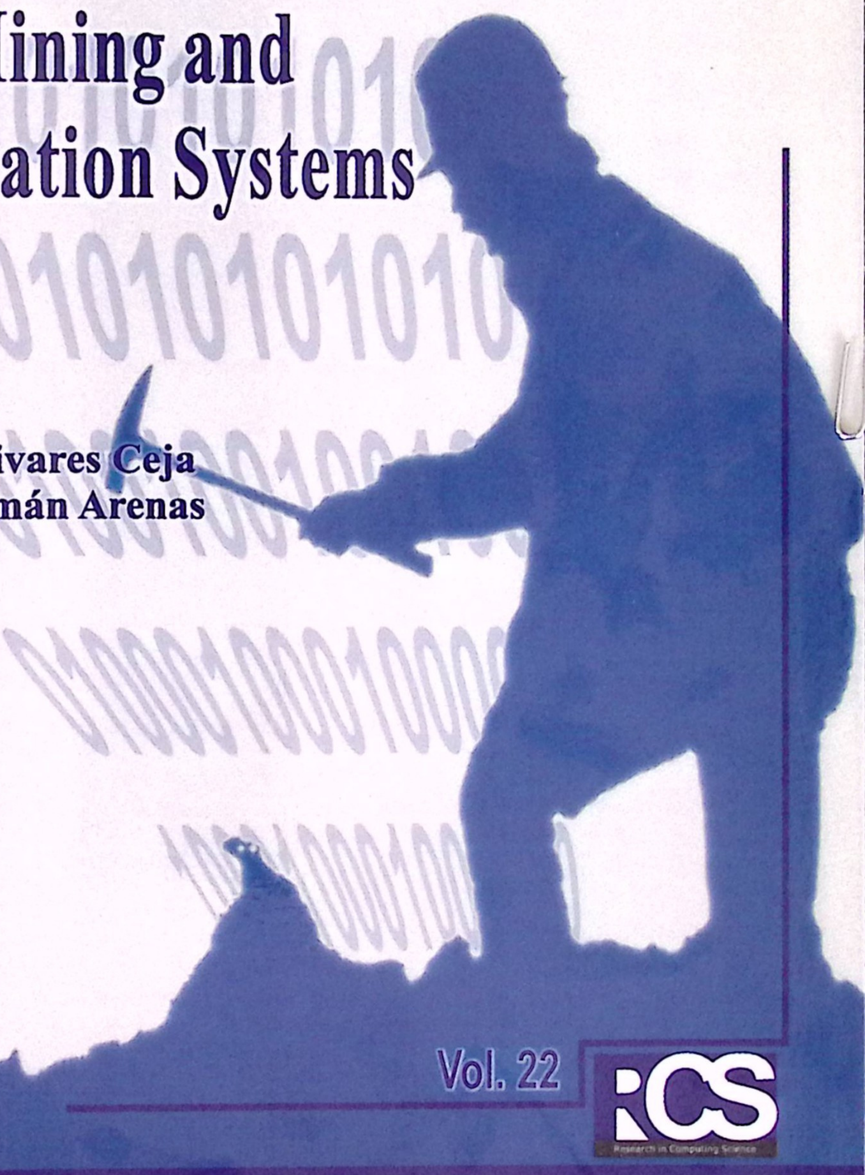
RESEARCH IN COMPUTING SCIENCE

ISSN: 1870-4069

Special issue: Data Mining and Information Systems

Jesús M. Olivares Ceja
Adolfo Guzmán Arenas
(Eds.)



Vol. 22

RCS
Research in Computing Science

Special Issue: Data Mining and Information Systems

Research in Computing Science

Research in Computing Science es una publicación trimestral, de circulación internacional, editada por el Centro de Investigación en Computación del IPN, para dar a conocer los avances de investigación científica y desarrollo tecnológico de la comunidad científica internacional. **Volumen 22**, Noviembre, 2006. Tiraje: 500 ejemplares. *Certificado de Reserva de Derechos al Uso Exclusivo del Título* No. 04-2004-062613250000-102, expedido por el Instituto Nacional de Derecho de Autor. *Certificado de Licitud de Título* No. 12897, *Certificado de licitud de Contenido* No. 10470, expedidos por la Comisión Calificadora de Publicaciones y Revistas Ilustradas. El contenido de los artículos es responsabilidad exclusiva de sus respectivos autores. Queda prohibida la reproducción total o parcial, por cualquier medio, sin el permiso expreso del editor, excepto para uso personal o de estudio haciendo cita explícita en la primera página de cada documento. Impreso en la Ciudad de México, en los Talleres Gráficos del IPN – Dirección de Publicaciones, Tres Guerras 27, Centro Histórico, México, D.F. Distribuida por el Centro de Investigación en Computación, Av. Juan de Dios Bátiz S/N, Esq. Av. Miguel Othón de Mendizábal, Col. Nueva Industrial Vallejo, C.P. 07738, México, D.F. Tel. 57 29 60 00, ext. 56571.

Editor Responsable: *Juan Humberto Sossa Azuela*, RFC *SOAJ560723*

***Research in Computing Science** is published by the Center for Computing Research of IPN. **Volume 22**, November, 2006. Printing 500. Authors are responsible for the contents of their papers. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without prior permission of Centre for Computing Research. Printed in Mexico City, November, 2006, in the IPN Graphic Workshop – Publication Office.*

Volume 22
Volumen 22

Special Issue:
Data Mining and Information Systems

Volume Editors:
Editores del Volumen

Jesús M. Olivares Ceja
Adolfo Guzmán Arenas

Instituto Politécnico Nacional
Centro de Investigación en Computación
México 2006

ISSN: 1870-4069

Instituto Politécnico Nacional (IPN)
Centro de Investigación en Computación (CIC)
Av. Juan de Dios Bátiz s/n esq. M. Othón de Mendizábal
Unidad Profesional "Adolfo López Mateos", Zacatenco
07738, México D.F., México

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

Indexed in LATINDEX
Indexada en LATINDEX

Printing: 500
Impresiones: 500

Printed in Mexico
Impreso en México

Preface

This issue includes some papers of The International Congress on Data Mining and Information Systems (ICDIS 2006). Research in this field is intended to improve applications performance and automatically finding of anomalies, deviations and interesting phenomena in a sea of data.

The topics covered in this number are related to:

— Merging semantic information
— Measuring inconsistency of information
— Automatic generation of hypotheses
— Decision making using hierarchical information
— Evaluation of mined rules and data models
— Engineering requirements in Data Mining projects

The field is active and we foresee that many research results would be seen soon in commercial products, providing the advantages that authors are working on.

We thank to all people who participated in this event. The reader is welcomed to enjoy the contents of this volume.

November 2006

Jesús M. Olivares Ceja
Adolfo Guzmán Arenas

Table of Contents

Índice

Page/Página

Information Systems

Data Mining

Information Systems

Automatic Merging of Knowledge using Ontologies

Alma Delia Cuevas Rasgado and Adolfo Guzman Arenas
Center for Research in Computer Science, National Polytechnic Institute, Av. Juan de Dios Batiz, s/n, Zacatenco, 07738
Mexico City, Mexico
almadeliacuevas@gmail.com; aguzman@acm.org

Abstract. The fact that many people simultaneously construct the pages of the Web in an independent way, generates a great obstacle for the machines that track the information in it. Therefore, the concept of Semantic Network has been introduced. It provides a standardization of the information through markup languages (SGML, XML, etc.) where the user generates his own annotations, almost all of them as labels or syntactic rules. Relatively few of the languages have tried to represent and to manipulate the knowledge with methods of Artificial Intelligence. This paper proposes a structure (an ontology) more suitable to represent knowledge, with interesting contributions with respect to current languages (AML+OIL[5], RDF[8], OWL[12]). Also, this paper presents an automatic algorithm to match and merge two or more ontologies. This merging is important when it is desired to increase the knowledge in an ontology. In that way it is possible to accumulate the knowledge in an automatic way. The process of merging begins by obtaining the value of the similarity between each elements of the ontologies (through COM[1] Algorithm); later, the optimal matching is sought. Finally, the result defines the new ontology. This process is performed totally by the computer. That is to say, the user does not take part in this process, as it happens in current merging algorithms (OntoMerge[6], FCA-Merge[9], Chimaera[11], Prompt[13], If-Map[14]). In the merging, the OM Algorithm solves problems of contradiction and reorganization of the final ontology. The efficiency of the algorithm of fusion is demonstrated through several examples.

Keywords: Knowledge base, Internet, Ontology merging

1 Introduction

These days computers are not anymore isolated devices but they are important entry points in the world-wide network that interchanges knowledge and carry out business transactions. Nowadays, using Internet to get data, information and knowledge interchange is a business and academic need. Despite the facilities to have access to the Internet, people face the problem of heterogeneous sources because there are not

© Jesús Olivares, Adolfo Guzmán (Eds.)
Data Mining and Information Systems.
Research in Computer Science 22, 2006, pp. 3-14

suitable standards in knowledge representation. This paper addresses this need of businesses and academia.

Many answers that people require involve acceding several sources in the Internet and then they merge manually the acquired information in a reasonable way. Merging the information is an important task and many languages and tools (DAML+OIL[5], RDF[8] and OWL[12]) have been developed to describe and process Internet content but the languages lack enough expressiveness to detail knowledge representation.

It is required that computer decipher the information (said, in a document written in a natural language) and convert it to a suitable notation (its knowledge base) that preserves relevant knowledge. This knowledge base can be an ontology. Ontology is an information technology that manages the knowledge through nodes that are joined with each other through relations, to describe a knowledge domain. Current works that merge ontologies (OntoMerge[6], FCA-Merge[9], Chimaera[11], Prompt[13], and If-Map[14]) rely on the user to solve the most important problems found in the process. This paper describes two important contributions to obtain better advantages of the Web resources:

1. A new notation to represent knowledge using ontologies, called OM (Ontology Merging) Notation and
2. An automatic algorithm to merge ontologies called OM Algorithm- That is, without human intervention

The OM notation provides several improvements to current languages of definition of ontologies. Two of them are: (a) the new type of relation called *Partition*; (b) a node or concept can also be defined as a relation.

Likewise, the merging algorithm that we will explain is totally automatic. This algorithm solves by itself all the problems found in the process. That is to say, the user does not take part in the process.

2. OM Notation

In the context of sharing knowledge, ontologies provide a clear, syntactic and formalized structuring of a set of nodes also called concepts that are related to each other, under a knowledge domain and that is common to many people and machines.

OM Notation represents ontologies through a structural design with labels similar to XML. Theses labels identify the description of the concepts and their relations. The labels and their descriptions are shown on table 1.

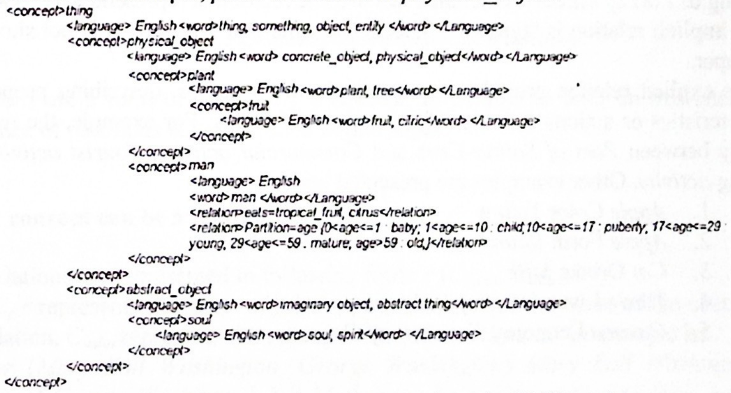
The binary and n-ary relations are described in OM Notation. That is, a relation can have more of one value and these could be concepts. For example, *Zebra* concept has a relation *Color* that is connected to two elements *White* and *Black*.

<concept> *c* </concept>	Where *c* represents the name of the concept.
<language> *l* </language>	Where *l* represents the language in which the words are defined.
<word>$w_1,w_2...w_n$</word>	Where $w_1,w_2...w_n$ represent the words that describe the concept *c*.
<arity> *a* </arity>	Where *a* is a positive number that describes the arity of the concept *c*.
<relation> *n* = *v* </relation>	Where *n* represents the name and *v* represents the value of the relation. The value *n* and *v* are concepts. The *v* can be a list if the relation has more of a value.
<part> *c* </part>	Concept that contains this relation *is part of* the concept *c*.
<member> *c* </member>	Concept that contains this relation *is member of* the concept *c*.
<subset> *c* </subset>	Concept that contains this relation *is subset of* the concept *c*.
<type> *c* </type>	Concept that contains this relation *is a type of* the concept *c*.

Table. 1. Labels used in the OM Notation.

The relations are properties or characteristics of the node or concept where they are defined. An example of this called relation *Eat* is shown in figure 1.

Other relations exist, such as hyponymous relation, that are expressed through concepts nested. For example, *plant* is a subset of *physical_object*.



```
<concept>thing
        <language> English <word>thing, something, object, entity </word> </Language>
        <concept>physical_object
                <language> English <word> concrete_object, physical_object</word> </Language>
                <concept>plant
                        <language> English <word>plant, tree</word> </Language>
                        <concept>fruit
                            <language> English <word>fruit, citric</word> </Language>
                        </concept>
                </concept>
                <concept>man
                        <language> English
                        <word>man </word></Language>
                        <relation>eats=tropical_fruit, citrus</relation>
                        <relation>Partition=age {0<age<=1 : baby; 1<age<=10 : child,10<age<=17 : puberty, 17<age<=29 :
                        young, 29<age<=59 . mature, age>59 : old,}</relation>
                </concept>
        </concept>
        <concept>abstract_object
                <language> English <word>imaginary object, abstract thing</word> </Language>
                <concept>soul
                    <language> English <word>soul, spirit</word> </Language>
                </concept>
        </concept>
</concept>
```

Figure 2 Representation of an ontology in OM Notation.

Relations are Implicit and Explicit. The Implicit relation indicates a structural relation (parent-son). For example, the relation "part of" exists between holonymous and meronymous sets. Ontology with nodes and relations is shown in the figure 2. The circles and arrows denote nodes and relations respectively. A set is holonymous of another when its semantic notion represents the whole of an object; therefore *bicycle* is holonymous of *handle-bar.* A set is Meronymous when it represents a part of an object; therefore *handle-bar* is meronymous of *bicycle.*

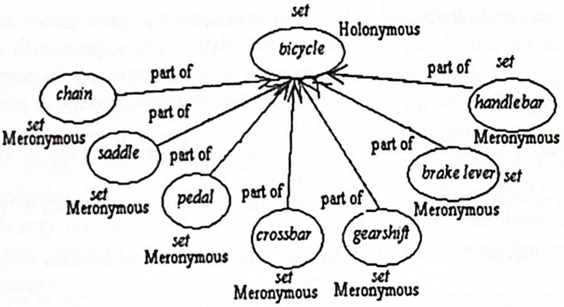


Figure 2. Graphical representation of ontology with "part of" relation.

Other implicit relations exist, such as: Hyperonymous and Hyponymous, where a term is hyperonymous of another if meaning of the first one includes the second one. The set *Port* as hyperonymous of *Port of Mexico* is an example of this, because the meaning of *Port of Mexico* is included in *Port*; the relation is represented as <subset>. Other implicit relation is "type of". This is the same that "subset". It is not shown in this paper.

The explicit relation provide more semantic to the nodes, describing properties, characteristics or actions that identify a concept of others. For example, the relation *activity* between *Port of Salina Cruz* and *Commercial activity*, *Tourist activity* and *Fishing activity*. Other examples are presented here:

1. *Apple* Color *Yellow*
2. *Apple* Form *Round*
3. *Cat* Drinks *Milk*
4. *Turtle* Lives *Intertropical zone*
5. *Oaxaca* Economy *Economy of Oaxaca*

2.1. Relation of type Partition

A partition is a collection of sets such that whatever two elements of this collection are mutually exclusive and all of them are collectively exhaustive.

Nowadays, partitions are not represented in languages [4], [10] and [12]. Partitions are represented of the following way:

Partition=$nomPart\{range_1{:}value_1;range_2{:}value_n,range_1{:}value_n\}$

Where *nomPart* represents the name of partition, *range* is the characteristic that distinguishes this set of the other sets of the partition. This element can be an interval,

a list of elements or simply a character. The *value* represents the value of the range, the name of interval, list or character. This can be a node or concept.
For example:
<relation>
Partition = *age {0<age<=1: baby;1<age<=10: child;10<age<=13: tennager;13<age<18: young;18<=age<40: adult;40<=age<60: mature;60<=age: old;}*
</relation>
The graphic representation of a partition is shown in figure 3: the small, black circle represei

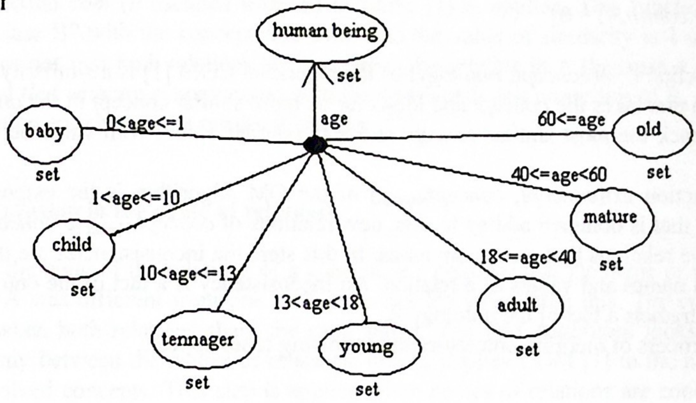


Figure 3 Graphical representation of a partition

Partitions are a form of classifying a concept, to be able to infer on this later. The inference of partitions is not included in this paper.

2.2. A concept can be a relation

The relations are represented in following form: *r* (C_{name}, C_{value})
Where, *r* represents the name of relation, C_{name} represents the name of the concept of the relation, C_{value} represents the concept value of the relation. An example is:
Mother (*Mary Ball Washington*, *George Washington*) *Mary Ball Washington* is *Mother* of *George Washington*, but *Mother* can be a concept that contains more information of the meaning of *Mother* and other concepts related to this. Other contributions exist but will no be explained in this brief space.

3. OM Algorithm for automatic merging of ontologies

Nowadays, several works that merging ontologies need the intervention the user for this important process, some of them are: [6], [9], [11], [13] and [14]. This algorithm is the unique (up to this days) that merges ontologies in an automatic form. The process to merge two ontologies, consists of the following general steps: Given 3 ontologies *A*, *B* and *C*, given concepts *a*, *b* and *c* that belongs to *A*, *B* and *C* respectively.

```
1. $a \in A$, to obtain $com(a,B)$
2. if $com(a,B) > 0$
       $b \in B$ with better $com(a,B)$
       to merge $a$, $b$ obtaining $c = ext(a,b_{relation})$
```

to obtain $c \in C$ to each pair (a,b) the resulting ontology is: $C = \{c\} \cup \{a\text{: } com(a,B) = 0\} \cup \{b\text{:} com(b,A) = 0\}$

The function *com*(*concept*, *ontology*) of the algorithm COM [1] is a similarity search function that takes the *concept* and looks for its more similar concept in the *ontology*, giving back the most similar *concept* and a *sv* (similar value) with value between 0 and 1.

The function *ext*(*concept*, $concept_{relation}$) of the OM Algorithm is the extension of concept that is obtained adding to this, new relations of $concept_{relation}$ to *concept* in *B* and those relations that are synonymous. In this step, the inconsistencies are detected between names and values of a relation. An inconsistency is a fact of the ontology *B* that contradicts a fact of the ontology *A*.

In the process of merging ontologies the following cases appear.

3.1. Verification of the arity in concept

The arity of a concept represents the number of values that the concept can take. If the concept takes only a value it is said that it is mono-valuated arity. For example, the arity of the concepts *Mother* and *Father* is mono-valuated; because a person can have a *Mother* and *Father* simultaneously.

A concept is a multi-valuated arity if this can take several values. For example, the *political position* that a person can carry out. OM Algorithm verifies the arity of concepts before copying the new relation in the resulting ontology. If this concept is a multi-valuated arity receives the new value; or else, tries to solve the problem using the Confusion [2] algorithm.

3.2. Union of a new relation

The union of a new relation in the resulting ontology implies the following:

a) The name and value of the relation in A are different from the name and value of the relation in B, that is to say; they are totally different concepts and they aren't synonymous.

b) The name and value of the relation in B are different from the name and value of the relation in A; that is to say, they are totally different concepts, they aren't synonymous.

3.3. Union of a relation with elements that are synonymous

In order to know if two concepts are synonymous, OM applies COM [1] Algorithm. This it gives back a message, a concept and a value of similarity. If the message is “Case B”, the given back concept is considered synonymous; the value of the similarity must be bigger or equal to 0.8 and minor or equal to 1.
Given two ontologies A and B to form one third C ontology, give the relation in A: *it escaped with* (*José Arcadio, a gypsy*) and the relation in B: *fled with* (*José Arcadio, a gypsy*).
The function *com* (*it escaped with, B*) of COM [1] is applied. This function gives back “Case B” with the concept *fled with* and the value of similarity is 1 and then OM does not fuse both relations but it enriches the relation in A (because *it escaped with* and *fled with* are synonymous) with the new words and properties of B, copying the relation enriched to the resulting ontology C.

3.4. Confusion in the name of relations

During the copy of the relations of a concept, it’s possible that the name of the relation in A was different from one in B more not the value from this. The confusion arises when both relations share the same value. The OM Algorithm looks for the synonymy between the names of relations; this is, applies COM [1] to the names of the involved concepts. This step is applied when names of relations are concepts. If COM [1] returns “Case B” then they are synonymous, otherwise they are not. There are other forms to find the synonymy between the names of relations, but because of lack of space they are not explained in this paper. If they are not synonymous OM solves the problem using Confusion [2]. For example:

Given a relation *r* in *A* with values: *Hydrology* (*Oaxaca, Main river of Oaxaca*).
Given a relation *r* in *B* with values: *River*(*Oaxaca, Main river of Oaxaca*)

A hierarchy of concepts is used where the names of the relations are represented. The figure 4 shows this hierarchy. In the hierarchy the number of levels is obtained. It is to say, the height of the tree is 2. The value of the Confusion [2] of using *River* instead of *Hydrology* is calculated, starting from the concept *River* and following a route up to *Hydrology*, counting the descendent levels and dividing the sum between the number of levels. In order to obtain the value of the confusion of using *Hydrology* instead of *River*, the descendent levels are added; that is to say, 1 is divided between 2. The result is 0.5.

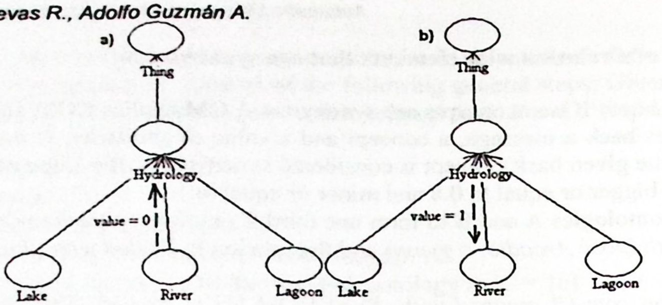


Figure 4 The Confusion of using *River* instead of *Hydrology* is 0. This is shown in a), and the Confusion of using *Hydrology* instead of *River* is 0.5. This is shown in b).

Finally OM chooses the smaller value of the confusion. In the example this is the name of relation in B; it is to say *River* (*Oaxaca, Main river of Oaxaca*).

3.5 Confusion in the value of the relations

If the Confusion [2] arises in the value of the relations, the arity of the name of relation is verified. For example:

Given the relation *r* in *A Birthplace* (*Benito Juárez, San Pablo Guelatao*)
Given the relation *r* in B: *Birthplace* (*Benito Juárez, México*)

The arity of *Birthplace* is mono-valuated, because it's not possible to be born at the same time in two different places, but the place can be specified. It's to say *San Pablo Guelatao* belongs to *Mexico*. Therefore, OM looks the synonymy of *San Pablo Guelatao* and *Mexico.* If this doesn't exist, OM apply Confusion [2], calculating firstly the height of the tree, the result is 5 according to what it shown in figure 5.

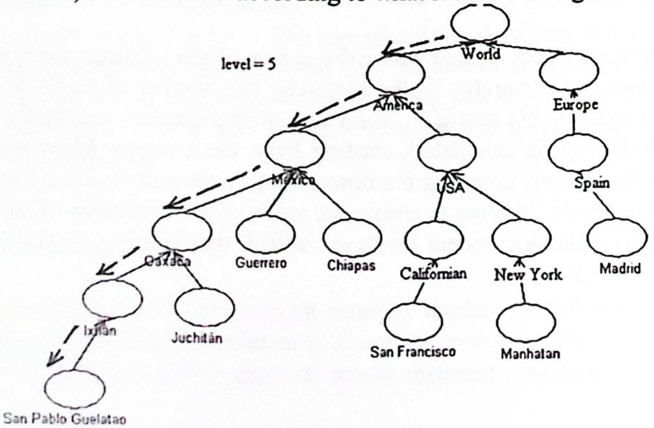


Figure 5. Graphical representation of the hierarchy that indicates the height of the tree.

Later, the value of Confusion using *Mexico* instead of *San Pablo Guelatao* is calculated, counting the number of descendent levels. The result is 3 divided between 5, obtaining the value of confusion 0.6. The value of the confusion of using *San Pablo Guelatao* instead of *Mexico* is obtained. This value is 0. Therefore OM decides to conserve the relation in A.

3.6. Union of partitions

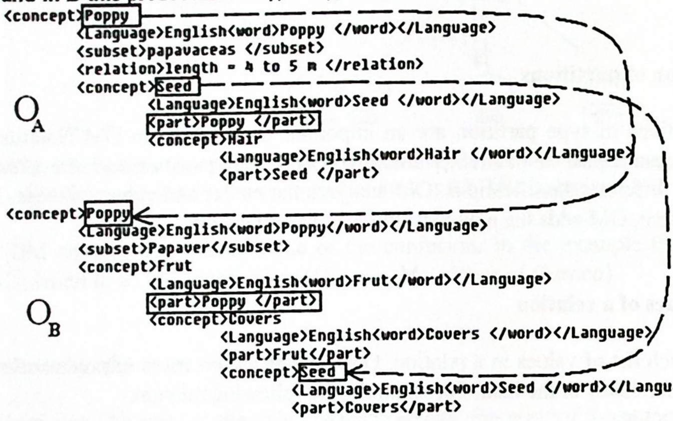
The relations of type partition are an important contribution in OM Notation. The case that can appear is: in another ontology, OM finds a partition with the same name but with different classifications. OM analyzes the ranges and values of these. If they are different, OM adds the new partition in C like a new one.

