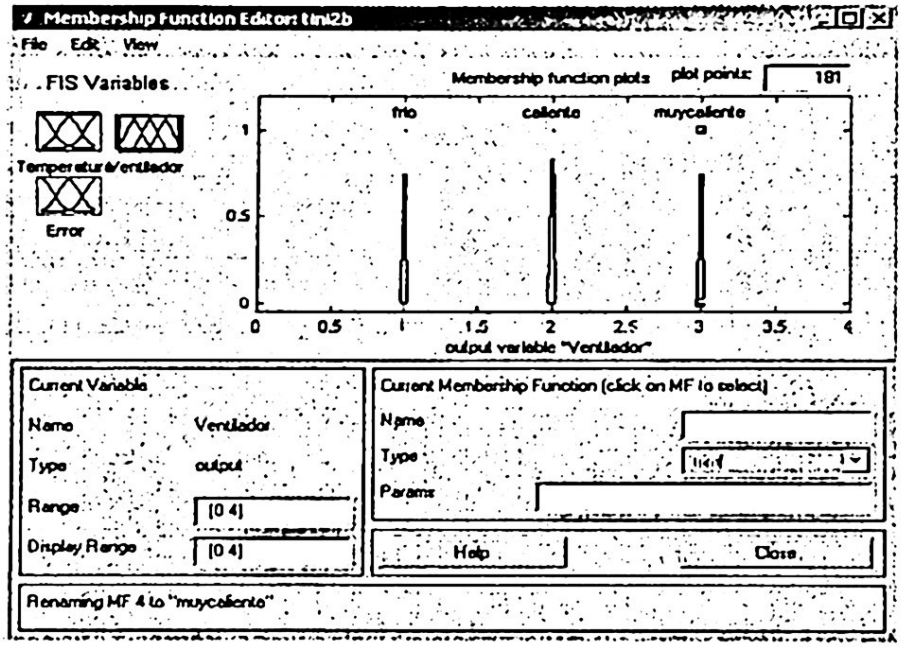


Fig. 8. Conjuntos difusos de la variable de salida *Ventilador*

Evaluación de reglas (Inferencia difusa). Determina que tan bien las condiciones de cada regla son satisfechas. Durante ciertos estados, la condición de algunas reglas son más satisfactorias que otras, de este modo, haciendo las reglas más aplicables que otras. En esta etapa se debe de establecer una serie de reglas de la forma SI – ENTONCES para indicar la acción a realizar en función del conjunto al que pertenece la entrada al sistema.

Una unidad de inferencia difusa consiste en un conjunto de variables de entrada, un conjunto de variables de salida y un mecanismo para la ejecución de un conjunto de reglas difusas.

De la tabla 1 podemos observar el comportamiento que tiene la variable de salida ventilador con respecto a las variables de entrada Temperatura y Error.

Tabla 1. Comportamiento de la variable de salida Ventilador.

Temperatura / Error	Frío	Templado	Ambiente	Tibio	Caliente
NG	apagado	vfrio	vfrio	vfrio	vfrio
NM	apagado	vfrio	vfrio	vfrio	vfrio

NP	apagado	vfrio	vfrio	vfrio	vfrio
PP	vcaliente	vcaliente	vcaliente	vcaliente	apagado
PM	vmuycaliente	vcaliente	vcaliente	vcaliente	apagado
PG	vmuycaliente	vmuycaliente	vcaliente	vcaliente	apagado

Analizando la tabla podemos deducir la reglas difusas que va a tener el controlador difuso, por ejemplo, si tenemos que la temperatura es templada y el error es negativo grande entonces el ventilador tendrá que estar encendido en frío. A continuación se muestra la regla difusa: if (Temperatura is Templado) and (Error is NG) then (Ventilador is frío).

En la Figura 9 se muestran las reglas difusas.

