APPLYING RELATIONAL DATABASE DEVELOPMENT METHODOLOGIES TO THE DESIGN OF LEXICAL DATABASES

F. Sáenz and A. Vaquero
Universidad Complutense de Madrid
Facultad de Informática UCM, Departamento de Sistemas Informáticos y Programación, E-28040 Madrid, Spain
{fernan, vaquero}@sip.ucm.es

ABSTRACT

We propose to apply relational databases (RDB) development methodologies to the design of lexical databases (LDB), which embody conceptual and linguistic knowledge. We represent the conceptual knowledge as an ontology, and the linguistic knowledge, which depends on each language, in lexicons. Our approach is based on a single language-independent ontology. Besides, we study some conceptual and linguistic requirements; in particular, meaning classifications in the ontology, focusing on taxonomies. We have followed a classical software development methodology for implementing lexical information systems in order to reach robust, maintainable, and integrateable RDB for storing the conceptual and linguistic knowledge.

KEYWORDS

Database Applications, Electronic Dictionaries, Ontologies, Lexicons.

1. INTRODUCTION

Weak attention has been paid on topics about development methodologies for building the software systems which manage LDB. We claim that the software engineering methodology subject is necessary in order to develop, reuse and integrate the diverse available linguistic information resources. Really, a more or less automated incorporation of different lexical databases into a common information system, perhaps distributed, requires compatible software architectures and sound data management from the different databases to be integrated. The database subject has already done a long way reaching a strong standardization, and supplying models and methods suitable for developing robust information systems. We apply relational database design methodologies to develop linguistic resources consisting of ontologies and lexicons. The conceptual knowledge is represented as an ontology, and the linguistic knowledge, depending on each language, is stored in its lexicon.

Subjects about electronic dictionaries for diverse natural language processing applications have been extensively studied [ZOC 03], [WIL 90], [WIL96], as well as LDB [MIL 95], world knowledge bases [LEN 90], ontologies in general [ONT], ontologies for computational linguistics [NIR], and the like. But there are no references on how these information systems have been developed and upgraded along their life. Moreover, tools for managing ontology-based linguistic information systems have been described [MOR 02], but there is no a declared software engineering approach for the development of these tools.

We have followed the classical relational database design based on the conceptual, logical, and physical models for building LDB, and software engineering techniques based on UML for building LDB interfaces (these are not described in this paper). The result is a methodology to develop information systems for building and querying lexical databases [SV 02]. Based on this methodology, we have developed software tools for authoring and consulting different kinds of linguistic resources: monolingual, bilingual and multilingual dictionaries.

Conventionally, dictionaries are conceived for human use and lexical databases are conceived for natural language processing (NLP) applications. Our methodology leads to friendly usable dictionaries, but structurally prepared to be easily embedded in computer applications, as we show along the paper.

The rest of the paper is organized as follows. Conceptual and linguistic requirements embodied in the lexical and ontological resources are first exposed in section 2, because of their relevance in building different lexical databases, such as electronic dictionaries, and distinguishing certain relevant aspects of our approach from others. The next section introduces how to apply the relational design methodology to develop LDB, and section 4 details its application to a bilingual dictionary. Finally, in section 5 certain conclusions are summarized and future work is foreseen.

2. CONCEPTUAL AND LINGUISTIC REQUIREMENTS

In this section, conceptual and linguistic knowledge incorporated in computing systems devoted to NLP are pointed out because of their relevance in the definition of the conceptual model showed below.

Regardless of the language, the knowledge in the discourse universe is conventionally divided in two classes: conceptual and linguistic. Terms and sentences refer to concepts, but they have particular structural and morphological features in each language. All of this information is not available in any dictionary, electronic or not, although it is the objective in the most exigent ontology-based linguistic Knowledge Bases, such as MikroKosmos [MIK].

In the next paragraphs, we limit the conceptual and linguistic knowledge to the level we are interested in. Then, we show the structure of these two kinds of knowledge, and how both are linked.

2.1 Lexicographic Order. From Paper to Electronic Dictionaries

No kind of term order is suitable for electronic dictionaries, because random direct access is better than alphabetical sequential one for human use. The first generation of electronic dictionaries [COW 99] is characterized by the direct access to terms, but the provided information and the ways for accessing to it differ from one dictionary to other, having unclear (not formally specified) structure and lack of declared development methodology. The new generation dictionaries intend to cover these holes.

