

Conceptual Modeling of Ontology-based Linguistic Resources with a Focus on Semantic Relations

¹Francisco Alvarez, ²Antonio Vaquero, ²Fernando Sáenz

¹Universidad Autónoma de Sinaloa

Ángel Flores y Vicente Riva Palacios s/n, 80000, Culiacán, Sinaloa, México

²Instituto de Tecnologías del Conocimiento

Facultad de Informática, C/Prof. José García Santesmases s/n, E-28040, Madrid, Spain

E-mail: fjalvare@fdi.ucm.es, vaquero@sip.ucm.es, fernan@sip.ucm.es

Abstract

Although ontologies and linguistic resources play a key role in applied AI and NLP, they have not been developed in a common and systematic way. The lack of a systematic methodology for their development has led to the production of resources that exhibit common flaws between them, and that, at least when it come to ontologies, negatively impact their results and reusability. In this paper, we introduce a software-engineering methodology for the construction of ontology-based linguistic resources, and present a sound conceptual schema that takes into account several considerations for the construction of software tools that will allow the systematic and controlled construction of ontology-based linguistic resources.

1. Introduction

Nowadays, one of the preferred ways of representing a body of domain knowledge is by means of an ontology. In fact, as Musen reports in (Musen., 2004), ontologies have become so popular that they have eclipsed almost everything else in applied AI including NLP. Consequently, we find ontologies pervasively used in areas such as legal information systems, text mining and information retrieval for bioinformatics and so on.

This new emphasis on ontologies is not surprising. It reflects the important role they play in structuring our collections of propositional knowledge to obtain a world model, over which, applications can “reason” in order to give the user a plausible answer to e.g. a query.

However, until now, ontologies have been developed relying on different foundations, which according to (Hovy, 2005), stem from philosophy, cognitive science, linguistics and AI/computational linguistics, and where computational foundations, under the form of a methodology for ontology building have received little attention.

Indeed, methodological discussion is necessary; if we are to provide the general enterprise of ontology building and relation creation, with rigor, systematicity and verification methods that would turn this effort from a subjective crafting enterprise into a science.

Without this, the user does not know how to choose between various alternative semantic theories and resources, and is forced to rely on unverifiable claims, the ontology builders’ reputation and/or erudition, and subjective preferences.

Although several methodologies (Gómez-Pérez, Fernández-López, Corcho, 2004) have been proposed, they do not offer a systematic way of constructing ontologies, but rather focus on knowledge acquisition

tasks.

In this paper, we present the first stage of a software engineering based methodology for ontology construction, that we have been developing through the years (Sáenz & Vaquero, 2002; Sáenz & Vaquero, 2005), namely the construction of a sound conceptual model which takes into account several aspects that are commonly overlooked in the development of ontologies.

The rest of the paper is organized as follows. In Section 2, we explain why a common methodology for construction is needed, why it must be software engineering based, and state that the first step for ontology building must be the construction of a sound conceptual schema. In Section 3, we expound our first consideration for the construction of the conceptual schema, by justifying why an ontology-lexicon structure is advantageous and necessary.

In Section 4, we give examples of some common problems in linguistic resources and ontologies that must be taken into account for the construction of the conceptual schema. In Section 5, we introduce an intermediate step to produce a conceptual dictionary prior to structuring the ontology with (ontology-specific) semantic relations. In Section 6, we explicate our solution to the problems exposed in section 4.

In Section 7, show how, by considering a small set of relations, bad modelling choices could have been prevented. In section 8, we present our conceptual schema. Finally, in Section 9, some conclusions and future work are outlined.

2. Software Engineering Problem-Solving for Ontology Construction

Until now the common trend in AI and NLP is to develop representation languages, ontology editors, and concrete

(ontological) resources, in a rush to implement and have results as soon as possible.

Nonetheless, better ways of building ontologies must be sought that provide two things: a) a systematic way (i.e. carried on using step-by-step procedures) of building ontologies; and b) a way of ensuring some sort of quality with respect to the intended goal(s) of the ontology by taking into account domain specific aspect as well as those of the problem(s) to solve.

We claim that in order to achieve this, ontology construction must follow a software engineering problem-solving approach, embodied, as we stated before, in the form of a methodology.

Indeed, if we want to avoid ontologies being always criticized as creative inventions of individuals (as linguistic rules used to be), then we must provide a general methodology that allows for their creation and maintenance (Wilks 2002).

However, a software engineering problem-solving methodology for ontology construction must not only borrow the vocabulary of software engineering and adapt it to carry on knowledge acquisition tasks (see the description of the Methontology (Gómez-Pérez, Fernández-López, Corcho, 2004) for an example of this).

