Development of an Electronic Dictionary based on Ontology

F. Sáenz and A. Vaquero

Departamento de Sistemas Informáticos y Programación, Universidad Complutense de Madrid, E-28040 Madrid, España

fernan@eucmax.sim.ucm.es, avaquero@sip.ucm.es

Abstract

In this paper, we present the methodology followed to develop tools for authoring and using a bilingual electronic dictionary. The general goal of our research line is to reach methodologies for the development of tools for creating Information Systems based on ontologies. Starting on this way, a conceptual model is defined, relying on the entity-relationship diagram. The model distinguishes between ontology and lexicons, supporting taxonomy links. The general model supports multiple lexicons in order to implement multilingual dictionaries. A particular model has been taken for implementing an English-Spanish bilingual dictionary. From this particular model, we have developed the electronic dictionary following a standard methodology for developing relational databases, and UML for formally describing the developing of the tools accessing the database. The tools are in the test stage following some refinements of the initial design.

1.Introduction

Lack of standardization is broadly felt as a very undesirable state into the community around ontologies, lexicons, and so on. For instance, standard terminology for a common reference ontology is a goal to be reached. But attention has not yet been paid on subjects about development methodologies for building the software tools supporting and handling those types of knowledge bases. We claim for this aspect of standardization as necessary in order to integrate the diverse available information systems of this kind now and in the future. An more or less automatized incorporation of lexical and ontological databases into a common information system requires compatible software architectures and uniform data management from the different databases to be integrated. With this vision in mind, paying attention to the software engineering aspects along the development of these kinds of systems from the beginning is mandatory.

Electronic dictionaries subsume the conventional ones (paper dictionaries and the like). Lack of the kind of dictionaries we propose has been felt, as [Fillmore,1992] states:

"... we imagine, for some distant future, an online lexical resource, which we can refer to as a 'frame-based' dictionary, which will be adequate to our aims. In such a dictionary (housed on a workstation with multiple windowing capabilities), individual word senses, relationships among the senses of the polysemous words, and relationships between (senses of) semantically related words will be linked with the cognitive structures (or 'frames'), knowledge of which is presupposed by the concepts encoded by the words."

Subjects about electronic dictionaries for diverse NLP applications have been extensively studied (See [Wilks,1995]), as well as Lexical Databases [Miller,1995], World Knowledge Bases [Lenat,1990], ontologies [MikroKosmos], and the like. But there are no references on how these information systems have been built, and generally, there ais no registered information about how they have been growing along their life.

2.Order, Classification, and Ontology

%Order -> classification -> taxonomy

Typically, monolingual dictionaries show an alphabetical order that can be seen as a simple term classification: terms are classified in singletons by its lexicographic form. Other possible less naïve classifications are derivative (root-shape), grammatical, and semantic. Derivative classifications [MaríaMoliner] are not common, and grammatical classifications are not intended for dictionaries. Finally, semantic classification groups terms by semantic categories (for instance, synonym and antonym dictionaries, or ideological dictionaries [Casares].) Semantic categories not also allow meaning[1] classification, but the more meaningful taxonomy of meanings. Conventional lexical databases, such as WordNet [Miller,1995], have term classification such as synonymy (grouped in the so called synsets.) Ontologies go beyond by playing the role of meaning classification [Nirenburg,1995]. Our proposal support a form of this important concept as will be explained along the paper.

% Order in taxonomy

Semantic categories are useless for term lookups since a meaning will correspond, in general, to a set of (synonym) terms[2]. However, it has an important role on both using and authoring dictionaries because each meaning of a given term (polysemy and/or homonymy) is precisely identified by its semantic category (categories from now on, for the sake of brevity), instead of the usual nonsense sequential number[3]. Therefore, semantic categories provide classification for meanings, and such classification can be arranged in a taxonomy. But this not straightforwardly implies a term order since meanings are abstract ideas that cannot be expressed in general by one distinctive word[4]. It is commonly acknowledged that the best order for lookups is lexicographic (a derivative classification is a counterexample for this, but it still keeps a lexicographical order by repeating entries and adding links.) Figure 1 resumes the order for taxonomies in a hierarchy; it shows a taxonomy of categories along with the set of terms belonging to each category. From this point of view, there is a complete lexicographic order (provided categories are identified with terms or phrases.) A hierarchy is a natural structure for meaning classification. Each node in the hierarchy corresponds to a category. In principle, every category in the hierarchy can be assigned to a term, no matter its hierarchy level. It must be noted that every category in the hierarchy contains at least the term that names the category, so that all categories are non-empty. On the other hand, the creation of new categories as intersection of several predefined ones should be avoided in a first instance, in order to reach compactness.

```
Category 1 → {Terms of Category 1}
  ├──Category 1.1→ {Terms of Category 1.1}
  │     └──Category 1.1.1→ {Terms of Category 1.1.1}
  │           ...
  ├──Category 1.2→ {Terms of Category 1.2}
  │  ...
  └──Category 1.n→ {Terms of Category 1.n}
...
```

Figure 1. A Taxonomy

In systems based on ontology, the concepts are generally represented with a taxonomy as an acyclic directed graph structure, which allows several parents for a child as well as a parent for several children, and therefore is richer than our representation. We restrict classification from graphs to trees in our semantic domain in order to reach compactness, although we may easily support it as it will be shown later.