3.7. Values of a relation

Given each list of values in a relation, OM verifies the presence of predecessors who cause redundancy in the data. For example, the following relation:
<relation> visited = `Istmo`, `Salina Cruz`, `Paris`, `France`, `Africa`</relation>
The analysis consists of verifying of sequential form each one of the values of the relation eliminating the predecessors of each concept in list. In this example `Istmo` is eliminated because `Salina Cruz` is member of the `Istmo` set. It's understood that if visited `Salina Cruz` then it visited `Istmo`.
If it's wanted to fuse two relations:
<relation> visited = Istmo, Francia, Frankford</relation> in O_A.
<relation> visited = Salina Cruz, Paris, Germany</relation> in O_B.
We would think that the result of the fusion would be:
<relation> visited = Istmo, Francia, Frankford, Salina Cruz, Paris, Germany</relation>
But OM would not return that fusion, since it compares the words of each one of the values of relation, if they are different compares the synonymy, if it does not exist then applies the algorithm of the Confusion to each one of the values of the relation and chooses the minus value of the confusion to fuse the relations. In such a way that the result would be:
<relation> visited = Frankford, Salina Cruz, Paris</relation>

3.8. Verification of redundant relations

During of fusion of ontologies, redundant relations are also copied. OM avoids that in the resulting ontology, redundant relations from a concept to another are made. The redundant relations arise when three concepts in C exist. For example, $c1_c$, $c2_c$ y $c3_c$ (c is the concept and the sub-index is the ontology to which it belongs) whose relations are the following: $c1_c$ is subset of $c2_c$, $c2_c$ is subset of $c3_c$ and $c1_c$ is subset of $c3_c$; the nested relation arises in: $c1_c$ is subset of $c3_c$ and OM eliminates it of ontology C. The nested relations do not only exist in those of type <subset>, also in those of type <part> and <member>. Figure 6 shows 2 ontologies A and B that merge with each other to obtain C. The lines represent the similarity between the concept origin in A (where it leaves) towards the concept destiny in B (where it point the arrow). In the figure 6 it's possible to observe that in A the concept *Seed* have as preceding *Poppy* and in B this predecessor its *great-grandfather*.



```
<concept>Poppy
          <Language>English<word>Poppy </word></Language>
          <subset>papavaceas </subset>
          <relation>length = 4 to 5 m </relation>
          <concept>Seed
                    <Language>English<word>Seed </word></Language>
                    <part>Poppy </part>
                    <concept>Hair
                              <Language>English<word>Hair </word></Language>
                              <part>Seed </part>
<concept>Poppy
          <Language>English<word>Poppy</word></Language>
          <subset>Papaver</subset>
          <concept>Frut
                    <Language>English<word>Frut</word></Language>
                    <part>Poppy </part>
                    <concept>Covers
                              <Language>English<word>Covers </word></Language>
                              <part>Frut</part>
                              <concept>Seed
                                        <Language>English<word>Seed </word></Language>
                                        <part>Covers</part>
```

Figure 6 A and B ontologies with the relations in *Poppy* that it will generate nested relation.

In figure 7 the result of the merge in ontology C appears, where the relation nested between the concepts has been eliminated.

```
<concept>Poppy
          <Language>English<word>Poppy</word></Language>
          <subset>Papaver</subset>
          <concept>Frut
                    <Language>English<word>Frut</word></Language>
                    <part>Poppy </part>
                    <concept>Covers
    O_C                       <Language>English<word>Covers </word></Language>
                              <part>Frut</part>
                              <concept>Seed
                                        <Language>English<word>Seed </word></Language>
                                        <part>Covers</part>
```

Figure 7 Representation of an ontology without the redundant relation.

3.9. Contributions of the OM Algorithm

1. Totally automatic, requires no human intervention.
2. It handles partitions as well as subsets.
3. It handles nodes (concepts) in an ontology that is described "shallowly" by just a word, a word phrase or a set of them.
4. Relation among nodes can also be concepts (nodes, that is).
5. It detects inconsistencies (contradictions) in the knowledge in ontology A versus the knowledge in B, using inconsistency measurements [7] and confusion [2].

6. It solves some of the contradictions detected in (5), through inconsistency measurement [7].

4. Tests

Tests have been merging ontologies in the domain of geographic zones, description of animals, biographies and description of tools, products and novels such as *Cien Años de Soledad* of Gabriel García Márquez. The ontologies were obtained manually from several documents describing that described, the same topic. The obtained ontologies were merged (automatically) by OM.

The validation of results has been made manually, although we are designing an automatic validation tool.

The work to be reported is a summary of the Ph D. thesis [3] of one of the authors, and uses COM, a software [1] that, given a concept *ca* in ontology *A* finds the most similar concept *cb* in ontology *B*, as well as its *similar value*.

Conclusions

A notation has been created to design ontologies. This notation presents some improvements made to languages of ontologies that exist in the Web. We also implemented OM, an algorithm to fuse ontologies; this algorithm does not process texts but it takes care to preserve the semantic of the merged ontologies. It detects the inconsistencies during the merge and it solves them. OM makes the fusion totally automatic. This is a great improvement to the fusion algorithms that are in the Web, since they perform the fusion in a semi-automatic form. It is to say, the user in them takes part in the important points of the fusion. OM notation and OM algorithm are part of the answer to the great necessity to make that the computer, as important entry point in the Web, can accumulate knowledge and make transactions of business without human intervention.

References

1. Adolfo Guzmán and Jesus Olivares, "Finding the Most Similar Concepts in two Different Ontologies", *Lecture Notes in Artificial Intelligence* LNAI 2972, Springer-Verlag. 129-138. ISSN 0302-9743, 2004
2. Adolfo Guzmán and Sergey Levachkine, "Hierarchies Measuring Qualitative Variables", Lecture Notes in Computer Science LNCS 2945, *Computational Linguistics and Intelligent Text Processing*, Springer-Verlag. 262-274, ISSN 0372-9743, 2004
3. Alma-Delia Cuevas-Rasgado. *Ontology Merging using semantic properties* Ph. D. thesis in progress. CIC-IPN, Mexico.
4. Asunción. Pérez , and Mary Carmen Suárez, *Evaluation of RDF[S] and DAML+OIL Import/Export Services within Ontology Platforms*. LNAI 2972, 109-118. 2004

5. D. Connolly, F. van Harmelen, I. Horrocks, D. L. McGuinnes, P. F. Patel-Schneider, L. Andrea Stein, *DAML+OIL Reference description*, March 2001, W3C Note 18 December 2001, http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218
6. D. Dou, D. McDermott, and Peichen Qi. *Ontology Translation by Ontology Merging and Automated Reasing*, Yale University, Computer Science Departament New Haven, CT 06520.
7. Edith Adriana Jimenez Contreras. *Quantifying inconsistencies in sentences (facts) with symbolic values*. Ph. D. thesis in progress. CIC-IPN, México.
8. F. Manola, E. Miller. *RDF Primer. W3C Recommendation*. 10 February 2004. http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
9. G. Stumme, A. Maedche. *Ontology Merging for Federated ontologies on the semantic web*. Institute for Applied Computer Science and Formal Description Method [AIFB] University of Karlsruhe D-76128 Karlsruhe, Germany.
10. Knowledge Interchange Format. *Draft proposed*, American National Standard [dpANS] NCITS. T2/98-004.
11. L. Deborah McGuinness, R. Fikes, J. Rice and S. Wilder, *The Chimaera Ontology Environment Knowledge*, System Laboratory CommerceOne Stanford University, Stanford, CA Mountain View, CA.
12. M. K. Smith, Electronic Data System, C. Welty, IBM Research, D. L. McGuinnes, Stanford University, *OWL Web Ontology Language Guide*, W3C Recommendation 10 February 2004.
13. N. Fridman Now and A. Mark. *PROMPT: Algoritm and Tool for Automated Ontology Merging and Alignment*, Stanford Medical Informatics, Standford University, CA.
14. Y. Kalfoglou and M. Schorlemmer. *Information-Flow-based Ontology Mapping*. Advanced Knowledge Technologies [AKT] Departament of Electronics and Computer ScienceUniversity of Southamptom. Advanced Knowledge Technologies [AKT] Centre for Intelligent System and their Applications The University of Edinburgh.

Measuring Inconsistency over a Hierarchy of Qualitative Facts

Adolfo Guzmán-Arenas, Adriana Jiménez-Contreras

Centro de Investigación en Computación (CIC),
Instituto Politécnico Nacional (IPN)
07799 Ciudad de México, MÉXICO

a.guzman@acm.org
adrianaj@sagitario.cic.ipn.mx

Abstract. In this article we present a model to compute the degree of inconsistency of a particular event. This situation is described through facts from observers, where each one of them informs on a fact. Contrary to the theory of Dempster-Schafer, all the observers are equally believable since if their observations differ, it will be for those different used observation ways. That is to say, if somebody has the situation in which he wants to investigate and to determine in which transport traveled Luis, and the informants report on what they observed, they will say that he traveled by airplane, bus, train, etc.; then with the model we can specify the most likely fact. We define the way to determine the disagreement of these facts and to determine which will be the value average that adjusts better. It is adjusted to the reported facts.

Keywords: Confusion, inconsistency, fact, observer, center of gravity.

1 Introduction

In this paper we present the study of situations or circumstances of reality, related to a particular aspect, where a serie of consistent or inconsistent observations is exposed as facts that describe the event. For such study we present a model that allows finding the inconsistency in that set of facts.

The data in those facts are qualitative in nature "Jon's hair is black"; more precisely, constants (such as "black") must belong to a hierarchy [6].

The model helps us to determine the degree of inconsistency of each fact in an event, using a function (confusion) in the hierarchy of facts and another function that computes the value of the inconsistency.

The facts are obtained through observers call reporters or informants, these facts can be located over a hierarchy of facts (qualitative values), on this hierarchy, a func-

© Jesús Olivares, Adolfo Guzmán (Eds.)
Data Mining and Information Systems.
Research in Computer Science 22, 2006, pp. 15-27

tion measures the confusion that arises when we use r instead of s, the intended or correct value. For example, about the confusion of using "America" instead of "México"In summary, we study existent facts about an event.

2 Antecedents

Inconsistency is a topic intensely studied in the area of computation. For example, in databases since the integrity is highly appreciated, so that measuring the inconsistency in the data is important.

Also inconsistency in the requirements stage of the development of a system is required, since it is impossible to design a system with inconsistent requirements [3].

Other investigations on this topic are carried out in the analysis of news using Classic logic [7], to find the inconsistency of news over a particular event [1]. Also others have used the Theory of Dempster-Shafer, also known as the Theory of Functions of Beliefs, which is a generalization of the Bayesian theory of subjective probabilities, where the idea is to obtain degrees of beliefs (informants are not reliable, they may lie) for a question and to combine such degrees when they are based on independent elements of evidences. For example, we want to know the probability rained in Mexico City on May 10, 2006; if Juan said that it rained, and Pedro said that it didn't rain. A subjective probability is assigned to the reliability of each people, these events are considered as independent and they combine these degrees of beliefs to determine if it rained or not. This is another form of finding inconsistency in a particular situation [10].

In this paper we solve the following:

- Given an event (set of facts) how certain is it? (To measure the certainty). That is to say, the list of facts will allow us to determine the consistency or inconsistency of these facts, and we will also find the must likely fact, that which generates the smallest uncertainty with respect to all facts in the set.

For example, the color of Luis' hair, an observer says that it is red, others say that it is light brown, light dark, blond and black respectively; it is required to find this set of facts, as well as to determine as close as possible the true color of Luis' hair.

- We want to analyze the consistency or inconsistency of this group of symbolic facts (colors). To denote the degree of inconsistency, we use the symbol σ. We want to compute σ.

3 Motivation

You can measure the weight of an object, its length, its volume, etc. For example, to measure the length of a door, where four workers take the measurement independently, the measures being of *2m*, *2.3m*, *3m*, and *2mts*. The must likely length is obtained by taking the average of all, which is *2.32m* in our example.

On the other hand, if the measurements are not numeric, then we have observations ("facts") of the particular event. For example, four people said:

- "Pedro's sweater is red",
- "Pedro's sweater is pink",
- "Pedro's sweater is clear",
- "Pedro's sweater is orange".

What is the color that makes more sense?, how to calculate the "average" of these facts?, how to combine the observed colors to determine which is the one that more approaches to the real color?, can we measure the degree of discrepancy among each one of these facts and the most likely real color?.

To find this "average", we place the reported colors in a hierarchy of colors. In this work we present a methodology that will allow to find the average of n qualitative variables.

These logic types except the diffuse logic have only two truth value, true and false, with no shades or gradations of truthfulness or falsehood. But the real world is more complicated. There are events that are not completely true or totally false, such as "the sky is blue" or "the weather is hot". Fuzzy logic solves this and provides degrees of veracity, by requiring a membership function whose range of values is $[0,1]$.

The Theory of Dempster-Shafer takes subjective probabilities for the observers. That is, for it people have different degrees of trust (some lie more than other).

The figure 1 shows the development of how to find the fact more commendable of a set of facts over a particular event. The observers inform of facts from a particular situation, after these facts are represented in a hierarchy and we calculate the inconsistency with the Model to measure the inconsistency (MMI).

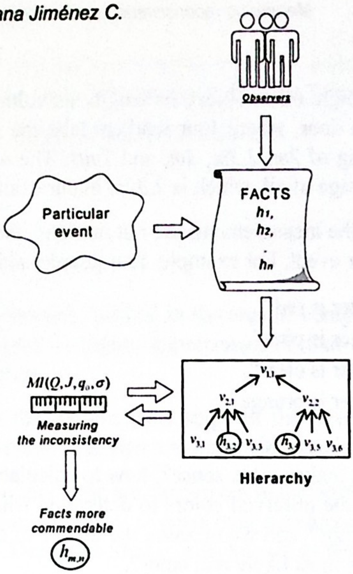


Figure 1. *Scheme to compute the inconsistency of a particular event.*

4 Previous works

Among the theories that have been dedicate to the study of the inconsistency in information, is the theory of Dempster-Shafter [10], a mathematical theory of the evidence that was introduced in the 70's and developed by Glenn Shafer and later extended by Arthur Dempster based on belief functions and commendable reasoning, which is used to combine pieces separated from information (evidences) to calculate the probability of an event.

The Theory of Dempster-Shafer is based on obtaining degrees of beliefs for a question from subjective probabilities, and combining such belief degrees when they are based on independent elements of evidences. In summary, to obtain the degree of belief, for a question (did a leaf fall in the car?) it assesses the probabilities of another question (is the testimony reliable?). The rule of Dempster begins with the supposition that the question for which it has probabilities is independent with regard to trials of subjective probabilities but this independence is only a priori; this disappears when the conflict is discerned among the different evidence elements. Contrary to Dempster-Shafer, in our work the **observers** that report on the facts have the same credibility (all say the truth) and if their facts (assertions) differ, it is due to errors or imprecisions in the observations, and not to a desire or impulse to lie. For instance, an observer saw Pedro at sunset time, so he reports "his sweater is orange", while other observer could only ascertain that "his sweater has a clear color" because the light was him.

To determine the fact with smaller inconsistency, in this work we use the hierarchies and the *Confusion* function [5], [9]. This function evaluates the similarity qualitative value *s* with regard to another *r*, both being represented in a hierarchy. For example, what is the confusion of using *dog* instead of *German Shepherd*?. We now give an example and the equations for determine the value of the function of *Confusion* (see figure 2).

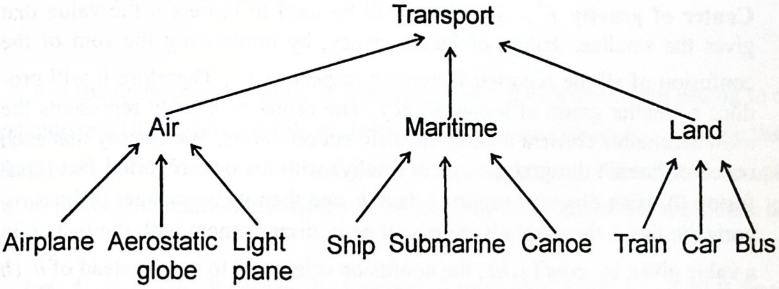


Figure 2. *Hierarchy of several types of transports for travel by air, water and land.*

The *confusion* of using *r* instead of *s*, for a hierarchy *H* (in this case the hierarchy of transports) the calculus is:

If *r*, *s* ∈ *H*, then the *confusion* in using *r* instead of *s*, written $conf(r,s)$, is:

- $conf(r,r) = conf(r,s) = 0$, when *s* is any ascendant of *r*.
- $conf(r,s) = 1 + conf(r, father\_of(s))$

To determine the confusion between *Transport* and *Air* is $conf(Transport, Air) = 1 + conf(Air, father\_of(Air)) = 1 + 0 = 1$. In this case, the confusion is *1*, because we are using *Transport* instead of *Aereo*. Due to the location in that, it is in the hierarchy and the rules of *confusion*, we travel the tree, where the upward levels the value is *0* and for each descent it will be *1*. Now then, if we obtain $conf(Air, Transport) = 0$, due to *Air* is a *Transport.*

Exemplifying the way to use hierarchies and *confusion* we will give a better vision of what is being carried out in this work, since it is a fundamental part of this.

5 Development

The Model to measure the inconsistency (MMI), finds the inconsistency of a set of facts. A value is calculated, which is interpreted as the degree of inconsistency, if it is close to zero means little inconsistency.

Definitions

- **Fact** (atomic fact): It is a measurement (numeric value) or an observation (symbolic) of an aspect (characteristic or property) of the reality. For example: (Juan, height, 1.77m), (Juan, hair's color, black).
- **Observer** (reporter, informant): - A person reporting one or more facts coming from particular event.
- **Center of gravity** r^* .- This term will be used to represent the value that gives the smallest degree of inconsistency, by minimizing the sum of the confusion of all the reported facts with respect to r^*. Therefore it will produce a smaller grade of inconsistency. The center of gravity represents the most acceptable consent among the different observers. We can say that each observer doesn't disagree (it agrees totally) with his own reported fact (confusion *0*). If an observer reports a fact h and then its newspaper or boss reports the fact *j*, then that observer will be in disagreement with the fact j in a value given by $conf(j,h)$, the confusion originated to use *j* instead of *h* (*h* was the reported by the observer). r^* is the fact *j* that minimizes the joint dissatisfaction or disagreement among the observers, or in fact, among the facts reported by the observers. r^* is the value *j* that minimizes $\sum_{i=1}^{n}(j,h_i)$, when using each reported value h_i instead of the most likely value r^*. n represents the reported observations.
- sigma σ .- It is the average of the additions of confusions. σ gives us idea of the average of dissatisfaction or disagreement that the observers have whose facts have you "summarized" reporting a single value r^* instead of $\{h_1, h_2, ..., h_n\}$. These observers reported h_i that differ something from the most likely value r^*. Each observer i has a certain dissatisfaction expressed by $conf(r^*, h_i)$. The average of those dissatisfactions is σ. n is the number of observations made from the event.

$$\sigma = \frac{\sum_{i=1}^{n}(r^*, h_i)}{n}$$

- Confusion. It has been defined in page 5 [5].

5.1 The Model to measure the inconsistency (MMI)

Let a model defined by a fourthtuple

$$(Q, J, q_0, \sigma),$$

where

Q, is a set of facts $Q = \{h_0, h_1, ..., h_n\}$.

J is the hierarchical relation of concepts, where the elements that belong to the relation are (h_a, h_b) and fulfill h_a is the immediate ascendant h_b. J' is the that fulfills the condition h_a is the immediate descendant of h_b. We will use the operator ϕ to denote the relation immediate ascendant and π to denote the immediate descendant.

Let $J^T = J \cup J'$, it defines the function of confusion $conf : J^T \rightarrow \{0,1\}$ like:

$$conf(h_a, h_b) = \begin{Bmatrix} 0\, si\, h_a\ \phi\ h_b \\ 1\, si\, h_a\ \pi\ h_b \end{Bmatrix} \qquad (1)$$

The function $asc : Q \rightarrow \{h_a \mid (h_a, h_b) \in J\}$ is defined like:

$$asc(h_a) = \begin{Bmatrix} h_b\ si\ h_a\ \phi\ h_b \\ \phi\ si\ h_a\ \not\phi\ h_b \end{Bmatrix} \qquad (2)$$

Let $conf_A : Q \times Q \rightarrow N$ the function of confusion[1] for anyone elements that belong to $Q \times Q$ is defined like:

$$conf_A(h_a, h_z) = \begin{Bmatrix} 0\, si\, h_a = h_z \\ 0\, si\, h_z = \phi \\ 0\, si\, h_z\ \pi\ h_y\ \pi \ldots \pi\ h_b, h_a \\ 1 + conf_A(h_a, asc(h_z)) \end{Bmatrix} \qquad (3)$$

Let $Q' \subset Q$ the set contains the facts of interest and Q'_p the set contains the facts of interest with more than one observation. That's to say $Q'_p = \{(h, p) \mid h \in Q', p\ is\ the\ number\ of\ observations\ over\ h\ (fact)\}$:

[1] $conf_A$ is analogous to the function $conf$ that is presented in the articles with references [6], [7].

$$conf_A^p((h_a,p_a),(h_z,p_z)) = \left\{ \begin{array}{c} 0\ si\ h_a = h_z \\ 0\ si\ h_z = \phi \\ 0\ si\ h_z\ \pi\ h_y\ \pi \ldots \pi\ h_b\ \pi\ h_a \\ 1 + conf_A(h_a, asc(h_z))[p_z] \end{array} \right\} \qquad (4)$$

ϕ is the empty set,

p_z represents the weight of h_z,

q_0 represents the highest node in the hierarchy and the beginning of this,

r^* defines the hierarchical value (fact) that minimizes the addition of the function of confusion:

$$\min \sum_{i=1}^{n} conf(r^*, h_i) \qquad (5)$$

σ is the value that represents the inconsistency. If $\sigma = 0$ then the inconsistency does not exist in the facts h_i, that they are contained in r^*, and σ is calculated like:

$$\sigma = \frac{\min(\sum_{i=1}^{n} conf(r^*, h_i))}{n} \qquad (6)$$

σ in equation (6) can be interpreted as the confusion average that minimizes r^*.

5.2 Measuring the inconsistency of a set of facts and finding the most acceptable value

We show the way to find the degree of inconsistency of a group of facts that describe a particular event, which were provided by observers. We analyze these facts with a confusion function, which helps us to compute the center of gravity of the set of facts.

That is, the fact that generates the smallest average inconsistency or which is the most believable, could be call it also the less lying or the less erroneous.

Example

We want to determine which animal is the pet of John, when the observers reported the following facts:

John has a siamese cat
John has a siamese cat
John has a feline

John has a chihuahueño
John has a dog
John has a dog
John has a Xoloitzcuintle
John has a domestic cat
John has a eagle

Once we have the list of facts, locate them in a hierarchy (the hierarchy is designed specialized according to the knowledge of some external source, it is "general knowledge"). The figure[2] 3 shows a hierarchy J, that includes the qualitative variables that were obtained of the facts, where the observations are represented by an $*$ (asterisk):

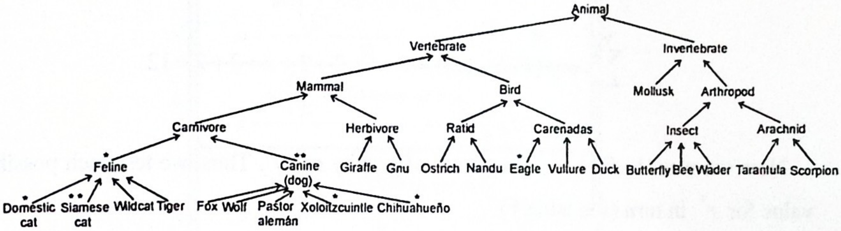


Figure 3.*Hierarchy of animals, where appear the facts from the above list.*

The set of facts is:

$$Q' = \{Feline, Domestic\ cat, Siamese\ cat, Dog, Xoloitzcuintle, Eagle, Chihuahueño\},$$

but there are two observations that represented that *John has a Siamese cat* and a *dog*, where the highest node in the hierarchy is:

$$q_0 = Animal$$

To determine the possible center of gravity, we must to calculate the confusions of *Feline* with each value in the set Q':

[2] The $conf(Feline, Siamese\ cat)$ is counted in one unit by each level that goes down in the tree from the node *Feline* to the node *Siamese cat*, the levels that ascend they don't count.

$$conf(Feline, Feline) = 0 \ (using\ Feline\ instead\ of\ Feline),$$

$$conf(Feline, Domestic\ cat) = 1 \ (using\ Feline\ instead\ of\ Domestic\ cat),$$

$$conf(Feline, Siamese\ cat) = 2 \ (using\ Feline\ instead\ of\ Siamese\ cat),$$

$$conf(Feline, Dog) = 2 \ (using\ Feline\ instead\ of\ Dog),$$

$$conf(Feline, Xoloitzcuintle) = 2 \ (using\ Feline\ instead\ of\ Xoloitzcuintle),$$

$$conf(Feline, Eagle) = 3 \ (using\ Feline\ instead\ of\ Eagle),$$

$$conf(Feline, Chihuahueño) = 2 \, (using\ Feline\ instead\ of\ Chihuahueño).$$

The sum of confusions of $Feline \times Q'$ is (For the others facts, the sum of confusions is obtained like *Feline*):

$$\sum_{i=1}^{9} conf(Feline, h_i) = 0+1+2+2+2+3+2 = 12$$

Now we want to find r^*, the center of gravity of Q'. Thus, we test each possible value for r^* in turn (see table 1).

$\sum_{i=1}^{9} conf(r^*, h_i)$
$\sum_{i=1}^{9} conf(Feline, h_i) = 12$
$\sum_{i=1}^{9} conf(Domestic\ cat, h_i) = 11$
$\sum_{i=1}^{9} conf(Siamese\ cat, h_i) = 10$
$\sum_{i=1}^{9} conf(Dog, h_i) = 12$
$\sum_{i=1}^{9} conf(Xoloitzcuintle, h_i) = 11$
$\sum_{i=1}^{9} conf(Eagle, h_i) = 29$
$\sum_{i=1}^{9} conf(Chihuahueño, h_i) = 11$

Table 1. *Candidates for gravity center of the facts Q'.*

Now, we find the value that fulfills:

$$\min\left(\sum_{i=1}^{9} conf(r^*, h_i)\right) \qquad (a)$$

The value that fulfills equation (a) is $r^* = \textit{Siamese cat}$ with $\sigma = 1.11$. This means that the most plausible value for the pet of John is Siamese cat, it is the value that minimizes the discomfort (measured by the confusion) of all the observers, which reported a value and "the real value published" was $r^* = \textit{Siamese cat}$.

In the table 2, are showed the inconsistency degrees for all the elements Q'.