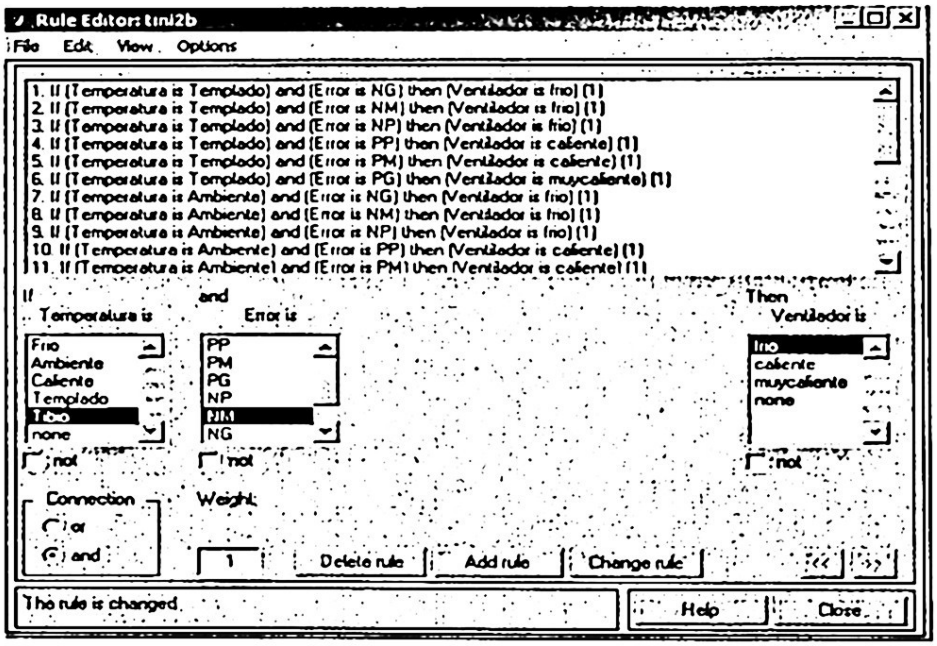


Fig. 9. Reglas difusas

Defuzificación. La defuzificación de las salidas consiste en obtener un valor numérico para cada una de las salidas del sistema a partir de los conjuntos difusos a los que pertenecen.

4 Desarrollo del sistema

La interfaz del sistema está siendo desarrollada en lenguaje de programación Java, la comunicación se llevará a cabo vía remota entre el cliente y el servidor. Del lado del cliente se tiene una aplicación independiente.

Modelado.- Diagrama General de Casos de Usos

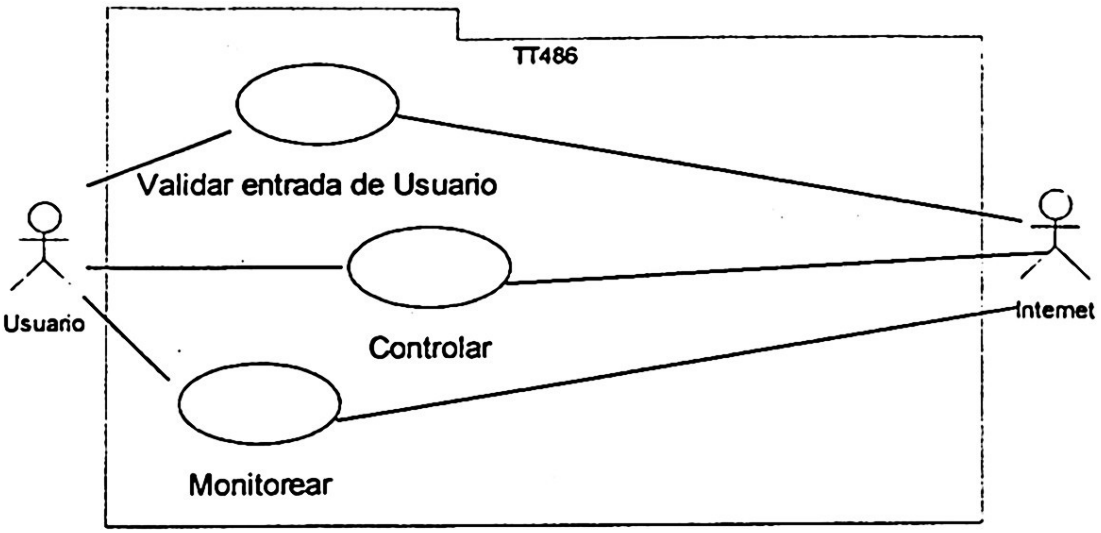


Diagrama de Casos de Uso de "Monitorear"