It must conceive ontology construction as process of elaboration, where the problem (e.g. building ontologies) is first represented at a high level of abstraction, and as the process progress, the statement of the problem moves from a representation of the essence of the solution toward implementation specific details, and finally to the construction of the ontology itself.

Thus, from a software engineering point of view, ontology construction requires a series of steps, in decreasing abstraction order, that take developers from essence to delivery, and that help them to develop the resource correctly.

However, in order to apply software engineering to the construction of ontology-based linguistic resources, it is mandatory to conceive these resources as information systems which are composed of a database and an application layer core (the rationale of this decision is explained in (Sáenz and Vaquero, 2005)) which allows the user and applications to interact with the conceptual and lexical data.

Otherwise, as stated in (Sáenz and Vaquero, 2005), the software engineering methods would only be applied (as it commonly happens) to the application layer (i.e. normally represented under the form of user interfaces).

Furthermore, if want to have any hopes for a more or less automated incorporation of different ontology-based linguistic resources into a common information system, perhaps distributed, we will require compatible software architectures and sound data management from the different databases to be integrated.

With this in mind we have developed a methodology that has the following steps:

- Represent the conceptual level using the E-R model.

- Apply the relational database design cycle in order to obtain the logical and physical models of the ontology-based lexical resource.
- Use UMLi (Pinheiro & Paton, 2003) to develop the interfaces for creation, management and querying.

We will focus here in the design stage (the construction of the conceptual schema of the resource by using the E-R model), in order to develop a common and sound database structure.

Conceptual modeling is an important milestone in software engineering, information systems development and database development. Developers need to know the conceptual model in order to develop an information system or database.

Unfortunately, the need for conceptual models in software engineering, and information systems development is often overlooked or simply disregarded (Pressman, 1999; Olivé, 2007), and a common practice is to begin by “coding the solution”, and as a result, interface, architectural, and data design just happen.

In our research, we have found that with a few countable exceptions (see (Vaquero, Alvarez and Sáenz, 2008) for a list and discussion) the same problem exists in the development of ontologies and linguistic resources in general, that is, there is no conceptual design (or conceptual modeling) and no software engineering approach in their development.

It can be argued that for ontology construction, methodologies such as the Methontology (Gómez-Pérez, Fernández-López and Corcho, 2004) offer a way to obtain a conceptual model. However, what they present in reality is a lightweight ontology, that later goes through a formalization process.

This same idea is shared by (Welty and Guarino, 2001), when they state that the accepted industrial meaning of ontology makes it synonymous with conceptual model, and that a conceptual model is an actual implementation of an ontology.

Nonetheless, for us, a conceptual model has a whole different meaning.

Following (Olivé, 2007) a conceptual model is a commitment to viewing domains in a particular way (e.g. objects, relationships and concepts). In our particular case, we assume, as (Chen, 1976) does that the “real world” consists (basically) of entities and relationships.

From this conceptual model (i.e. the E-R Model) we will obtain a conceptual schema, which following (Batini, Ceri and Navathe 1994, Connolly and Beg, 1999; Atzeni et al., 1999), is the description of the structure of the database at a high level of abstraction, without considering any implementation details, such as DBMS, application programs, programming languages or any other physical considerations.

Given that in Software Engineering and database development, conceptual schemas are the product of certain considerations with regard to the information

requirements of an application; in the following 4 sections we will make some considerations that will: a) determine the structure of our conceptual schema; and b) solve certain problematics common to most linguistic resources and ontologies in general.

3. An Ontology-Lexicon Structure

Since we aim for the production of ontologies to be used by NLP programs, we can advance here our first consideration: the schema will represent an ontology with a Mikrokosmos-like structure, that is, a single ontology with a language-specific lexicon mapped to it (Nirenburg, McShane and Beale, 2004).

On one hand, a lexicon is needed because for the majority of problems, we need more or less profound linguistic expertise to at least divide the problem into manageable parts.

For example, as (Hajic and Hajicova, 2007) explain, in machine translation today we need to lemmatize and identify (at least) “phrases”—indeed, very linguistic phenomena. For predicate extraction, we need to know at least some aspects of the syntactic structure of the sentence. And for word sense disambiguation, a lexicon must contain, in the word entries, specific information to help disambiguate a term (see (Nirenburg & Levin, 1992; Nirenburg, Mcshane and Beale, 2004) for examples of this).

On the other hand, and following (Mahesh & Nirenburg, 1995) we need an ontology for several reasons:

- To have a main repository for representing selectional preferences between concepts. This knowledge is invaluable for resolving ambiguities by means of e.g. a constraint satisfaction process.
- To enable inferences to be made from the input text using knowledge contained in the concepts, in order to resolve ambiguities, fill gaps in text meaning, and support metonymy and metaphor processing.
- To form a substrate upon which meanings in any language are grounded and constructed in the lexicon, by guaranteeing that every symbol used in representing lexical semantics is defined as a concept, is well-formed, and has known relations to all other concepts.