$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(r^*, h_i\right)\right)}{9}$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Feline, h_i\right)\right)}{9} = \dfrac{12}{9} = 1.33$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Domestic\ cat, h_i\right)\right)}{9} = \dfrac{11}{9} = 1.22$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Siamese\ cat, h_i\right)\right)}{9} = \dfrac{10}{9} = 1.11$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Dog, h_i\right)\right)}{9} = \dfrac{12}{9} = 1.33$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Xoloitzcuintle, h_i\right)\right)}{9} = \dfrac{11}{9} = 1.22$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Eagle, h_i\right)\right)}{9} = \dfrac{12}{9} = 1.33$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{9} conf\left(Chihuahueño, h_i\right)\right)}{9} = \dfrac{11}{9} = 1.22$

Table 2. *Inconsistency degrees for the elements of Q'*

In our example, r^* turned out to be the most specific fact (the fact deepest inside the hierarchy the fact furthest away from the root). This is not always the case. If five observations (facts) reporting *Dog* would have made $r^* = Dog$ with $\sum_{i=1}^{12} conf(Dog, h_i) = 12$ and $\sigma = \dfrac{12}{12} = 1$. For the rest of facts, the degrees of inconsistency are the following:

$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(r^{\bullet}, h_i)\right)}{12}$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Feline, h_i)\right)}{12} = \dfrac{15}{12} = 1.25$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Domestic\ cat, h_i)\right)}{12} = \dfrac{14}{12} = 1.16$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Siamese\ cat, h_i)\right)}{12} = \dfrac{13}{12} = 1.08$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Dog, h_i)\right)}{12} = \dfrac{12}{12} = 1$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Xoloitzcuintle, h_i)\right)}{12} = \dfrac{14}{12} = 1.16$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Eagle, h_i)\right)}{12} = \dfrac{32}{12} = 2.66$
$\sigma = \dfrac{\min\left(\sum_{i=1}^{12} conf(Chihuahueño, h_i)\right)}{12} = \dfrac{14}{12} = 1.16$

Table 3. *Inconsistency degrees for the elements of Q'.*

Conclusions

This model allows us, (1) to find the inconsistency in a set of facts; (2) to compute the degree of inconsistency of a set of facts. In (1) and (2) are carried out using hierarchies, instead of assigning subjective probabilities to the truth (reliability) of the observers, as Dempster-Schafer does or values that in some given moment they take us away from the reality of the facts.

Therefore, we can find the most commendable fact of a particular situation and a serie of inconsistency degrees. We no longer assert "these facts are inconsistent" or "these facts are consistent", as classic logic does. Now, we can say "these facts are consistent in degree x", where $x > 0$.

An obstacle can be that more complex facts are not managed, but that will be a future work.

References

1. Byrne, Emma; Hunter, Anthony. Man Bites Dog: Looking for Interesting Inconsistencies in Structured News Reports. Department of Computer Science. University College London, Grower Street. May 29, 2003
2. Carlson, Jennifer; R. Murphy, Robin: Use of Dempster-Shafer Conflict Metric to Detect Interpretation Inconsistency. Proceedings of the 21th Annual Conference on Uncertainty in Artificial Intelligence (UAI-05), AUAI Press, Arlington, Virginia. Pages 94-104. 2005
3. Easterbrook, Steve: Learning from Inconsistency. International Workshop on Software Specifications and Design. Proceedings of the 8th International Workshop on Software Specification and Design. Page 136. 1996.
4. Gabbay, Dov; Hunter Anthony: Making inconsistency respectable 1: A logical framework for inconsistency in reasoning. In Fundamentals of Artificial Intelligence Research, volume 535 of Lecture Notes in Computer Science, pages 19-32. Springer, 1991
5. Guzmán-Arenas, Adolfo; Levachkine, Serguei. Relatedness of the Elements of Hierarchies Partitioned by Percentages. Center of Computing Research- National Polytechnie Institute CIC-IPN.Ronny Lempel, Shlomo Moran, *Optimizing result prefetching in web search engines with segmented indices*, ACM Transactions on Internet Technology (TOIT), Volume 4 Issue 1, February 2004
6. Guzmán, A., and Levachkine, S. (2004) Hierarchies Measuring Qualitative Variables. Lecture Notes in Computer Science LNCS 2945 (Computational Linguistics and Intelligent Text Processing), Springer-Verlag
7. Hunter, Anthony: Measuring Inconsistency in knowledge via Quasi-classical Models. Eighteenth national conference on Artificial intelligence, p.68-73, July 28-August 01, 2002, Edmonton, Alberta, Canada
8. Hunter, Anthony: Reasoning with Contradictory Information using Quasi-classical Logic. Journal of Logic and Computation, volume 10, No. 5, 677-703, 2000
9. Levachkine, Serguei; Guzm\'{a}n-Arenas, Adolfo; Polo-de Gyves, Victor: The semantics of confusion in hierarchies: Theory and practice. In Proceedings of the 13th International Conference on Conceptual Structures: common semantics for sharing knowledge (ICCS 05). Kassel, Germany, 2005
10. Shafer, Glenn. A Mathematical Theory of Evidence. Princeton, N. Y. Princeton University Press, 1976
11. Shafer, Glen. Rejoinders to Comments on "Perspectives on the Theory and Practice of Belief Functions". International Journal of Approximate Reasoning, volume 6, No. 3, 445-480, 1992

[illegible] these facts are consistent [illegible] consistent in degree x", where x > 0.

An obstacle can be that more complex [illegible] managed, but that will be a future work.

References

1. Byrne, Emma; Hunter, [illegible] Structured News Reports. [illegible] Gower Street, [illegible] 2003

2. Carlson, Jennifer P.; [illegible] interpretation inconsistency. Proceedings [illegible] Artificial Intelligence (IAAI-[illegible]) [illegible] 2000

3. Easterbrook, Steve; [illegible] Specifications and Design. [illegible] Specification and Design, [illegible]

4. Gabbay, Dov; Hunter, Anthony. [illegible] for inconsistency [illegible] 535 of Lecture Notes in Computer Science, [illegible] Springer, [illegible]

5. Guerrero-Arenas, [illegible] Lorenzana, [illegible] by Fernandez [illegible] with segments [illegible] ACM [illegible] Volume [illegible] Issue 1, February 2004

6. Gueniche, A., and Levachkine, S. (2003) [illegible] Lecture Notes in Computer Science [illegible] Text Processing), Springer-Verlag [illegible]

7. Hunter, Anthony: Measuring Inconsistency in Knowledge via [illegible] eighth national conference on Artificial Intelligence [illegible] Edmonton, Alberta, Canada [illegible]

8. Hunter, Anthony. Reasoning with contradictory [illegible] Journal of Logic and Computation [illegible]

9. Levachkine, Serguei; Guzmán-Arenas, Adolfo [illegible] confusion in hierarchies: [illegible] Conference on Conceptual Structures [illegible] Dresden, Germany, 2003

10. [illegible] Glenn, A Mathematical Theory of Evidence [illegible]

Data Mining

Automatic Generation of Hypotheses Using the GUHA Method

K. Ramírez-Amaro, V. Ortega-González and J. Figueroa-Nazuno

Centro de Investigación en Computación
Instituto Politécnico Nacional
Unidad Profesional "Adolfo López Mateos"
Colonia Lindavista, C. P. 07738
México D. F.
kramirezb05@sagitario.cic.ipn.mx, eortegag631@ipn.mx, jfn@cic.ipn.mx

Abstract. The GUHA method for automatic generation of hypotheses is presented in this paper. This technique is based on mathematical logics and one of its advantages is not to assume any statistical distribution between the data. With the rules generated from GUHA it is possible to automatically extract models from data. This article is focused on the study of the interacting variables which are considered as indicators of social inequality, such as potable water and electricity availability among others. Using this data the results indicate us that some variables have a higher incidence than others. Furthermore, we demonstrate that GUHA is an interesting approach for obtaining automatic models from variables. This is experimentally evaluated on variables related to social inequality.

1. Introduction

The General Unary Hypotheses Automaton (GUHA) was first introduced by P. Háyek, I. Havel and M. Chytil [1]. GUHA is a method for automatic generation of hypotheses based on empirical data. GUHA is one of the oldest methods of data mining [1]. The principle of this method is to let the computer generate and evaluate all hypotheses and select those that are interesting from the point of view of the given data and the studied problem. It is important to mention that GUHA is not a method to verify previously formulated hypotheses.

GUHA systematically finds "all interesting hypotheses" from the point of view of a specific problem based on given data. This contains a dilemma: "all" means "as many as possible", and "interesting" implies to create "not too many" rules. To cope with this dilemma, one may try systematically different GUHA procedures. Once a specific procedure has been selected, it is necessary to adjust the values of its different parameters. All the specifications and results referred in this paper were obtained using the specific procedure known as GUHA-ASSOC.

This article is focused on the study of the interacting variables which are considered as indicators of social inequality. In order to evaluate experimentally the GUHA method, we analyzed 27 different social variables measured for each of the 32

© Jesús Olivares, Adolfo Guzmán (Eds.)
Data Mining and Information Systems.
Research in Computer Science 22, 2006, pp. 31-41

states of Mexican Republic. These variables are considered as indicators of social inequality. The data base used in this analysis was taken from the INEGI (Instituto Nacional de Estadística, Geografía e Informática) [10, 11 and 12].

2. GUHA Method

GUHA is a method for automatic formulation of interesting hypotheses supported by given data, and this is done by means of computer procedures. These hypotheses express statements concerning all variables from our sample. In general, the data can not guarantee the truthfulness of such hypotheses, but offer support and make them plausible.

The data to be processed can be represented as a rectangular matrix:

$$D := \left(d_{i,j}\right)_{m \times n} \tag{1}$$

where $d_{i,,j}$ is the value of the j-th attribute for the i-th object. Thus, the rows of matrix D correspond to the objects belonging to our sample and each column stand as a variable of interest, e.g., objects may be the states and attributes may be social variables.

It is important to keep in mind that GUHA produces multifactorial hypotheses. Therefore, these hypotheses express relations among single variables, pairs, triples, quadruples and further; and not only one-on-one relations.

3. General Procedure of GUHA

The hypotheses in GUHA-ASSOC exhibit the following structure "*A~S*" (properties of *A* are associated with *S*), e.g., smoking and cancer; where "~" stands as some notation of association for generalized quantifiers. *A* is called *antecedent* and *S* the *succedent* of the statement "*A~S*". A special case of association *A~S* is the implication of the form *A→S* ("*A* makes *S* likely"). Therefore, implicational quantifiers in some sense estimate the conditional probability *P(S |A)* [4].

Each generated hypothesis is evaluated as a statement on the data matrix. If the processed data matrix has no missing data items[1], each pair *A* and *S* produces the corresponding four-fold table:

Table 1. Four-fold table

Variable	S	¬S	Total
A	a	b	r:= a+b
¬A	c	d	s:= c+d
Total	k:= a+c	l:= b+d	n

[1] There exist some special considerations for handling missing values in GUHA; for more information see [5].

where a, b, c and d are the observed frequencies, defined as follows:

a:= *Freq(A & S)*; the number of objects in the data satisfying both A and S;
b:= *Freq(A & ¬S)*; satisfying A but not satisfying S;
c:= *Freq(¬A & S)*; not satisfying A but satisfying S;
d:= *Freq(¬A & ¬S)*; not satisfying both A and S;

and n is the number of objects in our data base, such that $n= a+b+c+d = k+l = r+s$.

On given data, each pair of boolean attributes (A, S) determines its own four-fold frequency table; the association of A with S is defined by choosing an associational quantifier "~".

GUHA uses generalized binary quantifiers which are sometimes referred to as "operators". The semantics of quantifiers are given by their *associated functions*: for each quantifier "~", there is an associated function $\mathrm{Tr}_{\sim}$ with the values 0 or 1. The associated functions operate on the frequencies of the different objects satisfying or not the given statement. For example, the associated function of the quantifier ∀ yields *1* if all the objects satisfy the statement; otherwise its value is 0.

There are several types of associational quantifiers, among them: *implicational quantifiers* (e.g., FIMPL) which formalize the association "many A are S"; *comparative quantifiers* (e.g., SIMPLE) which express the association "A makes S more likely, than ¬S does"; *symmetric associational quantifier* such as CHI-SQUARE $\sim^{\chi^2}_{\alpha}$, corresponding to the χ^2 asymptotic test of independence in four-fold tables with the significance level α [3]. Some quantifiers just express observations on the data; and some others serve as tests of statistical hypotheses with unknown probabilities.

Since we are interested in symmetric associations, the quantifier used in this article is CHI-SQUARE test (χ^2). The associational quantifiers are symmetric if satisfy the following:

If *(a,b,c,d)* is a four-fold table, $Tr_{\sim}(a,b,c,d) = 1$, then $Tr_{\sim}(a,c,b,d) = 1$

where "~" is the quantifier with the associated function $Tr_{\sim}$[4].

This quantifier has two input parameters: s and α, where s is the number of valid hypotheses and α is the level of significance.

Considering the input values of the associational quantifier CHI-SQUARE: $s \geq 2$, $\alpha \in (0,\ 0.5]$, an hypothesis is valid iff satisfies the conditions (2), (3) and (4) consecutively:

$$a \geq s \tag{2}$$

$$ad > bc \tag{3}$$

$$\chi^2 = \frac{n(ad-bc)^2}{k \cdot l \cdot r \cdot s} \geq \chi_1^2(1-2\alpha) \tag{4}$$

where $\chi_1^2(1-2\alpha)$ is the $1-2\alpha$ quantile of the χ^2 distribution with one degree of freedom [6]. Otherwise, the hypothesis is not considered valid.

Matrices with some missing data items can be processed. The user can choose one of three possible techniques for treatment of missing information: secured (the default choice), deleting or optimistic. These three possibilities in fact give triple meaning to a sentence *A~S* in a data matrix *D* with incomplete information. For more details see [7]. In this article data with missing items are not considered.

Now we briefly describe the GUHA-ASSOC procedure working with associational quantifiers such as CHI-SQUARE. The application of the procedure takes place in three main steps:

- *Preprocessing* – In this first step there is needed to define the following:
  - the data matrix,
  - it is necessary to establish whether a variable is considered as a antecedent or as a succedent and to define which rules of inference are to be selected
  - parameters determining syntactic form of antecedents and succedents to be generated,
  - minimal and maximal length of antecedents/succedents (number of literals),
  - the quantifier and its parameters,
  - preparing the internal representation of the data matrix in a suitable form for a quick generation and evaluation of hypotheses.

  For a whole comprehension on this step see Fig 1a and 1b.
- *Processing* – The main program produces all associations *A~S* satisfying the syntactic restrictions. The evaluation of these associations is supervised avoiding exhaustive search; hence, a group of "interesting" rules is produced. Semantics of hypotheses is determined by the selection of a quantifier and its parameters see Fig. 1c and 1d.
- *Postprocessing* – The output is formed by all generated hypotheses that have been found true and are not immediate consequences of previously found hypotheses. See Fig. 1e.

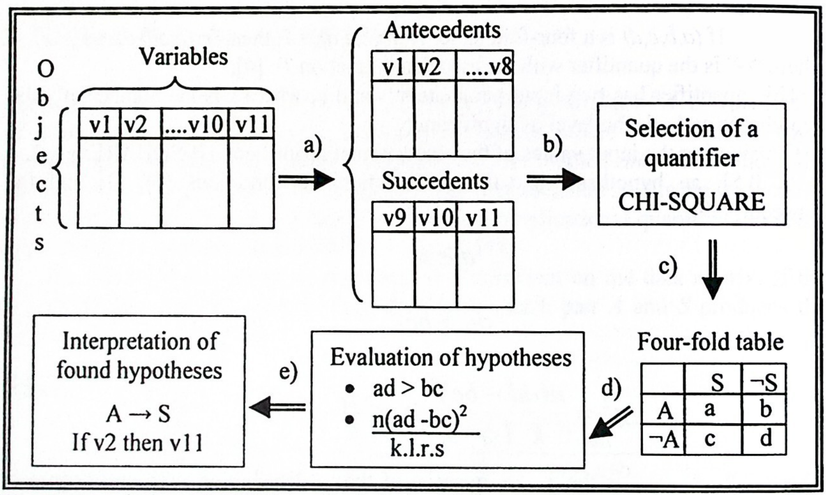


Fig. 1. Description of GUHA method. In the figure are shown the main stages of the GUHA method where the input is the matrix D and the output are the found hypotheses.

4. Experimental Analysis with Social Variables

There are many underlying factors for social inequality. Among them are: labour market, mortality rate, education, race, gender, culture, wealth accumulation, and development patterns. Social inequality refers to disparities in the distribution of economic assets and incomes.

As we saw in section 2, the data to be processed is represented as a rectangular matrix *D*, where the objects (rows) correspond to the 32 states of the Mexican Republic and the attributes (columns) stand as the social variables which are 27 [10, 11 and 12]. From these 27 variables (see Table 2), we define the first 24 as the antecedents and the last three as succedents. This division between antecedents and succedents is based on the literature [9]. After that, for each variable the objects are grouped into ten intervals according to their values. Subsequently, we select the quantifier CHI-SQUARE and its input parameters: $s = 2$ and $\alpha = 0.05$.

Table 2. Social Variables. The white boxes refer to antecedents and the gray ones to succedents.

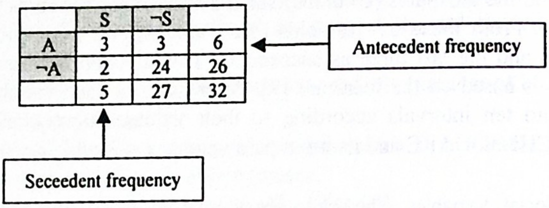
% Urban Population	Number of libraries	Total population (in thousands)	Annual rate of growth	% Illiteracy population
% Population without drainage	% Population without electricity	% Population with ground floors	% Population that speaks Indigenous languages	% Population without potable water
% Internal product	% Immigration rate	% Woman population earning up to 2 minimal wages	% Man illiteracy population	% Man population earning up to 2 minimal wages
% Emigration Rate	Index of corruption	Density of Population	% Woman illiteracy population	% Population up to 3rd grade of primary education
% Population up to 6th grade of primary education	% Population up to 1st grade of secondary education	% Population up to 3rd grade of secondary education	% Population beyond high school education	
Infant mortality rate	Degree of marginalization	Extreme poverty		

The next step is to generate the hypotheses and their corresponding four-fold table, for example:

Antecedent (*A*): *%Urban_popul* (53.9, 65.4]
Succedent (*S*): *Inf_mortality_rate* [5.9, 10]

Where () – defines an open interval and [] - defines a closed interval.

The following four-fold table is obtained from the data matrix and the antecedent and succedent previously defined:



	S	¬S	
A	3	3	6
¬A	2	24	26
	5	27	32

Antecedent frequency
Secedent frequency

As we can see there are three hypotheses that satisfying both *A* and *S*, i.e. three states of the Mexican republic satisfy if *%Urban_popul* (53.9, 65.4] then *Inf_mortality_rate* [5.9, 10].

The whole table indicate us that the hypotheses generated from the previous antecedent and succedent has a confidence of the 50% due to the fact that there are three objects that satisfy both *A* and *S*, but on the other hand there are three other objects that satisfy *A* but not *S* from the total of antecedents. The hypothesis that give us a remarkable information is the first one (*A~S*) with 3/6 of meaning, which is equivalent to 50% of confidence.

5. Experimental Results

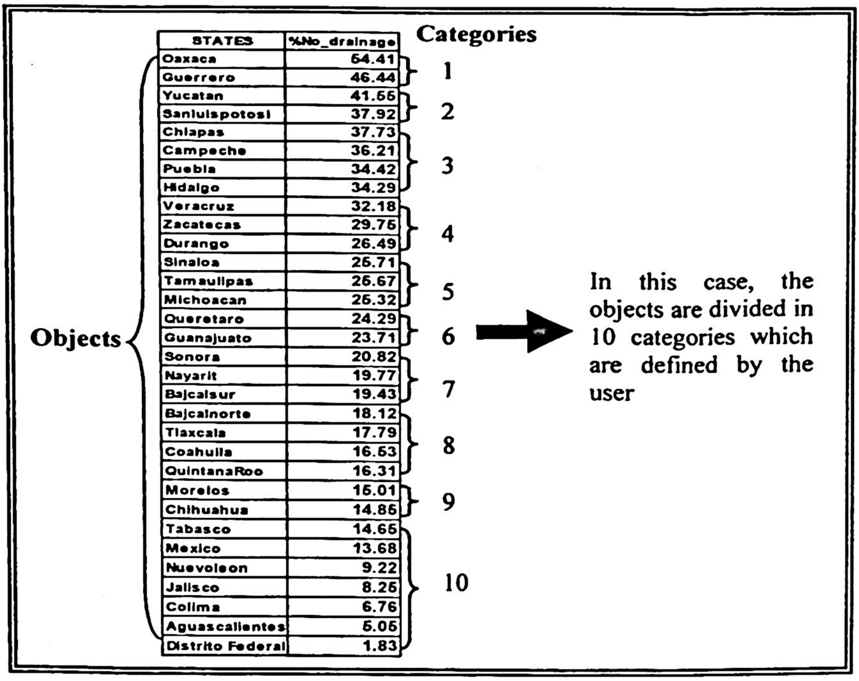
The objects re divided in ten intervals for each antecedent and succedent variables; each group was labeled for a better interpretation of the results. The categorization could be defined by the user or the program can do it by itself in two different ways: equidistant or equiprobable. In this analysis we decided to manually categorize the objects trying to conserve together those with similar properties. Fig. 2 can be seen as an example of the categorization for the %No_drainage variable.

After that, the ten categories are labeled as shows the Table 3.

Table 3. Labels for the Categories

1. Extremely low	2. Very low	3. Low	4.Middle low	5. Medium1
6. Medium2	7. Middle high	8. High	9. Very high	10. Extremely high

The results will be explained in the following subsections, they are divided in three parts due to the fact that we have previously defined three succedents. These hypotheses were selected arbitrarily among those labeled with the highest and lowest levels (corresponding to categories 1, 2 or 3 and 8, 9 or 10).



STATES	%No_drainage	Categories
Oaxaca	54.41	1
Guerrero	46.44	
Yucatan	41.55	2
Sanluispotosi	37.92	
Chiapas	37.73	3
Campeche	36.21	
Puebla	34.42	
Hidalgo	34.29	
Veracruz	32.18	4
Zacatecas	29.75	
Durango	26.49	
Sinaloa	25.71	5
Tamaulipas	25.67	
Michoacan	25.32	
Queretaro	24.29	6
Guanajuato	23.71	
Sonora	20.82	7
Nayarit	19.77	
Bajcalsur	19.43	
Bajcalnorte	18.12	8
Tlaxcala	17.79	
Coahuila	16.53	
QuintanaRoo	16.31	
Morelos	15.01	9
Chihuahua	14.85	
Tabasco	14.65	10
Mexico	13.68	
Nuevoleon	9.22	
Jalisco	8.25	
Colima	6.76	
Aguascalientes	5.05	
Distrito Federal	1.83	

In this case, the objects are divided in 10 categories which are defined by the user

Fig. 2. Example of the categorization of the variable %No_drainage.

As we mentioned in section 3, although an hypothesis satisfies the conditions (2), (3) and (4), GUHA does not guarantee its truthfulness. Some causes for this sort of situations could be:

- the quantity of objects from the matrix is insufficient,
- the categorizations are not appropriate,
- the confidence rate is less than 50% , etc.

In the following results it is possible to observe some hypotheses affected by one or a combination of the former causes.

5.1 Results of the succedent Infant Mortality Rate

The GUHA method generates 317 valid hypotheses from the succedent infant mortality rate. The most significant hypotheses are the following:

A) $A = Men\_2wage$ (low) → $S = Inf\_death$ (very low)

	S	¬S	
A	2	1	3
¬A	3	26	29
	5	27	32

Confidence = 2/3= 66.6%

The next step is to generate the hypotheses and their corresponding four-fold table, for example:

Antecedent (*A*): *%Urban_popul* (53.9, 65.4]
Succedent (*S*): *Inf_mortality_rate* [5.9, 10]

Where () – defines an open interval and [] - defines a closed interval.

The following four-fold table is obtained from the data matrix and the antecedent and succedent previously defined:

	S	¬S	
A	3	3	6
¬A	2	24	26
	5	27	32

Antecedent frequency

Sececdent frequency

As we can see there are three hypotheses that satisfying both *A* and *S*, i.e. three states of the Mexican republic satisfy if *%Urban_popul* (53.9, 65.4] then *Inf_mortality_rate* [5.9, 10].

The whole table indicate us that the hypotheses generated from the previous antecedent and succedent has a confidence of the 50% due to the fact that there are three objects that satisfy both *A* and *S*, but on the other hand there are three other objects that satisfy *A* but not *S* from the total of antecedents. The hypothesis that give us a remarkable information is the first one (*A*~*S*) with 3/6 of meaning, which is equivalent to 50% of confidence.

5. Experimental Results

The objects re divided in ten intervals for each antecedent and succedent variables; each group was labeled for a better interpretation of the results. The categorization could be defined by the user or the program can do it by itself in two different ways: equidistant or equiprobable. In this analysis we decided to manually categorize the objects trying to conserve together those with similar properties. Fig. 2 can be seen as an example of the categorization for the %No_drainage variable.

After that, the ten categories are labeled as shows the Table 3.

Table 3. Labels for the Categories

1. Extremely low	2. Very low	3. Low	4.Middle low	5. Medium1
6. Medium2	7. Middle high	8. High	9. Very high	10. Extremely high

The results will be explained in the following subsections, they are divided in three parts due to the fact that we have previously defined three succedents. These hypotheses were selected arbitrarily among those labeled with the highest and lowest levels (corresponding to categories 1, 2 or 3 and 8, 9 or 10).

If the percentage of urban population and the immigration rate are extremely low *then* the degree of marginalization is very high.

F) *A = %No_potable & %No_drainage & %No_electricity & %Ground_floors & %Illiteracy_P* (Extremely high) → *S = Marginal* (very high)

	S	¬S	
A	3	1	4
¬A	3	25	28
	6	26	32

Confidence = 3/4= 75%

	S	¬S	
A	2	0	2
¬A	4	26	30
	6	26	32

Confidence = 2/2= 100%

	S	¬S	
A	3	1	4
¬A	3	25	28
	6	26	32

Confidence = 3/4= 75%

	S	¬S	
A	3	0	3
¬A	3	26	29
	6	26	32

Confidence = 3/3= 100%

	S	¬S	
A	3	0	3
¬A	3	26	29
	6	26	32

Confidence = 3/3= 100%

If the percentage of population without potable water and the percentage of population without drainage and the percentage of population without electricity and the percentage of population with ground floors and the percentage of illiteracy population are extremely high *then* the marginalization grade is very high.

G) *A = %No_drainage & %No_electricity & %Ground_floors &* %IlliteracyP (extremely low) → *S = Marginal* (very low)

	S	¬S	
A	2	5	7
¬A	1	24	25
	3	29	32

Confidence= 2/7= 28.5%

	S	¬S	
A	2	0	2
¬A	1	29	30
	3	29	32

Confidence= 2/2= 100%

	S	¬S	
A	3	1	4
¬A	0	28	28
	3	29	32

Confidence= 3/4= 75%

	S	¬S	
A	3	0	3
¬A	0	29	29
	3	29	32

Confidence= 3/3= 100%

If the percentage of population without drainage and the percentage of population without electricity and the percentage of population with ground floors and the percentage of illiteracy population are extremely low *then* the degree of marginalization is very low.

