Hierarchy as a new data type for qualitative variables

Serguei Levachkine, Adolfo Guzman-Arenas[*]
Centro de Investigación en Computación, Instituto Politécnico Nacional
"Lopez Mateos" Campus, Mexico City, MEXICO
sergei@cic.ipn.mx, aguzman@ieee.org

SUMMARY. Qualitative variables take symbolic values such as *cat, orange, California, Africa.* Often these values can be arranged in levels of deeper detail. For example, the variable *place_of_birth* takes as level-1 values *Africa, Asia*... as level-2 values *Nigeria, Japan*... as level-3 values *California*, *Massachusetts*... These values are organized in a **hierarchy H**, a mathematical construct among these values. Over **H**, the following are defined: (1) the function *confusion* resulting when using a symbolic value instead of another; (2) the closeness to which object o **fulfills** predicate P; (3) a method which allows precision-controlled retrieval for relational databases whose objects have symbolic values.

INDEX TERMS: hierarchy, ontology, approximate queries, confusion, knowledge representation.

1. Introduction

What is the capital of Germany? *Berlin* is the correct answer; *Frankfurt* is a close miss, *Madrid* a fair error, and *sausage* a gross error. What is closer to a *cat*, a *dog* or an *orange*? Can we measure these errors and similarities? Can we retrieve objects in a data base that are close to a desired item? *Yes*, by arranging these symbolic (that is, non-numeric) values in a hierarchy. For the sake of completeness, four different definitions of hierarchy are

[*] Corresponding author.

given in §2. At least one of them is original. These definitions are important for understanding what a hierarchy is. However, the *confusion* does not depend on particular definition.

1. This arrangement allows the definition of *confusion* (§3.1) to measure the error when using one symbolic value associated with a node in a hierarchy in place of the (intended, correct) symbolic value associated with other node in the hierarchy. Variations of the definition are:
    a. When the values represent sets with some size, as population in {France, Italy, Spain, Sweden} we talk about *percentage hierarchies* (§3.1.1).
    b. When the values can be ordered, as temperature in {frigid, cold, warm, hot, burning} we talk about *ordered hierarchies* (§3.1.2).
2. *Confusion* is also defined (§3.3) for hierarchies whose nodes are associated with predicates (called variables in the paper) in addition to hierarchies whose nodes are associated with values of a variable (item 1).
3. *Confusion* among values (item 1) and among variables (item 2) enables measurement of how close a given object *o* fulfils a given predicate P (§3.2.2), and we write $P_\varepsilon(o)$ for this measure.

The main contributions are (1)-(3), pertaining to symbolic values. Of course, errors, distances and approximate answers are well understood and developed for *numerical* values.

The rest of the introduction discusses related work. Section 2 gives several definitions for hierarchies. Section 3.1 introduces *confusion*. Section 3.2 presents predicates on hierarchies. Discussion of overall paper's results is in Section 4 and conclusions form Section 5.

Related work. Artificial Intelligence, Natural Language and Knowledge Representation communities have been gauging the distance, proximity or "relatedness" between symbolic values. Relevant efforts:

A. **Hierarchies.** The concept of a (generalization) hierarchy is not new. Hierarchies are used in data warehousing and data mining; see, for instance, the H-sets of Bhin [2]. A practical use of hierarchies in symbolic processing is Clasitex [9], 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© [4], a recent development, uses a large taxonomy (although not a hierarchy) to classify text documents; a (distributed) crawler in it retrieves "external" documents residing elsewhere in the Web. If a document is about (Cf. Clasitex) war, Iraq and President Bush, its URL will be stored in these three nodes in the concept tree. Hierarchies are simpler than ontologies, albeit very useful [13, 21].

   The data modeling community, through the entity-relationship model, also organize items by their nature, properties and the relations among them.

B. **Natural Language.** Linguists (see, for instance, Proceedings of CICLING 04, LNCS **2945**, as referenced in [11]) have proposed many versions of semantic closeness, similarity, and other measures among words. Everett [6] identifies conceptually similar *documents* using a single ontology. Sidorov [8] does the same using a topic hierarchy: a kind of ontology. Montes y Gómez [19] builds trees of words, and by graph matching retrieves similar texts. Another common idea twisting around is to regard the representation space with a "universal" measure of proximity of space's elements and then an attempt to adapt it to different subject domains [16] [24]. Comments on this in §4.

WordNet [26] organizes information in logical groupings called synsets; each synset is a list of synonymous words or collocations (e.g., “fountain pen”, “take in”), and pointers that describe the relations between this synset and other synsets. A word or collocation may appear in more than one synset, and in more than one part of speech. The words in a synset are logically grouped such that they are interchangeable in some *context*. Nouns and verbs are organized into *hierarchies* based on the hypernymy/hyponymy relation between synsets. Two kinds of relations are represented by pointers: lexical and semantic. Lexical relations hold between word forms; semantic relations hold between word meanings. These relations include (but are not limited to), antonymy, entailment, and meronymy/holonymy. Additional pointers are used to indicate other relations.

Budanitsky [3] compares five measures of similarity or semantic distance in WordNet: Jiang and Conrath's measure (the best in the comparison: a spelling-corrector on real data); that of Hirst-St-Onge (seriously over-related), that of Resnik (seriously under-related [23]), and those of Lin [16] and of Leacock-Chodorow (in between). Note that all the measures except those of Hirst and St-Onge are *similarity* (not relatedness) measures considering only *the hyponymy hierarchy* of WordNet. The main problem (§4) with these approaches is that they use *distances,* thus obeying the symmetric property d(a,b) = d(b,a), while *conf* (§3.1) does not.

C. **Ontologies.** At least three approaches appear when measuring similarity or relatedness of concepts (nodes in the ontology):

1. *Syntactic approach.* Methods that take into account only the organization of the tree or data structure of the ontology; for instance [19], those based on XML, or the "ontology merging" of Protégè [20].
2. *Standard ontology.* Use of a common or agreed-upon ontology. Clearly, if different people (or agents) use the *same* ontology, similarities among concepts will be consistently measured across users. CYC [7] was an early attempt to build the concept tree for common concepts. A common ontology is predicted in [11]; conceptually similar *documents* are identified in [6] by using a single ontology. In contrast, point (3) following shows use of different ontologies.
3. *Measuring similarity across ontologies.* LIA, a language for agent interaction [10, 13, 21], has an ontology comparator COM, that maps a concept from one ontology into the closest corresponding concept in another ontology. COM is used in *sim* of §3.4. By repeated use of *sim*, the *degree of understanding* **du**(B, O_A) of agent B (with ontology O_B) about ontology O_A is found in [22].