2.2 Terms and Meaning. Polysemy and Synonymy

In every language there exists the well known naming problem [KAT 93], which consists of two elements: one is polysemy (under the synchronic point of view, that is, embodying polysemy itself and homonymy), by which a term can have several meanings; and the other is synonymy, by which one meaning can be assigned to different terms. We are going to study in the next section how to relate terms and meanings. The naming problem will be automatically solved by completely separating Lexicon from Ontology, as we shall see.

2.3 Semantic Relationships and Lexicon

Each meaning of a given term is precisely identified by its semantic category (category from now on, for the sake of brevity). Therefore, categories provide classification for meanings, and such classification can be arranged in a taxonomy [RK 02]. Here we do some remarks about the relationships among categories, meanings and terms. 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. We must also note that a category has a meaning described by a definition. This meaning is the extensional definition of the category. See [SV 02] for more details.

2.3.1 Lexical Databases

For a given language, we have a set of terms, meanings and categories holding certain relationships among them. Conventional LDB, such as WordNet [MIL 95], have term classification through synonymy

(grouped in the so-called synsets). LDBs based on ontological semantics go beyond by playing the role of meaning taxonomy and supporting more complex semantic relationships [NIR 95]. All of the relationships (meronymy, holonymy, hypernymy, hyponymy, and so on) represented in the more complete lexical databases, such as WordNet or EuroWordNet [EWN], are also represented in ontology-based databases, such as MikroKosmos; but in this case, all of the concepts and their relationships are present in the ontology, while each lexicon has the terms for each language and their linguistic arguments, as well as the links with the concepts into the ontology. The mapping between ontology and lexicon is the key for successfully coordinate all of the lexical and semantic relationships. This approach does full separation between ontology and lexicon, and if we now think of several languages, then the same ontology applies for each one of the lexicons.

2.3.2 Our LDB for Dictionaries

In our approach, relationships among terms from different languages come from considering jointly the involved Ontology-Lexicon schemes, as we will see later when considering the bilingual dictionary. In the dictionary here considered, the ontology only consists of one relationship which gives tree-structure to the conceptual taxonomy. A taxonomy is a natural structure for meaning classification. Each node in the taxonomy corresponds to a category. In principle, every category in the taxonomy can have meanings, regardless of its taxonomy level. It must be noted that every category in the taxonomy contains at least the term which names the category, so that all categories are non-empty. On the other hand, the creation of new categories as belonging to several predefined ones should be avoided, in order to reach a compact relationship as the taxonomy structuring backbone. Next sections show the development of a dictionary without overlapped classifications [RK 02], and only permitting tree-structured taxonomies. Since a meaning can belong to different categories, the extensional definition of categories is hold [SV 02].

When consulting or building dictionaries, there are a number of advantages in classifying meanings as taxonomies. First of all, meaning taxonomy is a useful facility for an electronic dictionary, because meaning classification embodies additional semantics, which provides more information to the user than that usually provided. As long as we know, this kind of facilities (meaning classification), normally used in conceptual modeling through ontologies [MCG 00], has not been implemented before into dictionaries.

One demanded facility in electronic dictionaries is the semantic relationship ‘See’ among terms. When a definition for a term A in a dictionary has the entry ‘See B’ (B is another term) it only refers to B, not the particular definition for B the author thought of, so that the user has to read all the definitions assigned to B until he reaches the intended one. Section 4 shows how we solve this problem in our approach.

3. DESIGNING LEXICAL DATABASES WITH RELATIONAL TECHNOLOGY

We understand lexical databases as information systems which are composed of a database core and an application layer which allows the user and applications to interact with the lexical data. On the one hand, having a database core instead of other file related approaches comes from well know issues in the database community (e.g., see classical texts as [SKS 02]). In particular, we do need integrity constraints for maintaining consistency when modifying data. On the other hand, the application layer should be understood as possibly containing user interfaces for both consulting and modifying lexical data, as well as NLP applications. When considering this two components of the information system, we do isolate data from applications, so that all consistency checking is encapsulated into the database core.

Both components should be developed following known software engineering methods. It is more likely to find these methods applied to the application layer, but, in general, we do not find them applied to the modeling of lexical databases.

