Graduated Errors in Approximate Queries using Hierarchies and Ordered Sets

Adolfo Guzman-Arenas, Serguei Levachkine

Centro de Investigación en Computación, Instituto Politécnico Nacional.

07738 Mexico City, MEXICO

a.guzman@acm.org, palych@cic.ipn.mx

Abstract. Often, qualitative values have an ordering, such as (very-short, short, medium-height, tall) or a hierarchical level, such as (The-World, Europe, Spain, Madrid), which are used by people to interpret mistakes and approximations among these values. Confusing Paris with Madrid yields an error smaller than confusing Paris with Australia, or Paris with Abraham Lincoln. And the "difference" between very cold and cold is smaller than that between very cold and warm.

Methods are provided to measure such confusion, and to answer *approximate queries* in an "intuitive" manner. Examples are given. Hierarchies are a simpler version of *ontologies,* albeit very useful.

Queries have a blend of errors by order and errors by hierarchy level, such as "what is the error in confusing very cold with tall?" or "give me all people who are somewhat like (John (*plays* baseball) (*travels-by* water-vehicle) (*lives-in* North-America))." Thus, retrieval of approximate *objects* is possible, as illustrated here.

1. Introduction

The type of mistakes and misidentification that people make give clues to how well they know a given subject. Confusing Ramses with Tutankamon is not as bad as confusing Ramses with George Washington, or with Greenland. Indeed, teachers often interpret these mistakes to assess the extent of the student's learning.

The paper formalizes the notion of *confusion* between elements of a hierarchy. Furthermore, this notion is extended to hierarchies where each node is an ordered set. *These are the main trusts of the paper.*

Some definitions follow.

Qualitative variable. A single-valued variable that takes symbolic values. ♦ As opposed to numeric, vector or quantitative variables. Its value cannot be a set, although such symbolic value may represent a set. Example: the qualitative variables (written in *italics*) *profession, travels-by, owns, weighs;* the symbolic values (written in normal font) lawyer, air-bone-vehicle, horse, heavy.

Partition. K is a partition of set S if it is both a covering for S and an exclusive set. ♦ The members of K are mutually exclusive and collectively exhaust S. Each element of S is in exactly one K_j.

Ordered set. An element set whose values are ordered by a < ("less than") relation. ♦ Example: {short, medium-length, long}. Example: {Antartica, Australia, Brazil, Ecuator, Nicaragua, Mexico, Germany, Ireland, Iceland}, where the relation "<" is "South of".

1.1 Hierarchy

For a node n in a tree, relations **father_of**(n)**, son_of**(n)**, brother_of, ascendant_of...** are defined, as expected. ♦

A **hierarchy** H is a tree whose root is a set S, and, if a node has sons, then these sons form a partition of their father. ♦ This paper deals with hierarchies whose set S is formed by symbolic values. Often, we give names (symbolic values, strings) to the different subsets of S. Often, we name the hierarchy H after the set S, and we speak of "the hierarchy S". Example: The Hierarchy H_1 of means of travel or transportation vehicles, whose root is the set S = {animal, foot, bike, motor-bike, 2-seat-car, 4-seat-car; van, bus, train, boat, ship, helicopter, airplane} is shown in Figure 1.

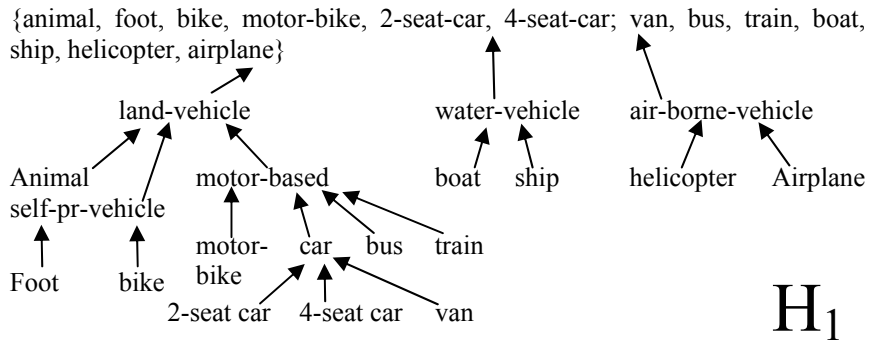

Figure 1. A hierarchy H_1 of transportation vehicles. Some qualitative values, like air-borne-vehicle, represent sets: {helicopter, airplane} in our example