Instead of using ontologies, this paper works on arbitrary *hierarchies* (§2). Why? Because the problem-oriented interaction can be easier to maintain if the hierarchical structure is not a priori rigid as in the case of common hierarchies or ontologies.

D. **Pattern Classifiers.** Our predicates with controlled precision or confusion (§3.2.1) are similar to Pattern Classifiers [18], but these classify *objects* according to the values of their properties, whereas hierarchies help to classify these *values*, when they are non-numeric.

E. **Distances and ultradistances.** Traditionally [1, 25], the representation space is regarded as a metric space with some "exotic" distance (e.g., ultrametric distance to measure the "distances" between members of a hierarchy). Thus, §2.4 develops ultrametric distances for hierarchies. However, often is not the case that such a distance meets the needs of the classification problem under consideration. Thus, we lean towards functions like *conf* (§3.1) that are not distances.

2. Theory

This section continues with the focus on distances of item (E) of §1: we show how to build an ultradistance from a hierarchy (§2.4.1), how to build a hierarchy from an ultradistance (§2.4.2), whereas in Section 3 we move to a new approach that does not use distances.

Element set E. A set whose elements are explicitly defined. ♦[1] *Example*: {red, blue, white, black, pale}.

Ordered set. An element set whose values are ordered by a < ("less than") relation. ♦ *Example*: {very_cold, cold, warm, hot, very_hot}.

Covering. K is a covering for set E if K is a set of subsets $s_i \subset E$, such that $\cup\ s_i = E$. ♦ Every element of E is in some subset $s_i \in K$. If K is not a covering of E, we can make it so by adding a new s_j to it, named "others", that contains all other elements of E that do not belong to any of the previous s_i.

Exclusive set. K is an exclusive set if $s_i \cap s_j = \varnothing$, for every $s_i, s_j \in K$. ♦ Its elements are mutually exclusive. If K is not an exclusive set, we can make it so by replacing every two overlapping $s_i, s_j \in K$ with three: $s_i - s_j$, $s_j - s_i$, and $s_i \cap s_j$.

Partition. K is a partition of set E if it is both a covering for E and an exclusive set. ♦

---

[1] This symbol means: end of definition.

Symbolic value. A value that is not numerical, vector or quantitative. ♦ Example: *red*.

Representation. A symbolic value v **represents** a set E, written v ∝ E, if v can be considered a name or a depiction of E. ♦ v is associated with E. Example: strings ∝ {violin, viola, cello, guitar}.

Qualitative variable. A single-valued variable that takes symbolic values. ♦ Its value cannot be a set,[2] although such value may *represent* a set.

father_of (v). In a tree, f is the father_of v if f is the node immediately following v in the path from v to the root. ♦ f is "the node from which v hangs." We say that v is a **son_of** (f). ♦ Similarly, **grand_father_of** (v), **brothers_of, aunt**, **ascendants**, **descendants**... are defined, when they exist. ♦ The **root** is the only node that has no father.

2.1 Hierarchy

Definition 1. A **hierarchy** H of an element set E is a tree whose root is E and if a node has sons then these form a *partition* of their father. ♦

Definition 2. For an element set E, a **hierarchy** H of E is a tree of nodes; each node *n* is either an element of E or a set of symbolic values v_i, for i=1,…n, where $v_i \propto E_i$, and {E_1, E_2,…, E_n} is a partition of E. ♦ Example: for E = {chair table bed shirt loafer moccasin hammer paintbrush broom saw}, a hierarchy is (figure 1) H_1 = {furniture ∝ {chair table bed} apparel ∝ {shirt shoe ∝{loafer moccasin}} tool ∝ {hammer brush ∝ {paintbrush broom} saw}}. A hierarchy groups E into smaller sets of alike symbolic values.

[2] Variable, attribute and property are used interchangeably. Some objects have an attribute (such as weight) while others do not: the weight of blue *does not make sense,* does not exist. A variable (*color*, *height)* describes an aspect of an object; its value (*blue*, *2 Kg*) is such description (symbolic value) or measurement (numeric value).

Definition 1 emphasizes that the nodes of H are sets, while for definition 2 the nodes are symbolic values (such as *furniture*). To reconcile, we use the former definition v ∝ E, where a symbolic value v represents a set E.

Some times, we add node "others" to certain level of a hierarchy, when we are not sure whether the nodes already present at that level will collectively exhaust their father. Thus, for instance, in Figure 1, we could add to the second level node "other_merchandise", if we are not sure that furniture, apparel and tool comprise all the merchandise we are interested.

A **hierarchical variable** is a single-valued qualitative variable whose values belong to a hierarchy. ♦ The data type of a hierarchical variable is hierarchy. Example: *trades_in* that takes values from H_1, as *trades_in = furniture, trades_in = broom.*

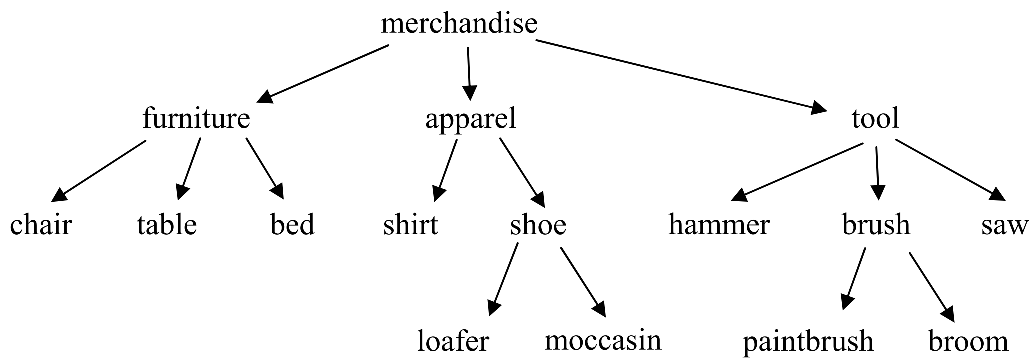

Fig. 1. Hierarchy H_1 of articles for sale.

2.2 Partitions of a finite set

Let E be a set of n elements. A **partition** P of E is a set of k subsets C_i of E such that (1) $C_i \cap C_j = \varnothing$; (2) $\cup_i C_i = E$. ♦

Two elements x and y of E are **equivalent** in a partition P if they belong to the same class C_i; this is denoted by xPy. ♦

Let **P**(E) be the set of all partitions of E; an **order relation** among the members of **P**(E), denoted by $<$, can be defined thus: for any two partitions P and P', P$<$P' iff $xPy \rightarrow xP'y$. Partition P is said to be **finer** than P'; it has more classes than P', i.e. k>k'. ♦ Example: let E={a,b,c,d,e,f}. Then E is less fine (i.e. coarser) than {{a},{b,c,d},{e,f}} which in turn is less fine than {{a},{b,c},{d},{e},{f}}.

