

Methodologically Designing a Hierarchically Organized Concept-Based Terminology Database to Improve Access to Biomedical Documentation*

Antonio Vaquero¹, Fernando Sáenz¹, Francisco Alvarez², and Manuel de Buenaga³

¹ Universidad Complutense de Madrid, Facultad de Informática, Departamento de Sistemas Informáticos y Programación, C/ Prof. José García Santesmases, s/n, E-28040, Madrid, Spain

² Universidad Autónoma de Sinaloa, Ángel Flores y Riva Palacios, s/n, C.P 80000, Culiacán, Sinaloa, México

³ Universidad Europea de Madrid, Departamento de Sistemas Informáticos, 28670 Villaviciosa de Odón. Madrid, Spain

{vaquero, fernan}@sip.ucm.es, fjalvare@fdi.ucm.es, buenaga@uem.es

Abstract. Relational databases have been used to represent lexical knowledge since the days of machine-readable dictionaries. However, although software engineering provides a methodological framework for the construction of databases, most developing efforts focus on content, implementation and time-saving issues, and forget about the software engineering aspects of database construction. We have defined a methodology for the development of lexical resources that covers this and other aspects, by following a sound software engineering approach to formally represent knowledge. Nonetheless, the conceptual model from which it departs has some major limitations that need to be overcome. Based on a short analysis of common problems in existing lexical resources, we present an upgraded conceptual model as a first step towards the methodological development of a hierarchically organized concept-based terminology database, to improve the access to medical information as part of the SINAMED and ISIS projects.

1 Introduction

Since the days of machine-readable dictionaries (MRD), relational databases (RDB) have been a popular device to store information for linguistic purposes. Relational database technology offers many advantages, being one of its more important ones the existence of a mature software engineering database design methodology. Nevertheless, most of the efforts aimed at developing linguistic resources (LR), whether they used RDB or not, have focused on content, implementation or time-saving issues, putting aside the software engineering aspects of the construction of LR.

* The research described in this paper has been partially supported by the Spanish Ministry of Education and Science and the European Union from the European Regional Development Fund (ERDF) - (TIN2005-08988-C02-01 and TIN2005-08988-C02-02).

Many authors use the term “software engineering” synonymously with “systems analysis and design” and other titles, but the underlying point is that any information system requires some process to develop it correctly. The basic idea is that to build software correctly, a series of steps (or phases) are required. These steps ensure that a process of thinking precedes action: thinking through “what is needed” precedes “what is written”. Although software engineering spans a wide range of problems, we will focus here on the database design aspects.

As it will be seen later, design issues are important when using RDB. Moreover, as we stated in [1], design is also important because in order to develop, reuse and integrate diverse available resources, into a common information system, perhaps distributed, requires compatible software architectures and sound data management from the different databases to be integrated. With that in mind, we have defined a methodology [1], for the design and implementation of ontology-based LR using RDB and a sound software engineering approach. Nevertheless, the conceptual model we propose as a point of departure of the methodology has some major limitations, which have to be overcome in order to create structurally sound LR.

In this paper, we will focus on the ontology representation limitations of our previous model (leaving the lexical side limitations for a future paper), and create a conceptual model of the ontological part, that overcomes such limitations as part of our efforts to have a solid foundation for action. Our final goal is to create a LR (a hierarchically organized concept-based terminology database) that will be part of an intelligent information access system that integrates text categorization and summarization, to improve information access to patient clinical records and related scientific documentation, as part of the SINAMED and ISIS projects [2].

The rest of the paper is organized as follows. In section 2, the advantages and disadvantages of RDB are pointed out, as well as the importance of database design in the construction of ontology-based LR. In section 3, some common problems of LR are summarized, and the need to develop methodologically engineered application-oriented LR is signaled. In section 4, the methodological gaps of past developing efforts are underlined. In section 5, a set of ideas intended to help developers to formally specify and clarify the meaning of concepts and relations are depicted. In section 6, a conceptual model that integrates the aforementioned ideas is introduced and described. Finally, in section 7 some conclusions and future work are outlined.

2 Designing LR Using RDB

RDB present a series of advantages that have been taken into account when used to construct databases for linguistic purposes [1, 3, 4, 5, 6]. From a software engineering point of view, their main advantage is that they provide a mature design methodology, which encompasses several design stages that help designing consistent (from an integrity point of view) databases. This methodology comprises the design of the conceptual scheme (using the Entity/Relationship (E/R) model), the logical scheme (using the relational model), and the physical scheme.