Hierarchies make it easier to compare qualitative values belonging to the same hierarchy (§2), and even to different hierarchies [COM in 4, 9].

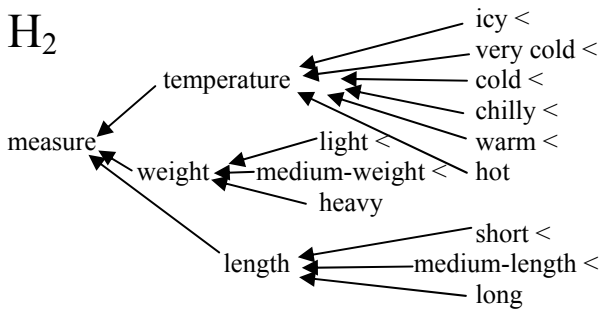

Figure 2. A hierarchy having some ordered sets: (short < medium-length < long), (light < medium-weight < heavy), (icy < very cold < cold < chilly < warm < hot)

A **hierarchical variable** is a qualitative variable whose values are nodes of a hierarchy. ♦ The data type of a hierarchical variable is hierarchy.

Example: *travels-by,* whose values are nodes of H_1 (figure 1). Example: *weighs,* whose values are nodes weight, light, medium and heavy of H_2. Note: hierarchical variables are single-valued. Thus, a value for *travels-by* can be water-vehicle, but not {boat, ship} although water-vehicle represents {boat, ship}.

It is also possible for a hierarchy to have some nodes that are ordered nodes. Example: Hierarchy H_2 of figure 2.

1.2 Previous related work

Hierarchies are used in data warehousing and data mining; see, for instance, the H-sets of [1]. The paper [7] enlarges these notions with greater mathematical background. [6] studies hierarchies where the relative proportion of each set in its father set is known. On the other hand, [9] deals mainly with *ontologies,* more elaborate data structures used for knowledge representation, of which CYC [2] was an early attempt to build an ontology for common concepts. A companion paper in this book [4], matches similar concepts in different ontologies. The thesis [8] describes how to map concepts from one ontology to another. A practical use of hierarchies is Clasitex [3], which finds the themes of an article written in Spanish or English. It uses the concept tree, and a word (not in the tree) *suggests the topic of* one or more concepts in the tree. BiblioDigital [5], a recent development, uses a large taxonomy (although not a hierarchy) to classify text documents.
Work described here is similar to Pattern Classifiers, but these classify *objects* according to the values of their properties, whereas hierarchies help to classify these *values*, when they are non-numeric.

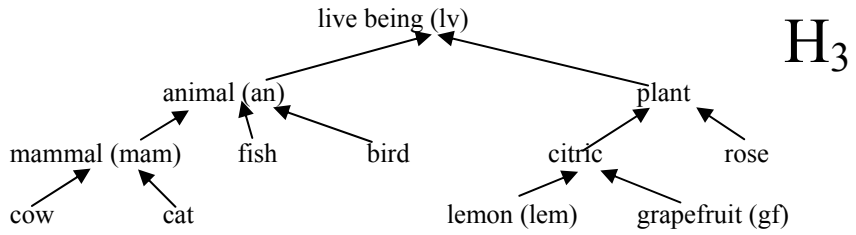

Figure 3. A hierarchy H_3 of living creatures (*lv*). *an* stands for animal; *mam* for mammal; *lem* for lemon, and *gf* for grapefruit. See table 1 below

2. Confusion in hierarchies

Who was the first Emperor of Mexico? "Agustin de Iturbide*"* is the correct answer; "Maximilian of Hapsburg" is a close miss, "Benito Juarez" a fair error, and "Mexico City" a gross error. What is closer to a cat, a dog or an orange? Can we measure these errors or similarities? Yes, with hierarchies of symbolic values.