It can be argued that this also can be done without following an ontological semantics approach, e.g. using the generative lexicon approach of (Pustejovski, 1991) or the SIMPLE framework (Lenci et al., 2000).

However, as (Nirenburg, Raskin & Onyshkevych, 1995) point out, although these approaches avoid the concept of ontology in their theoretical frameworks, they introduce elements of metalinguistic apparatus that play the same role as the ontology, or contain what would be more efficiently recorded in a single sufficient ontology, in order to make up for their sparsity of information in the

semantic substrate (McShane et al., 2004).

An issue that arises with an ontology-lexicon structure is the separation between ontology and lexicon. Nonetheless, we will not discuss it here as we are interested merely in the structure and advantages of such separation, and not in “a posteriori” knowledge representation issues.

However, as useful and necessary as an ontology can be, their construction must be done avoiding certain pitfalls. In the next section we will present a list of these issues that will undoubtedly shape our conceptual schema.

4. Some Common Problems in Linguistic Resources

Existing linguistic resources (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 schema that leads to a linguistic resource where some of these errors can be avoided.

We have found that most of the problems of past and present linguistic resources 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 (Ide and Veronis, 1993).

In WordNet, a widely used linguistic resource (and considered by some as an ontology), the “is-a” and “part-of” relations between synsets (WordNet’s representation of concepts) are not used in a consistent way.

For example, Burgun and Bondenreinder in (Burgun and Bondenreider, 2001) report that according to the taxonomic relations linking the hypernyms of “fever” in WordNet 1.6, “fever” ends up being categorized as a “psychological feature”.

Furthermore, as (Hirst, 2004) tell us, the is-a and hypernym relations are used interchangeably, in spite that although they are close in meaning, they are not the same.

In the biomedical domain, the UMLS (Unified Medical Language System) has circularities in the structure of its Metathesaurus (Bodenreider, 2001), and relations like “Conceptually-Related-To” that cannot be adequately interpreted (Montero, 2003).

In SNOMED-RT (a clinical terminology developed by the College of American Pathologists), we find an example of a mix of close-related relations that leads to an improperly structured taxonomy (Ceusters, Smith and Flanagan, 2003). The concept “testis” subsumes (correctly) the concepts “left testis”, “right testis” and “undescended testis”, but also “both testes”. We could accept “both testes” as being part-of another concept (e.g., “Testes”) denoting the mereological sum of the left and

right testes, but hardly as being a “Testis”.

Hence, besides producing resources with no methodology, what we have been producing are resources that have very general or imprecise relations that cannot be adequately interpreted, resources whose relations are subject to multiple interpretations, resources where the semantics of the relations are not fine-grained enough as to allow to differentiate between two relations that are close in meaning but are not the same, and improperly structured taxonomies.

Consequently, our conceptual schema must represent, at an abstract level, the solution to this particular set of problems, so as to allow us to implement a solution to counter the subjectivity with which semantic relations in these resources have been used, and to properly structure an ontology.

In the next two sections, we will describe a series of ideas that can help us solve these problems, and that will determine the final structure of our conceptual schema.

5. From Corpus to a Conceptual Dictionary

The first step in building an ontology for an specific application, is to find the set of relevant concepts representing an unstructured version of a domain model.

Several strategies can be followed to do this. However, it has been proved (Gómez-Gauchía, Díaz-Agudo, González-Calero, 2004; López Rodríguez, Tercedor Sánchez and Faber Benítez, 2006) that one of the best methods is to follow a middle-out strategy.

This strategy has as a goal, the compilation of domain and problem-related documents to form a corpus that can be used as a source of knowledge.

Once this is done, an statistical analysis can be performed on them, as well as other document processing techniques such as document parsing with XML (Gómez-Gauchía, Díaz-Agudo, González-Calero, 2004), in order to obtain a set of relevant concepts. Once this is done, the usual step is to immediately begin the construction of the ontology or the lightweight one mentioned before.

Nonetheless, for ontologies with the same structure as ours (i.e. an ontology-lexicon structure), we propose an intermediate step that will take developers, with the aid of an special tool, not from corpus to the ontology, but from corpus to a conceptual dictionary or primitive lexical-conceptual structure (PLCS) devoid of any ontology-specific relations (e.g. is-a, part-of, etc.).

The goal of is to capture and structure, through an iterative process, all the informal semantics of a domain, and to produce a representation as complete as possible of it.

In this PLCS, each concept will have an intensional definition given in natural language. Moreover, each intensional definition will contain a collection of user-stated keywords (i.e. they are part of the intensional definition) that will help the experts (i.e., by means of the authoring tool) to relate a concept to other concepts, and terms.