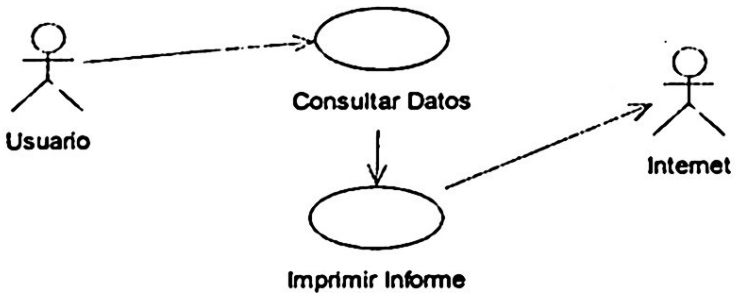
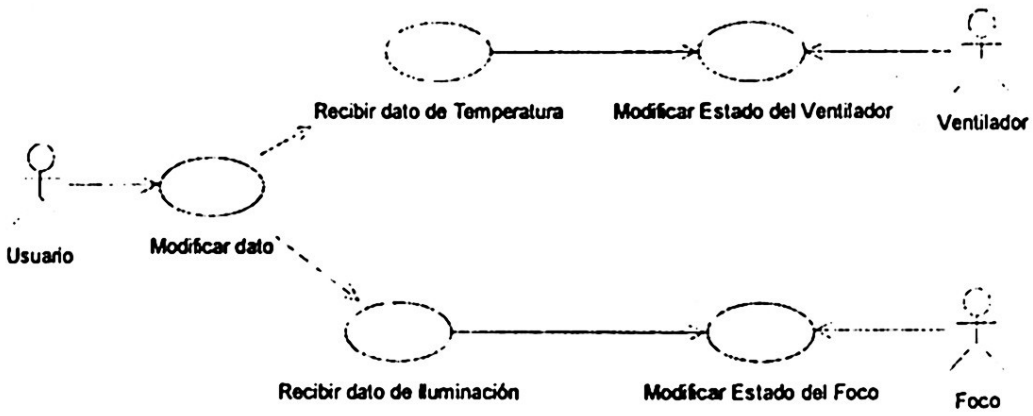


Diagrama de Casos de Uso de "Controlar"



En la figura 10 se puede observar la salida del controlador difuso, empleando el método de defuzificación centroide o centro de gravedad.

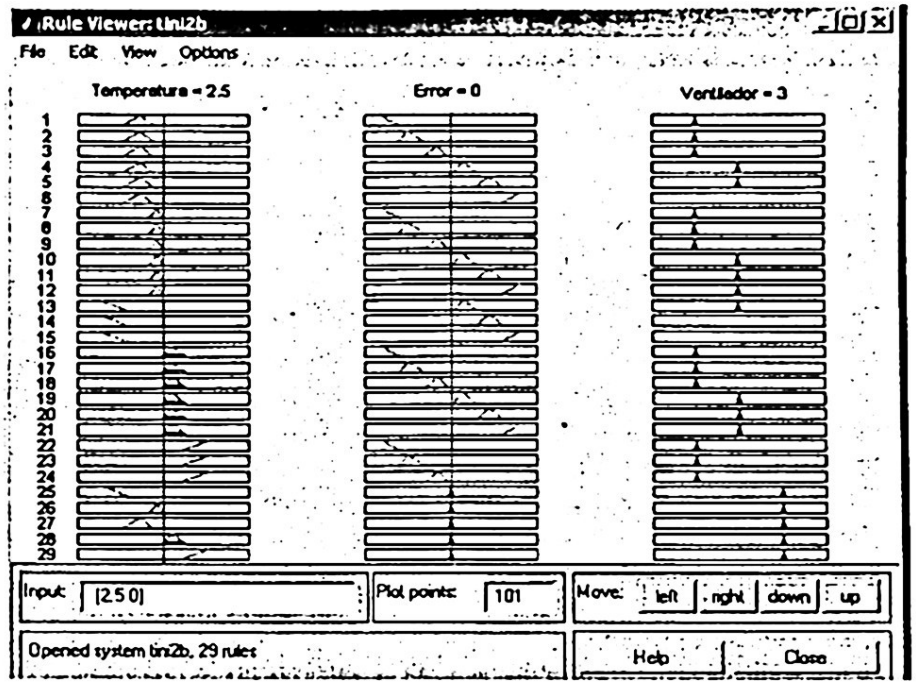


Fig. 10. Defuzificación.

5 Resultados y conclusiones

Al termino del trabajo terminal se obtendrá un prototipo de un sistema, por el cual, se monitorearán y controlarán algunas variables físicas, en este caso, iluminación y temperatura vía remota, es decir, un usuario mediante una computadora que tenga conexión a Internet podrá acceder al sistema mediante una aplicación independiente, ésta contendrá un menú con las opciones de consultar y modificar la iluminación o temperatura de una habitación. Si el usuario elige la opción de consultar, el sistema enviará un reporte general acerca del estado de la iluminación o temperatura de las habitaciones del lugar.