3.Practical Goals

One of the motivations behind building and using such an electronic dictionary is the possibility to develop specialised and limited dictionaries for the user's domain, that restrict the linguistic domain and make the categorisation of meanings and the definition of the taxonomy easier if compared with a general dictionary. There are a number of advantages in classifying meanings as a taxonomy. First, meaning taxonomy is a useful facility for an electronic dictionary because meanings embody additional semantics which provides more information to the reader (more than that of sequential numbers noted above.) Second, the system may also gain a new dimension

[1] Hereafter, we use the word "meaning" for referencing both meanings, concepts and senses.
[2] Nevertheless, there are other kinds of term lookups as ideological dictionaries show.
[3] However, meaning identifications by numbers also show a coarse classification; e.g. Tech. for Technical.
[4] The question is: Which is the best word to represent a meaning? In general, there are several (synonym) words representing the same meaning.

because it is possible to automatically generate specialised dictionaries under different categories (a sports dictionary may deal with soccer, tennis, or baseball dictionaries.) Third, it helps to develop a balanced dictionary by adding enough terms from different categories. Having the terms classified, it is easy to check out how many terms are under a given category. Fourth, it also helps to distribute the work between several authors by assigning categories to authors. A team of authors may develop a complete specialised dictionary by dividing the work by categories so that collaborative work is promoted. From an educational point of view, meaning classification can be done under a grammatical criteria, that is, categories can refer to grammatical properties of words (nouns, verbs, …), so that students can also learn grammatical aspects in this way.

There are different terminological databases (TDBs) built for different purposes. Some of them have incorporated the ontology structure and these very large databases are yet complete or almost complete. We are concerned in building databases for the restricted domain of multilingual dictionaries. Furthermore, we provide authoring and querying tools for both building and consulting the instanced database.

4.Terms, Meanings, and Categories

Our model obviously support the well-known naming problem [Katzenberg,1993], which consists of two elements: polysemy (under the synchronic point of view, that is, embodying polysemy itself and homonymy); and synonymy. But there are some considerations which have to be made in order to organize the elements of the electronic dictionary at the conceptual level. On the one hand, a given term can belong to several categories under different meanings. On the other hand, a given term can belong to several categories under the same meaning. Figure 2 shows two categories (C1 and C2) which respectively contain the meanings {M11, M12, M} and {M, M21, M22}. Each meaning has one or more terms associated. The term T2 is associated to meanings M12 and M21, which respectively belong to categories C1 and C2. We also show the term T that is assigned to meaning M, which belongs to both categories C1 and C2. Polysemy is present in T2, and synonymy is also present in T3, and T4, as it can be seen. T1 is neither polysemic nor synonym. TC1 and TC2 are the terms used to denote categories C1 and C2, respectively.

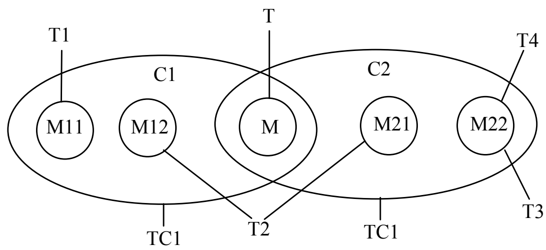

Figure 2. Relationships among categories, meanings and terms. Extensional definition

In this figure, the set of meanings {M11, M12, M} in C1 is the extensional definition of category C1. We must also note that a category has a meaning described by a definition. This figure does not embody this fact. In order to embody the meanings related to categories, we transform the scheme of figure 2 to the one depicted in figure 3. Now, C1 is the meaning of the category C1, and TC1 is the term assigned to such meaning, and the same applies to C2 and TC2. Then, we have one more meaning in each category. This meaning is the intensional definition of the category.

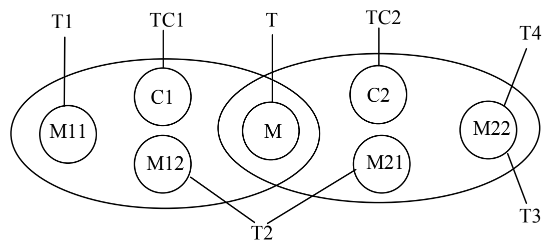

Figure 3. Relationships among categories, meanings and terms. Intensional definition

For a given language, we have a set of terms that holds the relationships with categories and meanings shown in figure 3. If we now think of several languages, the same applies for each one. Then, relationships between terms from different languages come from considering jointly the involved schemes .

5.Development of a bilingual dictionary

For building the bilingual English-Spanish dictionary we have followed a standard development methodology for, first, the design of the database supporting the introduced ideas, which will be detailed in forthcoming subsections, and, second, the design of the tools for accessing the database, which will be presented in the next section. In this paper, we focus on the database development, and, from the conceptual, logical, and physical designs, we show the first two for a relational database.