H) *A = %Urban_p & %GIB* (extremely high) → *Marginal* (very low)

	S	¬S	
A	3	0	3
¬A	0	29	29
	3	29	32

Confidence= 3/3= 100%

	S	¬S	
A	2	1	3
¬A	1	28	29
	3	29	32

Confidence= 2/3= 66.6%

If the percentage of urban population and the gross internal product are extremely high *then* the degree of marginalization is very low.

5.3 Results of the succedent Extreme Poverty

The GUHA method generates 317 valid hypotheses from the succedent Extreme Poverty. The most significant hypotheses of this succedent are the following:

I) *A* = *%No_potable* & *%Ground_floors* & *%Men_2wage* & *illiteracy_M* & *illiteracy_W* & *P3_Sec* & *%No_drainage* (extremely low) → *S* = *E_poverty* (extremely low)

	S	¬S	
A	3	1	4
¬A	1	27	28
	4	28	32

Confidence = 3/4= 75%

	S	¬S	
A	2	2	4
¬A	2	26	28
	4	28	32

Confidence = 2/4= 50%

	S	¬S	
A	2	1	3
¬A	2	27	29
	4	28	32

Confidence = 2/3= 66.6%

	S	¬S	
A	2	1	3
¬A	2	27	29
	4	28	32

Confidence = 2/3= 66.6%

	S	¬S	
A	2	1	3
¬A	2	27	29
	4	28	32

Confidence = 2/3= 66.6%

	S	¬S	
A	2	2	4
¬A	2	26	28
	4	28	32

Confidence = 2/4= 50%

	S	¬S	
A	3	4	7
¬A	1	24	25
	4	28	32

Confidence = 3/7= 42.8%

If the percentage of population without potable water and the percentage of population with ground floors and the man population earning up to two minimal wages and the percentage of man illiteracy population and the percentage of woman illiteracy population and the population up to 3rd grade of secondary education and the percentage of population without drainage are extremely low *then* the extreme poverty is extremely low.

J) *A* = *%No_potable* & *%No_electricity* & *Illiteracy_M* (extremely low) → S = *E_poverty* (extremely high)

	S	¬S	
A	2	2	4
¬A	1	27	28
	2	29	32

Confidence = 2/2= 50%

	S	¬S	
A	2	2	4
¬A	1	27	28
	3	29	32

Confidence = 2/4= 50%

	S	¬S	
A	2	1	3
¬A	1	28	29
	3	29	32

Confidence = 2/3= 66.6%

If the percentage of population without potable water and the percentage of population without electricity and the percentage of man illiteracy population are extremely low *then* the extreme poverty is extremely high.

6. Conclusions

By means of GUHA method we are able to obtain several hypotheses which relate different variables (in this specific case social inequality variables).

For each hypothesis we compute a percentage of confidence. Using this measure is possible to define if the hypotheses can be consider as truth or could be rejected, e.g., the hypotheses D states "*If* percentage of population without drainage is extremely low *then* infantile mortality rate is very high". As we can see, this particular hypothesis make no sense, and if we evaluate the value of its *% of confidence* we can see that shows a very low value compared with the other hypotheses. Thus, we define this hypothesis as non reliable.

On the other hand, the hypothesis E "*If* the percentage of urban population and immigration rate are extremely low *then* the degree of marginalization is very high" shows the highest reliability and can be considered to model some of these social variables.

Using the hypotheses with higher reliability we can design models that allow us to characterize these sorts of phenomena. Therefore, the GUHA method implicitly gives rise to these important models through the determination of specific rules with any assumption of statistical distribution between the data.

References

1. P. Hájek, I. Havel, and M. Chytil, *"The GUHA method of automatic hypotheses determination"*, Computing, no. 1, pp. 293–308, 1966.
2. R. Agrawal, H. Manilla, R. Sukent, A. Toivonen, and A. Verkamo, *"Fast Discovery of Association Rules"* in Advance in Knowledge Discovery and Data Mining, AAA Press, 1996, pp.307-328.
3. P. Hájek, M. Holeňa, J. Rauch, *"The GUHA method and foundations of (relational) data mining"*, Springer 2003, pp. 17-37.
4. P. Hájek, A. Sochorová, and J. Zvárová, "GUHA for personal computers", *Comp. Stat. Data Anal.*, no. 19, pp. 149–153, 1995.
5. P. Hájek, "Briefly on the GUHA method of data mining", Journal of Telecommunications and Information Technology, no. 3, pp. 112-114, 2003.
6. P. Hájek, T. Havránek, *"Mechanizing hypothesis formation- Mathematical foundations for a general theory"*, Springer Verlag, Heidelberg, 1978.
7. P. Hájek, *"The new version of the GUHA procedure ASSOC (generating hypotheses on associations)"*, Mathematical Foundations, COMPSTAT 1984, Proceedings in Computational statistics, Physica-Verlag, Wien, pp. 360-365.
8. T. Havránek, *"The statistical modification and interpretation of the GUHA method"*, Kybernetika 7, 1971, pp.13-21.
9. Documentación técnica de los indicadores Sociodemográficos, Archivo de metadatos, Consejo Nacional de Población Press., Diciembre de 2005. (www.conapo.gob.mx/)
10. INEGI. XII Censo General de Población y Vivienda 2000. Resultados preliminares. México, 2000.
11. INEGI. Estados Unidos Mexicanos. XII Censo General de Población y Vivienda, 2000. Tabulados Básicos. Tomo I. Aguascalientes, Ags., México, 2001.
12. INEGI. Encuesta Nacional de Ocupación y Empleo 2005, Indicadores estratégicos. Aguascalientes, Ags., 2005.

Hierarchical Fuzzy Logic Medical Database with Decision Algorithm for Metabolic Syndrome Diagnosis

I. Salgado [(1)] **, A. Rodríguez** [(1)] **, J. Chairez** [(1)] **, A. Zúñiga** [(2)] **, P. Santillan** [(2)] **.**

[(1)] Unidad Profesional Interdisciplinaria de Biotecnología del IPN.

[(2)] Instituto de Ciencias Médicas y Nutrición "Salvador Zubirán"

Email: jchairez@ctrl.cinvestav.mx

ABSTRACT

Probably, the most complicated aspect in medical care is an efficient and opportune diagnostic from the Medic. Few years ago this branch of medical attention was depending only on physician's experience and his knowledge, who were dedicated to the patients care. However, this practice has been changed since Data Base (DB) related and intelligent systems (like experts systems) have been used. In this document a Related DB for Nutriology Clinician Department of *Instituto Nacional de Ciencias Médicas y Nutrición (INCMNSZ),* is shown, whose objective is to classify the patient's information for a better clinical state visualization to be gotten. An appropriate designation for an operational risk scale that indicates the attention degree that must be applied in sick persons will be determinate by the mixed application of fuzzy logic and relational database. The information system response is based in a technique called Fussy Logic, which is applied in medical variables analysis in order to find the existence of Metabolic Syndrome in the patient, and in a direct test about their life quality, alimentation habits and Clinic History. The connection between the DB and Fussy System was created with Matlab and the ODBC Data Transfer System.

Introduction

The clinical record is the most important element in the medical treatment that can be applied to a patient by a physician, due the information included into it. Roughly speaking, the data set contained in the corresponding patient file, is constructed by the answers (given by a specific patient) to specific question designed by an expert medical group. The questions are related to the patient's health conditions, the sickness events over the entire life, a short version of the patient's family medical record, lifestyle, some relevant accidents, surgery interventions, allergies, vaccination history and any current medicine intake by the patient [1].

These groups of data are one of the most important aspects for the medic to take decisions in a correct way, diagnose efficiently the illness' patient and propose a solution to him. In this paper the CH is a key to follow the patients who are candidates to suffer *Metabolic Syndrome*. The metabolic Syndrome is defined by the OMS as an altered regulation of glucose or diabetes (it means an insulin unresponse that is defined as glucose make out under the last quartile levels for population in the study), besides 2 or more of the following components:

- High blood pressure.
- A triglyceride level above 150 mg/dl
- A High density lipoprotein level (HDL)
- Obesity or high IMC.

The exact cause of metabolic syndrome is unknown. Most researchers believe it is caused by a combination of the genetic makeup and lifestyle choices, including the types of eaten food and the level of physical activity. If the metabolic syndrome is diagnose, the body suffers a series of biochemical changes. Over time, these changes lead to the development of one or more associated medical conditions. The sequence begins when insulin, a hormone excreted from your pancreas, loses its ability to make your body's cells absorb glucose from the blood-your body uses glucose for energy. When this happens, glucose levels remain high after you eat. Your pancreas, sensing a high glucose level in your blood, continues to excrete insulin. Loss of insulin production may be genetic or secondary to high fat levels with fatty deposits in the pancreas [2]. Until today, the SM's diagnose is given for the Nutriology medic's experience. Actually an automatic or semiautomatic system, that can give us an appropriate and exactly diagnose, doesn't exist because each factor of SM has a dissimilar relative weigh and affect in a different way the metabolic patients state, which can't be so solved whit classic programming methods.

The Fuzzy Logic (FL) can associate uncertainly o eventually variables whit a mathematic function, which is called Membership Function, obtaining an associate grade inside a range between 0 and 1; that is one of the differences whit the classical binary theories, because the only can give two different values to the variables. The FL is conformed for four steps: Fuzzification, rules base, inference mechanism and defuzzification [3]. The FL is used in a lot of applications, like an uncertainly models controller, expert systems and decision systems whit partially defined variables. This is making whit an input's review, based in the *modus-ponens* structure: *condition-action*. An example applied to this work is the following:

If ... the patient has a high glucose level... so the patient has diabetes

These kinds of sentences are going to use to determinate SM and its related illness.

Methodology

The Methodology is described for the following steps:

I. Design of the Nutriology Department's Data Base

Microsoft Access was used to design de Data Base because it has an easy form to development the elements (tables, forms, queries, etc) of it. MatLab and the OBDC Data System were the tools to export and import data from the Data Base to the fuzzy system, which were done in Matlab. The DB has eight areas: *I.* Diagnose, *II.* Clinician Information, *III.* Life Style, *IV.* Poverty, *V.* Habits, *VI.* Metabolic Syndrome, *VII.* ECD and Comorbility, *VIII.* Nutritional Evaluation.

II. *Data export and import*

Matlab has a different kind of tools to link it whit external software, in other words matlab can import data from access to make an analysis of the DB. "*Visual Query Builder*" is software to take the information inside the DB tables and show it in a matrix form in MatLab.

The program code to import de information from the DB is the as follows:

```
1. logintimeout(10);
2. conn = database('bd2', '', '');
3. ping(conn);
4. curs = exec(conn, 'select * from prueba');
5. setdbprefs('DataReturnFormat','numeric');
6. curs = fetch(curs);
7. z = curs.Data;
```

Description of the code Line1. Specify the time connection to Matlab before send an error message. Line 2 and 3. "database" is the function to make the link between the DB and Matlab, the function contains the DB's name, user name and password The line 3 specifies the connection state. Line 4, 5, 6 and 7. The fields in a table are chosen whit the function shown in line 4. The Data Format Return are given by the function setdbprefs.

The program code to export information from Matlab to the DB is the following,:

```
1. setdbprefs('DataReturnFormat','cellarray');
2. d='MALA';
 3. exdata={d};
4. colnames(i) = {'calvida'};
5. insert(conn, Prueba, Colnames, exdata)
```

Line 1. Specify the format to return the data. Line 2 and 3. These lines show the data thal wll be export to access. Line 4. Shows the name of the DB field. Line5. To export the information to access is used the insert function which inserts a new line into the DB.

III. *Pre- Diagnostic fussy System Development*

A specialist medic and a lot of bibliography sources were consulted to make a correct diagnose, in special the National Cardiology journal was read and the diagnose algorithm was taken from this journal. This algorithm is shown in the picture 1 and describes the steps to follow to find evidence of SM. The diagnose standards are explained in the figure 1. The fuzzy system only analyzed the part in bold of the algorithm in figure 1, the variables to get involved used the parameters shows in table 1.

The figure 1 shows the diagnostic algorithm. The variables included in the MS description and its corresponding range is depicted in Table 1. The parameters defining the discourse universe are shown in the same table.

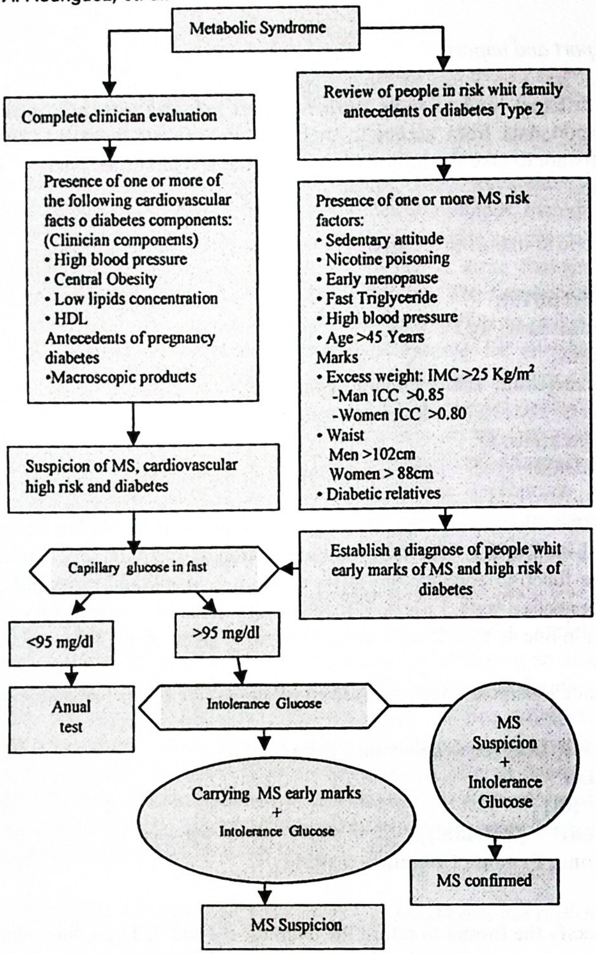

Metabolic Syndrome
Complete clinician evaluation
Review of people in risk whit family antecedents of diabetes Type 2
Presence of one or more of the following cardiovascular facts o diabetes components: (Clinician components)
• High blood pressure
• Central Obesity
• Low lipids concentration
• HDL
Antecedents of pregnancy diabetes
•Macroscopic products
Presence of one or more MS risk factors:
• Sedentary attitude
• Nicotine poisoning
• Early menopause
• Fast Triglyceride
• High blood pressure
• Age >45 Years
Marks
• Excess weight: IMC >25 Kg/m²
-Man ICC >0.85
-Women ICC >0.80
• Waist
Men >102cm
Women > 88cm
• Diabetic relatives
Suspicion of MS, cardiovascular high risk and diabetes
Establish a diagnose of people whit early marks of MS and high risk of diabetes
Capillary glucose in fast
<95 mg/dl
>95 mg/dl
Anual test
Intolerance Glucose
MS Suspicion + Intolerance Glucose
Carrying MS early marks + Intolerance Glucose
MS confirmed
MS Suspicion

Figure 1. Diagnose algorithm by

Following the WHO report (1998), the MS is componed by the following clinic settings *
Systolic Arterial Preassure ≥ 140 mmHg
Dyastolic Arterial Preassure ≥ 90 mmHg
ICC ≥ 0.85 cm for men and 0.80 cm for women
IMC ≥ 25 kg/m^2
Triglycerids ≥ 200 mg/dl
HDL < 35 mg /dl for men and 45 mg /dl for women

Table 1.

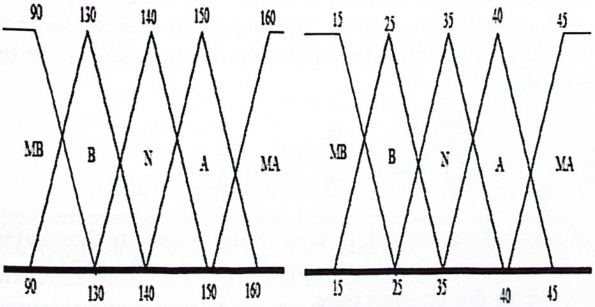
These are diagnostic criterion: It is considered that an abnormal fasting condition and with high glucose concentration, glucose intolerant or affected by 2-type diabetes mellitus had MS if and only if there is any possible association with both components enlisted above.

Fuzzy Algorithm Developement

There were designed 3 different fuzzy pre-diagnostic computer programs. Each one of them is dedicated to one of the following clinical conditions: low lipids concentration, obesity and hypertension. These three belong to the, so-called overall clinic evaluation where 6 different variables are used: Triglycerides concentration, HDL, IMC, ICC, SAP, DAP. It is important to mention that the designed program don't use all the mentioned variables in the Cardiology Journal because until this moment there is not a complete pre-diagnostic method. However, there are another illnesses associated to the considered variables that will be related with a specific health problem and can be added to the program after a hard analysis derived by medical protocols driven at INCMNSZ.

The automatic decision system (based on hierarchal fuzzy logic) is based on the classical fuzzy controller, i.e. the variables used in the fuzz pre-diagnostic system includes an output signal describing the possible sickness that is affecting to the patient, the membership function selected for each linguistic input and output variable (for the sake of simplicity, they were chosen as triangular, S and Z shapes as usual in mostly of fuzzy designs), the range assigned for each function. The fuzzy system designed to treat arterial pressure does not consider just the problems associated to high pressure, but the decreased arterial pressure too. So, it is possible to define a complete study on this important element into the health conditions given for the patient. Besides, a specific fuzzy controller to analyze and consider low lipids concentration concept has been implemented. This special fuzzy arrangement uses the triglycerides and the HDL concentration to define the inference result, and then to provide a correct diagnostic about the illness presence in the attended patient.

Inside the correct MS diagnose, the obesity is really important to specify the patient's risk due the suffered overweight. In particular, these fuzzy relationships are based on two simple concepts: a) the relation or the, so-called, hip-waist index and b) the corporal mass index. These both concepts are so useful to define in a unique form the obesity condition on the patient. An example of the fuzzy system process to diagnose one of the three illnesses mentioned (***Low Lipid Concentration***) is shown by the following membership functions (Figure 2) and its corresponding knowledge fuzzy matrix (Table 2).



The linguistic terms make reference to the variables concentration:

Very Low	(MB)
Low	(B)
Normal	(N)
Hight	(A)
Very High	(MA)

Figure 2. Membership functions for triglyceride and HDL

The fuzzy matrix to compare both clinician aspects gives us de rules to diagnose the illness.

Low Lipid Concentration		HDL				
		MB	B	N	A	MA
Triglycerides	MB	P	P	N'	N	N
	B	P	P	N	N	N
	N	P	N	N	N	N
	A	V	V	V	P	P
	MA	V	V	V	V	P

Table 2. Knowledge Fuzzy Matrix

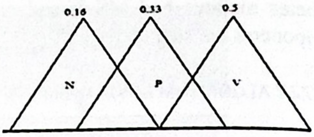


Figure 3. Output Membership Functions

The membership functions (Input and Output) and the matrix were suggested by the nutrition specialist at hospital after a detail revision of the article in the Cardiology Journal. The defuzification was made by the centroid method (equation 2).

IV. Designation for an operational risk scale Fuzzy System

The main variables taking place to assign the risky condition for the patient and to determinate the future attention level with the corresponding self-care are linked whit the MS diagnostic, but also include the meal habits, the carcinogenic tumors and a complete clinician analysis from the medicals to the patients. The next table illustrates the method followed in this part and the variables ranges are also depicted there. In this paper, just 4 different variables are taking into account to define the necessary medic attention. However, the real DB designed considers 39 different elements to produce a more confidence diagnostic. Once the variables are extracted of the DB, they are fuzzificated by the corresponding designed membership functions (Figure 4). The "risk" functions, for example, are classified in 1) minimum, 2) reasonable and 3) high patient risk. In general, this fuzzy structure is applied in all others variables.

Table 5.MS Variables

Variable	Minimum Risk	Reasonable risk	High Risk
Glucose	<90	<150	>300
Triglycerides	<150	<300	>500
Uric Acid	<6	>8	<3
Albumin	>3.5	>3.4	<2.9

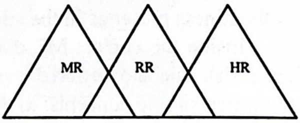


Figure 4. Input Membership fuctions

The ultimate evaluation (fuzzification) delivers a vector with three components (each one of them devoted to each considered variable). This important element, in fact, defines the medical assistance in view of the mixed membership function describing by the discrete algorithm, described below:

$$^{\circ}G = \left[\sum_{i=1}^{39} {}^{\circ}G_{MR(i)}, \sum_{i=1}^{39} {}^{\circ}G_{RR(i)}, \sum_{i=1}^{39} {}^{\circ}G_{HR(i)} \right] \quad (1)$$

The fuzzification method was selected as centroid, in view of the simplicity demanded for this job. It is important to remember that all this software should be working 24 hours and it should be attended many patients during short time.

$$y_q^{crisp} = \frac{\sum_{i=1}^{R} b_i^q \int_{Y_q} \mu_{B_q^i}(y_q)dy_q}{\sum_{i=1}^{R} \int_{Y_q} \mu_{B_q^i}(y_q)dy_q} \quad (2)$$

The given outputs correspond to the attention model: a) First time visiting the physician, b) First time and a predefined consequent attention and 3) Hard Nutritional Support (Figure 5).

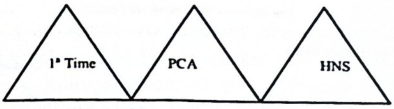


Figure 5. Membership function output

To make more exact this algorithm, the weighted fuzzy logic (to assign different values to each variable) was applied at this point. A complete version of the suggested method to treat the MS is depicted in Fig. 6.

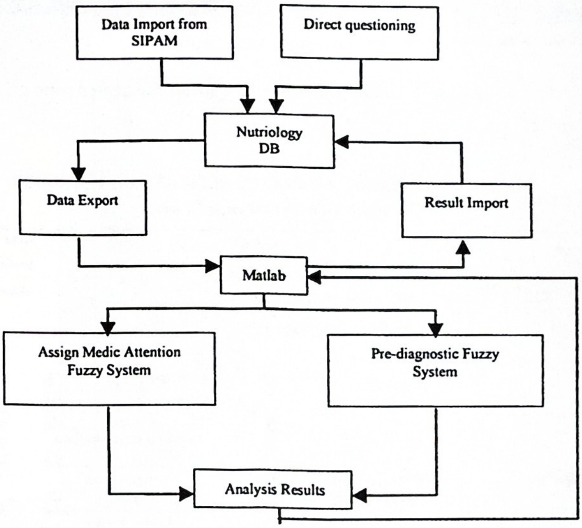
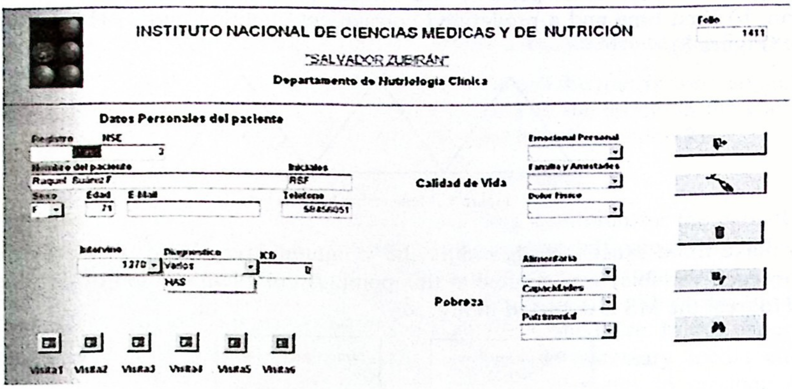


Fig. 6. Flow Chart for the overall fuzzy

Results

At this moment, the DB is receiving the patient's information, however, many designs related with it have been developed: for example the visual interfaces and the corresponding formularies. The final DB design was completed using the commercial software ACCESS and using the ODBC transferred method. Additionally, this database contains some special request that allows to assign numerical values for each linguistic variables associated with the original MS scheme proposed by the INCMNSZ's nutrition department (Fig. 7). Besides, the DB was build to attend each patient during its last six hospital appointments; this fact permits the physician to evaluate the advances or the falling backs in the patient's health. This aspect is novel because the current commercial databases just give a general view on the patient response but they do not suggest any kind of treatment and, obviously, many of them are not provided by an artificial intelligence algorithm to define the corresponding diagnostic.



INSTITUTO NACIONAL DE CIENCIAS MEDICAS Y DE NUTRICION

"SALVADOR ZUBIRAN"

Departamento de Nutriologia Clinica

Folio 1411

Datos Personales del paciente

Registro | NSE 3

Nombre del paciente: Raquel Suárez F

Iniciales: RSF

Sexo: F

Edad: 71

E Mail

Telefono: 58456051

Intervino: 1270

Diagnostico: Varios

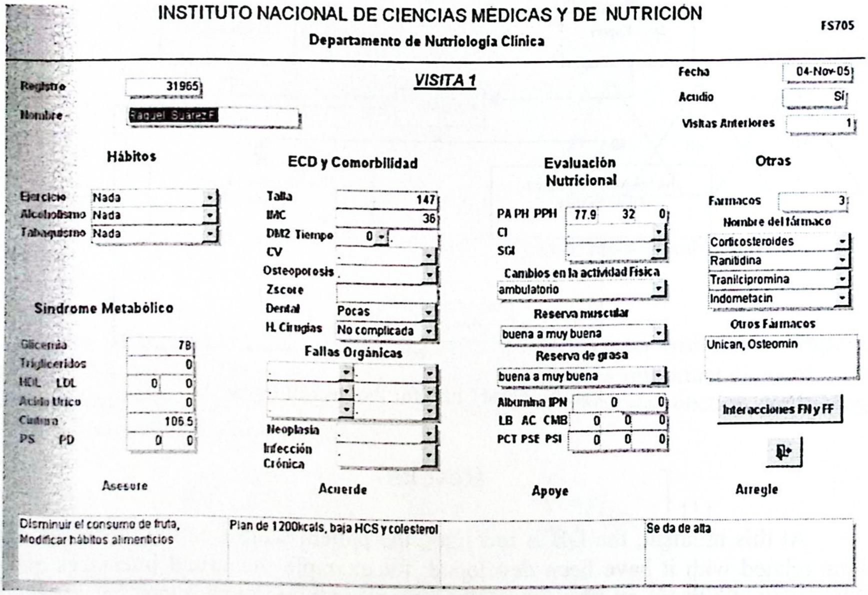
HAS

ICD: 0

Calidad de Vida: Emocional Personal; Familia y Amistades; Dolor Fisico

Pobreza: Alimentaria; Capacidades; Patrimonial

Figure 7. Lifestyle and general data for the treated patient.