Ahora bien, si elige modificar se mostrará un submenú con las opciones de temperatura o iluminación. Por medio de este submenú se podrán encender o apagar las luces de cualquier habitación o modificar la temperatura a la comodidad del usuario.

Referencias

[1] Tesis: "Implementación de un sistema de inferencia adaptable neuro – difuso para el control inteligente de un sistema de riego por goteo de un invernadero a través de una red". Con re conocimiento de validez oficial según acuerdo número 933068.

[2] Sistema de Control Vía WEB "Home Automatization". Consiste en el diseño de un sistema de control de seguridad vía Web utilizando la tecnología Linux mediante una PC.

[3] http://www.ibutton.com/TINI/

[4] http://www.maxim-ic.com

[5] http://www.national.com

Verificador de circuitos lógicos

Reyes Ríos Cabrera[1], Ricardo Chaparro Sánchez[2], Rafael Urretabizkaya Garbus[2]

[1] CIATEQ A.C. Centro de Tecnología Avanzada,
Manantiales 23ª, Fracc. Ind. B.Q., 76246, Querétaro,
México
[2] Facultad de Informática, Universidad Autónoma de Querétaro
Cerro de las Campanas, 76010, Querétaro,
México
rchapa@uaq.mx, reyes.rios@ieee.org

Resumen. El artículo presenta un verificador de circuitos lógicos, con capacidad para probar circuitos digitales de las familias estándar TTL y CMOS. Con este verificador se resuelve el problema de diagnóstico de circuitos dañados que existe en los laboratorios de electrónica de las escuelas, así como también en instituciones de investigación, constituyendo con una propuesta que aporta beneficios como bajo costo, fácil mantenimiento, manejo y escalable. Consta de dos partes un dispositivo electrónico y un software. Es de fácil manejo y cuenta con elementos como consulta de diagramas y temas de ayuda, además de ser expandible.

1 Introducción

En los laboratorios de pruebas con el uso y el paso del tiempo, los circuitos lógicos sufren deterioros o daños totales afectando directamente el funcionamiento de prácticas y proyectos realizados. Por ello, es de gran importancia conocer el estado de los circuitos antes de usarse o en el proceso de uso para detectar los errores que provocan, y con esto asegurar un funcionamiento correcto de lo que se esté haciendo.
Se han desarrollado muchos instrumentos y equipos que verifican los circuitos lógicos, sin embargo tienen algunas desventajas importantes como el desarrollado por D. M Brewer [1], donde utiliza dos microprocesadores Zilog y varios periféricos, elevando su complejidad tanto de construcción como de programación, generando que el mantenimiento sea complejo y que sólo alguien especializado pueda repararlo.
La propuesta presentada resuelve el problema de forma sencilla, ya que el programa reside en una PC. Al hacer todos los procesos se ejecuten en la computadora, la complejidad de la programación y del diseño electrónico se reduce considerablemente, aportando ventajas como gran cantidad de memoria, velocidad de procesamiento y un ambiente gráfico amigable.

Algunas de las características que presenta éste verificador son:

- Es seguro, dado que cuenta con protecciones en el hardware que garantizan que no se daña el verificador por errores en el manejo o descuidos del usuario.

A. Gelbukh, G. Sidorov, W. Olán, J. Vera (Eds.) Recientes avances en la ciencia de la computación en México, pp.194-201.© Centro de Investigación en Computación, IPN, México

- Es flexible y escalable, que puede adaptarse fácilmente para probar gran cantidad de familias lógicas. Tiene un manejo sencillo, tiene una interfaz de software para el usuario que cuenta con diferentes elementos amigables y sencillos.

2 Desarrollo

El sistema consta de un dispositivo electrónico que se conecta a la computadora y un software que lo controla, en esta sección se describen ambas.

En la figura 1, se muestra un diagrama general del sistema.

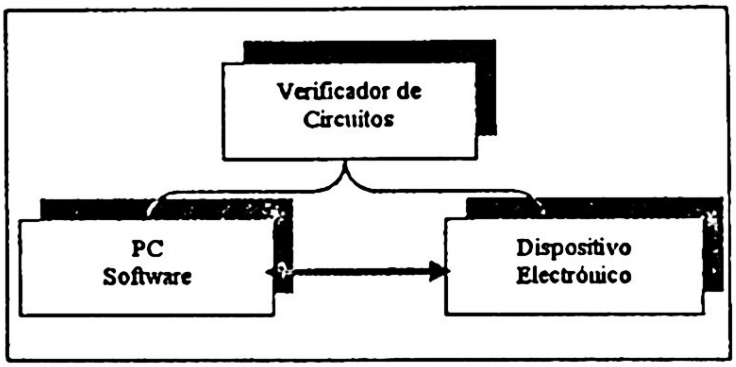


Fig. 1. Verificador de circuitos lógicos

El desarrollo se describe a partir de un planteamiento general de funcionamiento. La idea se basa en trasladar la complejidad de programación y electrónica a la computadora teniendo así mejores herramientas y más beneficios; reduciendo la complejidad y costo, tanto de desarrollo, como de mantenimiento del verificador, haciendo más fácil llevar a cabo versiones futuras, debido a que se traslada el problema a exclusiva programación en la PC, lo que hace posible un fácil crecimiento.