Over these intensional definitions we impose two

conditions: a) they must be as concise as possible and b) they must be formulated by experts. Figure 1 illustrates the approach.

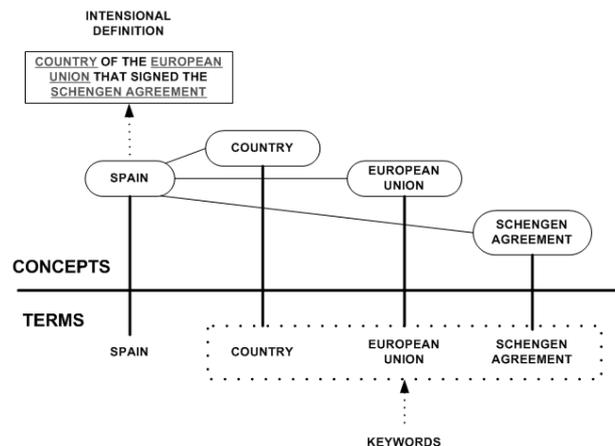


Figure 1: An Example of a PLCS

In Figure 1, the concept “Spain” has a definition with three keywords (i.e., “Country”, “European union” and “Schengen agreement”). These keywords are used to associate “Spain” to the “Country”, “European Union” and “Schengen agreement” concepts. Here, it is important to underline that the authoring tool will ensure the completeness of the PLCS, i.e. if in an intensional definition, the expert selects a keyword that does not exist, then, such a keyword must be created and included as part of the set of terms of the PLCS.

Furthermore, if the situation arises where a given term is polysemous then, the tools must bring forward this fact, and force the expert to make a choice between the concepts that the term denotes.

Finally, once the set of concepts and terms has been considered representative enough, developers can begin the construction of the ontology, by using an specific set of relations.

However, determining for one application which relations are these and how to obtain them is a knowledge acquisition task, and thus is out of the scope of these paper.

Instead, since what we are aiming for is the construction of a conceptual schema, we will focus next on a series of ideas to provide relations with specific semantics and prevent improper modeling choices in their usage while structuring the ontology.

6. Controlling Subjectivity in the Usage of Semantic Relations

Of the many difficulties in building a useful knowledge-based system (KBS), control and verification are one of the greatest challenges, and as we automate even more and more tasks the need for them becomes even more crucial. In fact, as (Hicks, 2003) tell us, the importance of verification in KBS cannot be overstated, specially if we want these systems to perform correctly in the long term.

The mistakes shown in section 3 clearly point out, that

in terms of ontology construction, semantic relations must be subject to control and verification, and that even seemingly harmless mnemonics e.g. “is-a” must be handled gingerly during the ontology construction process in order to minimize the effect of incorrect modelling choices and coherently structure an ontology.

Two approaches exist that try to deal with this problem (Welty & Guarino, 20001; Bachimont, Isaac & Troncy, 2002). Nonetheless, for the reasons we adduce in (Alvarez, Vaquero, Sáenz & Buenaga, 2007), these methods cannot fully work or not work at all, as they leave the semantics of the problem, the task and the application out of the equation, focus on concepts, not relations, to structure an ontology, and just take into account a handful of relations.

In spite of this, we share the idea that a set of conditions must be established to test whether two concepts can be linked by a given relation.

However, we differ in the way of enforcing such control and verification. In our opinion, this must be enforced through a set of properties defining the semantics of relations, and for all the relations to be used in the construction of the ontology, not just for a few basic ones.

Furthermore, the semantics must be in part determined by the domain, the problem and the task at hand (see (Alvarez, Vaquero, Sáenz and Buenaga, 2007) for an example of this), and not by an universal view of them.

Consequently, we divide these properties into algebraic properties denoting those properties needed to make valid syllogisms (e.g., transitivity, asymmetry, reflexivity, etc.), and intrinsic properties (i.e., those properties representing facts that are hard to formalize) encompassing domain-dependent, problem-dependent and task-dependent properties. Figure 2 illustrates this idea.

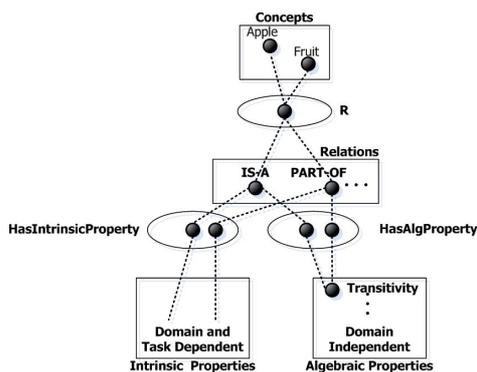


Figure 2: Algebraic and Intrinsic Properties

In addition, over these two sets of properties we apply two different criteria for their employment: a) algebraic properties since they are domain independent are invariable through all the graph; b) intrinsic properties on the other hand will depend on the depth level of the graph where the semantic relation is used.