A **lattice** structure for **P**(E) can be based on the order relation. For every pair of partitions P and P' there is a least upper bound (l.u.b.) $P \vee P'$, and a greatest lower bound (g.l.b.) $P \wedge P'$. ♦

Let us call P_k a partition of k classes where k is the level of P_k. A partition P' is said to **cover** a partition P if and only if P' results from combining *two* classes of P. ♦ Note that P'=bcd,a does not cover P=ab,c,d, because P' cannot be obtained from the *union* of two classes of P, which would in fact give P'_1=abc,d, P'_2=abd,c and P'_3=ab,cd, but *not* P.

A **chain** in the lattice is a sequence of partitions in order, e.g. $(P_1, P_2,...,P_j)$ where $P_1<P_2 <...< P_j$; the term is understood in the sense of an elementary chain in graph theory. ♦

If P_n is the finest partition, the **height** h(P) of P is the length of the chain joining P and P_n. ♦ If P_k is a partition of k classes, h(P)=n-k. It can be shown that all chains joining P and P' have the same number of elements, equal to the difference h(P) – h(P') between their heights, and (i) if P and Q each cover R, then $P \vee Q$ covers both P and Q; (ii) $h(P) + h(P') \geq h(P \vee P') + h(P \wedge P')$.

2.3 Hierarchy, equivalent definitions

Let E be a set of n elements, ℘(E) the set of all subsets of E and **P**(E) the **lattice** of the **partitions** defined by the **order relation** P < Q. Let CH be a complete **chain** in the lattice, i.e. a chain linking the **finest partition** P_n, of n elements, to the **coarsest partition** P_1=E.

Now we can give two additional definitions of a **hierarchy**.

Definition 3. A hierarchy is a set of partition classes constituting a complete chain, including in particular the set E itself and the n subsets formed by the elements of E. ♦

The passage from level k to level k-1 on CH corresponds to combining two classes. However, several levels can be passed over. Let P and Q be two non-consecutive partitions on CH, so that the classes of Q are either those of P or combinations of two or more classes of P. This leads to another direct definition.

Definition 4. A hierarchy is a subset H of ℘(E) such that (1) E∈H, (2) if x and y are elements of E, then {x},{y}∈H, (3) if h and h' are elements of H, then either h∩h'=∅ or h∩h'≠∅, in which case either h⊂h' or h'⊂h. ♦ Example*:* If E={a,b,c,d,e,f}, then figure 2 represents the hierarchy formed by the subsets {a},{b},{c},{d},{e},{f} with h_1=E, h_2={a,b,c,d}, h_3={e,f} and h_4={a,b,c}

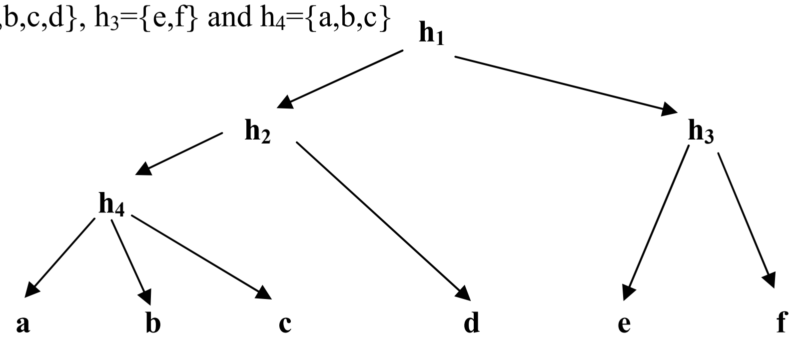

Fig. 2. Representation of a hierarchy.

2.4 Ultrametrics and clustering

A partial ordering of the elements of a hierarchy can be based on the inclusion relation and can be made a total ordering by the process of ascending a complete chain CH. In general, the same hierarchy can be defined by several different chains; thus in the example of figure 2 we can use three chains CH1, CH2 and CH3, with their nodes numbered 0,1,2,3,4 as follows:

CH1	*a,b,c,d,e,f*	*abc,d,e,f*	*abc,d,ef*	*abcd,ef*	*abcdef*
CH2	*a,b,c,d,e,f*	*abc,d,e,f*	*abcd,e,f*	*abcd,ef*	*abcdef*
CH3	*a,b,c,d,e,f*	*a,b,c,d,ef*	*abc,d,ef*	*abcd,ef*	*abcdef*
	0	1	2	3	4

Two elements of E occur in the same subset at a given node of CH, this being a partition of E. Given the chain, the node numbers characterize each pair of elements of E. We can now show how they can be used to define a special kind of distance.

2.4.1 Ultrametric distance from a hierarchy

If i, j and k are three elements of a set E, the **ultrametric distance** δ is defined as a function of E×E in R^+ as follows:

- $\delta(i,i)=0$,
- $\delta(i,j)=\delta(j,i)$,
- $\delta(i,j) \leq \max[\delta(i,k),\delta(j,k)]$ ♦ (1).

It is easy to prove the following *theorem*: every triangle is either isosceles or equilateral, with the base less than or equal to the equal sides.

So we might define a distance between elements of E by means of a chain of partitions, and it will be clear that this is an ultrametric distance in the sense just defined. It will also be clear that infinity of ultrametric distances can be defined so as to be consistent with the order imposed by the chain CH, and we must remember that the same hierarchy can be specified by several different such chains.

Conversely, an **indexed hierarchy** can be built (§2.4.2), given an ultrametric distance.

2.4.2 Constructing a hierarchy from an ultrametric distance

An ultrametric space is a couple consisting of a set E, finite or otherwise, and an ultrametric distance δ. In such a space, all triangles are isosceles or equilateral with the base less than or equal to the equal sides (see Theorem of §2.4.1). Conversely, if a generalized "distance" between any pair of elements of a set E is such that all triples give triangles having this property, this distance has the ultrametric property given by relation (1).

Let B be a sphere with center i and radius r, j an interior point and k a point on the surface. By hypothesis $\delta(i,j)\leq r$, $\delta(i,k)=r$. In the triangle (i,j,k) we must have $\delta(j,k)=r$. This means that any internal point j is equidistant from all points on the surface, or that *any internal point can be regarded as the center of the sphere*.