In our work, we focus on relational databases because of a number of reasons: they are widely used, efficient RDBMS (Relational Database Management Systems) are available, and a database design methodology has matured for them. The latter is the most important point we highlight, since it provides several design stages which help in designing consistent (from an integrity point of view) relational databases. This methodology comprises the design of the conceptual scheme (using the Entity/Relationship (E/R) model) and the logical scheme (using the relational model). A final stage is the physical scheme, which

is generally omitted in the literature since it depends tightly on the target RDBMS. Since in this paper we are not concerned with performance issues, we concentrate on the first two design stages, although we pose some remarks about the physical design.

We emphasize here the dependence between the design stages and the DB structure. Besides, the way to build a LDB comes through this dependence, as is expressed in section 4 after considering the constraints in section 3.1.

In other projects of LDB, when RDB techniques have been applied, there is no awareness of how this dependence is crucial to establish a development methodology and a formal common DB structure. We take two examples as representative samples. In [MOR 02], an E/R model is defined, but there is no expressed relation between the development stages and the DB creation. MILE [ABB 02] uses an E/R model in the lexical entry for automatically generating a RDB with different purposes. Our approach leads to very different E/R models, with less complexity. Besides, the development of their DB is not described neither the integrity constraints.

3.1. Constraints in Relational Design

The relational database design methodology is not only focused on representing data and their relations, but more important for us in this work, constraints about them. These constraints allow us to impose restrictions for both data and relations that any database instance must obey. Although these constraints can be implemented in the application layer, we advice against this. We claim that they must be implemented in the database core because consistency would be maintained by the RDBMS, instead the applications. By this, the constraints encapsulated into the database are independent from the applications. Next, we introduce the constraints at each design stage which are useful for our purposes.

The E/R model is the most common tool for the first design stage, the conceptual modeling, allowing several kind of constraints which we relate with the constraints needed in a lexical database, since there are several (philosophical) notions that such an ontology-based database has to represent (e.g., identity and membership [GW 00]). For instance, primary and candidate keys are used for the identity concept, i.e., given a class (entity set), every instance of the class (entity) can be unambiguously identified.

The relational model used in the second design stage, in particular offers several kinds of constraints, to implement those coming from the E/R model.

Constraints at the final database design stage, whose result is the physical scheme, depends on the RDBMS considered, but usually we find common constraints as well as constraint predicates (by means of the CHECK clause and triggers).

Because of the authoring nature of lexical databases we cannot impose all of the identified constraints (since there is absent information which can be known afterwards). Therefore, we are ought to provide consistency checking features to the lexical database authors. These features must inform the author about authoring constraints which are violated by the instance database. Such constraints which may be violated during the authoring are known as *soft* constraints, by contrast with the *hard* constraints that every database instance must hold at any time.

4. DESIGNING A LEXICAL DATABASE FOR A BILINGUAL DICTIONARY

As stated in former sections, we are interested in the representation of language information from an ontology point of view in order to build a lexical database, and, in this section, for a bilingual dictionary. First of all, we need to represent the meaning (concept) as a language independent entity, so that a set of terms (the so-called synonym set – synset in WordNet) in a given language is used to identify such a meaning. In this way, the synonymy property holds for the set of terms in a particular language related to a meaning. Further, a synset for each language can be found. Polysemy comes from the fact that a given term may be in different synsets for the same language (obviously related to different meanings). Finally, we are interested in classification of meanings, which can be represented with categories related to meanings, so that each meaning belongs to a category. If we restrict classifications to taxonomies, we have to impose a

constraint stating that a category can only have a parent category, and only one category (root category) can have no parent.

Since we are interested in an ontology-based lexical database, we must highlight some points. Meanings are directly related to categories, instead of terms. Synonymy is a set-oriented property of terms, and the set itself is related to a meaning, instead of each term in the set. A term in a synset belongs to a category via a transitive relation among the synset, the meaning the synset belongs to, and the category the meaning is classified under. In order to fulfill the intensional definition of categories explained in section 2.3, a meaning is needed for defining each category, and a non-empty synset is needed for such a meaning.

4.1. Conceptual Design for the Bilingual LDB

Following these premises, we propose the E/R scheme shown in Figure 5 (an upgrade from [SV 02]) as a result of the first stage design (conceptual modeling). In this figure (following some recommendations in [SKS 02]), entity sets are represented with rectangles, attributes with ellipses (those which form a primary key are underlined), and relationship sets with diamonds, which connect entity sets with lines. Undirected lines (edges) represent a many to many mapping cardinality. A one to many mapping cardinality from entity set A to entity set B is represented by an arc from B to A, meaning that an entity belonging to B is related at most with an entity in A. A total participation of an entity set in a relationship set is represented by double lines. Undirected lines also connect attributes to entity sets. Relationship set and entity set names label each diamond and box, respectively. Each side of a relationship set relating an entity set with itself is labeled with its role.