2.1 Confusion in using r instead of s, for a hierarchy H

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 measure conf, move from r to s in the hierarchy, and count the *descending* links from r to s, the replaced value. conf is not a distance, nor ultradistance.
Example: conf(r, s) in the hierarchy of Figure 3 is given in Table 1.

Table 1. conf(r, s), Confusion in using r instead of s, for hierarchy H_3. r runs down, while s runs to the right. Thus, the black **2** is the confusion of using an animal (*an*) instead of a cow, while the confusion of using a cow instead of an animal is 0. Values (nodes) of H_3 are ordered *width-first* in the table

	lv	an	plant	mam	fish	bird	citric	rose	cow	cat	lem	gf
lv	0	1	1	2	2	2	2	2	3	3	3	3
an	0	0	1	1	1	1	2	2	**2**	2	3	3
plant	0	1	0	2	2	3	1	1	3	3	2	2
mam	0	0	1	0	1	1	2	2	1	1	3	3
fish	0	0	1	1	0	1	2	2	2	2	3	3
bird	0	0	1	1	1	0	2	3	2	2	3	3
citric	0	1	0	2	2	2	0	1	3	3	1	1
rose	0	1	0	2	2	2	1	0	3	3	2	2
cow	0	0	1	0	1	1	2	2	0	1	3	3
cat	0	0	1	0	1	1	2	2	1	0	3	3
lem	0	1	0	2	2	2	0	1	3	3	0	1
gf	0	1	0	2	2	2	0	1	3	3	1	0

conf resembles our sense of "closeness" between these concepts. Examples:

conf (citric, plant) = 0; if I use citric instead of plant, the confusion is 0, since citrics are plants.

conf (plant, citric) = 1; giving a plant when I wanted a citric is a "small" error; giving a cow when I wanted a citric is a larger error (value 2). Using these gradations in errors, the paper later will produce responses to queries that are "very similar to x", or "somewhat similar to x", where x is a node or a predicate.

The confusion among two brothers, such as cow and cat, is 1. The confusion in using a son instead of its father is 0; the confusion in using a father instead of its son is 1. conf is not a symmetric function. In the next section we modify the confusion among two brothers to be a number ≤ 1, for brothers that belong to an ordered set.

Points to ponder. The confusion in using a live being instead of a plant is 1. Thus, conf (animal, plant) = conf (mammal, plant) = conf (cow, plant) = 1. This may seem odd, but it is not: cow, mammal, and animal are examples of live beings, and the confusion of using a live being instead of a plant is 1. Another example will perhaps be more convincing: Say that "wine" and "beer" are brothers, so that conf (wine, beer) = 1: if I am given wine when I wanted beer, the confusion is 1. But this is exactly the same confusion if I am given red wine instead of beer, or Riesling wine instead of beer, or chilled dry Riesling wine vintage 1999 instead of beer. It is always 1, no matter how "specialized" the wine or the live being is.

In the other direction, conf (citric, plant) = 0: if I am given a citric when I want a plant, the confusion is 0, because a citric *is* a plant. Another example: If I

am given a cold beer when I want a beer, the confusion is 0. Similarly, conf (Corona_beer, beer) = conf (chilled_Corona_beer, beer) = 0, since all these "specialized" types of beer are, nevertheless, beer.

Thus, conf (r, s), takes into account the relative position of nodes r and s in the hierarchy, *but only when going down* in our journey from r to s. When going up, no matter how far apart s is from r, conf is 0 "in the upwards part of the journey from r to s."