Similarly, it can be shown that if two spheres have a point in common then one must be included in the other; this follows from taking the common point as center for each sphere.

Let h denote the set of points of a sphere; all these points are equidistant from each other. If j is a point not belonging to h then, by $\delta(i,j)\leq\delta(j,k)$, it is equidistant from all points of h and this constant distance is greater than r. Thus h is an element of a **hierarchy** H deduced from the **ultrametric distance** δ. We can use this result to define an **algorithm for the construction of H from** δ:

Step 0. Set up the triangle table Δ of ultrametric distances between the elements of E.

Step 1. Find those elements $h_i,\ldots,h_k$ for which these distances are least (and equal, in the case of two elements only). Replace these in Δ by h_m. Recompute the distances between h_m and the other elements h_j:

$$\delta(h_m,h_m)=0, \qquad \delta(h_m,h_j)= \delta(h_i,h_j)$$

Step 2. If Δ has more than one column, repeat step 1, else end.

Example: Taking the example of figure 2, the successive tables (tables 1 to 5) are as follows.

Table 1	a	b	c	d	e	f
a	0	1	1	3	4	4
b		0	1	3	4	4
c			0	3	4	4
d				0	4	4
e					0	2
f						0

Table 2	h_4	d	e	f
h_4	0	3	4	4
d		0	4	4
e			0	2
f				0

Table 3	h_4	d	h_3
h_4	0	3	4
d		0	4
h_3			0

Table 4	h_4	h_3
h_2	0	4
h_3		0

Table 5	h_1
h_1	0

The hierarchy is the same as in Figure 2.

A knowledge of ultrametric distances enables us to construct a hierarchy and therefore to form a sequence of partitions of decreasing fineness along a chain CH of the partition lattice, i.e. to perform a classification of variable fineness. What is now required is to find that ultrametric distance which meets the needs of the **classification problem under con-**

sideration. Since, in general, the **data** of a problem consist of distances in the ordinary sense, the requirement is to obtain an ultrametric distance from ordinary distance.

2.4.3 Obtaining an ultrametric from a metric

Several algorithms that enable to us to derive an ultrametric which is close to the distance in terms of which the **problem data** have been proposed by Lance and Williams [14] and others [15].

2.4.4 Distances and linkage effect

In this section we define the **chain distance value** that can lead to the **linkage effect**.

Let E be a set of n elements and d(i,k) the distance between two elements. We define a path c_{jk} as a sequence of elements (j,…,q,…,k) and C_{jk} as the set of paths c_{jk}, and

$$\delta(c_{jk}) = \max_{q} d(q,q+1).$$

Thus δ is the length of the longest link in the path and we say that this defines the length of the path. Using the minimax criterion, in the set C_{jk} we find a path of shortest length, δ(j,k) say. This is an *ultrametric distance*, which we call the **chain distance value**. ♦

The use of a chain distance can lead to **linkage effect**, which appears in the nearest neighbor method of clustering (single linkage).

2.4.5 Clustering methods

The concepts of **hierarchy**, **ultrametrics** and **clustering** are closely linked.

It is natural to place different objects into the same cluster according to a criterion of *neighborhood*. As this proceeds, the partitioning becomes progressively less fine and new objects h of a hierarchy H, are created [11]. Use of the nearest neighbor or single linkage criterion can sometimes be dangerous because of the linkage effect, and for this reason we have to look for more satisfactory criteria which will take into account, among other things, the problem *context* under consideration. This is handled in next section.

2.5 Conclusions

Distances and ultradistances help us to establish a partial or total order among elements x,y,z,u ∈ E, this is written (x,y) ≤ (z,u) meaning that x resembles y more than z resembles u. If the number of elements of E is large, the establishment of this order may be difficult, so that a practical manner to order them (at least partially) *is to define a numerical function of similarity or dissimilarity (confusion)* that can be computed in terms of the attributes of every element of E: the **dissimilarity (confusion)** $\lambda(x,y)$ will be smaller the more closely x resembles y. Moreover, this function may not be a distance. That is developed in next Section.

3. Properties and functions on hierarchies

I ask for furniture, and they bring me a chair. Is there an error? Now, I ask for a saw, and a tool is brought. Can we measure this error? Can we classify or organize these values? Yes, by using hierarchies (figure 1). In this section, we measure the error (called *confusion*) when one symbolic value is used instead of another (the intended or correct value).

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

If r, s ∈ H, then [12] the **confusion** in using r instead of s, written conf(r, s), is:

- conf (r, r) = conf (r, any ascendant_of (r)) = 0.
- conf (r, s) = 1 + conf (r, father_of(s)). ♦[3]

[3] Without loss of generality we can define conf (r,s) = [1 + conf (r, father_of(s))] / height (H). In this case, 1 ≥ conf (r,s) ≥ 0

To measure conf, count the *descending* links from r (the replacing value) to s (the replaced or intended value). conf is not a distance, nor ultradistance. Example: conf(r, s) for the hierarchy H_2 of figure 3 is given in Table 6.

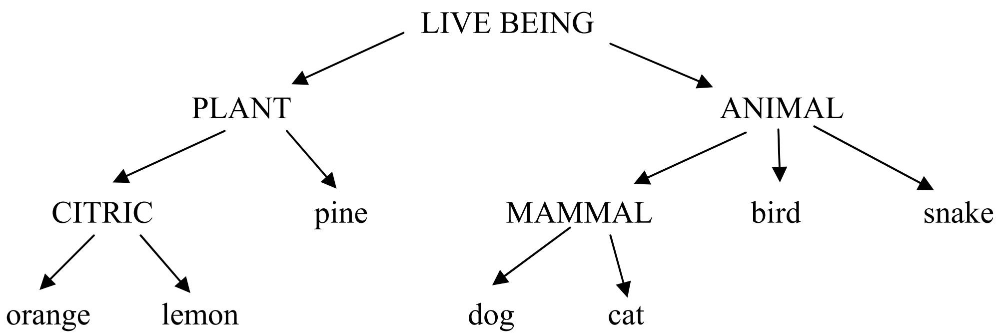

Fig. 3. A hierarchy H_2 of living creatures.