2.1 Dispositivo electrónico

El dispositivo se describe a partir del hardware, de las tareas y la comunicación con la PC.

2.1.1 Hardware. Se le considera una interfaz inteligente y está formada por dos microcontroladores (M1 y M2), los cuales, por medio de comunicación serial, se configuran y controlan por completo desde una PC. La interfaz inteligente tiene la característica de poderse configurar como se desee ya sean entradas o salidas digitales en cada una de sus terminales además de escribirse y leerse de la misma forma sin restricciones, teniendo la función de un puerto configurable.

Una vez hecho el análisis de los circuitos digitales se consideraron los siguientes requerimientos:

- Voltajes de 0V y 5V para marcar el cero y uno lógico respectivamente.
- 30 líneas Configurables como entradas o salidas digitales independientes.

- Puerto de comunicación.

A partir de los cuales se decidió usar el microcontrolador PIC16F877 de la marca Microchip [2], ya que cumple con las características.

El diagrama general del sistema, figura 2, cuenta con dos microcontroladores comunicados con una PC por medio de un bus serial (USART), un arreglo de buffers y una base de fuerza de inserción nula.

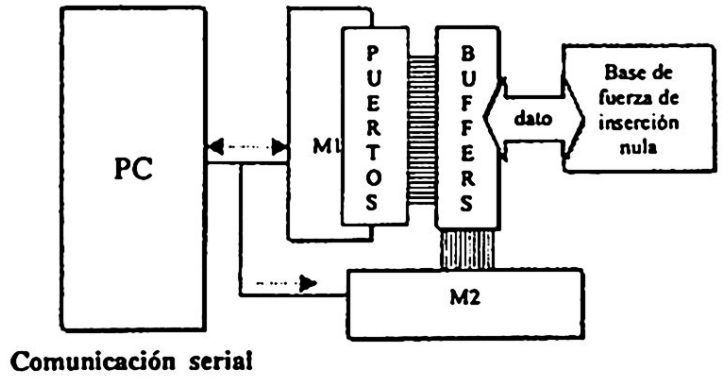


Fig. 2. Bloques del sistema

El arreglo de buffers configurables como entradas o salidas, se colocan donde los circuitos requieren energía, de modo que el circuito de prueba se encuentre siempre energizado por alguno de ellos, esto porque los microcontroladores no proporcionan la suficiente corriente para alimentar los circuitos.

Cada uno de los buffers conectados funcionan de la siguiente forma, figura 3.

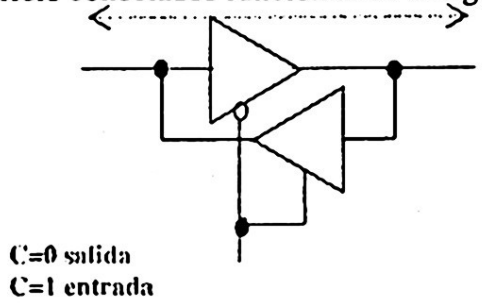


Fig. 3. Arreglo de Buffers

2.1.2 Tareas. M1 se encarga de las lecturas y escrituras en la base de prueba, M2 configura los buffers para determinar la función ya sea salida o entrada, figura 3, con respecto a M1.

De las 33 líneas de E/S digitales, que suman los puertos de M1; 2 se usan para la comunicación serial, 30 se distribuyen 15 y 15 en la base de fuerza de inserción nula y una más para un led indicador.

Los puertos de M1 se ordenan A, B, E, C, D, figura 4. Los puertos de M2 no se usan en su totalidad, sólo se usa el puerto B y D, que es con los que se configuran los buffers, y el resto es para escalamiento del dispositivo.

Fig. 4. Base de prueba.