With this idea, the experts must move from the PLCS described in the previous section to an ontology with the aid of an authoring tool, and relation by relation.

For instance, let us suppose that we are trying to develop an ontology to support a legal information system

to “assess claims for immigration” in “Schengen signatory countries”.

For example purposes, we will assume that only the is-a relation is needed to model the problem domain with the following properties: a) asymmetry, reflexivity and transitivity as algebraic properties; b) has borders, has constitution and has central government as domain-dependent intrinsic properties; and c) signed treaty of adhesion as task-dependent intrinsic property; and d) signed Schengen agreement as problem-dependent intrinsic property.

Based on these properties, the system could ask meaningful questions to the ontology developer in order to prevent inappropriate modelling choices

Hence, if a developer would want to link the concept “Spain” to the concept “Schengen agreement” using the “is-a” relation, the system would not allow such an operation, if “Spain” would not comply with all the algebraic properties and the intrinsic properties at that specific depth level. Figure 3 depicts this idea.

The ideas we presented in this sections, along with the ones from section 4 will form the basis for our conceptual model. However, before presenting it, we will introduce a group of ideas, that although will not alter the final conceptual model, will help developers to better structure an ontology.

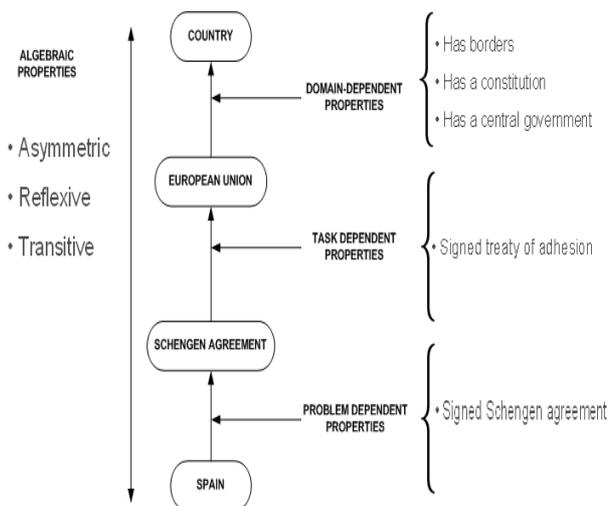


Figure 3: Algebraic and Intrinsic Properties in Action

7. Some Basic Relations for Ontology Structuring

Although we cannot determine in advance the full set of relations for a given application, what we can do is anticipate, the common relations for any application whatsoever: “member-of” and “is-a”.

The first relation that must be used is “member-of”, as we always begin constructing a domain model by identifying or selecting a set of objects of interest. Then the “is-a” relation appears by discovering the common properties to all the members belonging to a given extension set. Figure 4 exemplifies this idea.

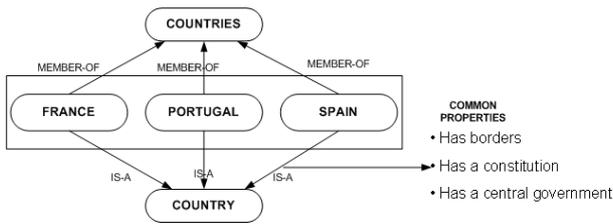


Figure 4: Using the “member-of” and “is-a” relations to build a domain model

To prove the usefulness of the common relations we mentioned above, let us remodel (without algebraic and intrinsic properties as it is a trivial example) the “both testes is-a testes” mistake” of (Ceusters, Smith and Flanagan, 2003) represented in Figure 5.

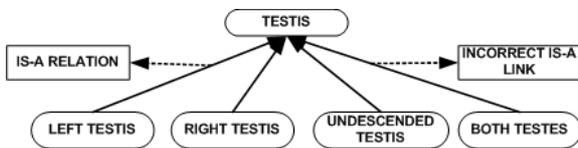


Figure 5: The “both testes is-a testis” mistake

We first have that the “undescended testis”, “left testis” and “right testis” concepts represent an extension concept of which they are members, denoted by the plural “testes”. Figure 6 illustrates this fact.

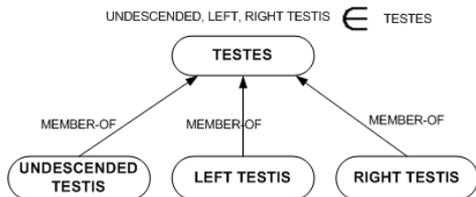


Figure 6: Structuring begins with the member-of relation

Then, the concept “testis” is created as a result of the abstraction of the common properties of all the “testes” and the “is-a” relation naturally appears. Figure 7 depicts the process.