5.1 Conceptual Design

The objective of this stage is to design a conceptual schema for the database with a model for representing semantic and lexical multilingual information. This is a sound conceptual model for the terminological database (TDB) which shall eventually hold the terms, definitions, meanings, and semantic categories. Since it is intended to deal with two or more languages (bilingual or multilingual dictionaries), we need to represent instances of terms, textual definitions, and textual semantic categories for each language, but, as meanings are not language dependent, we shall use unique representations for them.

The entity-relationship model is used to describe the conceptual model we propose, shown in figure 4. In this figure (following some recommendations in [Pressman,1997] [Silberschatz,1996]), entity sets are represented with rectangles, attributes with ellipses, and relationship sets with directed and undirected lines. If B has an incoming line from A, this denotes a one (A) to many (B) mapping cardinality. Double arrows denote many to many mapping cardinalities. Undirected lines denote both one to one mapping cardinalities, when connecting entity sets, and attributes of entity sets, when connecting attributes to entity sets. Relationship set names (not shown in this figure) label each line.

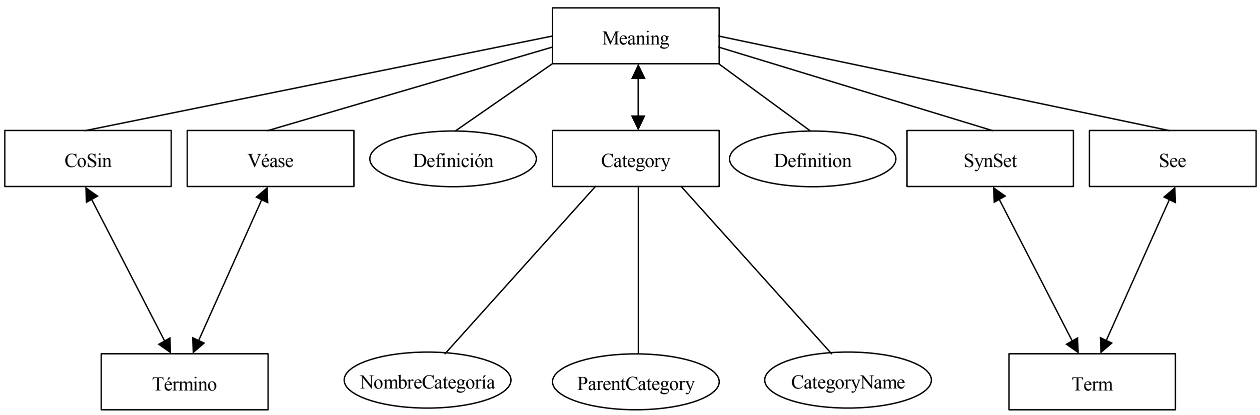

Figure 4. Entity-Relationship Model for an English-Spanish TDB

For the sake of clarity and conciseness, in this figure we show an instance of a multilingual terminological database for only Spanish and English languages, but it naturally derives from the general model depicted in figure 6 (where Li denotes the i-th language, $i \in \{1,..,N\}$.) In figure 5 we depict the entity set Meaning, the central entity set other entity sets rest on. In fact, this is the entity set which is language independent. The entity set SynSet denotes the English synonym set (SynSet - Synonym Set.) The relationship set between both entity sets is one to one. The entity set Term represents all the English terms that compose the terminological database. The relationship set between SynSet and Term is many to many since a synonym set contains several terms, and a term may be contained in several synonym sets (obviously, with different meanings.)

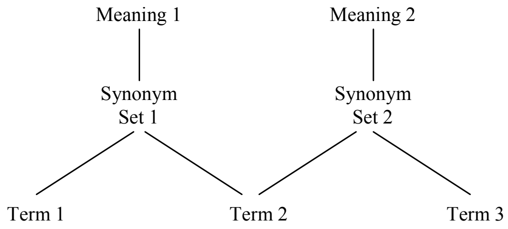

Figure 5. Polysemy and Synonymy related with the synonym sets

The entity set See denotes the set of English terms related under a given meaning. The relationship set between Meaning and See is one to one. The relationship set between See and Term is many to many, because a meaning may refer to several English terms, and one term may be polysemic. The entity set Category denotes the category each meaning belongs to. The relationship set between Category and Meaning is many to many since many meanings are in a category, and a meaning could be in several categories (this situation is expected to be reduced to the minimum since the goal is to keep the classification as disjoint as possible). This relationship set embodies the fact that our classification is not lexical (there is not a direct relationship between Category and Term) but semantic (we relate meanings to categories, i.e., we categorise meanings.) The entity set Category has three attributes: CategoryName, NombreCategoría, and ParentCategory. The first two correspond to the textual name of the category in each considered language, English and Spanish, respectively. The last attribute, ParentCategory, represents the links in the taxonomy by relating a category with its parent. Since each entity Category has a monovalued attribute for parent, this means that we restrict taxonomies to trees. If we change this attribute by a multivalued attribute, we allow a taxonomy graph instead of a tree. Meaning has two attributes: Definition and Definición, which correspond to the textual definition in the same considered languages. The remaining entity sets (CoSin, Véase, Término) are homologous to the respective entity sets (SynSet, See, Term.)