2.2 Confusion in using r instead of s, for a hierarchy with some ordered sets

In §2.1, the confusion between any two brother nodes is 1. For ordered sets, the confusion between any two brothers depends on how far they are in their ordering. If the ordered set has only one element e, then conf (e, e) = 0. If it has two elements, then conf (e1, e2) = 1. For ordered sets with more than two elements, n>2, the confusion between two contiguous elements is 1/(n-1). Figure 2 shows an example.

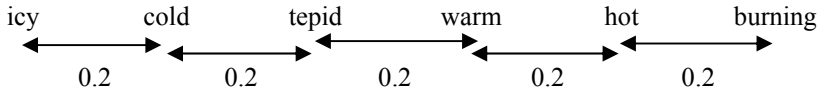

Figure 2. A set showing the confusion between its elements

Thus, conf (icy, cold) = conf (cold, icy) = 0.2; conf (cold, warm) = 0.4

For a hierarchy composed of sets some of which have an ordering relation (such as H_2), the confusion in using r instead of s, conf (r, s), is defined as follows:

- conf (r, r) = conf (r, s) = 0, when s is any ascendant of r.
- If r and s are distinct brothers,
  conf (r, s) = 1 if the father is not an ordered set; else,
  conf (r, s) = the relative distance from r to s = the number of steps needed to jump from r to s in the ordering, divided by the cardinality-1 of the father.
- conf (r, s) = 1 + conf (r, father_of(s)). ♦

This is like conf for hierarchies formed by (unordered) sets (§2.1; more at [6, 7]), except that there the error between two brothers is 1, and here it may be a number between 0 and 1. Example (for H_2): conf (short, measure) = 0; conf (short, length) = 0; conf (short, light) = 2; conf (short, medium-length) = 0.5; conf (short, long) = 1.

3. Queries and graduated errors

This section explains how to pose and answer queries where there is a permissible error due to confusion between values of hierarchical variables.

3.1 The set of values that are equal to another, up to a given confusion

A **value u is equal to value v, within a given confusion ε**, written $u =_\varepsilon v$, iff $conf(u, v) \leq \varepsilon$. ♦ It means that value u can be used instead of v, within error ε.
Example: If v = lemon (Figure 2), then

the set of values equal to v with confusion 0 is {lemon};

the set of values equal to v with confusion 1 is {citric lemon grapefruit};

the set of values equal to v with confusion 2 is {plant citric rose lemon grapefruit}.

Notice that $=_\varepsilon$ is neither symmetric nor transitive.

These values can be obtained from table 1 by watching column v ("lemon") and collecting as u's those rows that have conf ≤ ε.

That two values u and v have confusion 0 does not mean that they are identical (u = v). For example, the set of values equal to mammal with confusion 0 is {cow mammal cat}, and the set of values equal to live being (the root) with confusion 0 contains all nodes of H_3, since any node of H_3 is a live being.

3.2 Identical, very similar, somewhat similar objects

Objects are entities described by a set of (property, value) pairs, which in our notation we refer to as (variable, value) pairs. They are also called (relationship, attribute) pairs in databases. An object o with k (variable, value) pairs is written as $(o\ (v_1\ a_1)\ (v_2\ a_2) \ldots\ (v_k\ a_k))$.

We want to estimate the error in using object o' instead of object o. For an object o with k erhaps hierarchical) variables $v_1, v_2, .., v_k$ and values $a_1, a_2, ..., a_k$, we say about another object o' with same variables $v_1 \ldots v_k$ but with values a_1', $a_2', \ldots a_k'$, the following statements:

o' is **identical** to o if $a_i' = a_i$ for all $1 \leq i \leq k$. All corresponding values are identical. ♦ If all we know about o and o' are their values on variables $v_1, \ldots v_k$, and both objects have these values pairwise identical, then we can say that "for all we know," o and o' are the same.