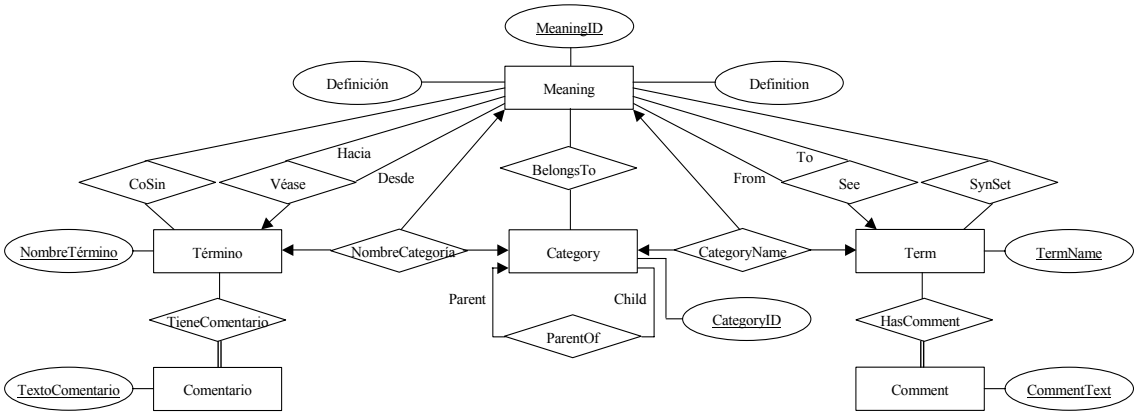

Fig. 1. Entity-Relationship Scheme for an English-Spanish LDB

In this figure, we show an instance of a simple bilingual lexical database for Spanish and English. In the following, we describe entity sets and its attributes, relationship sets, and constraints.

Entity Sets. The entity set Meaning is the central entity set other entity sets rest on and has three attributes: MeaningID (artificial attribute intended only for entity identification as shall be explained later), Definition and Definición, intended for the textual definitions of the meaning in both languages, English and Spanish, respectively. The entity set Term represents all of the English terms that compose the lexical database, and it has one attribute: TermName, which denotes the textual name of each term in this set. The entity set Category denotes the category each meaning belongs to, and it has one attribute: CategoryID (similar to MeaningID). The entity set Comment represents the comments about each term, and it has the attribute CommentText, which holds the textual comment for each term in this set. This entity arises from our need to develop a dictionary which can hold comments about terms in particular, not related to the concept itself (for instance, comments about the origins of the term). The entity and relationship sets from the Spanish language (CoSin, Véase, Término, TieneComentario, Comentario, and NombreCategoría) are homologous to the ones in English (SynSet, See, Term, HasComment, Comment, and CategoryName, respectively).

Relationship Sets. The relationship set SynSet between Meaning and Term denotes the English synonym set. The relationship set See denotes the semantic relationship ‘See’ among two meanings and a term (given a meaning, the user is referred to a representative term of another meaning, which is linked with the former via the 'See' relation). The relationship set BelongsTo between Category and Meaning is used to categorize

meanings, and it 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 categorize meanings). The relationship set ParentOf is used to represent taxonomies. The relationship set CategoryName is intended to relate a category with the term which names it, under the meaning that defines the category. The relationship set HasComment links comments with terms.

Constraints. Mapping cardinalities are as follows: SynSet is many to many since a synonym set may contain several terms, and a term may be contained in several synonym sets (obviously, with different meanings). The ternary relationship set See which connects Meaning (two times for the "from" and "to" parts) and Term is many to many because a meaning may refer to several English terms, and one term may be referenced by several meanings. BelongsTo 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 our goal in developing dictionaries is to keep the classification as disjoint as possible). ParentOf is one to many since a given category has only one parent, and a given category can have multiple children. CategoryName has cardinality one for the three entity sets related because terms, meanings, and categories are unique in this set. HasComment is many to many since a term may have several comments attached and a comment may refer to several terms

Note that there are less total participation constraints that one could expect, all of them derived from the incremental creation of a database instance, because of the following reasons. A meaning does not have to be categorized. A meaning does not have to have a term for its representation in *one* language (if we create a meaning, it is likely to have at least a term in a language for its representation, but not necessarily in both languages). A category may have no name (a term) in a given language provided that its name is defined in the other language. A category does not have to have related meanings. Finally, ParentOf has no total participation since a category may have no parent (the root category), and a category may have no children (leaf categories).