→ s

↓ r

conf	live being	ani-mal	plant	mam mal	bird	snake	citric	pine	cat	dog	lemon	orange
live being	0	1	1	2	2	2	2	2	3	3	3	3
ani-mal	0	0	1	1	1	1	2	2	2	2	3	3
plant	0	1	0	2	2	2	1	1	3	3	2	2
mam mal	0	0	1	0	1	1	2	2	1	1	3	3
bird	0	0	1	1	0	1	2	2	2	2	3	3
snake	0	0	1	1	1	0	2	2	2	2	3	3
citric	0	1	0	2	2	2	0	1	3	3	1	1
pine	0	1	0	2	2	2	1	0	3	3	2	2
cat	0	0	1	0	1	1	2	2	0	1	3	3
dog	0	0	1	0	1	1	2	2	1	0	3	3
lemon	0	1	0	2	2	2	0	1	3	3	0	1
orange	0	1	0	2	2	2	0	1	3	3	1	0

Table 6. Confusion in using row r instead of column s for the live beings of H_2.

The confusion thus introduced *catches the hierarchy semantics* and resembles reality. For example, the error when using *plant* instead of *live being* is 0, since all plants are live beings. conf(plant, live_being) = 0: there is no error if they ask you for a live being and you give them a plant. Giving a live being when asked for a plant has error 1; conf (live_being, plant) = 1. The confusion among two brothers, such as *orange* and *lemon*, is 1. The confusion when using the father instead of the son is 1. Using a son instead of the father gives 0. *conf* is not a symmetric function. Using specific things instead of general things produces low errors, see the column *animal*. Using general things instead of specific things produces high errors, see the row *live_being*. Using any descendant of a node *n* instead of *n* produces no error: observe the 0's in column *plant.* The lower triangular half has smaller errors than the upper triangular half of the table [4].

3.1.1 Confusion $conf^b$ for hierarchies that are formed by bags

Sometimes, the sizes of the sets that form the hierarchy matter. For example, for the hierarchy soldier ∝ {male_soldier, female_soldier}, our definition of *conf* would yield conf(soldier, male_soldier) = conf(soldier, female_soldier) = 1, when these numbers should be something like conf(soldier, male_soldier) = 0.02, conf(soldier, female_soldier) = 0.98, since approximately 98% of soldiers are male.

A **percentage hierarchy** H of E is a hierarchy in which the number of elements of E in each set of H is known. ♦ The nodes of H are not sets, but bags: unordered collection where repetitions are allowed.

[4] A loss of context appears if we use an ultrametric distance or any other symmetric function: these triangular parts would be equal.

For bags, the confusion in using r instead of s, $conf^b(r, s)$ should take into account the relative popularity of s in r, that is, $conf^b(r, s)$ = 1 – relative proportion of s in r. This is made precise in the following definition.

For percentage hierarchies, $conf^b(r, s) = 1 - (|E \cap r \cap s| / |E \cap r|)$. ♦ It is one minus the number of elements of E that are present in $r \cap s$, divided by the number of elements in E present in r. The intersection $\cap$ preserves repeated elements.

Example*:* For E = NorthAmerica(330M) ∝ {Canada(30M) ∝ {French_Canada(5M), English_Canada(25M)}, USA(200M), Mexico(100M)}, where the population in millions is indicated, we show in table 7 the confusion $conf^b$. For instance, $conf^b$(Canada, NorthAmerica) = 1 – (30M/30M) = 0; $conf^b$(NorthAmerica, Canada) = 1-(30M/330M)=1-.09 = 0.91. See table 7.

$conf^b$	**N**	**C**	**U**	**M**	**Fr_C**	**En_C**
N	0	0.91	0.39	0.7	0.99	0.93
C	0	0	1	1	0.63	0.17
U	0	1	0	1	1	1
M	0	1	1	0	1	1
Fr_C	0	0	1	1	0	1
En_C	0	0	1	1	1	0

Table 7. The confusion $conf^b(r, s)$ using row r instead of column s is shown for bags NorthAmerica, Canada, USA and Mexico, represented as N, C, U, and M.

3.1.2 Confusion $conf^L$ for hierarchies that contain lists

Sometimes, there is an order among the symbolic values that form a partition, as in H_3 ={microscopic, tiny, small, medium, large, gigantic}. In this case, some nodes in the hierarchy are lists (ordered sets), not just sets. For these, the confusion among two brothers should not be 1 (as the ordinary definition of *conf* will say), but a number between 0 and 1 related to the proximity of the two brothers. For a hierarchy composed of sets, some of which have an ordering relation, the confusion in using r instead of s, $conf^L(r, s)$, is defined as follows:

- $conf^L(r, r) = conf^L(r, s) = 0$, when s is any ascendant of r.
- If r and s are brothers,

  $conf^L(r, s) = 1$ if the father is not an ordered set; else,

  $conf^L(r, s)$ = 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.
- $conf^L(r, s) = 1 + conf^L(r, father\_of(s))$. ♦ Example*:* Refer to hierarchy H_3. $conf^L$ (microscopic, tiny)=1/5; $conf^L$(microscopic, small)=2/5; $conf^L$(microscopic, gigantic)=1.

  The rest of the paper will derive results for conf; those for $conf^b$ and $conf^L$ can be similarly derived.

3.2 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 $conf(u, v) \le \varepsilon$. ♦ v is the "correct" or intended value. It means that value u can be used instead of v, within error ε [12]. Example: Refer to figure 3. The set of values equal to CITRIC with confusion 0 is {CITRIC orange lemon}. The set of values equal to CITRIC with confusion 1 is {CITRIC orange lemon PLANT pine}. The set of values equal to *cat* with confusion 2 is {ANIMAL MAMMAL bird snake dog cat}. Notice that $=_\varepsilon$ is neither symmetric nor transitive.

3.2.1 Predicates on values of hierarchical variables

An object can be characterized by several (variable, value) pairs, and some of the variables may be hierarchical, such as (Sue (*lives_in* Canada) (*trades_in* furniture) (*size* small)). It is thus appropriate to define predicates that may hold for such objects. [5]

In this section we extend the notion of predicate to "predicate that holds within a confusion", by defining the set S of objects that satisfy predicate P within a given confusion ε.

P **holds for object o with confusion ε**, or P holds for o within ε, iff

- If P is formed by non-hierarchical variables, iff P is true for o.
- For pr a hierarchical variable and P 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).[6]
- If P is of the form P1 ∨ P2, iff P1 holds for o within ε or P2 holds for o within ε.
- If P is of the form P1 ∧ P2, iff P1 holds for o within ε and P2 holds for o within ε.
- If P is of the form ¬P1, iff P1 does not hold for o within ε. ♦

It is easy to see that if P1 holds for o within ε_1 and P2 holds for o within ε_2, then P1 ∨ P2 holds for o within $\min(\varepsilon_1, \varepsilon_2)$, whereas P1 ∧ P2 holds for o within $\max(\varepsilon_1, \varepsilon_2)$.