o' is **a substitute** for o if $conf(a_i', a_i) = 0$ for all $1 \leq i \leq k$. ♦ All values of o' have confusion 0 with the corresponding value of o. There is no confusion between a value of an attribute of o' and the corresponding value for o.

o' is **very similar** to o if $\Sigma\ conf(a_i', a_i) = 1$. ♦ The sum of all confusions is 1.

o' is **similar** to o if $\Sigma\ conf(a_i', a_i) = 2$. ♦

o' is **somewhat similar** to o if $\Sigma\ conf(a_i', a_i) = 3$. ♦

In general, o' is **similar$_n$** to o if $\Sigma\ conf(a_i', a_i) = n$. ♦

These relations are not symmetric.

Example 1 (We use hierarchies H_1, H_2 and H_3). Consider the objects

(Ann	(*travels-by* land-vehicle)	(*owns* animal)	(*weighs* weight))
(Bob	(*travels-by* boat)	(*owns* bird)	(*weighs* heavy))
(Ed	(*travels-by* water-vehicle)	(*owns* plant)	(*weighs* medium-weight))
(John	(*travels-by* car)	(*owns* cow)	(*weighs* light)).

Then Ann is similar$_4$ to Bob; Bob is very similar to Ann; Ann is somewhat similar to Ed; Ed is similar$_{3.5}$ to Bob;[1] Bob is similar$_6$ to John, etc. See Table 2.

Table 2. Relations between objects of Example 1. This table gives the relation obtained when using object o' (running down the table) instead of object o' (running across the table)

	Ann	**Bob**	**Ed**	**John**
Ann	identical	similar$_4$	somewhat similar	similar$_5$
Bob	very similar	identical	very similar	similar$_6$
Ed	similar	similar$_{3.5}$	identical	similar$_6$
John	substitute	similar$_4$	similar$_{2.5}$	identical

Hierarchical variables allow us to define objects with different degrees of precision. This is useful in many cases; for instance, when information about a given suspect is gross, or when the measuring device lacks precision. *Queries* with "loose fit" permit handling or matching objects with controlled accuracy, as exposed below.

3.3 Queries with controlled confusion

A table of a data base stores objects like Ann, Bob... defined by (variable, value) pairs, one object per row of the table. We now extend the notion of queries to tables with hierarchical variables,[2] by defining the objects that have property P within a given confusion ε, where $\varepsilon \geq 0$.

P **holds for object o with confusion ε**, written P_ε *holds for o*, iff

- If P_ε is formed by non-hierarchical variables, iff P is true for o.
- For *pr* a hierarchical variable and P_ε of the form (*pr* c),[3] iff for value v of property *pr* in object o, $v =_\varepsilon c$. [if the value v can be used instead of c with confusion ε]
- If P_ε is of the form $P1 \vee P2$, iff $P1_\varepsilon$ holds for o or $P2_\varepsilon$ holds for o.
- If P_ε is of the form $P1 \wedge P2$, iff $P1_\varepsilon$ holds for o and $P2_\varepsilon$ holds for o.
- If P_ε is of the form $\neg P1$, iff $P1_\varepsilon$ does not hold for o. ♦

The definition of P_ε *holds for o* allows control of the "looseness" of P or of some parts of P; for instance, the predicate (*plays* guitar)$_0$ will match people who play guitar or any of the variations (sons) of guitar (refer to Figure 4); (*plays* guitar)$_1$ will match those people just mentioned as well as people who play violin and harp.

---

[1] conf (water-vehicle, boat) = 1; conf (plant, bird) = 2; conf (medium-weight, heavy) = 0.5; they add to 3.5.

[2] For non hierarchical variables, a match in value means conf = 0; a mismatch means conf = ∞

[3] (*pr* c) in our notation means: variable *pr* has the value c. Example: (*profession* Engineer). It is a predicate that, when applied to object o, returns T or F.