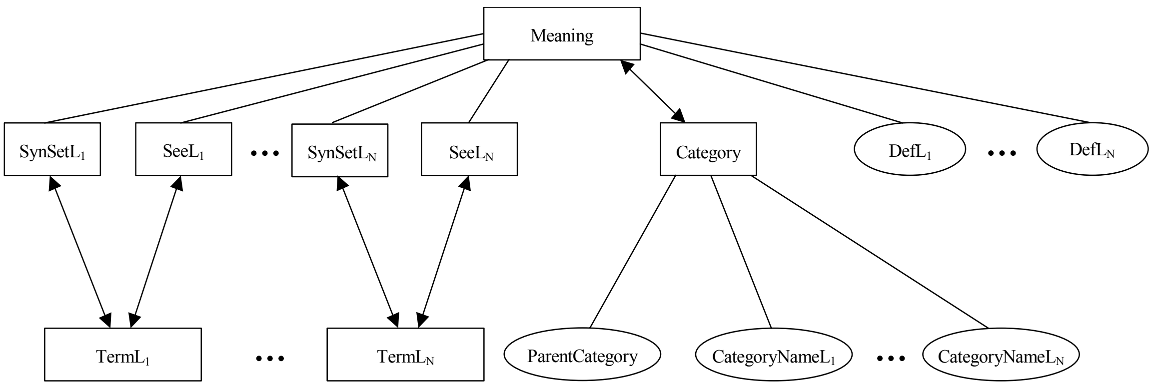

Figure 6. Entity-Relationship Model for a Multilingual TDB

An alternative model would consist of relationship sets for SynSet and See instead of entity sets, but we have selected entity sets in order to both highlight in a concise way the involved concepts, and simplify its presentation. Nevertheless, forthcoming stages of the development cycle show that the relations that SynSet and See represent are not implemented in the database schema as isolated tables. For the sake of clarity, the above diagrams do not show some needed attributes as the name of terms and the identifications of Meaning, SynSet, and See. Hidden attributes for entity sets conform the keys which univoquelly identify entities. Meaning and Category have several candidate keys, DefL*i*, and CategoryNameL*i*, respectively.

In addition to the entity-relationship diagram, a result of this stage is the definition of some constraints. Mapping cardinalities are partially showed in the diagram and we can further detail the participation of entities in relationships with elaborated participation cardinality. For instance, a Definition and a SynSet must be defined for each Meaning and also a Term have to be related because when one adds a new meaning he thinks in both a term and a textual definition which may define it. Note that this cannot be represented with the elements in the diagram since there are several languages (it could be when considering a monolingual dictionary). This participation constraint is not the same for See since, given a meaning, the definition of related terms (if any) can be delayed.

5.2 Logical Design

The objective of this stage is to design a logical schema for the database following the conceptual schema obtained in the former stage by using the relational model. The first step is to get a table from each entity set and from each relationship set, with indication of the primary keys by underlining attributes. For the bilingual dictionary we get, for instance:

Entity sets:	Meaning(MeaningID, Definition, Definición)	Meanings
	SynSet(SynSetID)	Synonym sets
	See(SeeID)	See sets
Relationship sets:	MeanSee(MeaningID, SeeID)	One to one relationship between Meaning and See
	In(MeaningID, CategoryID)	Many to many relationship between Meaning and Category
	MeanSyn(MeaningID, SynSetID)	One to one relationship between Meaning and SynSet

We see that there is redundant information (as usual) in entity sets related by a relationship set with one to one mapping cardinality so that they can be merged. In fact, we may arrange the entity-relationship model by substituting entity sets by relationship sets as figure 7 shows, in which diamonds enclose relationship names. After this rearrangement, which yields to entity set tables embodying some relationship sets, entity sets and relationship sets are:

Entity sets:	Meaning(MeaningID, Definition, Definición)	Meanings
	Term(TermID, TermName)	English terms
	Término(TerminoID, NombreTérmino)	Spanish terms
	Category(CategoryID, CategoryName, NombreCategoría)	Categories
Relationship sets:	In(MeaningID, CategoryID)	Many to many relationship between Meaning and Category. (Taxonomy definition)
	SynSet(MeaningID, TermID)	Many to many relationship between Meaning and Term. (A meaning related with its synonym set)
	See(MeaningID, TermID)	Many to many relationship between Meaning and Term. (A meaning related with its see set)
	And other two tables for the corresponding relationship sets for the Spanish side of the entity-relationship diagram.	

We have maintained in the logical design the attributes (fields) for identifying each entity because it allows both a more compact storage for relationships, and more efficient indexes (although physical design is not presented in this paper, we note here that indexes are created in all the key fields).