Example (refer to hierarchies H_1 and H_2 above):

Let the *predicates* be U = (*trades_in* furniture) ∨ (*owns* dog),
V = (*trades_in* furniture) ∧ (*owns* dog),
W = ¬ (*trades_in* apparel),
X = ¬ (*owns* bird)

[5] For variables that are not hierarchical, a match in value means conf = 0; a mismatch means conf = ∞.

and *objects* be (Carl (*trades_in* furniture) (*owns* snake)),
(Don (*trades_in* apparel) (*owns* citric)),
(Ed (*trades_in* shoe) (*owns* ANIMAL)),
(Fred (*trades_in* table) (*owns* orange)),
(Gal (*trades_in* tool) (*owns* PLANT)),
(Hal (*trades_in* hammer) (*owns* dog)).

Then we have the results of table 8.

	U holds within ε for:	**V holds within ε for:**	**W holds within ε for:**	**X holds within ε for:**
ε = 0	Carl, Fred, Hal	(nobody)	Carl, Fred, Gal, Hal	(all)
ε = 1	(all)	Hal	(nobody)	Don, Fred, Gal
ε = 2	(all)	Carl, Ed, Hal	(nobody)	(nobody)

Table 8. How the predicates U, V, W, X hold for several objects.

3.2.2 Fulfillment of a predicate by an object

In above example, how well an object such as Ed fulfils a predicate such as V? Looking at column V in table 8, V starts holding for o at ε=2. No smaller ε will do.

Object *o* **ε-fulfills** predicate P at threshold ε, if ε is the smallest number for which P holds for *o* within ε. ♦ It is a non-negative integer defined between an object and a predicate. The closer is ε to 0, the "tighter" P holds for *o*. Compare with the membership function for fuzzy sets.

3.2.3 Confusion between two objects

Similar to §3.1, for two objects (o (pr_1 v_1) (pr_2 v_2).. (pr_k v_k)) and (o' (pr_1 v_1') (pr_2 v_2')… (pr_l v_l')) with the same hierarchical variables, we can define the confusion when object *o* is

[6] c is the intended or correct value, and it is assumed to be a value (a constant) in the hierarchy of pr. If c is a constant that does not belong to the hierarchy of pr, then (pr c) is always false. If c were not a constant, but another variable, so that P is of the form (pr var), for instance (*size* *trades*_in), ["I want the objects where *size* and *trade_in* have the same value"] the value is false unless pr and var take values on the same hierarchy. In this case, the value of var is the intended value.

used instead of *o'* as the sum of the individual confusions conf(v_i, v_i') between their respective qualitative values. *o'* is the intended or correct object.

Confusion when object o is used instead of o'. CONF (o, o′) = Σ_i conf(v_i, v_i'). ♦ CONF (written in capital letters) is not symmetric, and is not a distance.

3.3 Confusion between variables (not values) that form a hierarchy

What could be the error in "Carl is a relative of Mary," if all we know is "Carl knows Mary"? And if what we know is "Carl is the grandfather of Mary"? That is, in addition to *qualitative values* forming a hierarchy, it is possible for the *variables* themselves to form a hierarchy. Example*:* hierarchy H_4.

```
handles ∝
          {trades_in ∝  {buys sells rents}
           carries   ∝  {transports_by_surface flies ships}
           maintains ∝  {oils fixes }
          }
```

When considering H_4 for (Carl (*trades_in* furniture) (*owns* snake)) of example § 3.2.1, he surely (*handles* furniture) with confusion 0, (*rents* furniture) with confusion 1, (*rents* merchandise) with confusion 1, and (*rents* table) with confusion 1+1=2. Thus, we can extend the definition of predicate with confusion to have variables that are members of a hierarchy, by adding another bullet to the definition of §3.2.1, thus:

- If P is of the form (var c), for var a variable member of a hierarchy, iff ∃ variable var_2 for which (var_2 c) holds for o within ε – conf (var_2, var), where var_2 also belongs to the hierarchy of var. ♦ The confusion of the variables adds to the confusion of the values.

Example: For (Don (*trades_in* apparel) (*owns* citric)) of E_2, predicate (*handles* apparel) holds with confusion 0; (*trades_in* apparel) holds with conf=0, (*buys* apparel) holds with conf=1, (*buys* shoe) holds with conf= 1+1=2, and (*buys* shoe) ∧ (*owns* lemon) with confusion = max(2, 1) = 2. Predicate (*transports* apparel) holds for Don with conf=1+1=2.

3.4 Confusion $conf^{do}$ for values in different ontologies

It is possible that values v_1 and v_2 belong to hierarchies coming out of *different* element sets E_1 and E_2 expressed as hierarchies H_1 and H_2. If these hierarchies can be converted into ontologies (to be called O_1 and O_2), and thus map values v_1 and v_2 (v_2 is the intended value) into objects (concepts) $c_1 \in O_1$ and $c_2 \in O_2$, we can use function *sim* of [13] to find $conf^{do}(v_1, v_2)$. Function sim(c_1, O_1, O_2) finds *sv*, a number between 0 and 1 expressing the similarity between a concept c_1 in O_1 and its most similar concept $c'_2 \in O_2$. It also finds such c'_2. Thus, to find $conf^{do}(v_1, v_2)$, execute these steps:

1. Find c'_2=sim(c_1, O_1, O_2), the concept c'_2 most similar to c_1, as well as *sv*.
2. Since c_2 and c'_2 belong to the same ontology O_2, find ℓ = length of the path going from c_2 to c'_2 in the O_B tree.
3. $conf^{do}(v_1, v_2) = (1+\ell)/sv$.♦ The steps are depicted in figure 4. See also [21].

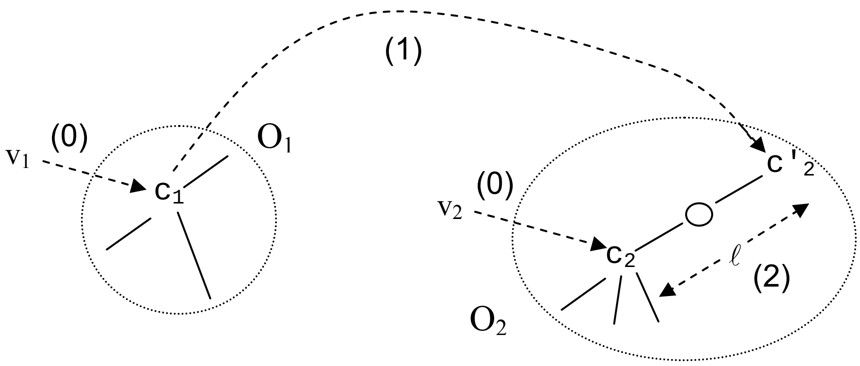

Figure 4. Steps to find $conf^{do}(v_1, v_2)$ for values in different ontologies: (0) map v_1, v_2 into c_1, c_2; (1) find c'_2 and *sv*; (2) find ℓ; (3) (not shown) compute $(1+\ell)/sv$.