However, RDB have various drawbacks when compared to newer data models (e.g., the object-oriented model): a) Impossibility of representing knowledge in form of rules; b) Inexistence of property inheritance mechanisms; and c) Lack of expressive power to represent hierarchies. In spite of this, by following a software engineering approach, that is, by paying attention to the database design issues [4], most of these drawbacks can be overcome, and thus, let us take advantage of all the benefits of RDB.

For instance, in [5] we can see how an UML (object-oriented) model is implemented within a RDB in a way that supports inheritance and hierarchy. Another similar example is found in [7], where the authors reproduce the structure of the Mikrokosmos ontology, using the E-R model. Other models [3, 8], although machine translation oriented follow a purely linguistic approach, and are not intended to overcome any of the limitations of the relational data model.

As it can be deduced, we have focused on the limitations of RDB to represent ontologies. There are several reasons why we have done that. First, our work is focused on the design and implementation of ontology-based LR using RDB [1]. Second, it has been proved by [9] that the use of a hierarchically organized concept-based terminology database, improves the results of queries on clinical data, and such is the goal of our projects. Third, we agree with [4, 10, 11, 12], when they state that the computationally proven ontological model, with two separated but linked levels of representation (i.e. the conceptual-semantic level and the lexical-semantic level) is our best choice for linguistic knowledge representation.

We have only found one reference, of a development effort that follows our software engineering approach for the development of ontology-based LR: the aforementioned work of [7]. The difference between our model [1] and the one in [7] is that ours only follows the ontological semantics ideas of Mikrokosmos; it does not recreate its frame-based structure. Nevertheless, although the model in [7] replicates the powerful ontological structure of Mikrokosmos in a RDB, it inherits all its problems (some will be described in the next section). As for the model we present in [1], it has a thesaurus-like structure where the concepts of the ontology are linked by a single implicit and imprecise relation; a situation that is problematic and severely limits the model, as it will be shown next.

3 Some Common Problems in LR

It is relatively easy to create a conceptual model of a LR. As seen in the previous section, this has already been done. However, existing LR (ontology-based or not) are plagued with flaws that severely limit their reuse and negatively impact the quality of results. Thus, it is fundamental to identify these flaws in order to avoid past and present mistakes, and create a sound conceptual model that leads to a LR where some of these errors can be avoided.

Most of the problems of past and present LR have to do with their taxonomic structure. For instance, once a hierarchy is obtained from a Machine-Readable

Dictionary (MRD), it is noticed that it contains circular definitions yielding hierarchies containing loops, which are not usable in knowledge bases (KB), and ruptures in knowledge representation (e.g., a utensil is a container) that lead to wrong inferences [13]. WordNet and Mikrokosmos have also well-known problems in their taxonomic structure due to the overload of the is-a relation [14, 15]. In addition, Mikrokosmos represents semantic relations as nodes of the ontology. This entails that such representation approach where relations are embedded as nodes of the ontology is prone to suffer the same is-a overloading problems described in [14, 15], as well as the well-known multiple inheritance ones (figure 1 illustrates this point by showing part of the Mikrokosmos ontology). In the biomedical domain, the UMLS has circularities in the structure of its Metathesaurus [16], because of its omnivorous policy for integrating hierarchies from diverse controlled medical vocabularies whose hierarchies were built using implicit and imprecise relations. Some of the consequences of these flaws, as well as additional ones have been extensively documented in [10, 11, 14, 17, 18, 19, 20, 21] for these and other main LR.