2.1.3 Comunicación y operaciones básicas. La computadora hace la petición de lo que necesita, envía un 0 para hacer una petición de lectura y un 1 para la escritura a la interfaz inteligente. Al escribir M1 recibe dos bytes, el primero indica la dirección y el segundo el dato, figura 5a. Al leer se recibe un byte, que indica la dirección de la cual el M1 debe leer y regresar a la computadora el valor que se encuentre en esa dirección, figura 5b.

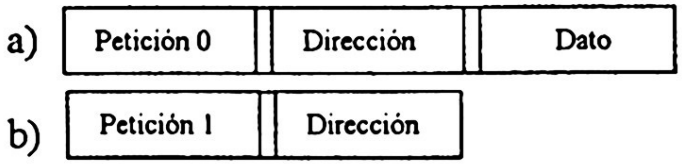


Fig. 5. Gráfica de las operaciones (sin protocolo).

La comunicación se llevo a cabo por medio del puerto USART de la interfaz inteligente, configurada a una frecuencia de 19200 bps, usando la norma RS232. Utiliza un protocolo de tres o cuatro bytes según la función, utilizando como byte de inicio, el código ASCII de la diagonal, los bytes de datos y como fin de trama el código de la diagonal invertida, figura 6.

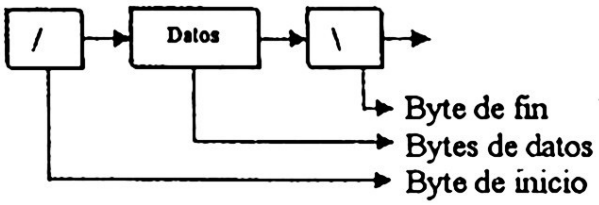


Fig. 6. Protocolo de comunicación

El protocolo es el mismo para los dos microcontroladores M1 y M2, lo que hace la diferencia es un bit de paridad enviado, de esta forma se define cual microcontrolador es el que tiene que hacer la operación, 0 para el M1 y 1 para el M2 (marca y espacio respectivamente).

2.1.4 Seguridad. El hardware está protegido contra cortos circuitos entre el circuito de prueba y la interfaz inteligente por medio de resistencias de 470Ω que mantienen a

los microcontroladores con una impedancia constante, aun si los voltajes no coinciden.

El verificador es el siguiente, figura 7

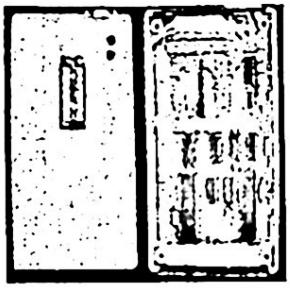


Fig. 7. Verificador de Circuitos lógicos.

2.2 Descripción del software

El software diseñado fue escrito en el lenguaje C++ Builder, [3] y [4], éste cuenta con varias opciones, figura 8.

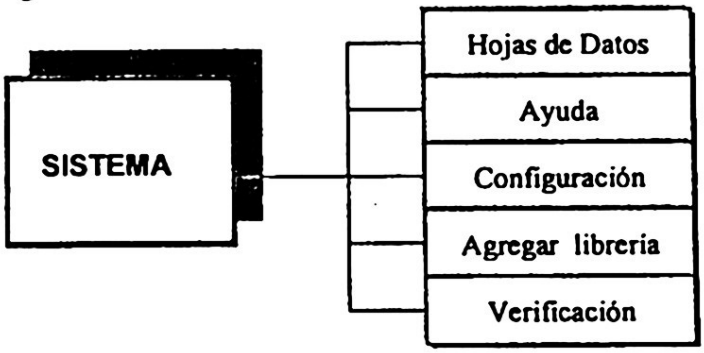


Fig. 8. Contenido de la Interfaz de Software

La primera opción nos muestra las hojas de datos de los circuitos de la familia TTL, en formato PDF, así como otros diagramas de uso común [5] y [6].

La parte de ayuda señala todos los procedimientos importantes en el sistema así como la interpretación de resultados y los pasos para llevar a cabo la reparación del verificador.

La sección de configuración nos permite fijar la frecuencia de trabajo y el puerto de comunicación con el dispositivo verificador (puertos COM1 ó COM2).

2.2.1 Agregar librerías. Las librerías de los circuitos de prueba son archivos que usa el sistema cuando lleva a cabo la verificación del funcionamiento. Maneja una librería por circuito, y esta contiene valores y características particulares.

Las librerías están guardadas de acuerdo a las operaciones que los circuitos necesitan de entrada y lo que se espera de respuesta, distribuidas en forma de vectores, donde si el circuito cuenta con 14 pines será un vector de 14 posiciones. En cada vector se encuentran los valores lógicos que determinan todas combinaciones necesarias para su funcionamiento.