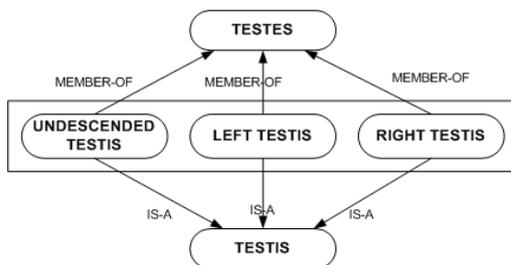


Figure 7: Using the “member-of” and “is-a” relations to build a domain model

After that, it becomes clear that “both testes” is a special subset of the extension set “testes”, and that the next relation that naturally appears is “component-of”, as shown in Figure 8.

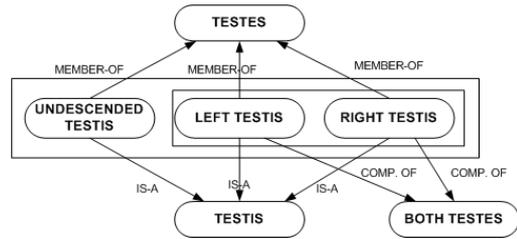


Figure 8: Using the “component-of” relation to build a domain model

The rationale behind Figures 6, 7 and 8 is that we must exhaust the set-theoretic relations before moving to other relations that are not set-theoretic like the “adjacent to” that we described in (Alvarez, Vaquero, Sáenz & Buenaga, 2007).

It also implies: a) that a single relation (i.e., “is-a” or subsumption) will always mislead us; b) that member-of and “is-a” should always be contemplated; and c) that tackling the complexity of domains must be done relation by relation.

8. Conceptual Schema for a Monolingual Ontology-based Linguistic Resource

The embodiment of the ideas presented in the previous sections is represented by the E-R model of Figure 9. In this model relations are treated as concepts via the specialization-generalization construct. The entity set “Relations” represents all the relations in the ontology. The entity sets “AlgebraicProperties” and “IntrinsicProperties” denote the group of algebraic and intrinsic properties that a given relation can have respectively.

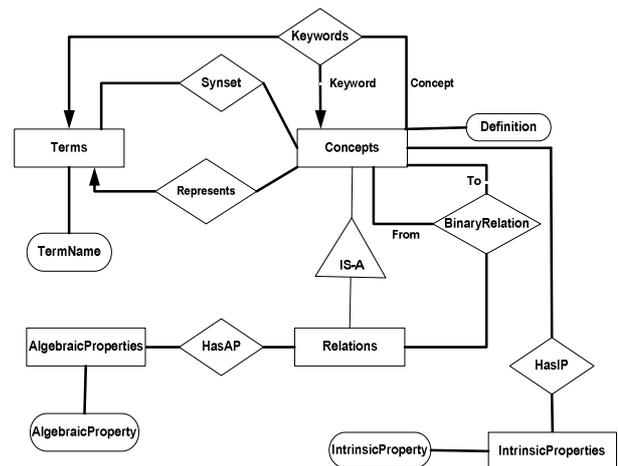


Figure 9: Conceptual Model of a Monolingual Ontology-based Linguistic Resource

The relationship set “BinaryRelation” is used to state that a concept is related to another concept via a binary semantic relation. Its cardinality is many to many because a concept can be linked to many concepts, and a concept can be related to other concepts through several relations.

The relationship sets “HasAP” and “HasIP” represent the idea that a relation can have a set of algebraic

properties. Both sets are many to many because a given relationship can have several algebraic or intrinsic properties and the same algebraic or intrinsic property can be present in different relations.

On the lexical-conceptual side, we have that the entity set Terms represents all the terms composing the linguistic resource, and each term is denoted by a name represented by the attribute "TermName". The entity set Concepts denotes the concept to which a given term is mapped and the attribute Definition characterizes the intensional definition of a given concept.

The relationship set "Synset" represents a set of terms mapped to a given concept. "Synset" has a many to many cardinality in order to denote synonymy and polysemy: a concept may be denoted by several terms and a term may embody several concepts.

The relationship set "Represents" is used to state that for a given set of synonyms, there is a term which is representative of the concept denoted by such a set. "Represents" has a one to one cardinality because we assume that only one term from the set of synonyms can be the representative one, and that it is unlikely that the same term may be representative of a different set of synonyms.

The relationship set "Keywords" asserts that a concept can be related to another concept by means of a single term (i.e., one of the keywords in the definition) and that this single term denotes a concept. "Keywords" has a many to many cardinality in its recursive side because a concept can be related (i.e., through its keywords) to one or more concepts, and is one to many between "Terms" and "Concepts" because each of the many concepts is denoted by a single given term.