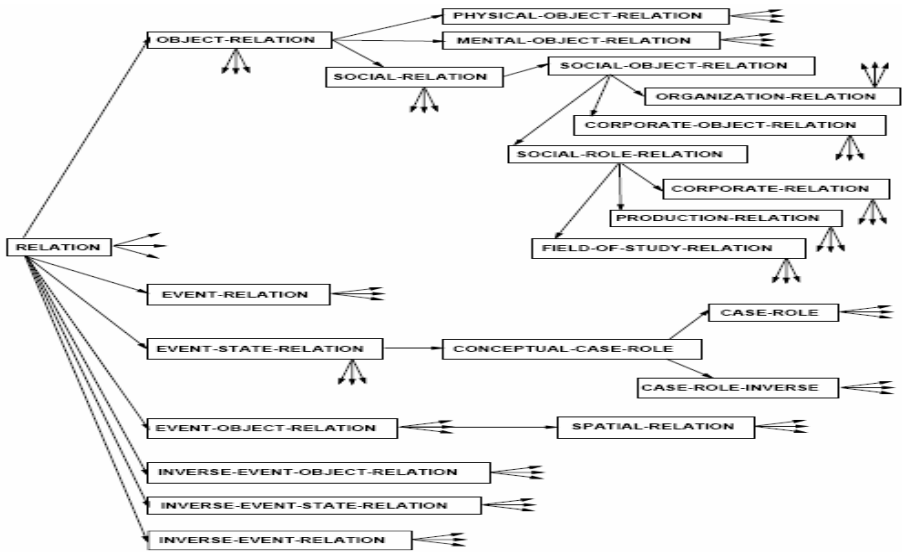


Fig. 1. Embedded Relations in the Mikrokosmos Ontology

3.1 Methodologically Engineered Application-Oriented LR

We have come a long way from the days of MRD. However, still today, the focus is on coverage and time-saving issues, rather than on semantic cleanness and application usefulness. Proof of this are the current different merging and integration efforts aimed at producing wide-coverage general LR [21, 22], and the ones aimed at (semi)automatically constructing them with machine learning methods [23, 24]. However, no amount of broad coverage will help raise the quality of output, if the

coverage is prone to error [11]. We should have learned by now that there are no short cuts, and that most experiments aimed at saving time (e.g., automatically merging LR that cover the same domains, or applying resources to NLP that are not built for it, like machine-readable dictionaries and psycholinguistic-oriented word nets) are of limited practical value [25]. Furthermore, in the current trend of LR development issues such as how to design LR are apparently less urgent, and this is haphazard. More attention must be paid on how LR are designed and developed, rather than what LR are produced.

The experience gained from past and present efforts clearly points out that a different direction must be taken. As [18] pointed out back in the days of MRD: “rather than aiming to produce near universal LR, developers must produce application-specific LR, on a case by case basis”. In addition, we claim that these LR must be carefully conceived and designed in a systematic way, according to the principles of a software engineering methodology. This is especially true if RDB are to be used as a knowledge representation schema for LR.

4 Methodological Gaps in the Development of LR Using RDB

Since we are interested in the development of a LR using RDB, it is worth mentioning that all the cited efforts in section 2, although they produced useful resources, they forgot about the methodological nature of RDB. They all stopped at the conceptual design stage. Thus, there is not a complete description of the entities, relationships and constraints involved in the conceptual and logical design of the DB.

The methodology we propose in [1] encompasses all of the database design phases. Nonetheless, the conceptual model from which it departs has several problems with respect to ontology representation; mainly, its does not foresee any control and verification mechanism for clarifying the semantics of relations, a problem that as seen in section 3 is of main concern.

Hence, if we are to design a hierarchically organized concept-based terminology database using RDB, our conceptual model must take also into account the semantic relations issue. As a first step, we enhance the conceptual model presented in [1] as shown in the next section.

5 Refining the Semantics of Concepts and Relations

In order to give our first step towards the enhancement of the conceptual model, we need to clearly state what are the elements that will be abstracted and represented in our upgraded conceptual model, that will help us to: a) build application-oriented LR (as pointed out in section 3.1); and b) avoid the problems present in existing LR as described in section 3.

These elements are concepts, properties of concepts, relations, and algebraic and intrinsic properties of relations. They will help an ontology developer to specify for concepts and relations formal and informal semantics that clarify the intended meaning of both entities in order to avoid the problems discussed in section 3.

Informal semantics are the textual definitions for both concepts and relations, as opposed to formal semantics that are represented by the properties of concepts and relations.

However, the fact that these elements will be part of the enhanced conceptual model does not imply that they are an imposition but rather a possibility, a recommendation that is given to each ontology developer. In the following, we detail the elements surrounding the basic element of our model: concepts.

5.1 Properties of Concepts

These are formal semantic specifications of those aspects that are of interest to the ontology developer. In particular, these specifications may be the metaproperties of [15] (e.g., R, I, etc.) In our application-oriented approach to LR development, only the properties needed for a concrete application domain should be represented. These properties play an important role in the control of relations as it will be seen later.

5.2 Relations

Instead of relations with an unclear meaning (e.g. subsumption), we propose the use of relations with well-defined semantics, up to the granularity needed by the ontology developer. Moreover, we refuse to embed relations as nodes of the ontology (because of the problems commented in section 3) or to implicitly represent any relation as it is done in Mikrokosmos with the is-a relation. We call these, explicit relations. This represents a novelty and an improvement when compared to similar design and implementation efforts as [7] based on RDB. In the next two subsections, we will describe the elements that help clarifying the semantics of relations.