Tabla 1. Contenido de las librerías

Contenido de la librería
Número de terminales
Terminal de voltaje
Terminal de tierra
Vector de configuración de entradas y salidas
Tabla lógica del circuito
...
...
...

Ejemplo para el circuito SN74LS14, figura 9.

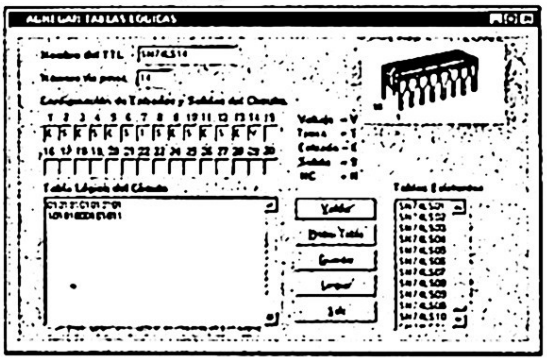


Fig. 9. Interfaz para agregar tablas lógicas

El software necesita como entrada el nombre del circuito y los datos mencionados, tabla 1. Cuando se ingresa la tabla lógica, un cero indica que es entrada en el circuito de prueba y un uno que es salida. Los microcontroladores no cuentan con datos, son únicamente una interfaz inteligente que sirve de intermediario entre el circuito de prueba y la computadora.

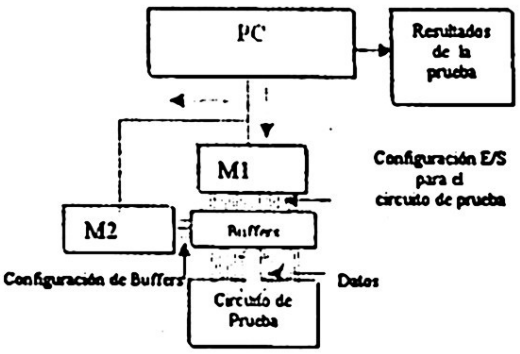


Fig. 10. Bloques de la verificación de un circuito

2.2.2 Verificación de funcionamiento. El primer paso es configurar M2, donde se define el funcionamiento de los buffers (entrada o salida con respecto a M1), figura 10, obteniendo los valores de la librería del circuito que se esté probando.

Después se configura M1, determinando las terminales como entradas o salidas para que coincidan con el circuito de prueba y en base a ello hacer la verificación. Una vez hecha la configuración comienza un ciclo terminando hasta que la tabla lógica del circuito es aplicada a éste en su totalidad. El procedimiento de aplicación de la tabla lógica se basa en las operaciones definidas de lectura y escritura, tomando la tabla lógica por bloques y distribuyéndola en la base de fuerza de inserción nula por medio de M1, escribiendo cada vector, leyendo las respuestas del circuito y comparando con resultados esperados. Finalmente muestra los resultados en la interfaz del software.

2.2.3 Opciones de prueba. Son dos una activa y pasiva; La primera consiste en una verificación constante e indefinida, finaliza cuando el usuario lo decida. La pasiva es sólo una escritura y lectura de la aplicación de la tabla lógica usándose una sola vez, determinando así el estado del circuito.

2.3 Análisis de resultados

El verificador general de circuitos lógicos desarrollado logró obtener de 100 pruebas activas realizadas a diferentes circuitos, 100 aciertos, además de otras 100 pasivas donde se obtuvieron resultados satisfactorios.

En el peor de los casos (donde la tabla es muy grande) se obtiene un tiempo de prueba de 15 segundos, mientras que el tiempo promedio es de 3 segundos por compuerta.

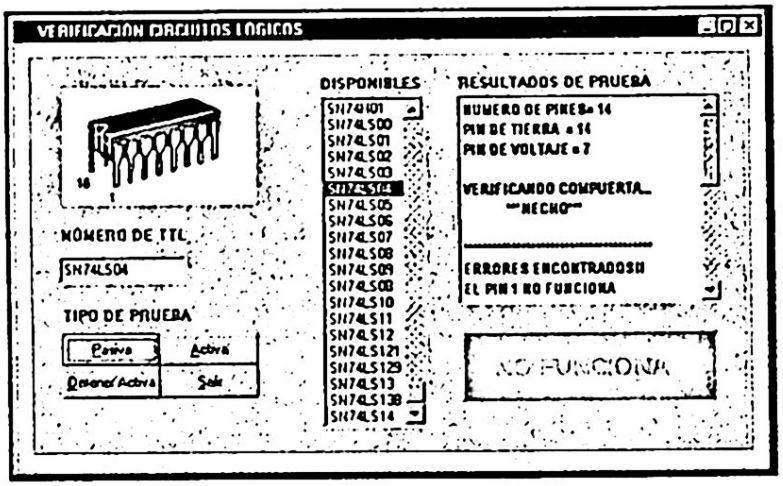


Fig. 11. Interfaz para la de verificación.