9. Conclusions and Future Work

In this paper, we have presented part of a software engineering approach for ontology development, where "thinking precedes action", in order to obtain a conceptual schema, for an ontology-based linguistic resource, that will be represented in a relational database.

The goal of the conceptual schema is not only to represent the information requirements of a given application in terms of lexical and conceptual knowledge, as it is done in any of the linguistic resources presented in (Vaquero, Alvarez & Sáenz, 2008), but also to provide:

- An structured and systematic approach to represent and organize all the conceptual and lexical items needed for an specific application
- The essentials for the implementation of mechanisms for control and verification of semantic relations in ontology construction.

Consequently, we have proposed a different ontology development approach divided in 2 levels.

In the first, we move from a corpus to a conceptual dictionary or PLCS where the informal semantic of the domain is represented and structured but devoid of ontology-specific relations.

In the second, the ontology is structured in a relation by relation basis, beginning from a small group of set theoretic relations, and by using the algebraic and intrinsic properties to prevent the experts from making inappropriate modeling choices.

These ideas must be included in the ontology development tools, much as it is done in Intelligence Augmentation Systems and Decision Support Systems (Paraense, Gudwin & Gonçalves, 2007), in order to have a computational system performing decision making, in terms of the usage of relations, based on the cooperation provided by an ongoing dialogue between a human user and a computer system.

This cooperation results in the appropriate structuring of the ontology, by means of a computational processing power applied to specific points in the human thought process which suffer from some sort of flaw or inefficiency.

Nonetheless, relational database by themselves only provide mechanism to express a handful of constraints: mainly referential integrity and cardinality constraints.

Consequently, we are currently studying the use of additional technology that will help us preserve the structure and format of the resource without moving to other file-related approaches (e.g., XML), and to counter the lack of expressivity of relational databases.

The options we are considering are the use of DataLog (García-Molina, Ullman & Widom, 2002), Answer Set Programming (Lifschitz, 2002), and the Maude System (Clavel et al., 2003).

10. References

- Álvarez, F., Vaquero, A., Sáenz, F., & Buenaga, M. (2007). Neglecting Semantic Relations: Consequences and Proposals. In Proceedings of the IADIS International Conference on Intelligent Systems and Agents (ISA 2007), part of the IADIS Multiconference on Computer Science and Information Systems (MCCSIS 2007), pp. 99-108, IADIS Press.
- Atzeni, P. C7eri, S., Paraboschi, S., & Torlone, R. (1999). Database Systems: Concepts, Languages and Architectures. McGraw-Hill International.
- Bachimont, B., Isaac, A., Troncy, R. (2002). Semantic Commitment for Designing Ontologies: A Proposal. Lecture Notes in Computer Science 2473, pp. 114-119.
- Batini, C., Ceri, S. & Navathe, S. (1994). Conceptual Database Design: an Entity-Relationship Approach. Addison-Wesley.
- Bodenreider, O. (2001). Circular Hierarchical Relationships in the UMLS: Etiology, Diagnosis, Treatment, Complications and Prevention. In Proceedings of the AMIA Symposium.
- Burgun, A. and Bodenreider, O. (2001). Aspects of the Taxonomic Relation in the Biomedical Domain. In Proc. of the 2nd International Conference on Formal Ontologies in Information Systems.
- Ceusters, W., Smith, B., & Flanagan, J. (2003). Ontology and Medical Terminology: Why Description Logics Are Not Enough. In Proceedings of the Conference: Towards an Electronic Patient Record (TEPR 2003).