A consistent LDB should hold total participation for the former constraints but they should be considered as soft constraints since they can be violated during the authoring process. We can identify other soft constraints which cannot be expressed with E/R-related constraints. For instance, a given meaning must have synsets in *both* languages in order to find translations, categories must be arranged in a tree, and all of the categories must have names in *both* languages. These constraints which cannot be expressed with E/R constructors are known as predicate constraints.

All of the attributes, but Definition and Definición, are primary keys. This means that they have an existence constraint automatically attached. But, if we consider that, for instance, a meaning is added to the database, it can be from any of the two languages, i.e., the LDB designer may have an English or Spanish definition for it. Although we can think of the attributes Definition and Definición as candidate keys, they cannot be since the null value will be, in general, in any of them. Therefore, an extra attribute is needed for identifying this entity set, which we call MeaningID. In the physical model, these attributes must have a type for identifiers (such as the sequences or autonumbers). From the discussion above, we should also impose soft existence constraints (for instance, there should be a definition for each meaning) and hard uniqueness constraints (each definition must be different) for Definition and Definición.

Our proposal presented in this paper assumes a simple bilingual dictionary for the sake of clarity. However, we have developed a more involved conceptual model for multilingual dictionaries, which abstracts the relationship and entity sets for each language, so that similar relationship and entity sets are not replicated (e.g., Term vs. Término) [SV02].

4.2. Logical Design for the Bilingual LDB

The objective of this stage is to design a logical scheme for the database from the conceptual scheme 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 underlined attributes. For the bilingual dictionary, we get the following (showing only the English part):

Objects in the E/R model	**Relations in the Relational Scheme**
Entity sets:	Meaning(MeaningID, Definition, Definición)

	Term(TermName)
	Category(CategoryID)
	Comment(CommentText)
Relationship sets:	SynSet(MeaningID, TermName)
	See(FromMeaningID, ToMeaningID, TermName)
	BelongsTo(MeaningID, CategoryID)
	ParentOf(Parent, Child)
	CategoryName(CategoryID, MeaningID, TermName)
	HasComment(TermName, CommentText)

As hard constraints, this stage inherits the constraints identified in the first design stage and adds referential integrity constraints for all of the relationship sets wrt. entity sets, and functional dependencies as:

- Definition → MeaningID, Definición (Meaning)
- Definición → MeaningID, Definition (Meaning)
- MeaningID → TermName, CategoryID (CategoryName)

As soft constraints we identify existence constraints for all of the non-primary key attributes, mapping cardinalities (a meaning should only belong to a category), total participation (for all of the entity sets but Term in HasComment and Término in TieneComentario), and predicate constraints (the ones noted in the previous design stage).

4.3. Notes about the Physical Design

Although we do not focus here in the physical design, we should note several issues regarding the translation from the logical scheme into the physical scheme.

From a performance point of view, tables should not be identified by long fields. Attributes in the relational model as CommentText and TextoComentario are expected to have long names. Since a primary key is implemented with an index with the same key, this key should be as small as possible in order to boost both index search and referential integrity constraint checking. Therefore, a relation coming from the logical scheme with a long key should be translated into a table with a new short key, as we can usually find in most relational designs. Nevertheless, the original primary key must be indexed (as a candidate key) in order to maintain consistency, and moreover the key for this index must be unique and without nulls.

Another source of performance gain comes from simplifying the number of tables. For instance, a table coming from a relationship set with one to many mapping cardinality can be merged with the table for the entity set with a participation of one. This would be the case of ParentOf, which can be merged into the table Category by adding the field Parent to this table. However, any modification of the relational scheme should be guided by a cost analysis of the involved operations (queries). This issue is not in the scope of this paper, although it plays an important role in performance when building large LDB.

A total participation can be implemented at this stage with the so-called deferred referential integrity constraints, which are intended for specifying that, for instance, HasComment.CommentText should be in Comment.CommentText and viceversa. Functional dependencies are hard constraints which can be implemented via triggers, which are activated each time a modification in the target table is performed. Finally, soft constraints should be coded as stored procedures in order to maintain the consistency checking at the database level, so that users and applications can test the current consistency degree of the database instance. In addition, they must also be implemented via SQL statements, because of the same reason stated for hard constraints.