5.3 Algebraic Properties of Relations

The meaning of each relation between two concepts must be established, supported by a set of algebraic properties from which, formal definitions could be obtained (e.g., transitivity, asymmetry, reflexivity, etc.). This will allow reasoning applications to automatically derive information from the resource, or detect errors in the ontology [26]. Moreover, the definitions and algebraic properties will ensure that the corresponding and probably general-purpose relational expressions are used in a uniform way [26]. Tables 1 and 2 (taken from [26]) show a set of relations with their definitions and algebraic properties.

Table 1. Definitions and Examples of Relations

Relations	Definitions	Examples
$C \text{ is-a } C_1$	Every C at any time is at the same time a C_1	<i>myelin is-a lipoprotein</i>
$C \text{ part-of } C_1$	Every C at any time is part of some C_1 at the same time	<i>nucleoplasm part-of nucleus</i>

Table 2. Algebraic Properties of Some Relations

Relations	Transitive	Symmetric	Reflexive
Is-a	+	-	+
part-of	+	-	+

5.4 Intrinsic Properties of Relations

How do we assess, for a given domain, if a specific relation can exist between two concepts? The definitions and algebraic properties of relations, although useful are not enough. As [15] point out, we need something more. Thus, for each relation, there must be a set of properties that both a child and its parent concept must fulfill for a specific relation to exist between them. We call these properties, intrinsic properties of relations. For instance, in [15] the authors give several examples (according to their methodology) of the properties that two concepts must have so that between them there can be an is-a relation.

6 Designing the Conceptual-Semantic Level of the Concept-Based Terminology Database

In this section, we present the conceptual model (an E/R scheme upgraded from our model in [1]) shown in figure 2, for the conceptual-semantic level of our future terminology database as a result of the first design phase, where all the ideas described in section 5 have been incorporated. However, as it was previously established, the model will reflect only the ontology part of our future hierarchically organized concept-based terminology database.

The entity set Concepts denotes the meaning of words, and it has two attributes: ConceptID (artificial attribute intended only for entity identification), and ConceptDefinition, intended for the textual definition of the meaning (informal semantics). The entity set ConceptProperties represents the set of formal properties described in section 5.1, and it has one attribute: ConceptProperty used to represent each property.

The entity set Relations represents the set of relations that can exist in an ontology, and it has two attributes: Relation that captures the textual name of each relation (e.g., is-a, part-of, etc.), and RelationDefinition for the textual definition of relations (informal semantics) as illustrated in table 1.

The entity set AlgebraicProperties represents the properties of relations (formal semantics) as seen in table 2, and it has one attribute: AlgebraicProperty that denotes each algebraic property. The entity set IntrinsicProperties conveys the set of properties mentioned in section 5.4 and has one attribute: IntrinsicProperty which represents each intrinsic property.

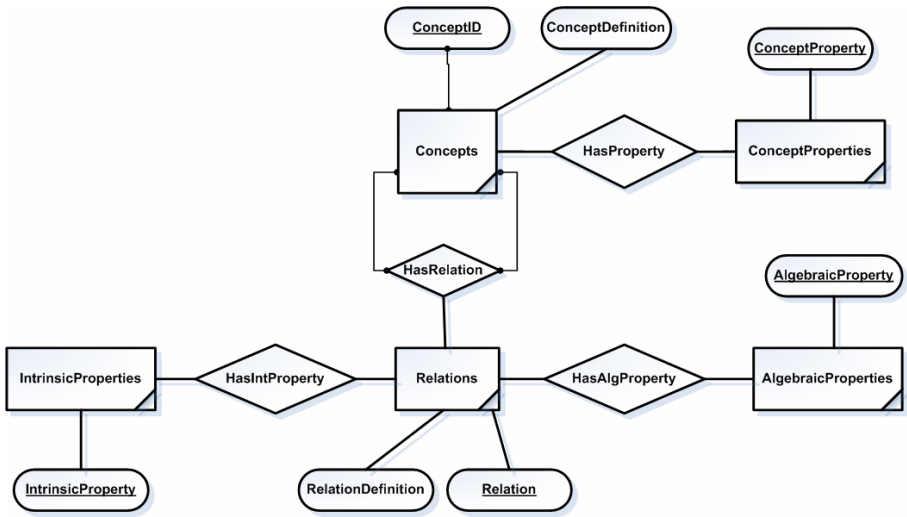


Fig. 2. Conceptual Model for an Ontology-Based LR

The relationship set **HasProperty** is used to assign properties to concepts. The ternary relationship set **HasRelation** is used to represent that two concepts in an ontology can be linked by a given relation. The relationship set **HasAlgProperty** is used to convey that relations could have attached a set of algebraic properties; the same applies for the relationship set **HasIntProperty**, but for intrinsic properties.