- Chen, P. (1976). The Entity-Relationship Model-Toward a Unified View of Data. *ACM Transactions on Database Systems*, Vol. 1, issue 1.
- Clavel, M., Durán, F., Eker, S. Lincoln, P. Martí-Oliet, N., Meseguer, J. & Talcott, C. (2003). The Maude 2.0 System. In *Proceedings of Rewriting Techniques and Applications. Lecture Notes in Computer Science 2706*, Springer-Verlag, pp. 76-87.
- Connolly, T. & Beg, C. (1999). *Database Systems: A Practical Approach to Design, Implementation and Management*, 2nd Edition. Addison.Wesley.
- García-Molina, H., Ullman, J. & Widom, J. (2002). *Database Systems: The Complete Book*. Prentice Hall.
- Gómez-Gauchía, H., Díaz-Agudo, B., González-Calero, P. (2004). Towards a Pragmatic Methodology to Build Lightweight Ontologies: a Case Study. In *Proc. of the IADIS International Conference on Applied Computing*, IADIS Press.
- Gómez-Pérez, A., Fernández-López, M., & Corcho, O. (2004). *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag London.
- Hajič, J. & Hajičová, E. (2007). Some of Our Best Friends Are Statisticians. In *Proceeding of the 10th International Conference (TSD 2007)*. Lecture Notes in Computer Science 4629 Springer-Verlag.
- Hicks, R.C. (2003). Knowledge-Based Management System-Tools for Creating Verified Intelligent Systems. *Knowledge Based Systems* 16, pp. 165-171.
- Hirst G. (2004). Ontology and the Lexicon. In Staab and Studer (eds), *Handbook on Ontologies*, Springer-Verlag, pp. 209-230.
- Hovy, E. (2005). Methodologies for the Reliable Construction of Ontological Knowledge. In *Proceedings of the 13th Annual Conference on Conceptual Structure (ICCS 2005)*, Lecture Notes in Artificial Intelligence, Vol. 3596, pp. 91-106.
- Ide, N., and Veronis, J. (1993). 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*.
- Lenci, A., Bel, N. Busa, F., Calzolari, N. Gola, E. Monachini, M, Ogonowski, A, Peters, I. Peters, W, Ruimy, N., Villegas, M, Zampolli, A. (2000). SIMPLE: A General Framework for the Development of Multilingual Lexicons. *LREC 2000*.
- Lifschitz, V. (2002). Answer set programming and plan generation. *Artificial Intelligence*, Vol. 138, pp. 39-54.
- López, C., Tercerdor, M., Faber, P. (2006): *Gestión Terminologica Basada en Conocimiento y Generación de Recursos de Información sobre el Cáncer: el Proyecto OncoTerm*. *Revista de Salud* 2(8).
- Mahesh, K. & Nirenburg, S. (1995). A Situated Ontology for Practical NLP. In *Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing. International Joint Conference on Artificial Intelligence (IJCAI-95)*.
- McShane, M., Zabłudowski, M., Nirenburg, S. & Beale, S. (2004). OntoSem and SIMPLE: Two multi-lingual world views. In *Proceedings of the ACL-2004 Workshop on Text Meaning and Interpretation*.
- Montero, S. (2003). Estructuración conceptual y formalización terminográfica de frasesmas en el subdominio de la oncología. *Estudios de Lingüística Española (ELiEs)*, Vol. 19.
- Musen, M. (2004). Ontologies-Necessary-Indeed Essential-but Not Sufficient. In Brewster, C. and O'Hara, K. *Knowledge Representation with Ontologies: The Present and Future*. *IEEE Intelligent Systems* 19(1): 77-79.
- Nirenburg, S. & Levin, L. (1992). Syntax-driven and ontology-driven lexical semantics. In Pustejovsky J. & Bergler, S. (Eds.), *Lexical Semantics and Knowledge Representation. Lecture Notes in Computer Science Vol. 627*, Springer Verlag, Berlin.
- Nirenburg, S., V. Raskin & B. Onyshkevych (1995).- *Apologiae Ontologiae*. In *Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)*.
- Nirenburg, S., McShane, M. and Beale, S. (2004). The Rationale for Building Resources Expressly for NLP. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC04)*.
- Olivé, A. (2007). *Conceptual Modeling of Information Systems*. Springer-Verlag Berlin Heidelberg.
- Pinheiro, P. & Paton, N. (2003). User Interface Modeling in UMLi. *IEEE Software*, Vol.20 No. 4, IEEE Computer Society, pp. 62-69.
- Pressman, R. (1999). *Software Engineering*. In Merlin Dorfman and Richard H. Thayer (Eds.) *Software Engineering*. Wiley-IEEE Computer Society Press.
- Pustejovsky, J. (1991). The Generative Lexicon. *Computational Linguistics* 17:4.
- Sáenz, F. & Vaquero, A. (2002). Towards a Development Methodology for Managing Linguistic Knowledge Bases. In *Research and development in intelligent systems XIX*. Springer, Cambridge (United Kingdom).
- Sáenz, F. & Vaquero, A. (2005). Knowledge Representation Issues and Implementation of Lexical Databases. *Second International workshop on UNL, other interlinguas and their applications, CICLing-2005, Research on Computing Science*, Vol. 12, pp. 430-442.
- Vaquero, A., Alvarez, F. & Sáenz, F. (2008). Representing Computational Dictionaries in Relational Databases. In Gouws, R.H., U. Heid, W. Schweickard & H.E. Wiegand (eds). 2009. *Dictionaries. An international encyclopedia of lexicography. Supplementary volume: Recent developments with special focus on computational lexicography: Chapter XVI: Models for the Representation of Dictionaries: The Form Aspect*. Mouton de Gruyter Publishers.
- Welty, C. & Guarino, N. (2001). Supporting ontological analysis of taxonomic relationships. *Data and Knowledge Engineering* vol. 39(1), Elsevier.
- Wilks, Y. (2002). Ontotherapy: or How to Stop Worrying about What There is. Invited presentation, *Ontolex 2002, Workshop on Ontologies and Lexical Knowledge Bases (OntoLex)*.