In order to prove that the schemas above are normalized according to the Boyce-Codd normal form, we note the functional dependencies for each entity or relationship (parentheses enclose the entity set or relationship set they correspond to):

TermID → TermName (Term)
MeaningID → Definition, Definición (Meaning)
Definition → MeaningID, Definición (Meaning)
CategoryID → CategoryName, NombreCategoría, ParentCategory (Category)
CategoryName → CategoryID, NombreCategoría, ParentCategory (Category)
(And similar others).

We see that left sides of functional dependencies are always superkeys, so that the database schema agrees the Boyce-Codd normal form.

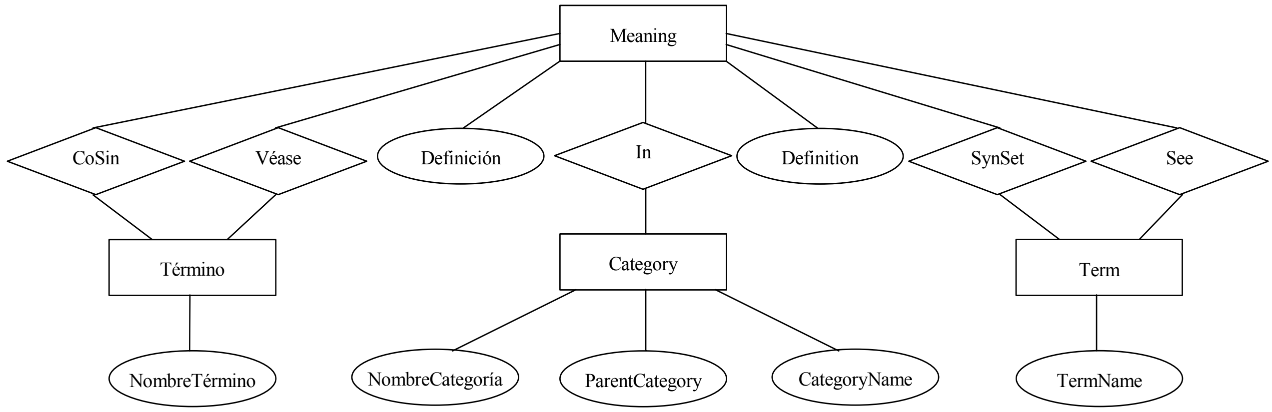

Figure 7. Modified Entity-Relationship Model

5.2.1 Integrity Constraints

We present here some integrity constraints which must be imposed to the relational database in order to achieve consistency.

Entity sets:

- Term(<u>TermID</u>, TermName).
  - Primary key constraint: TermID.
  - Domain constraint: TermName does not allow nulls.
  - Referential integrity constraint: Cascade deletions are not allowed in TermID for SynSet and See tables.
- Category(<u>CategoryID</u>, CategoryName, NombreCategoría, ParentCategory).
  - Primary key constraint: CategoryID.
  - Domain constraints: CategoryName, NombreCategoría, and ParentCategory do not allow nulls.
  - Referential integrity constraints: Category.ParentCategory → Category.CategoryID. A parent category have to be defined before its use (the root node points to itself). Cascade deletions are not allowed in CategoryID for In table (i.e., categories are not allowed to be deleted if they have related meanings).
  - Integrity constraints: The taxonomy has only one connected component (i.e., it has to be a tree, not a forest). In order to implement the intensional definition of the category, a meaning and a term have to be defined in Meaning and Term tables.
- Meaning(<u>MeaningID</u>, Definition, Definición).
  - Primary key constraint: MeaningID.
  - Domain constraints: Definition or Definición can be null, but not at the same time.
  - Referential integrity constraint: Cascade deletions are not allowed in MeaningID for In and See tables.

Relationship sets:

- In(<u>MeaningID</u>, <u>CategoryID</u>).
  - Primary key constraint: MeaningID, CategoryID
  - Referential integrity constraints: In.MeaningID → Meaning.MeaningID, In.CategoryID → Category.CategoryID
- SynSet(<u>MeaningID</u>, <u>TermID</u>).
  - Primary key constraint: MeaningID, TermID
  - Referential integrity constraints: SynSet.MeaningID → Meaning.MeaningID, SynSet.TermID → Category.CategoryID
- See(<u>MeaningID</u>, <u>TermID</u>).
  - Primary key constraint: MeaningID, TermID
  - Referential integrity constraints: See.MeaningID → Meaning.MeaningID, See.TermID → Term.TermID

We can summarize these constraints as follows: each time a new category is defined, the links between the category have to represent a tree structure. This is checked by allowing only one autoreferencing node and imposing the referential integrity constraint. Categories have to be defined both the English and Spanish textual identification. Each new meaning has to be identified by at least one textual description (English or Spanish) in order to allow a description to be delayed. Moreover, an English or Spanish term has to be assigned to its synonym set.

The category a meaning belongs to can be assigned later, as well as possible terms for the see set. All this delayed information have to be detected in an automatic way in order for the author to resolve pendings.