What do we mean by "P holds for o" when we do not specify the confusion of P? If P and o are *not* formed using hierarchical variables, the meaning is the usual meaning given in Logic. Nevertheless, if P or o use hierarchical variables, then by "P holds for o" we mean "P_0 holds for o". This agrees with our intuition: predicate (*owns* chord-instrument), given without explicitly telling us its allowed confusion, is interpreted as (*owns* chord-instrument)$_0$, which will also match with a person owning an electric-guitar, say.

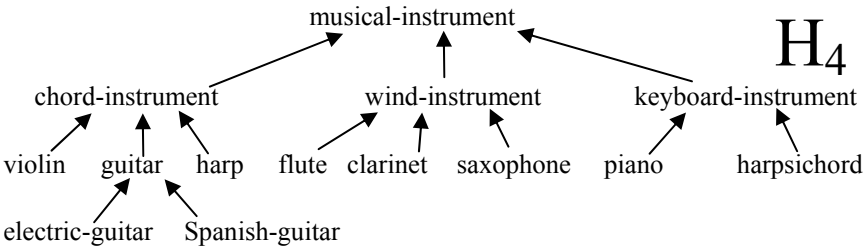

Figure 4. A hierarchy of musical instruments

Example 2 (refer to hierarchies and persons of Example 1). Let the predicates

P = (*travels-by* bike) $\vee$ (*owns* cow),
Q = (*travels-by* helicopter) $\wedge$ (*owns* cat),
R = $\neg$ (*travels-by* water-vehicle).

Then we have that P_0 holds for John; P_1 holds for John, P_2 holds for {Ann, Bob, John}, P_3 holds for {Ann, Bob, Ed, John}., as well as P_4, P_5,...

We also have that Q_0 holds for nobody; Q_1 holds for nobody; Q_2 holds for {Ann, Bob, John}; Q_3 holds for {Ann, Bob, Ed, John}, as well as Q_4, Q_5...

We also have that R_0 holds for {Ann, John}; R_1 holds for nobody, as well as R_2, R_3, R_4...

From the definition of *P_ε holds for o,* it is true that $(P \vee Q)_\varepsilon = (P_\varepsilon \vee Q_\varepsilon)$. This means that for $(P \vee Q)_a = (P_b \vee Q_c)$, a = min (b, c). Similarly, for $(P \wedge Q)_a = (P_b \wedge Q_c)$, we have a = max (b, c).

Accumulated confusion. For compound predicates, a tighter control of the error or confusion is possible if we require that the accumulated error does not exceed a threshold ε. This is accomplished by the following definition.

P **holds for object o with accumulated confusion ε**, written *P^ε holds for o*, iff

- If P^ε is formed by non-hierarchical variables, iff P is true for o.
- For *pr* a hierarchical variable and P^ε of the form (*pr* c), iff for value v of property *pr* in object o, $v =_\varepsilon c$. [if the value v can be used instead of c with confusion ε]
- If P^ε is of the form P1 $\vee$ P2, iff $P1^\varepsilon$ holds for o or $P2^\varepsilon$ holds for o.
- If P^ε is of the form P1 $\wedge$ P2, iff there exist confusions a and b such that a+b = ε and $P1^a$ holds for o and $P2^b$ holds for o.

- If P^{ε} is of the form $\neg P1$, iff $P1^{\varepsilon}$ does not hold for o. ♦

Example 3: For Q = (*travels-by* helicopter) ∧ (*owns* cat), we see that Q^0 holds for nobody; Q^1 holds for nobody; Q^2 holds for nobody; Q^3 holds for John; Q^4 holds for {Ann, Bob, John}; Q^5 holds for {Ann, Bob, Ed, John}, as well as Q^6, Q^7...

Closeness. An important number that measures how well object o fits predicate P_ε is the smallest ε for which P_ε(o) is true. This leads to the following definition.