INSTITUTO NACIONAL DE CIENCIAS MÉDICAS Y DE NUTRICIÓN

Departamento de Nutriología Clínica

FS705

VISITA 1

Registro: 31965

Nombre: Raquel Suárez F

Fecha: 04-Nov-05

Acudio: Sí

Visitas Anteriores: 1

Hábitos

Ejercicio: Nada

Alcoholismo: Nada

Tabaquismo: Nada

Síndrome Metabólico

Glicemia: 78

Trigliceridos: 0

HDL LDL: 0 0

Acido Urico: 0

Cintura: 106.5

PS PD: 0 0

ECD y Comorbilidad

Talla: 147

IMC: 36

DM2 Tiempo: 0

CV

Osteoporosis

Zscore

Dental: Pocas

H. Cirugias: No complicada

Fallas Orgánicas

Neoplasia

Infección Crónica

Evaluación Nutricional

PA PH PPH: 77.9 32 0

CI

SGI

Cambios en la actividad Física: ambulatorio

Reserva muscular: buena a muy buena

Reserva de grasa: buena a muy buena

Albumina IPN: 0 0

LB AC CMB: 0 0 0

PCT PSE PSI: 0 0 0

Otras

Farmacos: 3

Nombre del fármaco: Corticosteroides; Ranitidina; Tranilcipromina; Indometacin

Otros Fármacos: Unican, Osteomin

Asesore: Disminuir el consumo de fruta, Modificar hábitos alimenticios

Acuerde: Plan de 1200kcals, baja HCS y colesterol

Apoye

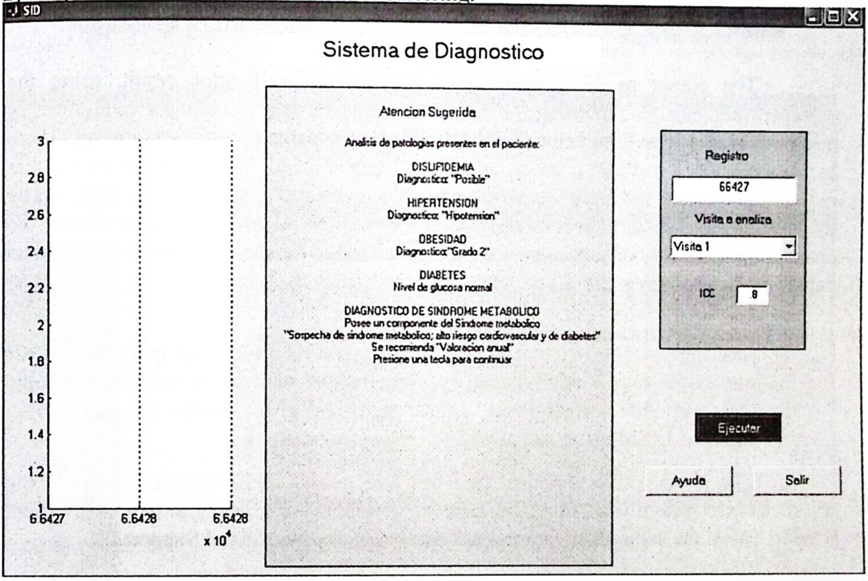
Arregle: Se da de alta

Figure 8. Clinical variables for each patient.

The figure 7 shows the DB view corresponding to the general data for each patient as well as those related with the lifestyle carried out by him or her. The next figure 8 demonstrates the frequently questions supplied by the medic to the patient and his/her clinical studies that were suggested before. Of course, these elements are used as the input data for the decision systems generated by the fuzzy arrangement.

The DB was linked with MATLAB in order to analyze the stored data. It is important to note the numerical representation considered at this point because this procedure allows defining any statistical method to derive an alternative form for the same diagnostic (one given by a "possibility condition" and other described by statistics developments).

A graphical interface was made to present the numeric analysis by GUIDE (Graphical User Interface Development Environment) of Matlab. This interface allows the medicals a fast and easy manipulation of the software. The figure 9 shows the diagnose into the graphical interface called DIS (Diagnose Interface System) concerning at SM diagnose. The official report provided by the diagnostic process is shown in figure 9, where the three specific illnesses (Arterial Pressure, Obesity and Low Lipids Concentration) are analyzed. Finally the software introduced here gives a couple of additional information: 1) a possible therapy to define and complete the diagnostic and 2) the possible confirmation for the MS suffering.



Atencion Sugerida

Analisis de patologias presentes en el paciente:

DISLIPIDEMIA
Diagnostico: "Posible"

HIPERTENSION
Diagnostico: "Hipotension"

OBESIDAD
Diagnostico:"Grado 2"

DIABETES
Nivel de glucosa normal

DIAGNOSTICO DE SINDROME METABOLICO
Posee un componente del Sindrome metabolico
"Sospecha de sindrome metabolico; alto riesgo cardiovascular y de diabetes"
Se recomienda "Valoracion anual"
Presione una tecla para continuar

Registro: 66427

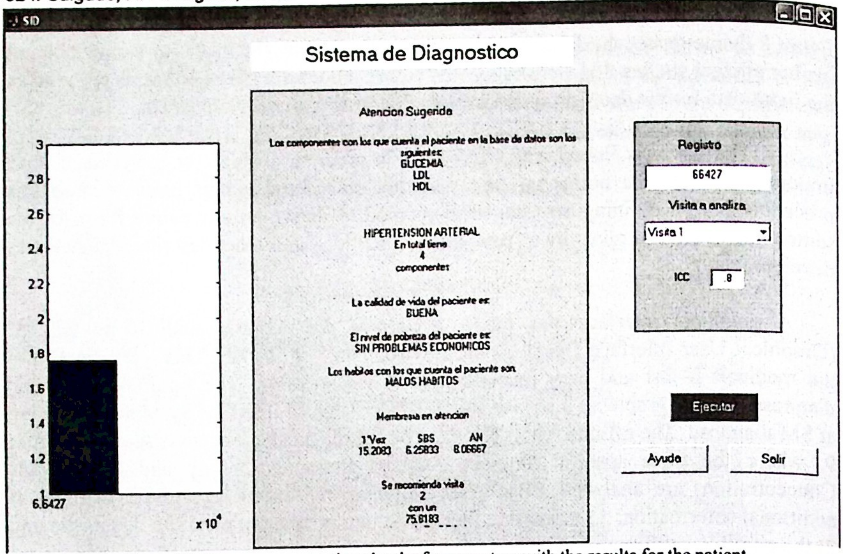
Visita a analiza: Visita 1

IDC: 8

Figure 9. General final view for the pre-diagnostic algorithm.

The next figure demonstrates the numerical inform given by Matlab alter the data were imported from the DB. This particular analysis was realized for a patient given in the previous diagnosis analysis (Figure 9).

- Sex: Female. This patient is affected by the MS illness and by the type II Diabetes Mellitus (bad glucose regulation), high HDL, uric acid outside of the normal ranges and obesity (classified as level 1).



Sistema de Diagnostico

Atencion Sugerida

Los componentes con los que cuenta el paciente en la base de datos son los siguientes:
GLICEMIA
LDL
HDL

HIPERTENSION ARTERIAL
En total tiene
4
componentes

La calidad de vida del paciente es:
BUENA

El nivel de pobreza del paciente es:
SIN PROBLEMAS ECONOMICOS

Los habitos con los que cuenta el paciente son
MALOS HABITOS

Membresia en atencion

1 Vez	SBS	AN
15.2083	6.25833	8.06667

Se recomienda visita
2
con un
75.6183

Registro
66427
Visita a analizar
Visita 1
ICC 8

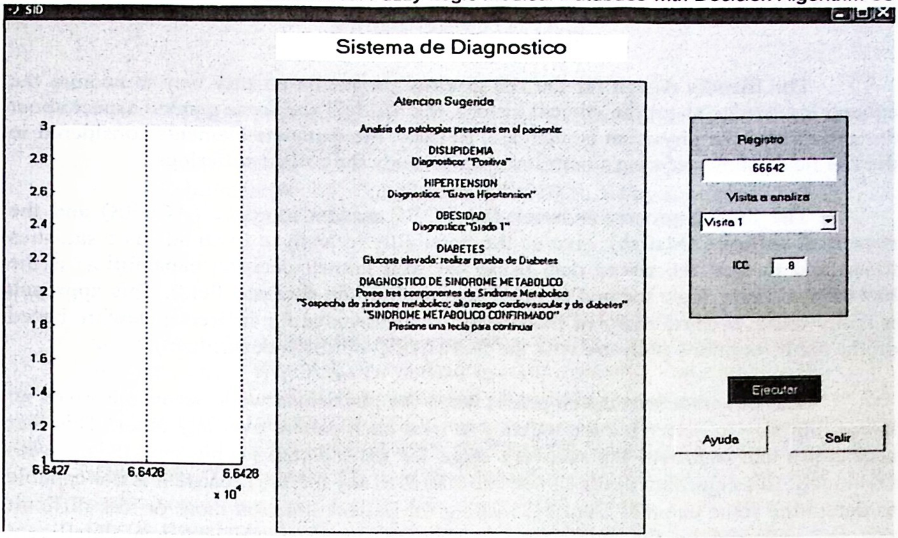
Figure 10. Response screen given by the fuzzy system with the results for the patient number 1. It should be noted the suggested therapy included at the bottom of the specific report.

The output in the graphic represents the defuzzification result, using the designed membership functions, the linguistic variables selected and the overall fuzzy inference model. The final value (1.7581) indicates the patient requires a new session to be attended with a medium urgency (possibility) determined by a 75.18 %. If the number in the response is 3, indicates the patient should be keep at the hospital to be observed by the medics and 1 with 0 % indicates the patient can be released for the treatment. To prove the workability of the suggested method, a second case is considered. The clinical values for this patient are:

- Sex: Female. She is a patient in death danger whit a lot of complications in her metabolic state. There is some evidence related with tumor activity, chronicle infection and a generalized organic fail. The given results are depicted in figures 11 (SM diagnose) and 12 (risk scale allocation).

The given results demonstrated this patient should be attended "before" the results derived before (patient 1), because the possibility or urgency to treat this patient is 90.91 %. At the same time, the results suggest a Risky Nutritional Support.

The analysis derived by the MS pre-diagnostiv system allows stating a well defined methodology to provide a range or degree of attendance and to define repeatability concepts to treat the MS illness, due the numerical algorithm introduced in this study.



Registro: 66642

Visita a analiza: Visita 1

ICC: .8

Atencion Sugerida

Analisis de patologias presentes en el paciente:

DISLIPIDEMIA
Diagnostico: "Positiva"

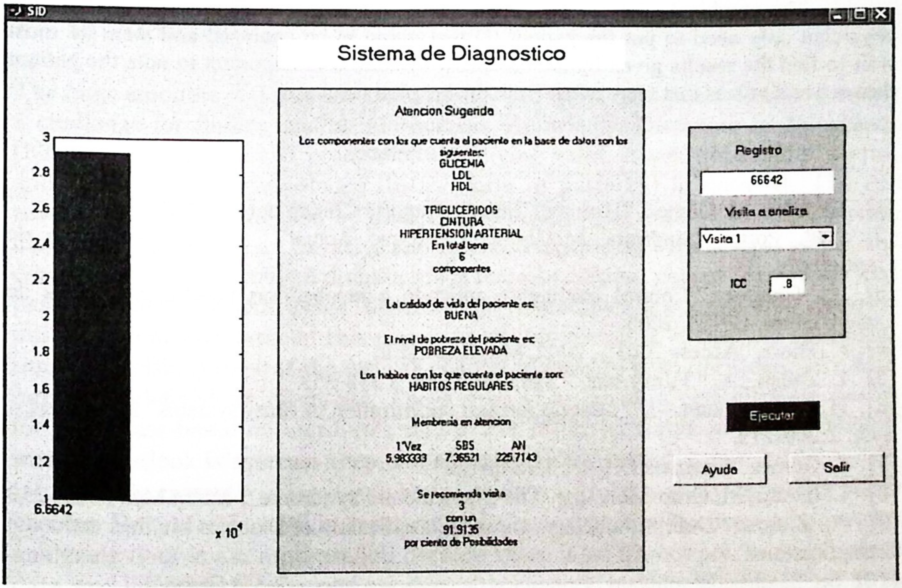
HIPERTENSION
Diagnostico: "Grave Hipertension"

OBESIDAD
Diagnostico:"Grado 1"

DIABETES
Glucosa elevada: realizar prueba de Diabetes

DIAGNOSTICO DE SINDROME METABOLICO
Posee tres componentes de Sindrome Metabolico
"Sospecha de sindrome metabolico: alto riesgo cardiovascular y de diabetes"
"SINDROME METABOLICO CONFIRMADO"
Presione una tecla para continuar

Figure 11. Results derived by the second patient's data from the diagnose algorithm.



Registro: 66642

Visita a analiza: Visita 1

ICC: .8

Atencion Sugerida

Los componentes con los que cuenta el paciente en la base de datos son los siguientes:
GLICEMIA
LDL
HDL

TRIGLICERIDOS
CINTURA
HIPERTENSION ARTERIAL
En total tiene
6
componentes

La calidad de vida del paciente es:
BUENA

El nivel de pobreza del paciente es:
POBREZA ELEVADA

Los habitos con los que cuenta el paciente son:
HABITOS REGULARES

Membresia en atencion

```
1°Vez      SBS       AN
5.983333   7.36521   225.7143
```

Se recomienda visita
3
con un
91.9135
por ciento de Posibilidades

Figure 12. Results given by the risk analysis fuzzy system

Conclusions

The friendly design for the DB gives to the medic an easy way to acquire the enough knowledge about the clinical history, the lifestyle and some general aspect about the patient that the physician is attended. Besides, the complete elements considered in the DB fields allow defining a better manner to study the patient sufferings.

The linking process between the ODBC system interface (ACCES) and the numerical software (Matlab), give us the possibility to analyze (with advance statistics tools) and to treat the stored data using the well known decision capabilities of the hierarchical fuzzy logic method (a novel approach in the database field). This approach is really useful because many of the elements considered at the inference step are based on the medic opinions (they are with the INCMNSZ' nutrition department).

The MS treatment development helps the physicians work because it gives an interesting way to determine the urgency to treat each patient over any other. This fact accelerates and improves the recovery stage for the affected people on MS, as well. Obviously, this algorithm never can be substitutes to any medic, because it is not capable to determine some external elements making the patient situation more or less difficult that the indicated by the stored data in the DB, however, this artificial intelligent algorithm can be useful to help the MS treatment and to improve the people health.

The graphical interface provide a easier way to drive the software because the physician only need to put the patient ID that going to be analyzed and then, he must wait to find the results given by the automatic system. It is important to note the patient data are real values that were taken from the hospital database.

References

[1] I. Castro, M. Gámez. "Historia Clínica", Reporte Clínico del HNCMNSZ

[2] R. Porto. "Síndrome Metabólico. Enfoque Actual" , Revista Mexicana de Cardiología, 2001.

[3] I. Chairez "Control inteligente aplicado a incubadoras neonatales", Tesis de Licienciatura, UPIBI, 2003.

[4] C. Pérez, "Access 2003" , 2002, Mc. Graw Hill.

[5] L. Zadeh, :A: " Fuzzy sets. " 1965 Inf. Cont. 8:338-353

[6] H. Zimmermann. –J. " Description and optimization of fuzzy systems" Int. J. Gen. Syst. 2:209-215.

[7] S. Gómez "Sistemas Difusos Jerárquicos"

[8] L. Groop, M. Orho-Melander. The dysmetabolic Syndrome. J Intern Med 2001; 250

[9] A. Zimmet "Definition, Diagnosis and Classification of Diabetes Mellitus and its Complications"

[10] Isselbacher, Braunwald "Principios de medicina Interna", Mc Graw-Hill.

On Querying Inductive Databases to Mine Decision Rules

Omar Nieva-García and Edgard Benítez-Guerrero

Laboratorio Nacional de Informática Avanzada, A. C.
Rebsamen No. 80, Col. Centro, CP 91000, Xalapa, México
ong0027@lania.edu.mx, ebenitez@lania.mx

Abstract. This paper introduces the MINE DECISION RULE extension to SQL for mining classification rules. It allows the user to express his/her mining requirements and to use the resulting rules to classify unseen data. To enable the evaluation of an inductive query Q incorporating a MINE DECISION RULE expression, a typical object–relational algebra has been augmented with the MineDR operator to mine decision rules. To evaluate Q, it is first translated into a query tree with nodes containing operators in this augmented algebra, then the query tree is transformed into an execution plan which is finally executed. A prototype system supporting our approach is also presented.

1 Introduction

The huge amounts of data that are currently produced in digital format represent a challenge for finding useful information. Knowledge Discovery in Databases (KDD) is the non-trivial process of identifying valid, novel, potentially useful, and understandable knowledge (in the form of patterns) in data [1]. The extracted knowledge can then be used to characterize the data or to classify new, unseen data. KDD is an iterative and interactive process with several steps: understanding of the problem domain, data transformation, pattern discovery, and pattern evaluation and usage. Data Mining techniques are applied to discover patterns from raw data. In this paper we are interested in extracting classification (or decision) rules of the form IF-THEN to classify new, uncategorized data into a pre-defined set of classes. These rules then create a classification model for each class based on attributes values. For instance, a rule might say that if weather outlook is overcast then one can play Golf.

There is currently a large number of tools to help the analysts in the KDD process. However, they fail in supporting the complete KDD process adequately: analyzing data is a complicate job because there is no framework to manipulate data and patterns homogeneously. Recently, Inductive Databases (IDBs) have been proposed to remedy this situation. In this framework, a database contain, in addition to the raw data, (implicit or explicit) patterns about the data [2]. The discovery of patterns can then be viewed as a special kind of database querying and, in this context, query languages and associated query evaluation and

© Jesús Olivares, Adolfo Guzmán (Eds.)
Data Mining and Information Systems.
Research in Computer Science 22, 2006, pp. 55-65

optimization techniques are being proposed. Current research in this area has focused on inductive querying of association rules [3, 4], sequential patterns [5] and clusters [6].

The expression and evaluation of queries to mine classification rules have been partially studied on the inductive database context. Some languages have been proposed, such as DMQL [7], which provides primitives for extracting rules and other kinds of patterns, AXL [8] that gives a user the possibility of extending SQL to introduce complex algorithms such as data mining functions, and DMX [9] that also provides expressions to support a variety of mining techniques, including rule induction. These languages are important because they introduce decision rule mining into the traditional database framework. However, in these proposals, it is not clear how a query is processed, how the extracted rules can be manipulated and how they can be used to classify new data.

In this paper, we propose the MINE DECISION RULE extension to SQL for mining classification rules. It enables the user to express his/her rule mining requirements such as the data source and the constraints that a rule must satisfy to be considered in the final result. In order to manipulate homogeneously data and rules, a typical object-relational data model is used [10]. For processing queries using MINE DECISION RULE, we have extended a modified version of the Object-Relational algebra presented in [10] with the MineDR operator (Ξ) to mine decision rules. This operator help us to produce and process algebraic expressions to manipulate data and decision rules in the same framework. To experiment our approach, we have implemented the *DRMiner* prototype system to show the capabilities of our language and test query processing techniques.

This paper is organized as follows. Section 2 overviews related work. Section 3 briefly describes the OR model and introduces the elements extending this model to represent decision rules. Section 4 presents MINE DECISION RULE. Section 5 summarizes the OR algebra and explains the MineDR operator. Section 6 describes the DRMiner prototype system. Finally, Section 7 concludes this paper and introduces our future work.

2 Related Work

Extracting decision rules from small datasets is a problem that has been studied for years. However, mining rules from large databases poses new challenges. In order to discuss relevant related work, we have classified it in two categories: rule learning algorithms, and inductive query languages and processing.

Research on rule learning algorithms has traditionally focused on improving the heuristic search and the functions for rule evaluation. In general, algorithms follow a covering strategy, i.e., an algorithm searches for a rule that explains a part of its training instances (pre-classified data), separates these instances and repeats the search for a rule until no instances remain. Popular rule learning algorithms are R1 [11], PRISM [12], CN2 [13], PFOIL [14] and RIPPER [15]. A useful analysis of these algorithms can be found in [16].

Inductive query languages (such as MSQL [4] and MINE RULE [3]) have mainly focused on association rule mining. Regarding the extraction of decision rules, there are three proposals: DMQL [7], AXL [8] and DMX [9]. DMQL is a language providing expressions to carry out an extensive set of data mining tasks, and in particular it allows the user to generate rules to classify data according to one or more attributes. It also allows the user to select and filter source data from a table. However, DMQL does not provide support for rule filtering and other post-processing operations (such as cross-over) and, for this, ad-hoc tools have to be externally provided.

Another proposal is the AXL language. It allows the user to add some extensions to SQL for expressing data mining tasks such as classification. By defining a new aggregate function to implement a classifier, a classification technique can be expressed in a SQL language expression. However, its main disadvantage is that the mining algorithm has to be implemented, involving significant code rewriting.

DMX (Data Mining Extension), like DMQL, provides several Data Mining techniques. DMX is divided on a Data Definition Language (DDL) and a Data Manipulation Language (DML). Using the DDL, the user has to define the data model schema (using specific data types) and the proper algorithm to use, according to the data mining task. The user then utilizes DML sentences to handle the mining model and to perform prediction tasks. Let us remark that the resulting mining model is a "black box", unless the user queries its description using specific DML sentences.

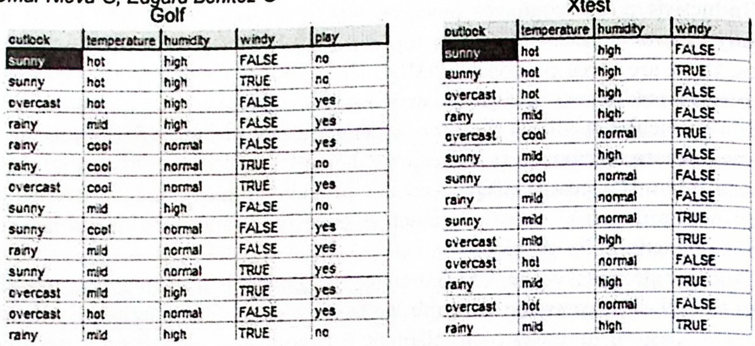
There are other works that are useful to understand the KDD process and the IDBs framework. These works are related to efficient mapping of patterns in memory [17], development of general primitives for data mining tasks on query languages [18–20] and development of a formal theory for IDBs [21, 22].

3 Data and Rules Model

This section introduces the model for data and decision rules. To represent them homogeneously, the Object-Relational (OR) data model presented in [10] is used. In the following, the OR model is briefly introduced and then the data types proposed to represent rules are explained.

The basic components of the OR model are types. An OR database schema consists of a set of row types $R_1, \ldots, R_m$, and each attribute in a row type is defined on a certain type, which can be a built-in type, an abstract data type (ADT), a collection type, a reference type or another row type. An object-relational database D on database scheme OR is a collection of row type instance sets (called OR tables) $ort_1, \ldots, ort_m$ such that for each OR table ort_i there is a corresponding row type RT_i, and each tuple of ort_i is an instance of the corresponding row type R_i.

Let us consider for instance the database shown in Figure 1. It contains two tables: the Golf table and Xtest table. The Golf table stores a set of weather conditions that can be used to decide if one can play Golf or not. Its row type



Golf

outlook	temperature	humidity	windy	play
sunny	hot	high	FALSE	no
sunny	hot	high	TRUE	no
overcast	hot	high	FALSE	yes
rainy	mild	high	FALSE	yes
rainy	cool	normal	FALSE	yes
rainy	cool	normal	TRUE	no
overcast	cool	normal	TRUE	yes
sunny	mild	high	FALSE	no
sunny	cool	normal	FALSE	yes
rainy	mild	normal	FALSE	yes
sunny	mild	normal	TRUE	yes
overcast	mild	high	TRUE	yes
overcast	hot	normal	FALSE	yes
rainy	mild	high	TRUE	no

Xtest

outlook	temperature	humidity	windy
sunny	hot	high	FALSE
sunny	hot	high	TRUE
overcast	hot	high	FALSE
rainy	mild	high	FALSE
overcast	cool	normal	TRUE
sunny	mild	high	FALSE
sunny	cool	normal	FALSE
rainy	mild	normal	FALSE
sunny	mild	normal	TRUE
overcast	mild	high	TRUE
overcast	hot	normal	FALSE
rainy	mild	high	TRUE
overcast	hot	normal	FALSE
rainy	mild	high	TRUE

Fig. 1. Example database

is composed by the attributes *outlook*, *temperature*, *humidity*, *windy* and *play* (the attribute class) which are all of atomic types. The Xtest table stores data related to weather conditions but without specifying a class for each tuple.

The result of the processing of the MINE DECISION RULE expression is a table named by default *Decision-Rules* which has the following row type:

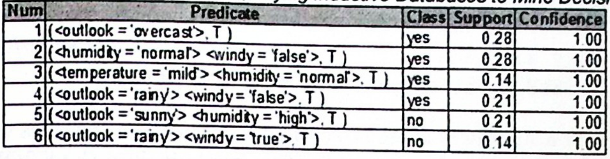
decision-rule (idrule: *integer*, predicate: *dr-pattern*, class: *string*, support: *float*, confidence: *float*)

where *idrule* is a unique identifier for a rule, *predicate* is the "body" of the rule to predict a class (of type *dr-pattern* which will be defined later), *class* is the label to be assigned and *support* and *confidence* are two accuracy measures of each rule. The type *dr-pattern* is an Abstract Data Type (ADT) defined as:

dr-pattern(Equalityset:set(Equality), Ordered:boolean,
  create: Func(boolean,set(Equality),boolean),
  get-rule: Func(string),
  exist: Func(boolean,string))

A *dr-pattern* has two attributes: *Equalityset* and *Ordered*. *Equalityset* is a non-empty set of values of type *Equality*, where each *Equality* represents a selector of a rule, e.g. <humidity = "high">. The value of the *Ordered* attribute indicates the form in which the rules are tested against the data (more on this in Section 6). There are three functions defined for *dr-pattern*. The *create* function is a constructor that takes as input a set of *Equality* and a boolean value and returns a boolean value indicating success or failure. The *get-rule* function generates as a string the body of a conjunctive rule from the selectors in *Equalityset*. Finally, the *Exists* function searches a string inside *Equalityset* and returns a boolean value indicating the success or failure of the search.

The table shown in Figure 2 contains a set of rules describing the behavior of the Golf data. For instance, rule number one says that when attribute *outlook* is



Num	Predicate	Class	Support	Confidence
1	(<outlook = 'overcast'>, T)	yes	0.28	1.00
2	(<humidity = 'normal'> <windy = 'false'>, T)	yes	0.28	1.00
3	(<temperature = 'mild'> <humidity = 'normal'>, T)	yes	0.14	1.00
4	(<outlook = 'rainy'> <windy = 'false'>, T)	yes	0.21	1.00
5	(<outlook = 'sunny'> <humidity = 'high'>, T)	no	0.21	1.00
6	(<outlook = 'rainy'> <windy = 'true'>, T)	no	0.14	1.00

Fig. 2. A table containing a set of decision rules

overcast then the predicted class is *yes* with a *support* of 0.28 and a *confidence* of 1.0, i.e., one can play Golf.

