Conceptual Design for Domain and Task Specific Ontology-Based Linguistic Resources*

Antonio Vaquero[1], Fernando Sáenz[1], Francisco Alvarez[2], and Manuel de Buenaga[3]

[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. Regardless of the knowledge representation schema chosen to implement a linguistic resource, conceptual design is an important step in its development. However, it is normally put aside by developing efforts as they focus on content, implementation and time-saving issues rather than on the software engineering aspects of the construction of linguistic resources. Based on an analysis of common problems found in linguistic resources, we present a reusable conceptual model which incorporates elements that give ontology developers the possibility to establish formal semantic descriptions for concepts and relations, and thus avoiding the aforementioned common problems. The model represents a step forward in our efforts to define a complete methodology for the design and implementation of ontology-based linguistic resources using relational databases and a sound software engineering approach for knowledge representation.

1 Introduction

Existing linguistic resources (LR) can be used as a source of knowledge by any natural language processing application. However, most of these LR were developed focusing on coverage and implementation issues rather than on questions of design. This approach to LR construction has yield LR with huge quantities of information but poorly structured. A situation that severely limits their reuse and integration (with other LR), and has proven to be a major obstacle to obtain better results.

We claim that design issues are an important part of the construction of a LR, because in order to develop, reuse and integrate diverse available LR, into a common

* 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).

R. Meersman, Z. Tari et al. (Eds.): OTM 2006, LNCS 4275, pp. 855 – 862, 2006.

information system (perhaps distributed), requires compatible software architectures and sound data management from the different databases to be integrated. Hence, under this view, a LR must be carefully designed before any implementation is made, by following a software engineering approach.

With that in mind, we have developed a methodology based on relational databases (RDB) and software engineering principles, for the design and implementation of ontology-based LR [1]. The methodology already proposes a conceptual model (an E-R schema). However, the model has to be modified if it is to be used to create structurally sound LR. In this paper, we present an upgrade of this conceptual model as a refinement of the representational power of our previous model. Nevertheless, we only present here the first stage of the classical RDB design (i.e., conceptual design) and only for the ontology side of the model. Thus, leaving the other two stages (i.e., logical and physical) and the lexical side of the model for future papers.

The rest of the paper is organized as follows. In section 2, we point out the importance of conceptual design in the construction of linguistic RDB, and explain our decision to use an ontological model for knowledge representation. In section 3, some common problems of LR are summarized, and the need to develop application-oriented LR is signaled. In section 4, the methodological gaps of past developing efforts that used RDB are underlined. In section 5, we depict a set of ideas intended to help developers to formally specify and clarify the meaning of concepts and relations in more detail. In section 6, a conceptual model that integrates the aforementioned ideas is introduced and described. Finally, in section 7 some conclusions are outlined.

2 Conceptual Design of LR Using RDB

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. However, as shown in [2, 3, 4] a careful design (i.e., conceptual modeling) can overcome these drawbacks, and let us take advantage of all the benefits of using RDB technology to design and implement linguistic databases [1, 2, 3]. In addition, and following [4, 5, 6, 7], we believe that the computationally proven ontological model with two separated but linked and concurrently developed levels of representation (i.e. the conceptual-semantic level and the lexical-semantic level) is our best choice for linguistic knowledge representation.

3 Some Common Problems in LR

It is relatively easy to create a conceptual model of linguistic knowledge. 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 these errors can be avoided.

Most of the problems of past and present LR have to do with their taxonomic structure. For example, 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 [8]. WordNet and Mikrokosmos have also well-known problems in their taxonomic structure due to the overload of the is-a relation [9, 10]. 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 [9, 10], as well as the multiple inheritance ones. In the biomedical domain, the UMLS has circularities in the structure of its Metathesaurus [11], because of its omnivorous policy for integrating hierarchies from diverse controlled medical vocabularies. Some of the consequences of these flaws, as well as additional ones have been extensively documented in [5, 6, 9, 12, 13, 14, 15, 16] for these and other main LR.

3.1 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 efforts aimed at producing wide-coverage general LR [16, 17], and the ones aimed at (semi)automatically constructing them from texts [18, 19]. However, no amount of broad coverage will help to raise the quality of output, if the coverage is prone to error [6]. 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 [20]. 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 [13] 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 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 phase and then presented the interface(s) of their respective resources. 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 stages. 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 relations, a problem that as seen in section 3 is of main concern.

Therefore, if we are to design an ontology-based LR using RDB, our conceptual model must take also into account the semantic relations issue. Thus, 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 [10] (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. This represents a novelty and an improvement when compared to similar design and implementation efforts as [4] based on ontological semantics and 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 [21]. Moreover, the definitions and algebraic properties will ensure that the corresponding and probably general-purpose relational expressions are used in a uniform way [21]. Tables 1 and 2 (taken from [21]) show a set of relations with their definitions and algebraic properties.