5. CONCLUSION

Continuing with the refinement of our development methodology of information systems for LDB, an elaborated and sound design method has been presented here. The design has been tested and used to complete the development of certain information systems to build and consult monolingual, bilingual and multilingual dictionaries.

The advantages of applying software engineering principles and methods to information systems for lexical databases are indeed evident. Moreover, by using the resulting tools, the LDB authoring is friendly simple, and the inserted information has to accomplish certain constraints (consistency, non recurrence, ...) controlled by the system, helping the authoring process (avoiding violation of hard constraints and reporting the violation of soft constraints). Besides, the integration of diverse LDB built with these tools is assured by the migration tools developed for this purpose. In addition, the resulting dictionaries are friendly usable and supply very useful semantic information to the reader. Finally, as future work, we foresee:

- Refining the design and development methodology from the current state, in order to take into account other possible structures of the taxonomy (for instance, graph-shaped classifications), providing to the ontology with support for explicit generalized relationships, and admitting more linguistic information in the terms of the lexicons.
- Developing new information systems according to the required characteristics of the LDB to come in the future.
- Studying the application of the methodology to the integration of heterogeneous LDBs, interoperability among them, and so on.
- Building LDBs structurally prepared to be easily embedded in NLP applications.
- Applying the tools to formal and informal Education with the aim of building individual or community dictionaries.

Acknowledgments

This work has been partially supported by the Spanish CICYT project number DPI2002-02924.

REFERENCES

[ABB 02] Atkins S., Bel N., Bertagna F., Bouillon P., Calzolari N., Fellbaum C., Grishman R., Lenci A., MacLeod C., Palmer M., ThurmairR., Villegas M., Zampolli A. (2002) "From Resources to Applications. Designing TheMultilingual ISLE Lexical Entry". In Proceedings of LREC 2002, Las Palmas, Canary Islands, Spain.

[COW 99] A. P. Cowie “English Dictionaries for Foreign Learners. A History”. Oxford. Clarendon Press, 1999.

[EWN] http://www.uva.nl/EuroWordNet.html

[KAT 93] B. Katzenberg and P. Piela, "Work Language Analysis and the Naming Problem", Communications of the ACM, Vol. 36, No. 4, June 1993.

[LEN 90] D.B. Lenat, and R.V. Guha, “Building Large Knowledge-Based Systems”, Reading, Massachussets, Addison-Wesley, 1990.

[MCG 00] Deborah L. McGuinness. "Conceptual Modeling for Distributed Ontology Environments," In the Pro-ceedings of The Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000), Darmstadt, Germany, August 14-18, 2000.

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

[MIL 95] G. Miller, “WordNet: A Lexical Data Base for English”, Communications of the ACM, Vol. 38, 11, 1995.

[MOR 02] A. Moreno, and C. Pérez, “Reusing the Mikrokosmos Ontology for Concept-based Multilingual Terminology Databases”, Proceedings of LREC2000, 2002.

[NIR 95] 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.

[NIR] S.Nirenburg and V.Raskin, « Ontological Semantics »- In http://crl.nmsu.edu/Staff.pages/Technical/sergei/book.html

[ONT] http://www.ontology.org/main/papers/iccs-dlm.html

[RK 02] C. Raguenaud and J. Kennedy, “ Multiple Overlapping Classifications: Issues and Solutions”. 14th International Conference on Scientific and Statistical Database Management (SSDBM'02). Edingburgh, Scotland, 2002.

[SV 02] Sáenz, F. & Vaquero, A. “Towards a Development Methodology for managing Linguistic Knowledge Bases”. Proceedings ES’2002. Springer-Verlag, 2002. pp 453 – 466.

[SKS 02] A. Silberschatz, H.F. Korth, S. Sudarshan, "Database System Concepts", WCB/McGraw-Hill, 2002.

[WIL 90] 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.

[WIL 96] Y. Wilks, B. M. Slator, and L.M. Guthrie, "Electric words: Dictionaries, Computers and Meanings". MIT Press. Cambridge, 1996.

[ZOC 03] M. Zock and J. Carroll “Les dictionaires électroniques”. TAL, Vol. 44, 2. 2003.