Actualmente se está trabajando en una tercera etapa, donde el proyecto, podrá identificar los dispositivos automáticamente; también se modificará para llevar a cabo la prueba y programación de memorias y arreglos lógicos, ya que teniendo las bases actuales, sólo basta agregar librerías en el formato establecido para llevar a cabo la programación o verificación que se desee.

3 Conclusiones

Se ha desarrollado un sistema verificador de circuitos lógicos, que resuelve el mismo problema que otros modelos comerciales [7], pero a un costo mucho menor, además de que para su mantenimiento no se requieren conocimientos avanzados.

Referencias

1. PC BASED TESTER FOR TTL & CMOS LOGIC DEVICES Written & Designed by D. M. Brewer 1996. http://freespace.virgin.net/darren.brewer/project.htm
2. Microchip Data Sheet PIC16F87X, 28/40-Pin 8-Bit CMOS FLASH microntroles.
3. Chirs H. Pappas, William H. Murray III "Manual de Borland C++ 4.0", McGraw-Hill, 1997.
4. Francisco Charte, "Programación con C++ Builder 5", Anaya Multimedia
5. Data Sheet Book 2 - J. P. M. Steeman, 1988, Elektuur B.V., Beek L, Netherlands, reproduksi dan distribusi Indonesia: 1989, PT. Elex Media Komputindo, Jakarta Indonesia.
6. Texas Instruments, "Logic Selection Guide and Databook CD-ROM", Advanced System Logic Products, January 1997.
7. Jameco Electronics, "Linear IC Tester", Octubre – January 2001-2002, Belmont California U.S.A.

Índice de autores

Comité editorial del libro

Dra. Sofía Natalia Galicia Haro (CIC-IPN, UNAM, México)
Dra. Elizabeth Pérez Cortés (UNAM, México)
Dr. Adolfo Guzmán Arenas (CIC-IPN, México)
Dr. Alexander Gelbukh (CIC-IPN, México)
Dr. Grigori Sidorov (CIC-IPN, México)
Dr. Carlos A. Coello Coello (CINVESTAV, México)
Dr. Cuauhtémoc Lemus Olalde (CIMAT, México)
Dr. Andrei Tchernykh (CICESE, USA)
Dr. Sergio Rajsbaum (UNAM, México)
Dr. Mario Piattini (UCLM, España)
Dr. José Miguel Alonso (EHU, España)
Dr. Manuel Montes y Gómez (INAOE, México)
Dr. António Rito Silva (UTL, Portugal)
Dr. Luis Alberto Pineda Cortés (UNAM, México)
Dr. Dominique Decouchant (IMAG, France)
Dr. Aurelio López López (INAOE, México)
Dr. Macario Polo Usaola (UCLM, España)
Dr. Manuel Palomar (UA, España)
Dr. Arturo Trujillo (Canon Research Centre, Inglaterra)
Dr. Anselmo Peñas Padilla (UNED, España)
Dr. Erick Cantú Paz (CASC, USA)
Dr. Michel Adiba (IMAG, Francia)
Dr. Leo Joskowicz (HUJI, Israel)
Dr. Carlos Martin Vide (URV, España)
Dr. Paolo Rosso (UPV, España)
Dr. Vicente Pelechano (UPV, España)
Dr. Ralf Steinberger (Joint Research Centre, Italia)
Dr. Oscar Pastor López (UPV, España)
Dr. Daniel Panario (Carleton University, Canada)
Dr. Carlos Gonzalez Ochoa (Purdue University, USA)
MC. Jorge Arnulfo Quiane Ruiz (IMAG, Francia)
MC. Julia Sambarino (CIC-IPN, México)
MC. Manuel Mejia Lavalle (IIE, México)

Recientes avances en la ciencia de la computación en México
Impreso en los Talleres Gráficos
de la Dirección de Publicaciones del
Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, DF.
Mayo 2004. Edición: 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 7 años. Los jóvenes mexicanos, continuando con la gran tradición, presentan al mundo sus obras, con pasión y amor.

ISBN: 970-36-0149-9

970-36-0149-9