4. Discussion

Starting the discussion, let us look again at the ordering and similarity of the elements of finite sets. A reflexive and transitive relation can be defined for the set F of all pairs of elements in E x E, which is called a **partial order**. With $x,y,z,u \in E$, this is written $(x,y) \leq (z,u)$ meaning that x resembles y more than z resembles u. Such a relation does not necessarily apply over the complete set F because they may be pairs of elements that are really not comparable. If it does apply over the complete set it becomes a **total order**.

It is difficult in practice to set up such a partial order if the number of elements of E is large, and if it is possible it is difficult to make this order without running the risk of generating contradictions. In fact, *the only practical way to establish a partial order is to define a numerical function of similarity or dissimilarity (confusion)* that can be computed in terms of the attributes of every element of E: the **dissimilarity (confusion)** $\lambda(x,y)$ will be smaller the more closely x resembles y.

The same partial order can be generated by any of an unlimited number such functions. Some dissimilarity functions, however, may not be distances (Cf. §3). Thus, by definition, a distance d(x,y) satisfies: (a) d(x,x) = 0; (b) d(x,y) = d(y,x); (c) d(x,y) ≤ d(y,z) + d(x,z). Relation (a) can be satisfied by making min $\lambda = 0$, which is simply a change of origin, and for (b) it is sometimes necessary to make λ symmetrical. Relation (c) may not hold, however, although it can be made to hold by adding a sufficiently large constant λ_0 to the values, whilst retaining $\lambda(x,x) = 0$. The corresponding partial ordering will not be changed. Thus the following general statement can be made: *it is always possible for a finite set E to make simple modifications that will transform a dissimilarity into a distance measure without affecting the partial order.*

In early approaches [25], the measures of similarity between variables have been considered using *Kendall, Hamming, Russell & Rao, Jaccard, Kuzlinsky, Yule,* and other distances, and a *contingency table*. These result, however, insufficient for variables that take symbolic values.

In context of the **identification problem**, distance measures for objects and concepts have been proposed too. The idea of classification carries with it in the implication that a **descriptor** or **symbolic description** can be defined for each class and will in some sense be representative of the class; possible descriptors are *a symbolic description of the class; a feature represented as a relational structure.* If X is the representation of an **object**, we can define a **concept** A_i as an entity which is such that an ordering of the couples (X,A_i) can be established for i = 1,...,k. A concept is not necessarily associated with one particular class; this may be so, e.g. in the case of descriptors, but in other cases the concepts may overlap, as for example for probabilities. The "distance" $D(X,A_i)$ between an object and a concept can be used effectively as a measure of similarity or dissimilarity. In other words, the "object-concept" distance D is used as a characteristic function. The following functions have been used: *probability, fuzzy assignment, inertia and potential, nearest neighbors (q-NN)*, etc. The main problem with these approaches is that they often (always) omit the context of a classification problem under consideration.

Let f(X;A) be a measure of similarity or dissimilarity between the representation of an object X and a concept A. Such a measure limited to a single A would be of little interest; if for example, A were the symbolic description of a class, there would clearly be at least two classes, A and ¬A ("not-A"). However, it can happen that the concepts A do not coincide with the classes. Let $\{A_i\}$ be the set of concepts under consideration: the assumption that these concepts can be grouped together into a set implies that the set operations of union ($\cup$) and intersection ($\cap$) can be applied. The A_i are not necessarily disjoint, i.e. several can apply simultaneously to a single object, although in the very special case that each identifies one class they are clearly disjoint. Let @ be the algebra generated by the A_i and operations: $\cap$, $\cup$, and ¬; the elements of @ form the **interpretation space**. If the concepts A_i are expressed by predicates, the operations are written as $\wedge$ and $\vee$ respectively and we have Stone's theorem: *all distributive logic systems are homomorphic to a distributive lattice of subsets of a set.* We may recall that if p,q and r are predicates then, $\wedge$ is distributive with respect to $\vee$.

Given f(X;A) and f(X;B), the *fundamental question* is: what are the values of $f(X;A\cap B)$ and $f(X;A\cup B)$? This can be answered in terms of two laws or operations: (i) an additive law $\oplus$, homomorphic with $\cup$, and (ii) a multiplicative law $\otimes$, homomorphic with $\cap$ and distributive over $\oplus$, and the *answer* is [15, 25] (Cf. §3.2.1)

$$f(X; A\cup B) = f(X;A) \oplus f(X;B),$$

$$f(X;A \cap B) = f(X;A) \otimes f(X;B) \qquad (2).$$

Union and intersection of ontologies and corresponding measures for them will be considered in a forthcoming work.

4.1 Summing up

We can emphasize in this summary the following key points:

- The *context* of a problem under consideration may be lost when attempting to define a distance on hierarchies of symbolic values (to measure closeness between hierarchical elements) that holds its partial (total) order.
- We show (§3.1) a way that takes into account the problem context, represents the intrinsic data semantics and expresses the similarity (dissimilarity) function in terms of the data attributes.
- Such approach permits to know the set of values that are equal to another up to a given confusion (§3.2), how close an object fulfils a predicate, and how close an object is to another (§3.2.3).
- Confusions and similarity functions for values in different ontologies can be defined as well (§3.4).

4.2 Applications

A) To queries, either to retrieve objects that hold for a predicate to a given threshold [5], or to measure the **closeness** of an object to a certain predicate (definition of §3.2.2).

B) To handle partial knowledge. Even if we only know that Carl trades in furniture, we can productively use this value in precise searches (example of §3.2.1).

C) As an approximation to the manner in which people use gradation of symbolic values (ordered sets), to provide less than crisp, but useful, answers.

D) As a measure of the **confusion** between two values (§3.1), such as in the answer "the capital of Germany is Frankfurt" (Cf. Introduction), versus "the capital of Germany is sausage."

E) To compare *attributes* (as opposed to values) that are similar, but not equal, such as my_neighbor and my_acquaintance (Cf. §3.3).

F) As an alternative to fuzzy sets, using $P_\varepsilon(o)$ (§3.2.2) as the membership function of a set.

G) As a supervised pattern classifier, by using CONF(o, o') of §3.2.3 between two objects. See also [18] for another approach to classifiers of qualitative data.