4 The Mine Decision Rule Extension to SQL

This section introduces our MINE DECISION RULE extension to SQL for extracting classification rules. We have considered some features of the MINE RULE language [3] in the design of MINE DECISION RULE: (a) selection of relevant data, (b) definition of specific structures and (c) definition of conditions to filter rules. The syntax is as follows:

```
MINE DECISION RULE [<target>]
WITH <attribute> AS CLASS
FROM <table> | <sub-query>
[WHERE <conditions>]
```

The MINE DECISION RULE clause produces a new table that can be optionally renamed as indicated by < *target* >. The WITH < *attribute* > AS CLASS clause specifies the < *attribute* > that contains the classes to be predicted. The FROM clause defines the data source (a table or a subquery) for decision rule mining. Finally, the WHERE clause is optional and let the user specify < *conditions* > to filter a set of rules to get only those of interest.

In the following, we present some examples to show how MINE DECISION RULE can be used to express different queries. First, let us consider the query *Retrieve a set of decision rules to know when one can play Golf (or not) considering play as attribute class* (**Q1**).

```
MINE DECISION RULE Xmodel
WITH play AS CLASS
FROM Golf
```

In this inductive query, the resulting table is renamed as *Xmodel*. The WITH clause specifies the attribute that has the classes to be predicted, *play* in this case. This line is mandatory because rules are built around this attribute. In this example, the FROM clause defines as source the Golf table. With this query, the user retrieves all possible rules with their respective support and confidence.

Now consider the query *Retrieve a set of decision rules based on attributes outlook and windy, to know when one can play Golf or not* (**Q2**).

```
MINE DECISION RULE Xmodel
WITH play AS CLASS
FROM ( SELECT outlook, windy, play FROM Golf )
```

In this case a sub-query is used to extract a dataset and the rules will consider only the attributes *outlook* and *windy*. This is specified in the FROM clause.

Let us suppose now that a domain expert wants to consider the query *Retrieve a decision rules set with the minimum support of 0.25, to know when we can play Golf or not* (**Q3**)

```
MINE DECISION RULE Xmodel
WITH play AS CLASS
FROM Golf
WHERE support >= 0.25
```

The resulting table is filtered in the WHERE clause to retrieve the rules having a computed support equal to 0.25 or above.

Now, let us consider the query *Classify the data on the Xtest table based on rules extracted from the Golf table, considering play as class attribute* (**Q4**).

```
SELECT *
FROM Xtest AS XT, ( MINE DECISION RULE
                    WITH play AS CLASS
                    FROM Golf ) AS XM
WHERE PredictionJoin( XT, XM )
```

This query shows that it is possible to join (using a the PredictionJoin function) a table T containing uncategorized data (Xtest in this example) with a table TR (that can be the result of the evaluation of a MINE DECISION RULE expression) containing a set of rules. The result of a query of this kind is a table containing the tuples of T already classified by the rules in TR.

5 Query Processing

For processing inductive queries using MINE DECISION RULE, we have extended the OR algebra introduced in [10] with the MineDR operator to mine decision rules. In the following, we briefly explain the OR algebra and then introduce MineDR.

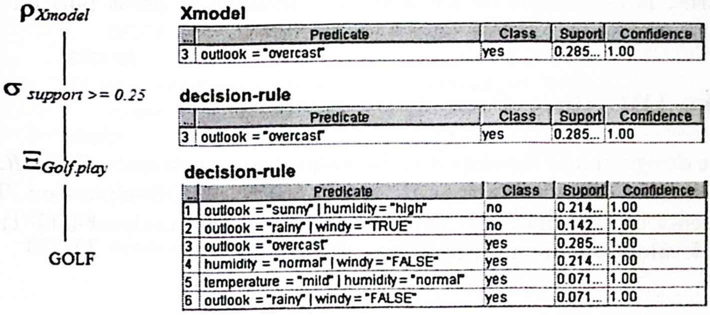
Expressions in the OR algebra consist of OR operands and OR operators. An OR operand is either an OR table, a path expression or the result of another operation. In this section, we simply use the term "table" to refer to all the possible operands, as long as no distinction is necessary. The set of OR operators consists of the object-relational counterparts of basic relational operators – select (σ), join ($\bowtie$), Cartesian product($\times$), project (π) –, set operators – union ($\cup$), difference ($-$), intersection ($\cap$)), nest (ν), unnest (υ)–, and special operators to handle row type object identity – map(ϕ) and cap(δ) –. To the original operator set, we incorporate the operators group-by (F) and rename (ρ).

To this algebra we have added the MineDR operator to mine decision rules, which is noted as follows:

$$\Xi_{A_n}(r)$$

where r is an input table with schema $(A_1, A_2, ..., A_{n-1}, A_n)$ such that A_k $(k = 1 \dots n-1)$ are general attributes and A_n is a special attribute representing a label or class for each tuple in r. The MineDR operator computes as result a table of row type *decision-rule* (see Section 3) containing a set of rules extracted from table r.

To illustrate the concepts and operations related to the MineDR operator, the evaluation of queries Q_3 and Q_4 is explained in the following. Let us consider first the query Q_3, *Retrieve a decision rules set with the minimum support of 0.25, to know when we can play Golf or not.* In this query, the classification is based on the attribute *play*.



Xmodel

...	Predicate	Class	Suport	Confidence
3	outlook = "overcast"	yes	0.285...	1.00

decision-rule

...	Predicate	Class	Suport	Confidence
3	outlook = "overcast"	yes	0.285...	1.00

decision-rule

...	Predicate	Class	Suport	Confidence
1	outlook = "sunny" | humidity = "high"	no	0.214...	1.00
2	outlook = "rainy" | windy = "TRUE"	no	0.142...	1.00
3	outlook = "overcast"	yes	0.285...	1.00
4	humidity = "normal" | windy = "FALSE"	yes	0.214...	1.00
5	temperature = "mild" | humidity = "normal"	yes	0.071...	1.00
6	outlook = "rainy" | windy = "FALSE"	yes	0.071...	1.00

Fig. 3. Query Tree for Q_1

Figure 3 shows the query tree for Q_3. As input to MineDR (Ξ), it is necessary to indicate the name of the table containing the data set (in this case the Golf table) from which rules will be generated, and the name of the attribute classifying the data (*play* in this example). Next, MineDR produces a table containing a set of decision rules. In this query, the WHERE clause is used to filter the rules to obtain only those that have a support equal to 0.25 or above, and thus it is necessary to introduce the select (σ) operator in the query tree.

Now consider the query Q_4 where a set of decision rules is applied: *Classify the Xtest table based on rules extracted from the Golf table, considering play as the class attribute.* In the query tree in Figure 4, a new table is produced by the MineDR operator by extracting decision rules from the GOLF table. At the left side of the tree is the Xtest table with tuples representing instances to be classified. To obtain the result of the query (classified instances), the Prediction Join of the resulting table of MineDR with Xtest is executed. At the top of the

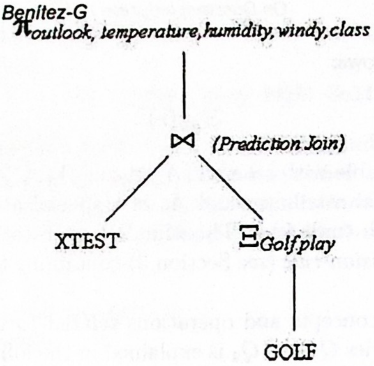


Fig. 4. Query Tree for Q_3

query tree, it is possible to see that all attributes of Xtest plus an attribute containing the assigned class have been projected.

6 The DRMiner Prototype System

We have designed and developed in Java a prototype system, called *DRMiner*, to process queries considering the MINE DECISION RULE expression. The main components of DRMiner are: (a) User interface, (b) Analyzer and Translator, and (c) Evaluation engine (See Figure 5).

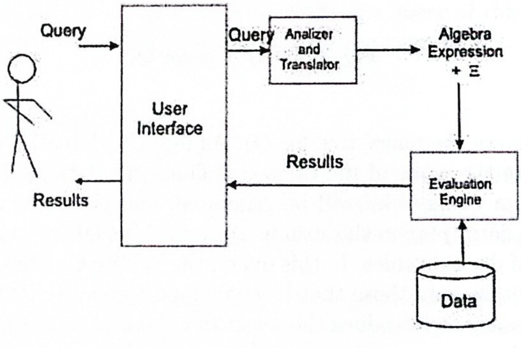


Fig. 5. DRMiner System Architecture

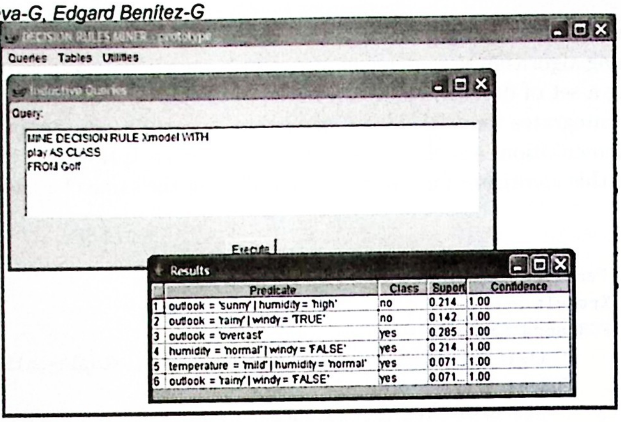
Once a user types a query, the Analyzer verifies its syntax and translate it into a query tree with nodes containing operators in the OR algebra augmented with MineDR. This query tree is passed to the evaluation engine, which uses a rule learning algorithm to implement MineDR to classify data and returns a table containing a set of decision rules as a result of the query. Our evaluation engine currently integrates the PRISM [12] algorithm, because it easy to find standard code implementations and documentation on the Internet. However, it is possible to change this algorithm for another accomplishing the same classification task.

```
Function PREDICTIONJOIN (table T, table M)
Xresult = $\oslash$
FOR each tuple $t_T = (a_1, a_2, .., a_n) \in T$
   classified = false; class = null; C = $\oslash$; xtuple=null;
   FOR each tuple $t_M \in M$
      IF satisfies($t_T$,$t_M$) THEN
         class = $t_M$.class
         classified = true
         IF $\neg t_M$.predicate.ordered THEN add(C,class)
         ELSE break
      ENDIF
   ENDFOR
   IF $\neg t_M$.predicate.ordered AND classified THEN
      class = solve_controversy(C)
   ENDIF
   xtuple = $(a_1, a_2, ..a_n$, class)
   add(Xresult,xtuple)
ENDFOR
RETURN Xresult
```

Fig. 6. Prediction Join Algorithm

The DRMiner evaluation engine also implements a prediction join operation to classify uncategorized data according to a set of rules. Figure 6 shows the algorithm. The input is a table (T) with unseen instances (tuples) and a table (M) with the decision rules to predict the class of each tuple of T. To classify a new tuple, each rule is tested on the tuple and, when the conditions of a rule are satisfied, then a class is found. If rules are ordered then the first class found is assigned to the tuple, otherwise all possible classes are found and then the controversy is solved to assign only one class to the tuple. The variable *xtuple* stores each time a tuple $t_T \in T$ plus its assigned class. If none of rules fires, the algorithm assigns the tuple a *null* class. The result is the $Xresult$ table.

Finally, Figure 7 shows the DRMiner user interface. It is possible to type a query to retrieve a table containing a set of rules. All queries mentioned in this paper can be executed "as is" in *DRMiner.* For instance, in Figure 7, we can see that a basic query to *Retrieve a set of decision based on the Golf table and considering play as attribute class* is introduced and the results are shown.



```
MINE DECISION RULE λmodel WITH
play AS CLASS
FROM Golf
```

	Predicate	Class	Suport	Confidence
1	outlook = 'sunny' | humidity = 'high'	no	0.214	1.00
2	outlook = 'rainy' | windy = 'TRUE'	no	0.142	1.00
3	outlook = 'overcast'	yes	0.285	1.00
4	humidity = 'normal' | windy = 'FALSE'	yes	0.214	1.00
5	temperature = 'mild' | humidity = 'normal'	yes	0.071	1.00
6	outlook = 'rainy' | windy = 'FALSE'	yes	0.071	1.00

Fig. 7. DRMiner user interface

7 Conclusion and Future Work

This paper introduced the MINE DECISION RULE extension to SQL for mining classification rules. It enables the user to express his/her requirements on the extraction of decision rules from pre-classified data and apply the mined rules over new, unclassified data. To evaluate queries incorporating MINE DECISION RULE, the MineDR operator has been integrated into a typical OR algebra. MineDR takes as input a source table and a specific class attribute, and produces a table containing a set of rules. The DRMiner prototype system has been developed to test our ideas.

Our future work includes research on query optimization. It will be focused on deciding which algorithm, from a possible set of rule learning algorithms, is the most suitable for a classification task, according to data features such as the number of attributes, data volumes and the presence/absence of noise.

Acknowledgements. The authors thank the reviewers of this paper for their useful comments. The work of Mr. Nieva-García has been supported by the Mexican National Council of Science and Technology (CONACYT).

References

1. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R.: From Data Mining to Knowledge Discovery: An Overview. AAAI/MIT Press (1996)
2. Mannila, H.: Inductive Databases and Condensed Representations for Data Mining. In: Proceedings of the 1997 International Symposium on Logic Programming (ILPS'97), MIT Press (1997) 21–30
3. Meo, R., Psaila, G., Ceri, S.: A New SQL-like Operator for Mining Association Rules. The VLDB Journal (1996) 122–133

4. Imielinski, T., Virmani, A., Abdulghani, A.: Datamine: Application Programming Interface and Query Language for Database Mining. In: Proc. of the 2nd Int'l Conference on Knowledge Discovery and Data Mining (KDD-96). (1996) 256–262
5. Benítez-Guerrero, E., Hernández-López, A.R.: The MineSP Operator for Mining Sequential Patterns in Inductive Databases. MICAI 2006: Advances in Artificial Intelligence, 5th Mexican International Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence **4293** (2006) 684–694 Springer-Verlag.
6. Ordonez, C., Cereghini, P.: SQLEM: Fast Clustering in SQL using the EM Algorithm. In: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, ACM (2000) 559–570
7. Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., Zaiane, O.: DBMiner: A System for Mining Knowledge in Large Relational Databases. In: Proc. of the Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96), Portland, Oregon (1996) 250–255
8. Wang, H., Zaniolo, C.: Using SQL to Build New Aggregates and Extenders for Object-Relational Systems. In: Proc. of the 26th International Conference on Very Large Data Bases (VLDB '00), Morgan Kaufmann Publishers Inc. (2000) 166–175
9. Chaudhuri, S., Narasayya, V., Sarawagi, S.: Extracting Predicates from Mining Models for Efficient Query Evaluation. ACM Transactions on Database Systems **29**(3) (2004) 508–544
10. Li, H., Lui, C., Orlowska, M.: A Query Systems for Object-Relational Databases. In: Proc. of the 9th Australasian Database Conference (ADC'98). (1998) 39–50
11. Holte, R.C.: Very Simple Classification Rules Perform Well on Most Commonly Used Datasets. Machine Learning **11** (1993) 63–91
12. Cendrowska, J.: PRISM: An Algorithm for Inducing Modular Rules. International Journal of Man-Machine Studies **27**(4) (1987) 349–370
13. Clark, P., Boswell, R.: Rule Induction with CN2: Some Recent Improvements. In: Proc. Fifth European Working Session on Learning, Springer (1991) 151–163
14. Mooney, R.J.: Encouraging Experimental Results on Learning CNF. Machine Learning **19**(1) (1995) 79–92
15. Cohen, W.W.: Fast Effective Rule Induction. In: Proc. of the 12th Int'l Conference on Machine Learning, Tahoe City, CA, Morgan Kaufmann (1995) 115–123
16. Furnkranz, J., Flach, P.: An Analysis of Rule Learning Heuristics. Technical Report CSTR-03-002, Department of Computer Science, University of Bristol (2003)
17. Raedt, L.D.: A Perspective on Inductive Databases. SIGKDD Explorations Newsletter **4**(2) (2002) 69–77
18. Sattler, K.U., Dunemann, O.: SQL Database Primitives for Decision Tree Classifiers. In: Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM'01), ACM Press (2001) 379–386
19. Hinneburg, A., Lehner, W., Habich, D.: COMBI-Operator: Database Support for Data Mining Applications. In: Proceedings of 29th International Conference on Very Large Data Bases. (2003) 429–439
20. Geist, I., Sattler, K.U.: Towards Data Mining Operators in Database Systems: Algebra and Implementation. In: Proceedings of 2nd International Workshop on Databases (DBFusion 2002). (2002)
21. Boulicaut, J.F., Klemettinen, M., Mannila, H.: Modeling KDD Processes within the Inductive Database Framework. In: Proc. of the First Int'l Conf. on Data Warehousing and Knowledge Discovery (DaWaK '99), Springer (1999) 293–302
22. Raedt, L.D., Jaeger, M., Lee, S., Mannila, H.: A Theory of Inductive Query Answering. In: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), IEEE Computer Society (2002) 123

A Performance Evaluation of Vertical and Horizontal Data Models in Data Warehousing

Victor Gonzalez-Castro[1], Lachlan M. MacKinnon[2,] David H. Marwick[1]

[1]Heriot-Watt University, School of Mathematical and Computer Sciences, Edinburgh, U.K
{victor,dhm}@macs.hw.ac.uk

[2] University of Abertay Dundee, School of Computing and Creative Technologies, Dundee, UK. l.mackinnon@abertay.ac.uk

Abstract. In Data Warehouse (DWH) environments administrators commonly face the following problems: exponential growth of the DWH; massive storage requirements; excessive long query response times; excessive Extract, Transform and Load data (ETL) times; big batch processing windows to backup and restore the environment; and increasing complexity of DBA tasks. We propose the use of alternative data models utilising a vertical approach to data storage (Binary-Relational, Triple Store-Associative) as opposed to the traditional horizontal storage approach used by the relational model, as a better approach for DWH environments. We present an impartial evaluation of these models using an extended TPC-H benchmark, the extensions taking into account ETL times, Storage requirements, and Backup / Restore times plus Queries response times. These extensions represent common issues in a production DWH environment, and to the best of our knowledge, are not considered in any existing benchmark.

1. Introduction

From the early days of data processing systems through the development of relational databases up to the present day, data has been stored and processed following a horizontal approach, where data is stored in records or relations with *n* number of fields or attributes. This approach has been called the N-ary storage model (NSM) by Copeland [6] and Direct Image Systems (DIS) by Date in [7].

Other researchers have focused on a vertical approach to store and manage the data and abandon the traditional record structure. The idea of vertical storage models is not new, but its application on Data Warehousing environments is novel.

In 1985 G.Copeland published a paper "A Decomposition Storage Model" (DSM) [6] which follows a *Binary-relational* approach. This approach formed the basis on which Boncz et.al. [4][22] developed MonetDB [11] Stonebraker et.al are building C-Store [21] and it is also the base model of SybaseIQ [22]. In the fundamental paper presented in 1988 by G. Sharman [18] they defined the *Triple Store Model* that was further developed and enriched by P.King in the Tristarp project [24]. S. Williams used this work as the basis to create his *Associative Model of Data* and thence build the SentencesDB [25].

© Jesús Olivares, Adolfo Guzmán (Eds.)
Data Mining and Information Systems.
Research in Computer Science 22, 2006, pp. 67-78

Another approach that abandons the record structure and follows a vertical approach is presented in [7] by Date, where the *Transrelational* ™ Model is described. The authors have implemented the essential algorithms and reported its behaviour in [11].

2. Identified Problems

The Relational Model is the predominant model used in commercial DBMS and of the vast majority of companies use Relational products. RDBMS have been demonstrated to be very successful in transactional environments. However RDBMS have since been used to create Data Warehouses without questioning if this is the best approach to manage this type of system. The following problems have been identified in Relational Data Warehouses:

- Data Warehouse grows at an exponential rate [8]
- The Data Base explosion phenomenon [14] is hard to control or eliminate
- Poor management of data sparsity [10]
- Low Data Density [10]
- Huge amounts of disk storage are required [26]
- The cost of storage and its maintenance are not negligible [8] and the storage itself could be up to 80% of the entire system cost [26]
- Long query response times is one of the main user complains, an average 17% of the sites using OLAP tools, with the worst case reaching 42% of the sites that use Oracle Discoverer [17]
- Long periods of time to Extract, Transform and Load (ETL) data [10]
- Big batch processing windows to Backup and Restore the environment [10]
- Increasing complexity of the Data Base Administration tasks

Different approaches (approximate queries [1], materialized views [2], Iceberg cubes [3], dwarf cubes [19], bit map indexes [20]) have been researched to tackle these problems but the fundamental reason has yet to be addressed properly: The horizontal approach used by the Relational model to store data is not the best approach for data warehouses. We propose the use of alternative models that abandon the traditional record structure and follow a vertical approach to store and manage data to be used in Data Warehouse environments. Boncz et al [4] have been working with this approach using a Binary-Relational approach, but it is necessary to benchmark other data models which use vertical approaches, and also all the daily tasks that are involved in a production data warehousing environment. To the best of our knowledge, any of the existing benchmarks consider the whole data warehousing cycle.

In order to do this we propose to extend the TPC-H benchmark and to consider the whole Data Warehousing cycle. The extended benchmark is explained section 3 and the results for each model are presented in section 5.

The Authors have published performance metrics of the behaviour of the alternative data models in [10], [11],[13].

3. Extension to the TPC-H Benchmark

In order to carry out the Performance Evaluation of the selected models in a Data Warehouse Environment, it was necessary to select a benchmark that: can be considered useful to measure different models; well defined; impartial; complete; and general accepted in the research and commercial communities. The following benchmarks have been analysed.

- The 007 benchmark [5] is designed for Object Oriented Data bases, none of the vertical models rely on OODBMS.
- The APB-1 (Analytical Processing Benchmark) created by the OLAP council favours products based on cubes and does not consider relational vendors, and as mentioned before the bigger enterprise Data Warehouses in production are based on Relational Products.APB-1 lacks wide acceptance and has been declining in the last few years [16]
- The Drill down Benchmark [4] designed by Boncz et. al. at the Institute for Mathematics and Computer Science Research at the Netherlands (CWI) been used to benchmark CWI's MonetDB vs. other RDBMS, but lacks wide acceptance.
- The Transaction Processing Council (TPC) has a suite of benchmarks that are widely accepted in industry and have been widely used by the research community. Each of the Benchmarks is targeted to different computing environments but focused on Relational DBMS.

As no existing benchmark satisfies all the criteria, we propose to extend the TPC-H benchmark [23] to consider the complete cycle of an Enterprise Data Warehousing Environment.

Two of the TPC benchmarks (H and R) are targeted to Decision Support Systems which are typical applications on a Data Warehousing Environment. The TPC-H benchmark was chosen because it offered the best constructs to evaluate pristine Data Models and not technology. The only type of auxiliary structures allowed in TPC-H are indexes on primary and foreign keys (it should be remembered that indexes are not part of the Relational model) which makes it more restrictive. In contrast, TPC-R allows the use of extended Relational technology, like indexes over any column, join indexes materialized views, pre-aggregates computation, and practically any technology that the DBMS can have to improve performance.

TPC-H was design to run on Relational based products; its schema consists of 8 tables and a workload of 22 queries which are typical queries in a DW environment. The queries are evaluated with different DW sizes, called the Scale Factor (SF) [23].

TPC-H considers times to load data and execute queries, but it does not consider other tasks that are important in DW environments. The proposed extensions follow the philosophy of the TPC-H Power test, where all queries are executed sequentially. In Table 1 a comparison between the metrics considered in the extended TPC-H and the original TPC-H benchmarks is presented. We refer to a database created without using any auxiliary performance structure as pristine mode.

Table 1. TPC-H and Extended TPC-H pristine models metrics

Metric	Extended TPC-H	TPC-H
Extraction times from transactional systems	Yes	No
Transformation times to conform to the target data model	Yes	No
Input files sizes measurement	Yes	No
Load data times in to the Data Warehouse	Yes	Yes
Database tables sizes after load (Data Base size)	Yes	Yes
Data Density Measurement	Yes	No
Queries execution times (Pristine mode)	Yes	No
Data Warehouse Backup time	Yes	No
Backup size	Yes	No
Data Warehouse restore time	Yes	No

There is another set of metrics that are useless while evaluating data models because they are technology improvements over the relational technology (indexing, statistics computation and query optimizer effects). However, these were measured to provide pragmatic performance metrics for real DW environments (Table 2).

Table 2. TPC-H and Extended TPC-H technology specific metrics

Metric	Extended TPC-H	TPC-H
Index creation times	Yes	Yes
Index sizes	Yes	Yes
Query times (with indexes)	Yes	No
Statistics computation times	Yes	Yes
Query times (with indexes & statistics)	Yes	Yes
Query times (with statistics without indexes)	Yes	No

4. Experimental Design

This experiment covers Relational, Binary-relational and Associative-TripleStore models. The results for Transrelational had been reported by the authors in [11]. All DBMSs are used with the default parameters, further tuning can be done to all DBMSs, but for the objectives of our research we wished to consider raw performance. The TPC-H data set was used with two different scale factors SF=0.1 (100MB), SF=1 (1GB). An ETL tool was developed to read the tables from a relational data source and generate Flat Files according to the structure and characteristics required for the Binary-relational and Associative models. The loading features of the tool are based on the bulk loaders of each target DBMS. The machine used for the experiment has 1 Pentium IV@1.60 GHz, 512 MB RAM, Cache size 256 KB, Bus Speed 100 MHz and O.S. Fedora Core 2 system V 2.4.9-12.

5. Results and Analysis

Extraction, Transformation times and flat files sizes can be considered constants for the tested models, due the fact that differences are small no matter the scale factor used, the results are in Table 3 where a linear behaviour is observed. Relational measures are included because when building Data warehouses it is common to extract data from Transactional systems built on relational DBMS and loaded into the Data Warehouse (in this case the TPC-H data base).

Table 3. Extraction,Transformation times and input files sizes

	Scale Factor	Relational	Binary-Relational	Associative
Extract. & Transf. time (min)	100MB	2.5	2.6	2.7
Extract. & Transf. time (min)	1 GB	27.6	29.5	27.5
Generated Flat File (MB)	100MB	102.8	98.3	100.0
Generated Flat File (MB)	1 GB	1049.6	1004.9	1021.7

After extraction the corresponding input files were loaded into each target DBMS. **Loading times** are in Fig. 1. For Associative with SF=1GB times for Orders and LineItem are estimated after loading 300,000 records (Orders) and 10,000 records (LineItem) because their actual processing times were too long. The best loading times are achieved for the Binary-relational model while the worst times were for Associative, being several orders of magnitude greater (note the use of logarithmic scales on the y-axis).