6.Interface and Functionalities of the Tools

This section briefly describes the interfaces and functionalities of the user tool and the author tool we have developed.

6.1 The User Tool

We have developed an user tool, a query interface which allows us to easily recover the information about both English and Spanish terms as well as their relationships from the terminological database. This database holds the terms, categories, their attributes, and the relationships. The interface allows the user to navigate the semantic categories, also allowing to retrieve the relevant information of any term (definition, other related terms, translation, synonyms, …)

The Start window of this tool allows the user to select the base language (i.e., the source language for translations and for representing dialogs) among the available languages by pressing its button (from now on, we consider a bilingual dictionary so that it is unnecessary to select the source language or the target language.)

This action pops up the Semantic Category window, as shown in figure 8; its left pane shows the semantic categories structured as a tree, and the right pane, all the words under the highlighted semantic category. The total number of terms is showed on top of the right pane. The nodes in the tree can be clicked in order to expand or contract semantic categories subtrees. A text box is used for term lookups so that the closest word to the substring typed is shown in the right pane. Pressing Enter or double-clicking the highlighted word yields to the Query window. This window shows the relevant information about the selected term: its definition, comments, the list of semantic categories it belongs to (the one corresponding to the shown definition is highlighted), the synonym set, and the list of related terms. It also displays a navigation history. It is possible to select another semantic category in this window, which results in updating all the relevant information. Direct access to the terms in both the synonym and related terms windows is allowed by double-clicking.

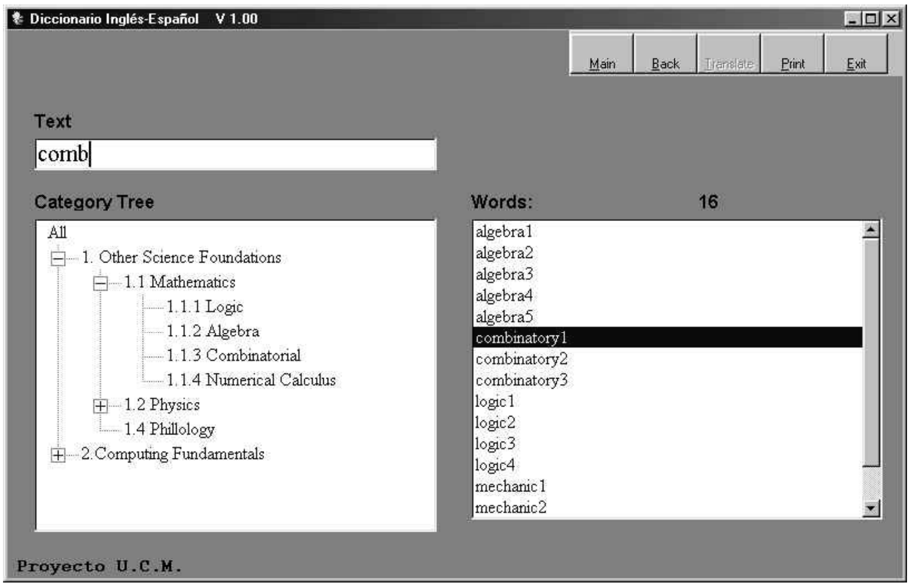

Figure 8. Semantic Category Window

The Semantic Category window has a control box with buttons to activate the return to the Start window, navigate backwards, translate the selected word, print, and exit the interface. The Translate button offers one of the main functionalities of this interface, i.e., the translation from the (source) base language to the target language and, when pushed, it pops up the Translation window (figure 9). This window shows a first field for the term in the first language, and a second field for the term in the second language. There are also navigation buttons for searching other terms in the same semantic category under an alphabetical order. It is possible to translate from the first or from the second language by using two buttons which express the two possible translation directions. Also, the Go to buttons allow us to go to the Semantic Category window for the selected term. This completes the overall description of the functionalities of the user tool.

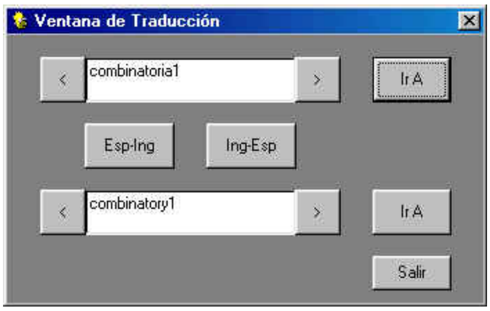

Figure 9. Translation Window

6.2 The Author Tool

The author tool allows the author to add new terms to the terminological database, and all the relevant information, such as its definition, semantic categories, meanings, synonym sets, and related terms. We have developed a Spanish user interface for this tool (easily rewritable to other language), and it consists mainly of one Author window, as shown in figure 10. It has several management areas (indicated by superimposition in this figure) which are explained next.