H) As an alternative to data mining, where approximate answers are usually useful.

I) As an accelerator of applications (G) and (H), if we store in cache memory the queries and results of previous tasks, and compare the (predicates of the) new queries to these cached data, before embarking in new searches [17].

5. Conclusions and suggestions for further work

5.1 Conclusions

The notions of hierarchy and hierarchical variable make possible to measure the *confusion* when a value is used instead of another. This creates a natural generalization for predicates and queries. The notions were introduced and developed for hierarchies formed by sets, but they can be extended to bags and lists, too.

The concepts and examples given here have practical applications, since they mimic the manner in which people process symbolic values.

In a subsequent paper we will describe a mathematical apparatus and further properties of functions defined in §3. See also [13, 22] for *ontologies* instead of hierarchies.

5.2 Suggestions for further work

1. Extend definition 3.2 (the sets of values equal to another value) to fuzzy sets.
2. Extend definition 3.3 (confusion between *variables,* not values) to fuzzy sets.
3. Extend queries of 3.2.2 to queries (search) in text, with the help of Clasitex [9].
4. Idem to queries in maps.
5. Idem to queries in semi-structured data.

Acknowledgments

The advice of the referees was very useful. Helpful discussions were held with Prof. Victor Alexandrov, SPIIRAS-Russia, Dr. Jesus Olivares and Prof. Gilberto Martinez, CIC-IPN. Work herein described was partially supported by NSF-CONACYT Grant 32973-A and Project CGPI-IPN 20010778. The authors have a SNI *National Scientist* Award.

References

1. V. Alexandrov. *Developed systems in science, technique, society and culture.* (Saint Petersburg State Technical University, Russia, 2000). In Russian.
2. N.T. Bhin, A.M. Tjoa, and R. Wagner. Conceptual Multidimensional data model based on meta-cube. *Lecture Notes in Computer Science* **1909** (Springer-Verlag, 2000) 24-31
3. A. Budanitsky and G. Hirst. Semantic Distance in WordNet: An Experimental, Application-oriented Evaluation of Five Measures. In: *Proc. North American Chapter of the Association for Computational Linguistics* (NAACL-2000), Pittsburgh, PA.
4. V.P, de Gyves and A. Guzman-Arenas. A distributed digital text accessing and acquisition system. *BiblioDigital*. ©SoftwarePro International. In: *Lecture Notes in Computer Science* **3061** (Springer-Verlag 2004) 274-283.

5. V.P. de Gyves. *Precision-controlled retrieval of objects with symbolic values, in a relational database.* B. Sc. Thesis in preparation, UPIICSA-Instituto Politecnico Nacional. In Spanish.
6. J. Everett, D Bobrow, *et al.* Making ontologies work for resolving redundancies across documents. *Comm. ACM* **45** (2) (2002) 55-60.
7. D. B. Lenat and R. V. Guha. *Building Large Knowledge-Based Systems*. (Addison-Wesley 1989).
8. A. Gelbukh, G. Sidorov, and A. Guzman-Arenas. Document comparison with a weighted topic hierarchy, In: *Proc. DEXA-99, 10th International Conference on Database and Expert System applications,* Workshop on Document Analysis and Understanding for Document Databases (Florence, 1999) 566-570.
9. A. Guzman. (1998) Finding the main themes in a Spanish document. *Journal Expert Systems with Applications* **14** (1/2) (1998) 139-148.
10. A. Guzman, J. Olivares, A. Demetrio and C. Dominguez. Interaction of Purposeful Agents that use Different Ontologies. In: *Lecture Notes in Artificial Intelligence* **1793** (Springer-Verlag 2000) 557-573.
11. A. Guzman and S. Levachkine. Graduated errors in approximate queries using hierarchies and ordered sets. In: *Lecture Notes in Artificial Intelligence* **2972** (Springer-Verlag 2004) 139-148.
12. S. Levachkine and A. Guzman-Arenas. Hierarchies Measuring Qualitative Variables. In: *Lecture Notes in Computer Science* **2945** (Springer-Verlag 2004) 262-274.
13. A. Guzman and J. Olivares. Finding the Most Similar Concepts in two Different Ontologies. *Lecture Notes in Artificial Intelligence* **2972** (Springer-Verlag 2004) 129-138.
14. G.N. Lance and W.T. Williams. Mixed-Data Classificatory Programs I – Agglomerative Systems. *Australian Computer Journal* **1** (1) (1967) 15-20.
15. S. Levachkine. *Qualitative data conversion and representation*. MIEM (Moscow Institute of Electronics and Mathematics, 1994) Report (in Russian).
16. D. Lin. An Information-Theoretic Definition of Similarity. *Proc. 15th Int. Conf. on Machine Learning* (ICML 1998) Madison, Wisconsin.
17. G. Martinez-Luna. *Incremental data mining.* Ph. D. Thesis in preparation. Centro de Investigacion en Computacion, Instituto Politecnico Nacional. In Spanish.
18. F. Martinez-T and A. Guzman. The logical combinatorial approach to Pattern Recognition, an overview through selected works. *Pattern Recognition* **34** (2001) 741-751.
19. M. Montes-y-Gómez, A. Lopez-Lopez, and A. Gelbukh. Information Retrieval with Conceptual Graph Matching. In: *Lecture Notes in Computer Science* **1873** (Springer-Verlag 2000) 312-321.
20. N. Noy, R.W. Fergerson and M.A. Musen. *The knowledge model of Protégè-2000: combining interoperability and flexibility.* Stanford Medical Informatics Technical Report, Stanford University, 2000
21. J. Olivares. *A model of interaction among purposeful agents with mixed ontologies and unexpected events*. Ph. D. Thesis. Centro de Investigacion en Computacion, Instituto Politecnico Nacional, 2002. In Spanish. Available on line at http://www.jesusolivares.com/interaction/publica

22. J. Olivares and A. Guzman-Arenas. Concept similarity measures the understanding between two agents. In *Proceedings of NLDB 04*. (2004)
23. P. Resnik. Disambiguating Noun Groupings with respect to WordNet Senses. In: Armstrong, S. et al. (eds.), *Natural Language Processing Using Very Large Corpora*. (Kluwer Academic Publishing, Dordrecht, 1995) 77-98.
24. P. Resnik. (1999) Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research* **11** (1999) 95-130.
25. J-C Simon. *Patterns and operators. The foundations of data representation.* (McGraw-Hill, 1984
26. WordNet: A Lexical Database for the English Language http://www.cogsci.princeton.edu/~wn/