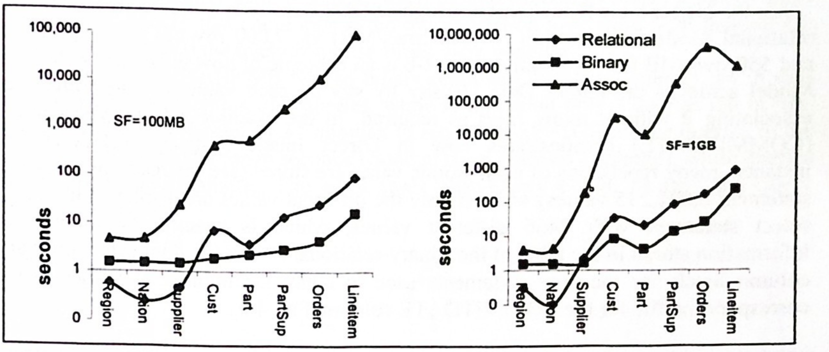


Fig. 1. Loading times

The best savings in time are achieved with the bigger tables. The Binary-relational Model instantiation requires an average of 76% less time to load the data set than the Relational Model instantiation.

When data is loaded into the Binary-relational Model instantiation, size reductions are achieved. The bigger reductions are in bigger tables, whereas in contrast the Associative model produces bigger table sizes (Fig. 2). The total data base size in the Binary-relational model instantiation has savings of 32% compared with the relational model instantiation, no matter the scale factor used, demonstrating linear scalability.

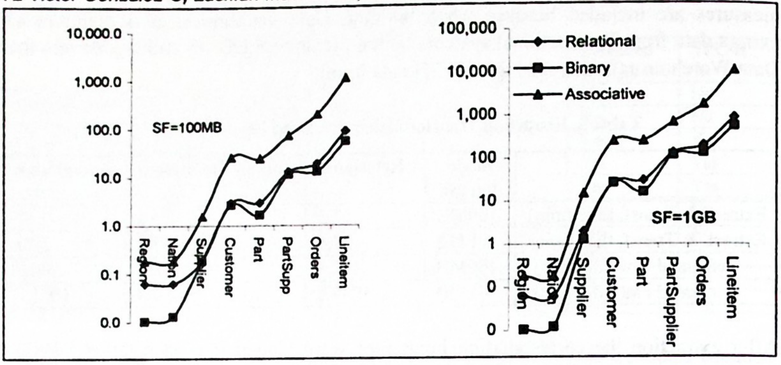


Fig. 2. Table sizes (MB)

Data Density is defined as the number of rows stored per Megabyte on disk. In Fig. 3 relative data density for each data model is presented. Observe the linear behaviour in this metric for each table. The Binary-relational model has the highest Data Density while Associative the lowest no matter the size of the table (Fig. 3).

For the bigger table (Lineitem), the highest Data Density is achieved by the Binary-relational Model instantiation (11,000 rows/MB) vs. 7,000 rows/MB for Relational and 550rows/MB for Associative. Fig. 4-b is an example of how the Binary-relational Model achieves the higher Data Density by storing each value just once and then associating it with as many rows as required. In the example a Date type column (COMMITDATE) demonstrates how in Direct Image systems, Relational for instance, many repetitions of each atomic value are stored (see the result of the select statement 6,001,215 values) while if only the different values are displayed in the 2nd select statement with 2466 different values, which is exactly the amount of information stored in the table of the Binary-relational Model (no duplicate values at a column level; see the unix statements used to count the number of values in the corresponding file for the COMMITDATE column, Fig. 4-b).

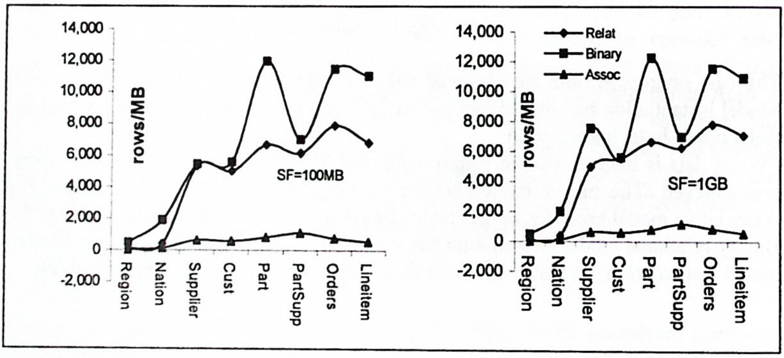


Fig. 3. Relative Data Density (rows/MB)

```
SQL> select count    (L_COMMITDATE)
        from LINEITEM;
-----------------
        6001215
SQL> select count(distinct L_COMMITDATE)
       from LINEITEM;
--------------------------
            2466
```

(a)

```
$ strings 34.theap >
L_COMMITDATE.lis
$ wc -l L_COMMITDATE.lis
2466 L_COMMITDATE.lis
$
```

(b)

Fig. 4. Data stored by relational (a) and binary-relational (b) models instantiations

Query Execution Times in Pristine Mode are shown in Fig. 5. As defined earlier pristine mode refers to loading data into the instantiations of the Relational and Binary-relational models without running indexing or statistics computation that have great influence over the RDBMS Query Optimizer, but that are not part of the relational model.

We did not evaluate more metrics for the Associative model, because it has the worst results in loading times and also in the size of the resulting Data warehouse, which are some of the key problems that we are trying to solve by using alternative data models.

The Query Language used was SQL, even though MonetDB offers a native query language called MIL that could thus avoid the translation time from SQL language to MIL, but even with it the query response times were superior.

On Fig. 5 with SF=100MB for the binary-relational model all queries ran in sub-seconds with the exception of 3 queries, while in relational only 3 queries ran in sub-seconds and Query 4 need 31.31 minutes to ran. With SF=1GB all the queries in the binary-relational model ran in seconds (none of them reached 1 minute), while queries in Relational ran in minutes. The worst cases were Query 4 with 14 days and Query 21 with 30 days.

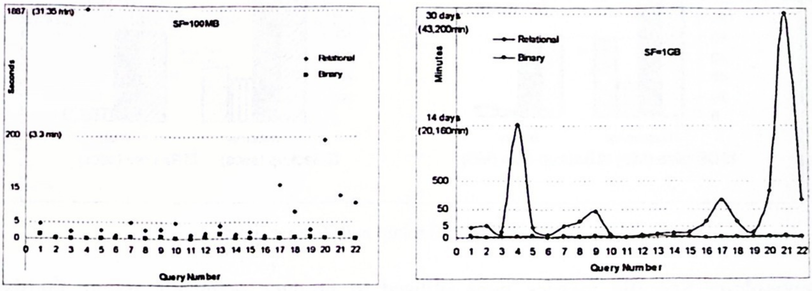


Fig. 5. Query Execution Times in Pristine Mode

The Total Query workload is the total processing time for the 22 queries, as can be seen in Fig. 6, the differences are considerable. With SF=100 MB, binary-relational required 9.5 seconds to process the 22 queries while relational required 36.1 minutes. With SF=1GB binary-relational took 3.8 minutes to process the 22 queries while relational took 43.8 days. These are the times using only relations (tables) as defined by the relational model and it is the way to compare model achievements, but of course in order to have pragmatic results we measure the queries using extended relational technology (indexes and statistics only) that are used extensively in relational products. The results are presented after the pristine metrics for backup and restore.

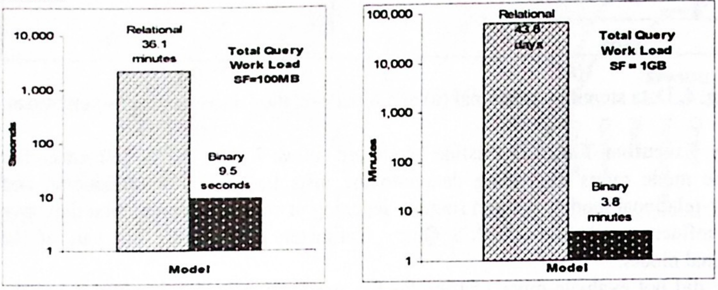


Fig. 6. Total Query Work Load in Pristine Mode

Another important fact that arose while running the queries is that the temporary space required to process the queries by the relational model instantiation grew to 1,487 MB, this size is more than the DB size itself (1,227 MB).

Backup and Restore times are other tasks in a DW environment that are frequently out of the main research focus in the Data Warehouse area. The results are presented in Fig. 7 for different Scale Factors, considering relational and binary-relational models; Binary-relational has better Backup/Restore times than Relational with around 80% less time, no matter the scale factor.

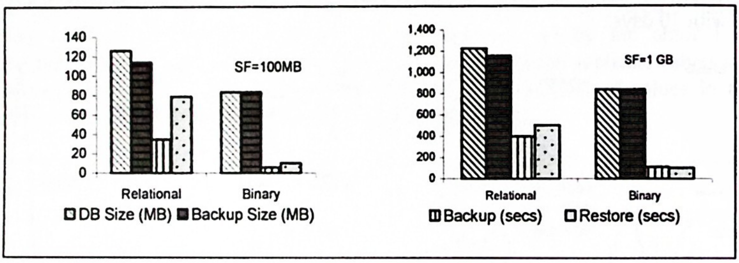


Fig. 7. Backup and Restore times

Technology Specific metrics were utilised to produce pragmatic results for the relational model. One way to improve performance on relational products is by indexing the tables and computing statistics but these need extra processing time and disk space. Fig. 8 shows the total processing time in Relational, which includes Data loading, Index creation and statistics computation. The optimization time (Index + Statistics) is not trivial compared with the required loading time. This optimizing time is not required by the Binary-Relational time. Apart from that, indexes required extra space to be store, Fig. 9 shows the disk space required by indexes.

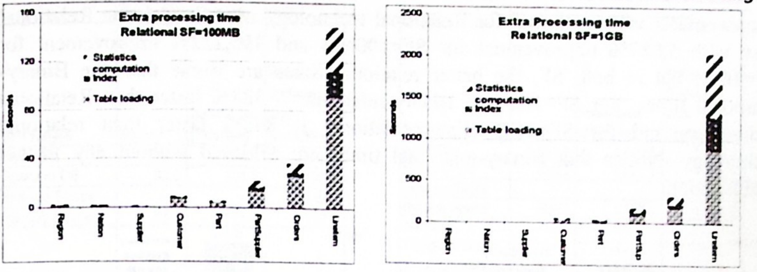


Fig. 8. Extra processing time to improve performance in Relational

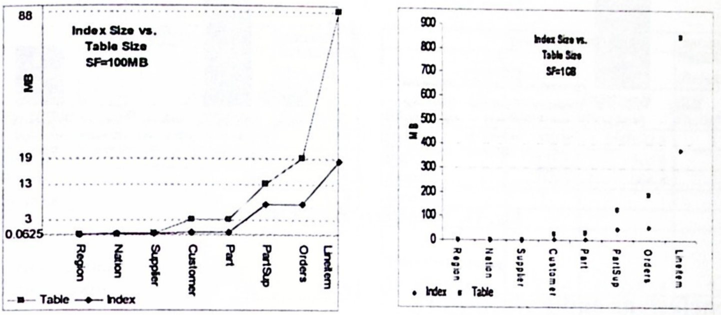


Fig. 9. Index space requirements

After creating indexes and computing statistics in order to help the Query optimizer to produce better execution plans, 3 scenarios were run: Queries executed only with indexes; Queries executed only with statistics; and Queries executed with both indexes and statistics, the results are in Fig. 10.

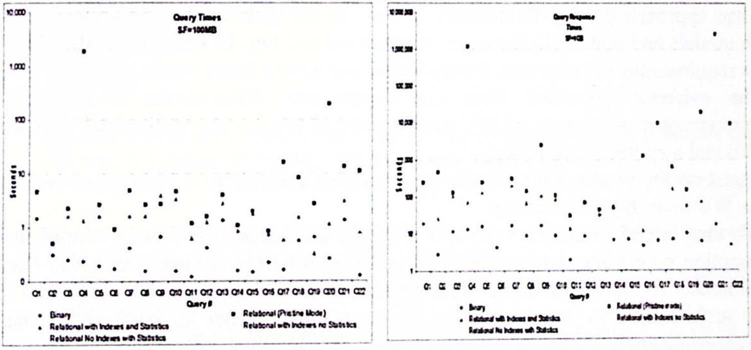


Fig. 10. Query response time with different scenarios to optimize relational

Fig. 11 shows the Total query work load considering the pristine mode of Relational plus the three scenarios described previously. In both scale factors huge improvements were achieved for Relational technology over the pristine Relational case with 5,887% improvement for SF=100MB and 195,222% improvement for SF=1GB; but in both SF, the better relational times are worse than the Binary-relational times. For SF=100 MB Binary-relational is 388% faster than Relational technology and for SF=1GB Binary-relational is 848% faster than relational technology. Notice that Binary-relational times are achieved without any further optimization.

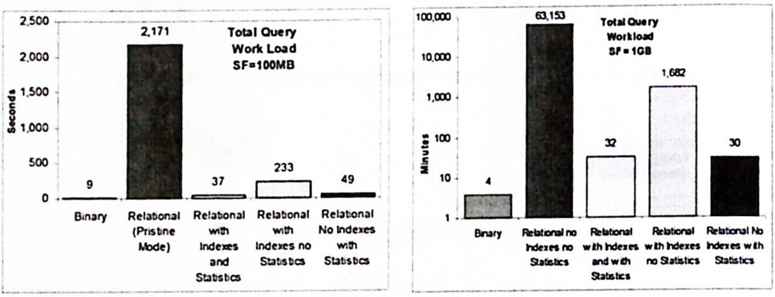


Fig. 11. Total Query Work Load

6. Conclusions and Future work

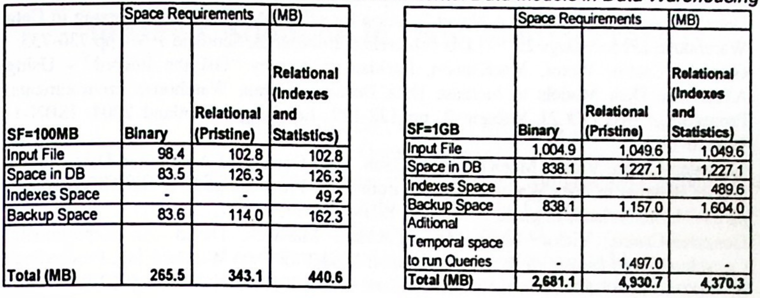
We are investigating the use of alternative data models which abandon the traditional N-ary approach or Record Structure to store and process data within Data Warehouse environments. According to the results achieved with our extended version of the TPC-H Benchmark, which considers a broader set of tasks that are common in Data warehousing environments, the future of N-ary horizontal storage approach models (Relational) can be certainly be challenged by a better approach which uses a vertical storage approach (Binary-Relational). In Fig. 12 the global space requirements for both models and both scale factors is summarized and Fig. 13 summarizes the global time requirements in both cases Binary-Relational has the better results.

The evidence presented from our experiments demonstrates a significant improvement in all aspects of DW performance of Binary-relational model over the traditional n-ry Relational model.

Based on the results achieved, the Binary-Relational Model is the best option for Data Warehousing environments.

We also intend to develop and test a hybrid architecture combining a relational transaction processing database as front end with a back-end binary-relational Data Warehouse. Assuming the same improvements in performance can be maintained in such architecture, we would propose this as a future model for commercial Data Warehousing applications.

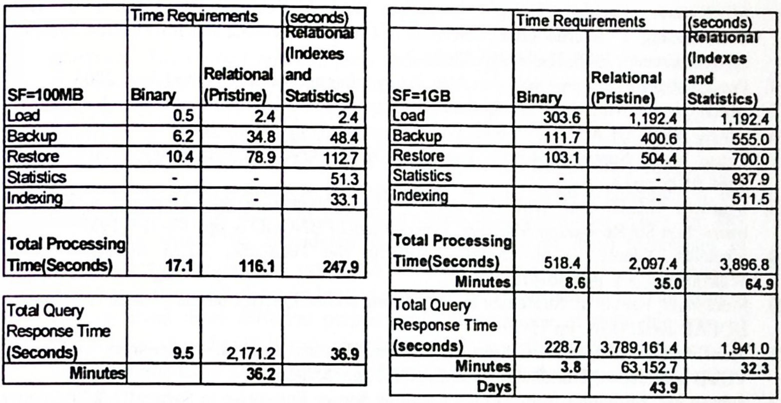
We also are going to test with real life Data Warehouses to verify that the results achieved by the Binary-Relational model are as good as they are with the synthetic data set of TPC-H.



	Space Requirements		(MB)
SF=100MB	Binary	Relational (Pristine)	Relational (Indexes and Statistics)
Input File	98.4	102.8	102.8
Space in DB	83.5	126.3	126.3
Indexes Space	-	-	49.2
Backup Space	83.6	114.0	162.3
Total (MB)	265.5	343.1	440.6

	Space Requirements		(MB)
SF=1GB	Binary	Relational (Pristine)	Relational (Indexes and Statistics)
Input File	1,004.9	1,049.6	1,049.6
Space in DB	838.1	1,227.1	1,227.1
Indexes Space	-	-	489.6
Backup Space	838.1	1,157.0	1,604.0
Aditional Temporal space to run Queries	-	1,497.0	-
Total (MB)	2,681.1	4,930.7	4,370.3

Fig. 12. Total Space Requirements



	Time Requirements		(seconds)
SF=100MB	Binary	Relational (Pristine)	Relational (Indexes and Statistics)
Load	0.5	2.4	2.4
Backup	6.2	34.8	48.4
Restore	10.4	78.9	112.7
Statistics	-	-	51.3
Indexing	-	-	33.1
Total Processing Time(Seconds)	17.1	116.1	247.9
Total Query Response Time (Seconds)	9.5	2,171.2	36.9
Minutes		36.2	

	Time Requirements		(seconds)
SF=1GB	Binary	Relational (Pristine)	Relational (Indexes and Statistics)
Load	303.6	1,192.4	1,192.4
Backup	111.7	400.6	555.0
Restore	103.1	504.4	700.0
Statistics	-	-	937.9
Indexing	-	-	511.5
Total Processing Time(Seconds)	518.4	2,097.4	3,896.8
Minutes	8.6	35.0	64.9
Total Query Response Time (seconds)	228.7	3,789,161.4	1,941.0
Minutes	3.8	63,152.7	32.3
Days		43.9	

Fig. 13. Total Time requirements

References

1. Acharya, Swarup et al. Aqua: A Fast Decision Support System Using Approximate Query Answers. Proceedings 25 VLDB, 1999. Edinburgh, Scotland. pp. 754-757.
2. Agrawal, D. et al. Efficient View Maintenance at Data Warehouses. ACM-SIGMOD 1997.pp 417.
3. Beyer, Kevin, et.al. Bottom-Up computation of Sparse and Iceberg CUBEs. Proceedings ACM-SIGMOD 1999. Philadelphia, USA. pp.359-370
4. Boncz, Peter et al. The Drill Down Benchmark. Proceedings of the 24th VLDB Conference, pp 628-632.
5. Carey, M. et al. The 007 Benchmark. ACM-SIGMOD 1993. Washington USA. pp 10-21.
6. Copeland, George P. Khoshafian, Setrag N. A Decomposition Storage Model. In Proc of the ACM SIGMOD Int. Conf. On Management of Data, pp 268-279, May 1985.
7. Date, C.J. An introduction to Database Systems. Appendix A. The Transrelational Model, Eighth Edition. Addison Wesley. 2004. USA. ISBN: 0-321-18956-6.

8. Datta, Anindya, et al. Curio: A Novel Solution for efficient Storage and Indexing in Data Warehouses. Proceedings 25th VLDB conference, Edinburgh, Scotland 1999. pp 730-733.
9. Gonzalez-Castro, Victor. MacKinnon, Lachlan. A Survey "Off the Record" - Using Alternative Data Models to increase Data Density in Data Warehouse Environments. Proceedings BNCOD 21 Volume 2. pp 128-129. Edinburgh, Scotland 2004. ISBN-1-904410-12-X
10. Gonzalez-Castro, Victor. MacKinnon, Lachlan. Data Density of Alternative Data Models and its Benefits in Data Warehousing Environments. Proceedings BNCOD 22 Volume 2. pp 21-24. Sunderland, England U.K. 2005. ISBN-1-873757-55-7.
11. Gonzalez-Castro, Victor. MacKinnon, Lachlan. Marwick, David. An Experimental Consideration of the use of the Transrelational Model for Data Warehousing. Proceedings BNCOD 23 pp 47-58. Belfast, Northern Ireland U.K. 2006. ISBN-3-540-35969-9.
12. MonetDB web site. http://monetdb/cwi.nl
13. Petratos, Panagogiotis. Michalopoulos, Demitrios (eds). Gonzalez-Castro, Victor. MacKinnon, Lachlan. Using Alternative Data Models in the Context of Data Warehousing. 1st International conference in Computer Science and Information Systems. Athens, Greece. 2005. ISBN-960-88672-3-1.
14. Pendse Nigel. Database explosion. http://www.olapreport.com Updated Aug, 2003.
15. Pendse, Nigel. Multidimensional data Structures. www.olapreport.com . March 19, 2001.
16. Pendse Nigel. OLAP Benchmarks. www.olapreport.com. March 2003.
17. Pense, Nigel. Summary Results from The OLAP survey 4. Microstategy 2005, COLL-0566 0105. pp12.
18. Sharman G.C.H. and Winterbottom N. The Universal Triple Machine: a Reduced Instruction Set Repository Manager. Proceedings of BNCOD 6, pp 189-214, 1988.
19. Sismanis, Yannis, et al. Dwarf: Shrinking the PetaCube. ACM SIGMOD 2002, Wisconsin, USA. pp 646-475.
20. Stockinger, Kurt et al. Strategies for Processing ad hoc Queries on Large Data Warehouses DOLAP 2002. USA. Pp 72-79.
21. Stonebraker, Mike, et.al. C-Store A Column Oriented DBMS. Proceedings of the 31st VLDB conference, Trondheim, Norway, 2005. pp. 553-564.
22. Sybase Inc. Migrating from Sybase Adaptive Server Enterprise to SybaseIQ White paper USA 2005.
23. TPC Benchmark H (Decision Support) Standard Specification Revision 2.1.0. 2002.
24. Tristarp project web site. www.dcs.bbk.ac.uk/~tristarp
25. Williams, Simon. The Associative Model of Data. 2nd Ed, Lazy Software Ltd. ISBN: 1-903453-01-1. 2003. www.lazysoft.com
26. Zukowski, Marcin. Improving I/O Bandwidth for Data-Intensive Applications. Proceedings BNCOD 22 Volume 2. pp 33-39. Sunderland, England U.K. 2005.

Requirements Engineering Techniques: Considerations for their Adoption in Data Mining Projects

José Gallardo[1], Claudio Meneses[1],
and Óscar Marbán[2],
[1] Departamento de Ingeniería de Sistemas y Computacion, Universidad Católica del Norte.
Av. Angamos 0610, Antofagasta, Chile
{jgallardo, cmeneses}@ucn.cl
[2]Facultad de Informática, Universidad Politécnica de Madrid.
Campus de Montegancedo s/n, Boadilla del Monte, Madrid, España.
omarban@fi.upm.es

Abstract. The correct and complete requisites specification is a key factor to the success for any project. Data Mining (DM) projects constitute decision-making support systems, and therefore the traditional Requirements Engineering techniques cannot be directly applied to them. This on-going research work presents an overview of the main models of the Requirements Engineering (RE) processes, and the most broadly used techniques in the development of the different phases involved in a RE model. Then the key issues that should be considered in the application of these techniques in Requirements Engineering processes for Data Mining projects are discussed. Also a proposition is done on how to structure the requirements in Data Mining projects from three different perspectives, in each one of them the type of information that should be captured is detailed, in order to particularly specify the requirements of DM projects, and generally in the decision-making support systems.

Keywords: Data Mining, Requirements Engineering Techniques, Requirements Elicitation.

1 Introduction

During the last years, a large number of Data Mining projects have been developed and it is expected that in the next decade this quantity will increase to 300%, as estimated in a report from GartnerGroup [6]. However, the execution of this type of projects faces serious problems, for example, they are never finished, or they are out of date or they are out of budget [23]. These problems are similar to those presented in the development of software applications [22], in what was named "software crisis", which was solved with the development of the Software Engineering discipline. In the Data Mining area, as a way of facing the generated problems, mainly due to a lack of standard or methodological guidelines for their development, a group of European companies which are pioneers in this type of projects (Teradata, SPSS, Daimler-Chrysler and OHRA), proposed in 1999 a reference guideline named CRISP-

© Jesús Olivares, Adolfo Guzmán (Eds.)
Data Mining and Information Systems.
Research in Computer Science 22, 2006, pp. 79-90

DM (Cross-Industry Standard Process for Data Mining) [5], that establishes a procedure for the systematic development of this type of projects. CRISP-DM is not the only guideline that has been proposed. Also, there are others, proprietary or open, like the one developed by SAS company, named SEMMA (Sample, Explore, Modify, Model, Assess) [19], DMAMC [10] or the 5 A's [16]. A survey developed by kdnuggets.com [12] shows that CRISP-DM is the most used one.

However, all the proposed methodologies for Data Mining projects development, lack of methods or techniques that allow us to appropriately educe the project requirements. More concretely, a mature process does not exist yet, that can be seen as a solid methodology. Although CRISP-DM establishes a group of activities that should be executed in the project, it does not establish with which techniques or models should be implemented.

In this paper, an overview of the Requirements Engineering (RE) processes and the main techniques used in each phase of the process is done, and a discussion of the main issues to be considered before adopting them in the construction process of the requirements document in a Data Mining project.

2 Models for the Requirements Engineering (RE) Process

Currently, Requirements Engineering is a technique used by many specialists for the construction of the Requirements Document, which should be the starting point for the correct design and implementation of a system, no matter its nature.

In the RE process, certain fundamental activities can be identified that should be developed to build a document for specifying the requirements. These activities are: requirement elicitation, analysis, specification and validation, and they serve as the foundation in the proposal of different models.

In [8] a model of the RE process is proposed, based on the activities of elicitation, specification and validation, as represented in Figure 1. The main elements of the outlined diagram shown in Figure 1 are briefly described in [2]:

- *Elicitation:* It is the process of acquiring the relevant knowledge, necessary to produce the requirements model in a problem domain, by means of the communication with clients, users of the system, and those involved with the project. After having obtained an initial group of requirements, they should be analyzed and represented in a technical language, in order to avoid inconsistencies and ambiguities.
- *Specification:* The elicitation process provides the entrance for the requirements specification process. The product is a specification model, or the models corresponding to different points of view. These models formalize the group knowledge or of the people involved in the project. The requirements specification also has a double purpose: on one hand, it serves as an agreement among the group involved in the project, for the problem to be solved, and on the other hand, it serves as a model for continuing with the following step.
- *Validation:* The validation is the activity that checks if the requirements specification is done according to the clients expectations. In this stage, it takes

place the integration and final validation of what was done in the previous stages, giving as final result the Requirements Document.