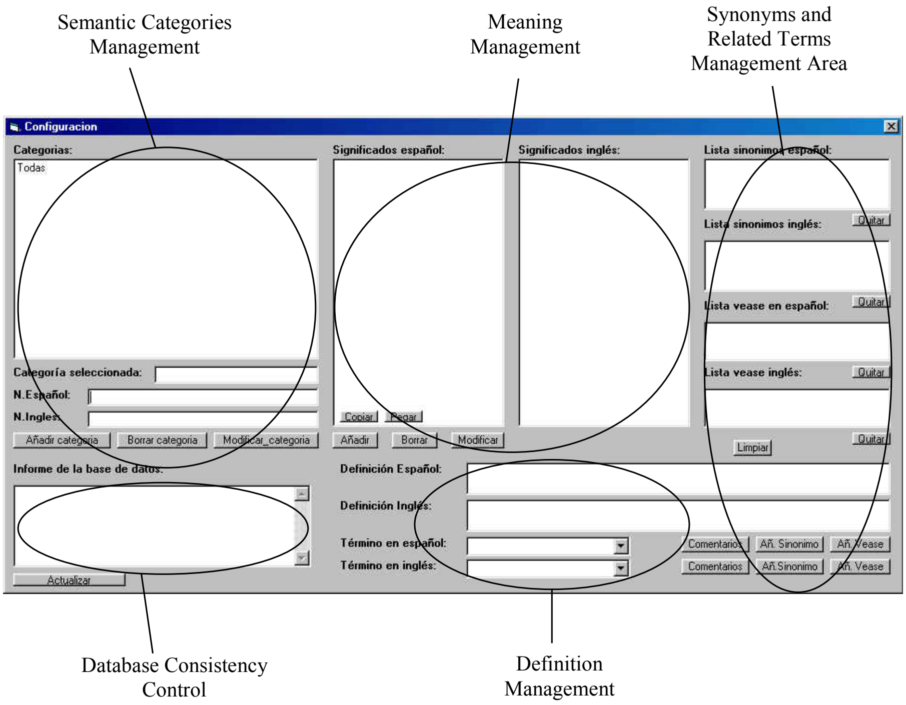

Figure 10. Author Window

- **Semantic Categories Management**
  This area is intended for managing all the operations related to semantic categories, as illustrated in figure 11 with a fragment of a taxonomy. It has several controls: a hierarchical view of the semantic categories (with expand/collapse functionality), text fields for the semantic category names (English and Spanish), and the buttons Add Category, Delete Category, and Modify Category. The insertion point when adding a new semantic category is the highlighted semantic category, and the Spanish and English texts for the semantic category name must be typed in the aforementioned text fields.

Figure 11. Semantic Category Management Area

- **Meaning Management**
  The area for meaning management, illustrated in figure 12, consists of two lists for the meanings in both languages and the buttons Add, Delete, and Modify for addition, deletion, and modification of meanings, as well as buttons for edition (Copy and Paste buttons.) These lists show the meanings in the form Term -> Definition for the highlighted category, so that one can see several meanings for the same term. Moreover, when a pair Term -> Definition is selected, the corresponding Term -> Definition translation is automatically highlighted (if present); there is a one to one mapping between meaning representation in all the languages. It should also be noted that meanings, which are language independent, are shown with the *best* representation we have in a given language, i.e., a pair Term -> Definition, since there are no other pair Term -> Definition2 with the same meaning (note that is the same term in both pairs.)