Table 1. Definitions and Examples of Relations

Relations	Definitions	Examples
C is-a C_1	Every *C* at any time is at the same time a C_1	*myelin is-a lipoprotein*
C part-of C_1	Every *C* at any time is part of some C_1 at the same time	*nucleoplasm part-of nucleus*

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 [9, 10] 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 [10] 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 for a LR

In this section, we present a conceptual model (an E/R scheme upgraded from our model in [1]) for the conceptual-semantic level of an ontology-based LR 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 in the introduction, the model will reflect only the ontology part of the LR.

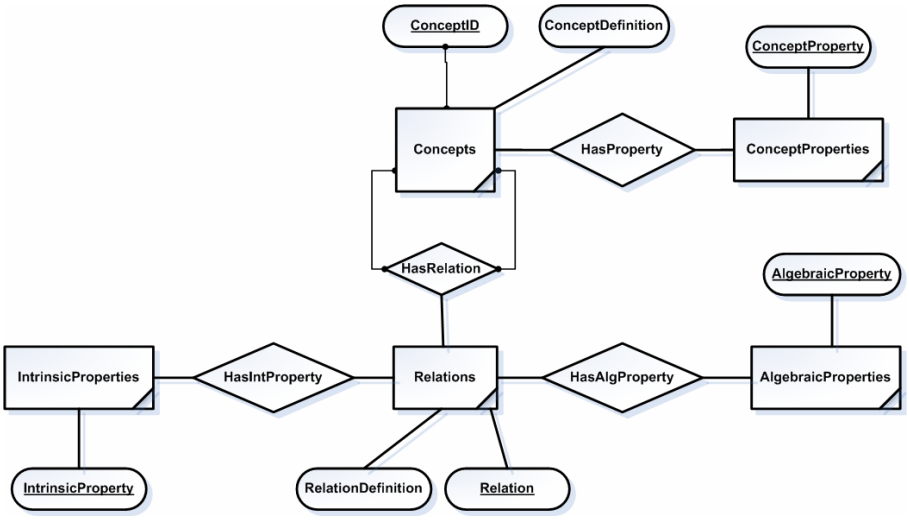

Fig. 1. Conceptual Model for an Ontology-Based LR

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.

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

We have pointed out that an important and normally put aside step in the development of linguistic databases is the conceptual modeling or conceptual design step. With that in mind, we have used a semantic data model (i.e., the E-R model) to create a conceptual model (departing from the one in [1]), which accounts for a set of ideas that could help developers to create domain and task specific ontology-based LR, where the use of semantic relationships can be controlled. 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). In this paper, 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 [2].

Moreover, 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 [18, 19] or b) merging different LR into a definitive one [16, 17]. 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 development of its graphical user interface(s). However, the majority of the management software tools for LR are just briefly described, and although some are extensively described [4, 14], there is no declared software engineering approach for their development [1]. Although not covered in this paper, our methodology encompasses this aspect too.

References

1. Sáenz, F. and Vaquero, A. Applying Relational Database Development Methodologies to the Design of Lexical Databases. Database Systems 2005, IADIS Virtual Multi Conference on Computer Science and Information Systems, (2005)
2. Moreno A. Diseño e Implementación de un Lexicón Computacional para Lexicografía y Traducción Automática. Estudios de Lingüística Española, vol(9), (2000)
3. Hayashi, L. S. and Hatton, J. Combining UML, XML and Relational Database Technologies - The Best of all Worlds for Robust Linguistic Databases. Proceedings of the IRCS Workshop on Linguistic Databases, (2001)
4. 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.

5. 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)
6. 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, (2005)
7. Cimino, J. Desiderata for controlled medical vocabularies in the twenty-first century. Methods of Information in Medicine, 37(4-5):394—403, (1998)
8. 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)
9. 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.
10. Welty, C. and Guarino, N. Supporting ontological analysis of taxonomic relationships", Data and Knowledge Engineering vol. 39(1), (2001) pp 51-74.
11. Bodenreider O. Circular Hierarchical Relationships in the UMLS: Etiology, Diagnosis, Treatment, Complications and Prevention. In Proceedings of the AMIA Symposium, (2001)
12. Bouaud, J., Bachimont, B., Charlet, J. and Zweigenbaum, P. Acquisition and Structuring of an Ontology within Conceptual Graphs. In Proceedings of the ICCS'94 Workshop on Knowledge Acquisition using Conceptual Graph Theory, (1994)
13. Evans, R., and Kilgarriff, A. MRDs, Standards and How to do Lexical Engineering. In Proceedings of the Second Language Engineering Convention, (1995) pp. 125–32.
14. 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)
15. Martin, P. Correction and Extension of WordNet 1.7. In Proc of the 11th International Conference on Conceptual Structures, (2003) pp 160-173.
16. 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)
17. Philpot, A., Hovy, E. and Pantel, P. The Omega Ontology. In IJCNLP Workshop on Ontologies and Lexical Resources, (2005) pp 59-66.
18. 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 the 22nd International Conference on Machine Learning, (2005)
19. 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)
20. Nirenburg, S., McShane, M., Zabludowski, 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)
21. 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), (2006)