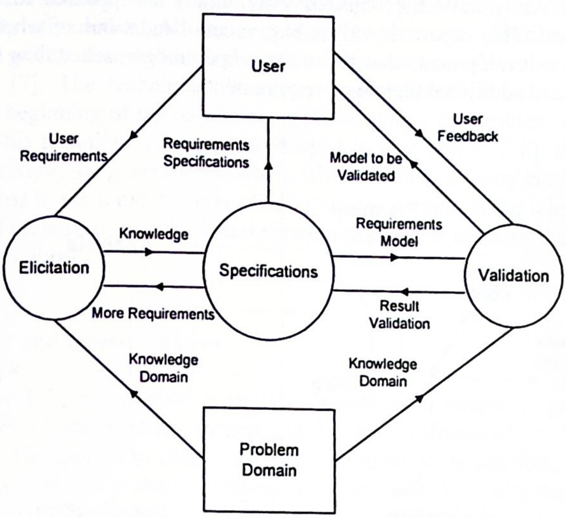


Fig. 1. General diagram of the Requirements Engineering Process

Sommerville [22] proposes another model for the Requirements Engineering process which is represented in Figure 2.

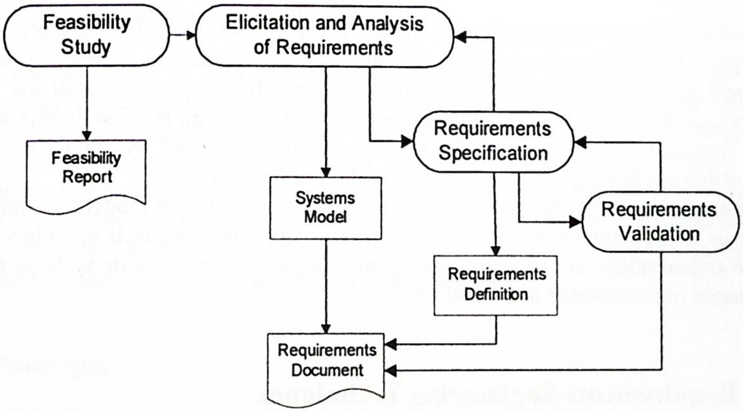


Fig. 2. The Process of Requirements Engineering

This process model incorporates a feasibility study as the first stage in the process, which represents a first approximation that receives as input, a brief description of the system to be developed and how this will be used by the organization. The objective of the feasibility study is to target aspects related to technical and economical feasibility of the project development, under the consideration that the project contributes to the organizational goals, and the way that the new system may be integrated to the existing systems.

When we already have the required information, we proceed to elaborate the feasibility report. This report should include recommendations of when to continue with the project development, changes in the scope, budget, scheduling of the system development, and additional high level requirements.

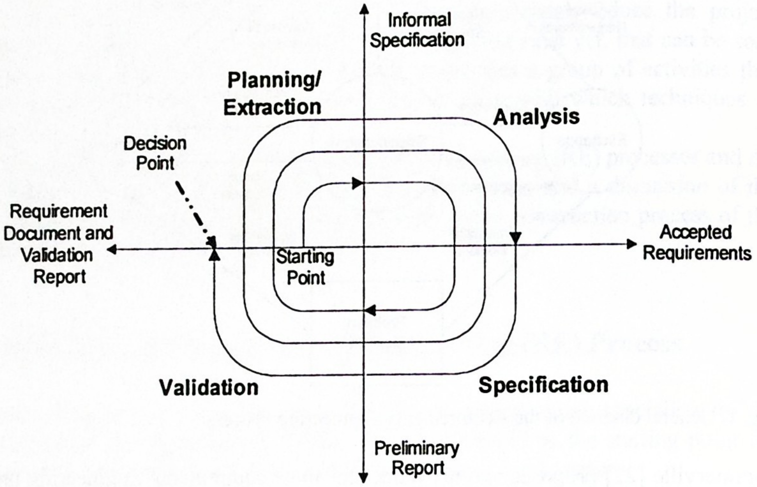


Fig. 3. Spiral Model of the RE process

A third model that we can analyze, is the one named “spiral model” [14], represented in Figure 3. The use of a spiral in this model is appropriate to represent that the different activities that constitute the model are repetitive activities, until the final acceptance decision of the specification requirements document is taken. In this sense, it is important to highlight that the process can be influenced by certain external factors that can bring us to an anticipating ending of the process.

Summarizing, it can be said that there does not exist a unique model that one can apply to all the requirements administration processes, since the facts and objectives of the organizations are different, and so are the type of projects or systems to be developed (operational or not operational).

3 Requirements Engineering Techniques

Independently of which RE process is used, the development of each phase is supported by a set of techniques which have been proposed and applied along time ([2], [4]). It follows a representative, although not exhaustive, review of the RE techniques most commonly used.

3.1 Brainstorming [1]

This technique is broadly used in different areas and it is basically based in the project team creativity stimulation. All the people involved should contribute with ideas, which should not be assessed until the end of the process, when there are no more contributions [7]. The technique allows us to generate different problem views, mainly at the beginning of the requirements phase, where the problem points of view are diffuse. This technique is usually developed in four phases [18]: the meeting or session preparation; the generation phase in which there is a freely contribution to all the ideas related to the topic; the consolidation phase, where all the relevant ideas are identified and organized; and the documentation phase, that contains the main aspects said and the conclusions.

3.2 Interviews and Questionnaires

This technique [13] is based on a series of questions to people or groups that are potential users of the system, carried out by the professional in charge of the requirements. The goal is to collect the more information as possible, which should not necessarily lead to a probable solution of the problem. Typically the questions are at a high level and the success of the use of this technique depends fundamentally in the interviewer's ability to get good answers and to interpret them correctly. In this technique one can identify three phases [17]: interviews preparation, the carry out of them, and the analysis of the data.

3.3 JAD (Joint Application Development)

JAD was developed by IBM in 1977, and is based on the following principles [18]: the groups' dynamics, the use of audiovisual help, the organized and rational process (meetings are developed during two to four days) and the documentation philosophy (in the meetings one works directly on the documents that are generated). This technique can be divided into two parts: the JAD/Plan whose purpose is to elicit and specify requirements, and the JAD/Design, in which the system design is approached. Its development is done in five phases: the project definition, the investigation, the preparation, the JAD session, and the final documentation.

3.4 Prototypes

Technique commonly used in the systems development. It allows the developer to build a model of the system that must be developed in the future [14]. The model is a simulation of the probable system, and subsequently is utilized by the end user. This technique allows getting the required information feedback so as to assess whether the system designed based in the requirements, allows the user to carry out its work in an efficient and effective way.

3.5 Hierarchical Analysis Process

The Hierarchical Analysis Process has as a fundamental objective, to solve quantitative problems, in order to facilitate analytic thought and metrics. This technique is divided into a series of tasks, such as:

- Finding the requirements that will be prioritized.
- Combining the requirements in rows and columns of a matrix.
- Doing comparisons of the requirements in the matrix.
- Adding the columns.
- Normalizing the sum of the rows.
- Calculating the averages.

These steps can be applied easily to a small quantity of requirements, nevertheless, for a large volume this technique is not the most adequate one.

3.6 Use Cases

This technique is based on the definition of certain functionalities that are expected from a system and that allow it to interact with something or someone. These functionalities are called *use cases*. A use case can be defined as: "a textual narrative description of the processes of a business or system" [15], in which the system is considered a black box and from which the actors obtain answers [4]. As actors, will be understood the people or other systems that interact with the system whose requisites are being described [20]. Initially, this technique was proposed in [11]. From this publication, the most recognized specialists in Object-oriented methods have agreed in considering the *use cases* as an excellent way of specifying the external behavior of a system. Due to this, the notation of the *use cases* was incorporated to the standard language of modeling UML (Unified Modeling Language) [3].

4 Considerations to Apply RE Techniques in DM Projects

The establishment of the requirements in a Data Mining project constitutes a fundamental task in order to specify and validate the services that the system should provide, as well as the restrictions with which the work itself should be developed. This process is essential, because the most common and costly errors that have to be corrected are a result of an inadequate Requirements Engineering process. Previously to the requirements specification is necessary to consider the following aspects:

i) To identify and know the objectives of the business (business model). Any Data Mining project should have as a final objective the generation of some type of benefit for the organization, either improving the efficiency of the business processes or discovering new sources of improvement.
ii) To identify the problem domain. In this context, the identification of the problem domain will allow to specify the area in which the Data Mining project will take

place. As an example, a general classification of problems types that a Data Mining project allows us to face are the following:

1. *Marketing.* It considers getting the greater quantity of related information to the business clients in order to establish potential clients, to determine who will buy, when and where, to improve the relationship with the clients, etc. That is, the result of the Data Mining project should allow planning in the best way the future marketing campaigns of the company.
2. *Market basket.* It considers the determination of product purchase patterns in the retail area. These patterns may trigger the initiation of guided promotion campaigns, new physical disposition of the items, multi-item pack offerings, offerings and promotions considering time patterns (e.g., day of the week), etc.
3. *Risks reduction.* In this case, Data Mining will allow an automatic evaluation of risks, base on previous experiences.
4. *Frauds detection.* Users will be able to obtain models that would allow discovering possible frauds in base to anomalous behaviors detection models.
5. *Quality control.* It considers the definition of models that will allow the precise and anticipated detection of faulty products.

iii) To map the business problem into a Data Mining problem. After identifying the domain of the problem, it should be mapped into a Data Mining problem, that is, it should be considered that each type of application of Data Mining is related to one or more type of tasks. The main types of Data Mining tasks are:

1. *Association.* It is basically used to discover relations among attributes. That is, the idea is to discover rules that identify behavior patterns, and it is largely used in the market basket domain.
2. *Time Sequences.* It is similar to the association task, but in this task the time variable is incorporated.
3. *Classification.* This task uses a collection of data to develop a model that will be used to classify new unseen data. The predicted variable is a nominal one.
4. *Regression.* This task is similar to classification, but in this case the predictive variable can take possible unlimited numeric values.
5. *Clustering.* This task is utilized in typical segmentation applications, and consists in a division of the data into collections of related data or groups, in which the data in each identified group are similar, i.e., they share a number of similar characteristics.

iv) To define the generic components that a Data Mining requirement should consider, such as:

1. *Data component.* This component should respond to the question on what data and with what structure (model of data) they are needed, in function of the algorithmic technique that will support the application of the Data Mining task.
2. *Interface component.* It should answer the question about the format in which will be visualized or presented the project results.
3. *Usability and correctness component.* It considers the way in which the project results should contribute to the business and user objectives, and the degree of accuracy that will provide the model.

4. *Understandability Component.* It considers the way in which the Data Mining model can be understood and will allow justifying the achieved results.
5. *Resources Component.* It should consider the available resources for the project, such as personnel (e.g., business expert, data expert, Data Mining team, technical aid), and hardware and software platforms.
6. *Not functional components.* They are reutilization requirements, development environments, results availability and quality (delivery times), security and legislation.

5 A Model of the RE Process for Data Mining Projects

Based on the considerations mentioned before, now is the turn of applying a RE model to construct the requisite document. Determining the needs for the project development is a complex process, and there doesn't exist unique and standard techniques that provide a framework for development that guarantees good results. Considering that a Data Mining project is essentially a decision support system, in which is important the adequate problem domain understanding, the construction of the business and data models, and the basic elements seen in the RE processes mentioned before, it is proposed a RE process model for data mining projects which is shown in Figure 4.

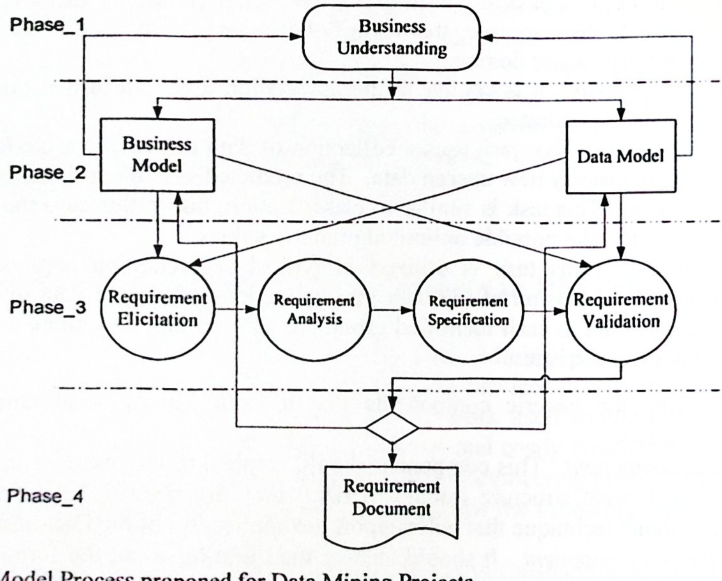


Fig. 4. RE Model Process proponed for Data Mining Projects.

This model is structured in the sequence of phases that should be carried out in order to generate the requisite document. The first phase, business understanding, considers the knowledge of the problem domain, the contexts, the organizational structure, the decision-making levels, and the own vocabulary of the business domain. In the second phase, the business decisional model and the data model are built, which are the input necessary to tasks to be performed in the third phase (requirements elicitation, analysis, specification, and validation). The fourth and final phase corresponds to the construction of the requisite document.

The selection of RE techniques to use for the execution of each phase of the RE proposed model, and the success of the results reached will depend finally on the clients/users, and on the development team and its experience in similar projects.

6 Structuring Data Mining Requirements

Data Mining projects involve different stakeholders, who express the purpose of the project, indicate the direction of the activities, and define the expectations in terms of information goals for the organization, in a similar way to the development of a project based on data analysis, such as a Data Warehouse system [21]. Therefore, the requisites generation in Data Mining projects should consider different perspectives, corresponding to different stakeholders, each one associated to different abstraction levels in the organization. Thus, we can distinguish three levels of abstraction in the generation of requisites for Data Mining: business point of view, user perspective, and technical staff (development team) view.

From the business point of view, the requisites of Data Mining should consider the identification of at least the following information:

- Business goals
- Business opportunities
- Business necessities to be satisfied
- Stakeholders and the generated value for them
- Criteria to decide to start / non-start the project
- Problem domain background
- An appropriate business case to illustrate the project impact

From the user perspective, we should be able to identify and describe the tasks that the user should be able to perform or improve using the project results. Among other information, the user requirements should consider:

- The user specific goals
- Business questions that may be answered with the project
- The process and business rules related to the user requirements
- The opportunities to improve certain business processes
- The way that the results should be used (user case)
- The way that the results should be tested (test case)
- The user profiles associated to the project results.

From the development people side (data miners), the requisites must have enough information about the specific tasks to be performed by them, the data sources, as well as the results attributes or characteristics. The technical requisites must be explicitly linked to business and user goals, specified before. Technical requisites should consider mainly:

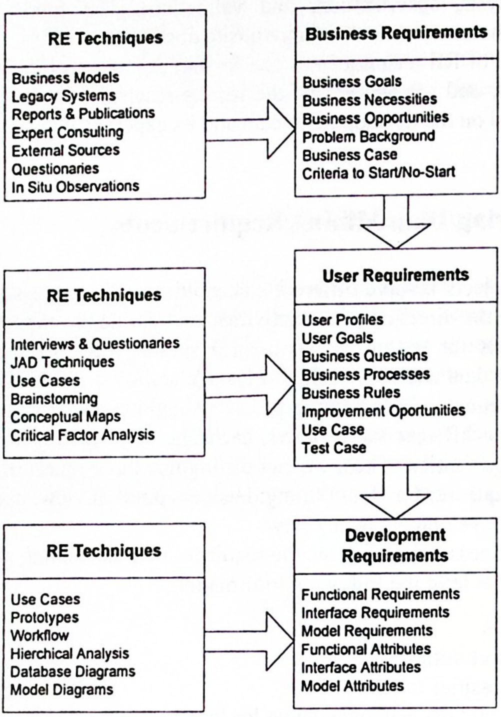


Fig. 5. Requisites structure for Data Mining and the corresponding RE techniques

a) *Functional Requirements,* which considers the tasks specification as extraction, loading, integration, cleansing and data transformation, as well as the identification of the type of task, data exploration, model generation to be carried out, and the process and results documentation.

b) *Interface Requirements,* that considers the characteristics specification of the interfaces to be developed during the project, such as database, software, and hardware interface

c) *Model Requirements,* which considers the identification of the type of model, its characteristics of understandability, ways to evaluate correctness and consistency of the generated model.

d) *Functional attributes,* that considers the definition of operational attributes, of performance, and of security of necessary tasks to be carried out by the development team of the project.

e) *Interface attributes,* that considers the definition of usability conditions and of form of the different interfaces required in the project.

f) *Model attributes,* that considers the establishment of criteria to generate the models and to evaluate its validity and acceptance.

Thus, we can summarize the different perspectives of the stakeholders involved in the definition of a requirement, to define a format for requirements acquisition in Data Mining projects, and identify associated RE techniques for each different vision or stakeholder perspective, as shown in Figure 5.

6 Conclusions

A Data Mining project has an exploratory nature, of decisional type and not of operational type that seeks to contribute value to the business through its data. Normally, the decision making processes are not well structured, with the consequent and inherent difficulty for modeling them. A Data Mining project is essentially approached to the comprehension and exploration of data and therefore, the requirements should be focused in determining the way the data influence in making decisions or in which way they impact in the decisions that are taken. Thus, to establish a unique criterion for the selection of the most appropriate techniques for its application in the different phases of the Requirements Engineering process is somewhat complex.

Our proposal constitutes a first step in order to establish a process model for the requirements definition in this type of project, under the consideration, that the different development processes models for Data Mining projects, such as CRISP-DM [5], SEMMA [19] or DMAMC [10], in spite of the fact that they present the requirements capture like a task to be performed, they do not indicate how to carry out this task, what techniques to utilize, neither the formats for the outputs that are proposed. Further work needs to be done in order to evaluate whether the proposed techniques are appropriate and useful to capture requisites in Data Mining projects.

The application of RE techniques in the proposed model should probable consider adaptations, and other aspects such as learning facility, cost, quality, completeness, time restrictions for its applicability, available personnel, etc.

Finally, we consider that it doesn't exist a valid and unique Requirements Management process and technique that can be applied in all the organizations and in all type of projects. Each organization should select or develop its process, in agreement with the type of product to be generated, to the organizational culture, and the level of experience and ability of the people involved in the Requirements Engineering process. Nevertheless, the identification of different perspectives of the same problem (obtaining of requirements in a Data Mining project) seeks to structure the process of RE, such that particular techniques can be applied to each perspective/stakeholder identified.

The next steps in this research work consider a detailed and formal description of the process here proposed the definition of criteria to evaluate the effectiveness of the proposed model, and its application to real Data Mining projects, in order to validate the model. This will allow us to illustrate, to refine, and to validate the proposed structure of Data Mining requirements.

References

1. Arango J. "Tormenta de Ideas", Colombia. Universidad EAFIT, 2002. D [en línea], disponible en: http://www.eafit.edu.co/tda/boletin/TORMENTA%20DE%20IDEAS.htm
2. Bahamonde J.M., Rossel R. "Un Acercamiento a la Ingeniería de Requisitos", Universidad Técnica Federico Santa María, 2003.
3. Booch, G., J. Rumbaugh, y I. Jacobson. "The Unified Modeling Language User Guide", Addison–Wesley, 1999.
4. Choque Guillermo, "Ingeniería de Requisitos", artículo de divulgación, Ingeniería de Software, Universidad Mayor de San Andrés, 2003.
5. Chapman P., (NCR), Clinton J., (SPSS) Kerber R., (NCR), Khabaza T. (SPSS), Reinartz T. (DaimlerChrysler), Shearer C. (SPSS), and Wirth R. (DaimlerChrysler). "CRISP-DM 1.0 step-by-step data mining guide", Technical report, 2000.
6. Dilauro, L. "¿What's nest in monitoring technology?", Data Mining finds a calling in centers", May 2000.
7. Gause, D. C. and G. M. Weinberg. "Exploring Requirements: Quality Before Design". Dorset House, 1989.
8. Jaap Gordijn, "Value-based Requirements Engineering Exploring Innovative e-Commerce Ideas", VRIJE UNIVERSITEIT, 2003.
9. Houghton Mifflin Company. "The American Heritage Dictionary of the English Language", 3rd Edition, Houghton Mifflin Company, Electronic Version. 1992.
10. http://www.isixsigma.com, "consulta sobre metodología 6-Sigma" [en línea], disponible en: http://www.isixsigma.com/sixsigma/six_sigma.asp.
11. Jacobson, I., M. Christerson, P. Jonsson, y G. Övergaard. "Object–Oriented Software Engineering: A Use Case Driven Approach ", Addison–Wesley, 4ta. edición, 1993.
12. Kdnuggets, http://www.kdnuggets.com, "consulta sobre metodologías utilizadas en Data Mining" , http://www.kdnuggets.com/polls/2004/data_mining_methodology.htm
13. Komer, P. "Dirección de la Mercadotecnia", Séptima Edición. España. Prentice Hall, 1993.
14. Kotonya G. and Sommerville I. "Requirements Engineering. Processes and techniques", USA. J. Wiley, 1998.
15. Larman C. "UML y Patrones, introducción al análisis y diseño orientado a objetos", Ed. Prentice Hall, 1999.
16. Martínez de Pisón Ascacibar, F.J. "Optimización mediante técnicas de minería de datos del ciclo de recocido de una línea de galvanizado", Tesis Doctoral, Universidad de La Rioja, Servicio de Publicaciones, 2003.
17. Piattini, M. G., Calvo-Manzano, J. A., Cervera, J., Fernández, L. "Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión". Rama, 1996.
18. Raghavan, S., G. Zelesnik, y G. Ford. "Lecture Notes on Requirements Elicitation", Educational Materials CMU/SEI–94–EM –10, Software Engineering Institute, Carnegie Mellon University, 1994. http://www.sei.cmu.edu
19. SEMMA, http://www.sas.com/technologies/analytics/datamining/miner/semma.html
20. Scheneider, G. & Winters. Applying Use Cases: a Practical Guide. Addison–Wesley, 1998.
21. Schiefer, J. List, B., Bruckner, R.M., A Holistic Approach for Managing Requirements of Data Warehouse Systems, Proceedings of the Eighth Americas Conference on Information Systems, 2002.
22. Sommerville, I., "Ingeniería de Software", 6ta. Edición, Ed. Addison Wesley, 2002.
23. Zornes A., META Group Research-Delta Summary, "The Top 5 Global 3000 Data Mining Trends for 2003/04", Enterprise Analytics Strategies, Application Delivery Strategies, Delta, 2061, March 26, 2003.

Author Index
Índice de autores

Editorial Board of the Volume

Comité editorial del volumen

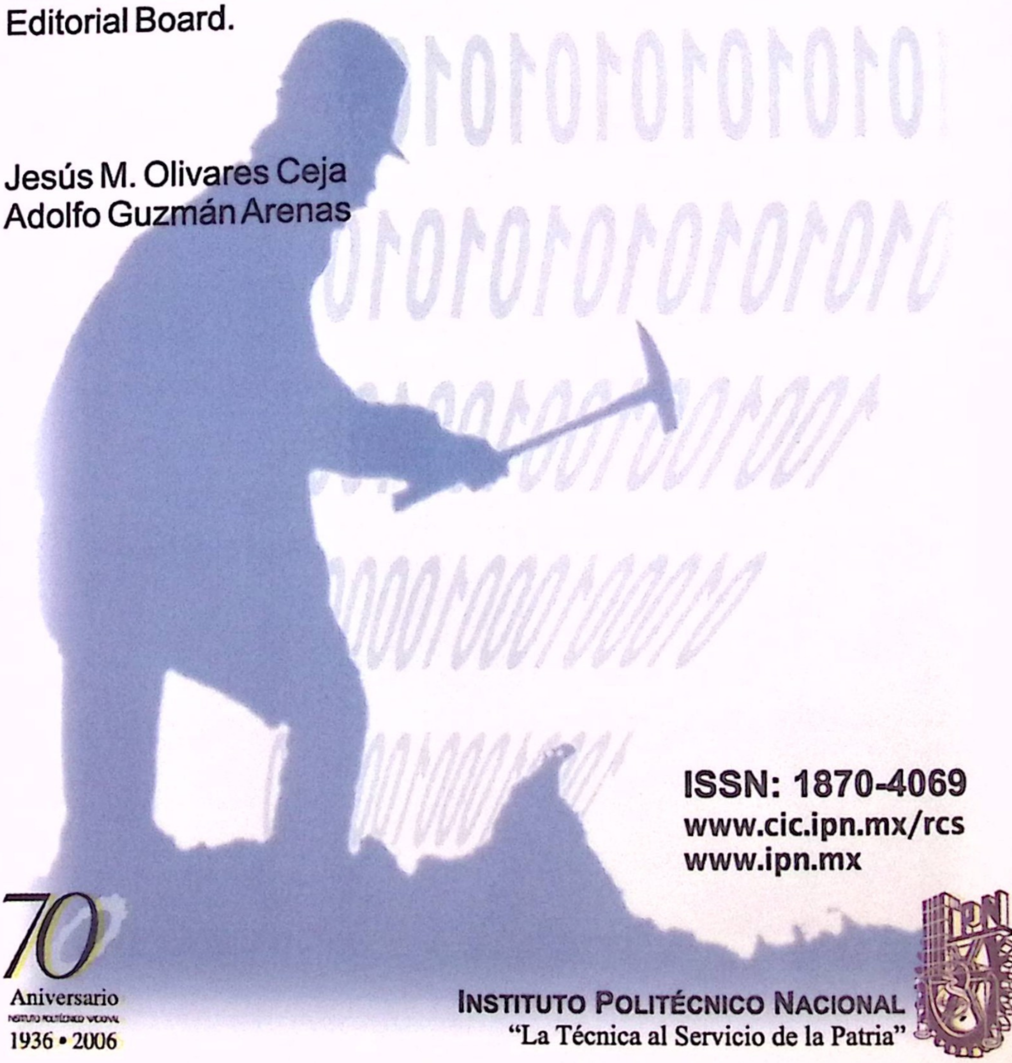
Impreso en los Talleres Gráficos
de la Dirección de Publicaciones
del Instituto Politécnico Nacional
Tresguerras 27, Centro Histórico, México, D.F.
Noviembre de 2006.
Printing 500 / Edición 500 ejemplares.

Information Systems is a field with many applications that current trends deals with semantic information processing and the Internet support. Data Mining deals with the problem of finding interesting patterns in information that currently is being produced by transactional information systems.

This special issue presents original research papers on semantic information processing applied to Information Systems and Data Mining.

We cordially thanks all people involved in the preparation of this volume that include paper authors, reviewers, editors and the Editorial Board.

Jesús M. Olivares Ceja
Adolfo Guzmán Arenas



ISSN: 1870-4069
www.cic.ipn.mx/rcs
www.ipn.mx

70 Aniversario
Instituto Politécnico Nacional
1936 • 2006

INSTITUTO POLITÉCNICO NACIONAL
"La Técnica al Servicio de la Patria"