7 Conclusions and Future Work

The use of RDB to represent lexical knowledge provides a complete software engineering methodological approach for the design of the database that will contain the LR. However, the approaches that use this technology sometimes only present an E-R schema and forget about the rest of the DB development stages or simply state that they use RDB. This is far from being adequate, as LR to be used by domain specific applications need to be developed in such a way that all the modeling choices are clearly stated and documented.

With that in mind, we have chosen to develop our future terminology database following a sound software engineering methodology. However, the proposed conceptual model of the methodology had some major limitations. In order to overcome them, we modified it based on an analysis of common problems in LR. The new model can now account for any number of ontological relationships (as long as they are binary), and we have incorporated a set of ideas that help designing application-oriented LR where the semantics of relations is clearly stated and the use of relations can be controlled (e.g., the model allows the integration of the OntoClean [15] method for evaluating taxonomies). Moreover, although we have selected RDB

to represent lexical and conceptual knowledge, the model is totally independent of any knowledge representation schema (i.e., databases or knowledge bases).

We still have to go through the logical and physical design stages of the database. However, we have taken a first step towards our final goal, by clearly stating and depicting the structure, scope and limitations of our future LR. Moreover, we have focused on the ontology side of the model; however, the lexical side of our previous model (see [1]) also needs to be upgraded as it is quite limited. Thus, we are considering the integration of the E-R model for the lexical side of an ontology-based LR proposed and described in [4].

A thing that must be clearly understood is that our efforts lean towards the establishment of a software engineering methodology for the design and implementation of ontology-based LR using RDB. However, it is not a methodology aimed at saving time by: a) constructing or extracting a LR from texts using machine learning methods [23, 24] or b) merging different LR into a definitive one [21, 22]. We follow a software engineering approach (where thinking precedes action) by focusing on analysis, design and reuse (as understood by software engineering) aspects. Thus, we apply the principled methods and techniques of software engineering (which guide the development of user-oriented, readable, modular, extensible, and reusable software) to the design and implementation of ontology-based LR.

Finally, a very important aspect in developing a LR is the availability of software tools for its enlargement and modification. However, the majority of the management software tools for LR are just briefly described, by pointing out their features, and although some are extensively described [7, 17], there is no declared software engineering approach for their development [1]. Although not covered in this paper, our methodology takes also into account this important aspect.

References

1. Sáenz, F. and Vaquero, A. 2005. Applying Relational Database Development Methodologies to the Design of Lexical Databases. Database Systems 2005, IADIS Virtual Multi Conference on Computer Science and Information Systems (MCCSIS), ISBN 972-8939-00-0, (2005)
2. Maña, M., Mata, J., Domínguez, J.L, Vaquero, A., Alvarez, F., Gomez, J., Gachet, D., De Buenaga, M. Los proyectos SINAMED e ISIS: Mejoras en el Acceso a la Información Biomédica Mediante la Integración de Generación de Resúmenes, Categorización Automática de Textos y Ontologías. En Actas del XXII Congreso de la Sociedad Española de Procesamiento del Lenguaje (SEPLN), (2006)
3. Bläser, B; Schwall, U and Storrer, A. Reusable Lexical Database Tool for Machine Translation. In Proceedings of the International Conference on Computational Linguistics -- COLING'92, volume II, (1992) pp. 510-516.
4. Moreno A. Diseño e Implementación de un Lexicón Computacional para Lexicografía y Traducción Automática. Estudios de Lingüística Española, vol(9). (2000)
5. Hayashi, L. S. and Hatton, J. Combining UML, XML and Relational Database Technologies - The Best of all Worlds for Robust Linguistic Databases. In Proceedings of the IRCS Workshop on Linguistic Databases. (2001).