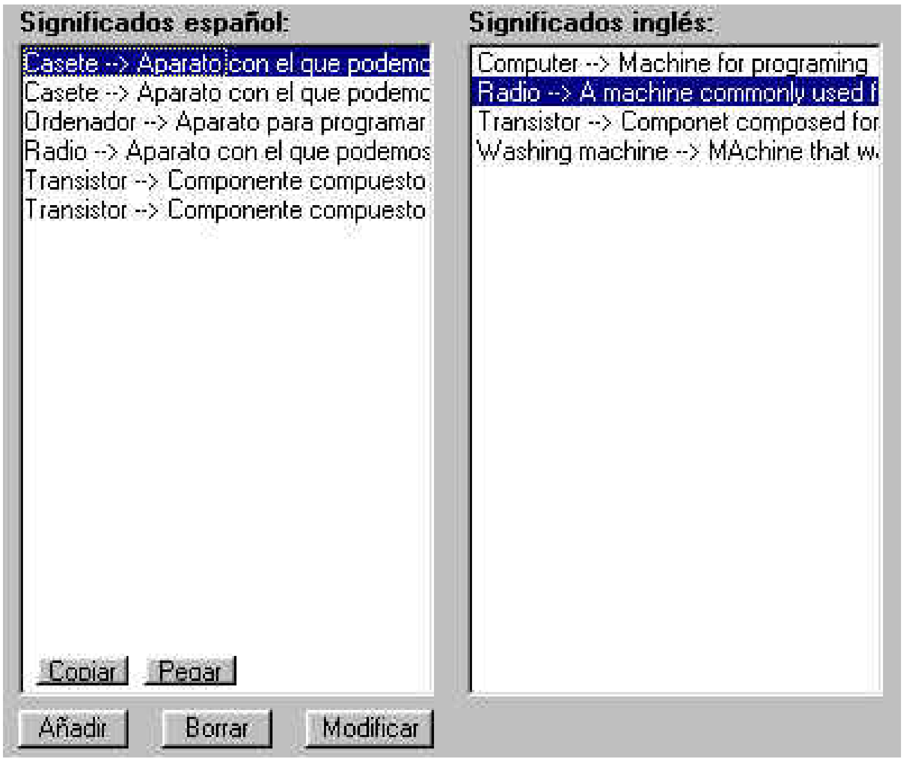

Figure 12. Meaning Management Area

- **Synonyms and Related Terms Management**
  The area on the right in figure 10 has four lists for the synonyms, and related terms in both languages which correspond to the highlighted meaning in the Meaning Management area.
- **Database Consistency Control**
  This area contains the button Update, which is used to obtain a report (text box Data Base Report) about consistency of the database (figure 11). Up to now, consistency detection only detects lack of textual definitions for terms, but it can be extended in order to detect other inconsistencies or omissions. This is quite important when authoring dictionaries, since a dictionary cannot be consistently built at each step, but it is constructively built from terms to relationships between terms (polysemy, synonymy.) For instance, this tool can be extended in order to give hints for detecting circular definitions (there are commercial dictionaries with this failure), for detecting possible lacks of synonym and related terms, and consistency constraints already note in subsection 5.2.1.
- **Definition Management**
  This area allows the user to define terms and textual definitions for meanings.

7.Conclusions

When developing tools to manage (authoring, modifying, querying) information systems for NLP applications, a sound methodology must be followed for reasons of portability and integrability. Here we have described the methodology applied along the development of authoring and using tools to manage a bilingual electronic dictionary based on ontology. The implemented system supports ontology concepts with a tree structure. Although its limitations, this methodology represents the beginning of the way to be followed in order to minimize and coordinate efforts, trying to create and use common linguistic resources.

An entity-relationship model has been designed for multilingual electronic dictionaries supporting ontologies in the form of taxonomies with a tree structure. The methodology applied (standard for relational databases) assures, by way of constraints and, in particular, functional dependencies, the consistency of the information.

Classification of meanings and a sound conceptual model are important for integrating multilingual knowledge bases based on ontologies.

Since the conceptual model we develop here supports ontologies, the implementation of information systems managing linguistic knowledge from that conceptual model must facilitate their integration into an ontology based lexical knowledge base.

Besides the conventional NLP applications, potential applications of these systems to education cannot be neglected, as we have shown thinking in authoring and querying of personal or common electronic dictionaries [Vaquero,2000].

% Future work

The presented tools can be enhanced in several ways. For instance, both the user and author tool can be deployed in a Web context in order to allow centralized information for queries, and, what is more important, to allow collaborative work at a distance. In addition, they can be extended with phonetic search.

% Further applications

As we are just at the beginning of the way, a long list of tasks must be made: to assume an entity-relationship model supporting acyclic directed graph-structured taxonomy, and then, adding to it support for general relationships between concepts and lexical knowledge for terms. And, from this conceptual model, continue applying sound development methodologies for reaching a robust system. We hope this way for methodology standardization will be followed by the people developing information systems for managing natural language knowledge.

8.References

[Casares] Ideological Spanish Dictionary.

[Fillmore,1992] C.J. Fillmore, and B.T. Atkins, “Toward a frame-based lexicon: The semantics of RISK and its neighbors”, Lehrer and Kittay, pp. 75-102, 1992.

[Katzenberg,1993]

[Lenat,1990] D.B. Lenat, and R.V. Guha, “Building Large Knowledge-Based Systems”, Reading, Massachussets, Addison-Wesley, 1990.

[MaríaMoliner] Derivative Spanish Dictionary.

[MikroKosmos] http://crl.nmsu.edu/Research/Projects/mikro/index.html

[Miller,1995] G. Miller, “WordNet: A Lexical Database for English”, Communications of the ACM, Vol. 38, 11, 1995.

[Nirenburg,1995] S. Nirenburg, V. Raskin, and B. Onyshkevich, “Apologiae Ontologiae”, Proceedings of the Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, Center for Computational Linguistics, Catholic University, Leuven, Belgium, pp. 106-114, 1995.

[Pressman,1997] R.S. Pressman, "Software Engineering. A Practitioner's Approach", McGraw-Hill, 1997.

[Silberschatz,1996] A. Silberschatz, H.F. Korth, S. Sudarshan, "Database System Concepts", WCB/McGraw-Hill, 1996.

[Vaquero,2000] Vaquero, A., Sáenz, F., and Barco, A. 2000. Improving the Language Mastery through Responsive Environments, SIIE’2000. Puertollano, 15-17 Nov. One of the 20 papers to be included in book form. Kluwer Academic Publishers.

[Wilks,1990] Y.A. Wilks, D.C. Fass, C.M. Guo, J.E. McDonald, T. Plate, and B.M.Slator, ”Providing machine tractable dictionary tools”. Machine Translation, 5, 1990, pp. 99-151.