The **closeness** of an object o to a predicate P_ε is the smallest ε which makes P_ε true. ♦ The smaller this ε is, the "tighter" P_ε holds.

Example: (refer to hierarchies, persons and predicates of Example 2) The closeness of P_ε to John is 0; its closeness to Ann is 2; to Bob is 2, and to Ed is 3. This means that John fits P_ε better than Ed. See Table 3.

Table 3. Closeness of an object to a predicate. Persons, hierarchies and predicates are those of example 2

	P_ε	Q_ε	R_ε
Ann	2	2	0
Bob	2	2	∞
Ed	3	2	∞
John	0	3	0

3.4 Conclusions

The paper shows a way to introduce ordered sets into hierarchies.
Hierarchies can be applied to a variety of jobs:

- To compare two values, such as Madrid and Mexico City, and to measure their *confusion* (§2), for instance in answering query "What is the capital of Spain?"
- To compare two objects for similarity, like Ann and Ed (§3.2), giving rise to the notions of **identical, very similar, similar…** objects (not *values*).
- To find out how closely an object o fits a predicate P_ε (definition of **closeness**, §3.3).
- To retrieve objects that fit imperfectly a given predicate to a given threshold, using P_ε *holds for* o (confusion, §3.3 and example 2), and P^ε *holds for* o (accumulated confusion, §3.3 and example 3).
- To handle partial knowledge. Even if we only know that Ed *travels-by* water-vehicle, we can productively use this value in controlled searches (Example 1 of §3.2).

Hierarchies make a good approximation to the manner in which people use gradation of qualitative values (ordered sets), to provide less than crisp, but useful, answers.

Ordered sets add a further refinement to the precision with which confusion can be measured and used.

Hierarchies can also be used as an alternative to fuzzy sets, defining a membership function for a set with the help of closeness.

They can also be employed as a supervised pattern classifier, by using definitions of §3.2 that measure how close two objects are, and by using definitions of P_ε and P^ε (§3.3).

In [7] we describe a mathematical apparatus and further properties of functions and relations for hierarchies. Instead, [4, 9] explain similar functions, relations and examples for *ontologies.*

Acknowledgments

Useful exchanges were held with CIC-IPN professors Jesus Olivares and Alexander Gelbukh, and with Dr. Victor Alexandrov, SPIIRAS-Russia. This work was partially supported by Project CGPI-IPN 18.07 (20010778) and NSF-CONACYT Grant 32973-A. The authors have a *National Scientist* Award from SNI-CONACYT.

References

1. Bhin, N. T., Tjoa, A. M, and Wagner, R. Conceptual Multidimensional data model based on meta-cube. In *Lecture Notes in Computer Science* **1909**, 24-31. Springer. (2000)
2. Lenat, D. B., and Guha, R. V. *Building large knowledge-based systems.* Addison Wesley. (1989)
3. Guzman, A. Finding the main themes in a Spanish document. *Journal Expert Systems with Applications*, Vol. **14**, No. 1/2, 139-148, Jan./Feb. (1998)
4. Guzman, A., and Olivares, J. Finding the most similar concepts in two different ontologies. Accepted in *MICAI 04.*
5. Guzman, A., and De Gyves, V. *BiblioDigital. A distributed digital library.* Work in progress. SoftwarePro International, Inc.
6. Levachkine, S., and Guzman, A. Confusion between hierarchies partitioned by a percentage rule. Submitted to *AWIC 04.*
7. Levachkine, S., and Guzman, A. Hierarchies as a new data type for qualittive variables. Submitted to *Data and Knowledge Engineering.* (2003)
8. Olivares, J. *An Interaction Model among Purposeful Agents, Mixed Ontologies and Unexpected Events*. Ph. D. Thesis, CIC-IPN. In Spanish. Available on line at http:// www.jesusolivares.com/interaction/publica (2002)
9. Olivares, J. and Guzman, A. *Measuring the comprehension or understanding between two agents*. CIC Report in preparation.