6. Wittenburg, P., Broeder, D., Piepenbrock, R., Veer, K. van der. Databases for Linguistic Purposes: a case study of being always too early and too late. In Proceedings of the EMELD Workshop. (2004).
7. Moreno, A. and Pérez, C. Reusing the Mikrokosmos Ontology for Concept-Based Multilingual Terminology Databases. In Proc. of the 2nd International Conference on Language Resources and Evaluation, (2000) pp 1061-1067.
8. Tiedemann, J. MatsLex: A multilingual lexical database for machine translation. In Proc. of the 3rd International Conference on Language Resources and Evaluation, (2002), pp 1909-1912.
9. Lieberman, M. The Use of SNOMED to Enhance Querying of a Clinical Data Warehouse. A thesis presented to the Division of Medical Informatics and Outcomes Research and the Oregon Health & Sciences University School of Medicine in partial fulfillment of the requirements for the degree of Master of Science. (2003)
10. Nirenburg, S., McShane, M. and Beale, S. The Rationale for Building Resources Expressly for NLP. In Proc. of the 4th International Conference on Language Resources and Evaluation, (2004).
11. McShane, M.; Nirenburg, S. and Beale, S. An implemented, integrative approach to ontology-based NLP and interlingua . Working Paper #06-05, Institute for Language and Information Technologies, University of Maryland Baltimore County, (2005)
12. Cimino, J. Desiderata for Controlled Medical Vocabularies in the Twenty-first Century. *Methods of Information in Medicine*, 37(4-5):394-403, (1998)
13. Ide, N., and Veronis, J. Extracting Knowledge Bases from Machine-Readable Dictionaries: Have we wasted our time? In Proc. of the First International Conference on Building and Sharing of Very Large-Scale Knowledge Bases, (1993)
14. Guarino, N. Some Ontological Principles for Designing Upper Level Lexical Resources. A. Rubio et al. (eds.), In Proc. of the First International Conference on Language Resources and Evaluation, (1998) pp 527-534.
15. Welty, C. and Guarino, N. Supporting ontological analysis of taxonomic relationships", *Data and Knowledge Engineering* vol. 39(1), (2001) pp. 51-74.
16. Bodenreider O. Circular Hierarchical Relationships in the UMLS: Etiology, Diagnosis, Treatment, Complications and Prevention. In Proceedings of the AMIA Symposium, (2001)
17. Feliu, J.; Vivaldi, J.; Cabré, M.T. Ontologies: a review. Working Paper, 34. Barcelona: Institut Universitari de Lingüística Aplicada. DL: 23.735-2002 (WP), (2002)
18. Evans, R., and Kilgarriff, A. MRDs, Standards and How to do Lexical Engineering. Proc. of 2nd Language Engineering Convention, (1995) pp. 125–32.
19. Burgun, A. and Bodenreider, O. Aspects of the Taxonomic Relation in the Biomedical Domain. In Proc. of the 2nd International Conference on Formal Ontologies in Information Systems, (2001)
20. Martin, P. Correction and Extension of WordNet 1.7. In Proc. of the 11th International Conference on Conceptual Structures, (2003) pp 160-173.
21. Oltramari, A.; Prevot, L.; Borgo, S. Theoretical and Practical Aspects of Interfacing Ontologies and Lexical Resources. In Proc. of the 2nd Italian SWAP workshop, (2005).
22. Philpot, A., Hovy, E. and Pantel, P. The Omega Ontology. 2005. In IJCNLP Workshop on Ontologies and Lexical Resources, (2005) pp. 59-66.
23. Makagonov, P., Ruiz Figueroa, A., Sboychakov, K. and Gelbukh, A. Learning a Domain Ontology from Hierarchically Structured Texts. In Proc. of Workshop "Learning and Extending Lexical Ontologies by using Machine Learning Methods" at 22nd International Conference on Machine Learning, (2005)

24. Makagonov, P., Ruiz Figueroa, A., Sboychakov, K. and Gelbukh, A. Studying Evolution of a Branch of Knowledge by Constructing and Analyzing Its Ontology. In Christian Kop, Günther Fliedl, Heinrich C. Mayr, Elisabeth Métais (eds.). Natural Language Processing and Information Systems. 11th International Conference on Applications of Natural Language to Information Systems, (2006).
25. Nirenburg, S., McShane, M., Zabudowski, M., Beale, S. and Pfeifer, C. Ontological Semantic text processing in the biomedical domain. Working Paper #03-05, Institute for Language and Information Technologies, University of Maryland Baltimore County, (2005)
26. Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, Mungall CJ, Neuhaus F, Rector A, Rosse C. Relations in Biomedical Ontologies. *Genome Biology*, 6(5), (2005)