

Semantic Relations: Modelling Issues, Proposals and Possible Applications

Francisco Alvarez¹, Antonio Vaquero², Fernando Sáenz², Manuel de Buenaga³, José María Gómez³

¹Universidad Autónoma de Sinaloa, Ángel Flores y Vicente Riva Palacio, s/n, C.P. 80000, Culiacán, Sinaloa, México
fjalvare@fdi.ucm.es

²Universidad Complutense de Madrid, Facultad de Informática, Departamento de Ingeniería del Software e Inteligencia Artificial
C/ Prof. José García Santesmases, s/n, E-28040, Madrid, Spain
{vaquero,fernan}@sip.ucm.es

³Universidad Europea de Madrid, Departamento de Sistemas Informáticos, E-28670, Villaviciosa de Odón. Madrid, Spain
{buenaga,jmgomez}@uem.es

Abstract

Semantic relations are an important element in the construction of ontology-based linguistic resources and models of problem domains. Nevertheless, they remain under-specified. This is a pervasive problem in both Software Engineering and Artificial Intelligence. Thus, we find semantic links that can have multiple interpretations, abstractions that are not enough to represent the relation richness of problem domains, and even poorly structured taxonomies. However, if provided with precise semantics, some of these problems can be avoided, and meaningful operations can be performed on them that can be an aid in the ontology construction process. In this paper we present some insightful issues about the representation of relations. Moreover, the initiatives aiming to provide relations with clear semantics are explained and the inclusion of their core ideas as part of a methodology for the development of ontology-based linguistic resources is proposed.

Introduction

One of the most overlooked aspects in the design and development of ontology-based linguistic resources (OBLR) and ontologies in general are semantic relations. Although they hold together the entities that represent a domain and shape its structure, semantic relations have not been given the attention they deserve. The focus has been on concepts, their properties and the operations that can be performed on them, rather than on the semantics of relations and the possible operations that could be performed on them.

Several problems arise from this overlook that severely compromise the reusability of OBLR. Hence, in order to avoid some of these problems, the semantic relations (whether taxonomic or not) used to link concepts in an OBLR must be provided with fine-grained semantics.

Providing relations with precise semantics is an important issue in the development of problem-solving OBLR. Tackling a precise task in-side a given domain requires precise semantics, not only at the concept level but also at the relation level. This is opposite to the current trend in the development of linguistic resources (LR) where the

focus is on coverage and time saving issues, rather than on semantic cleanness and application usefulness.

As part of the SINAMED and ISIS projects (Maña et al., 2006) for the integration of ontologies for summarization and categorization in the biomedical domain, we have pointed out: a) the need to provide the relations used in OBLR with definite semantics (Vaquero et al., 2006b) and b) to develop OBLR following a software engineering approach (Vaquero et al. 2006c). Nonetheless, although we have proposed a set of ideas for the representation of relations and their semantics, the results of our research stem mainly from an analysis of “task-neutral” LR (Vaquero et al., 2007).

In this paper, we present an analysis of the state-of-the-art in the representation of semantic relations in Software Engineering (SE) and Artificial Intelligence (AI), including the available methodologies to structure taxonomies. Our goal is to gain a better insight of this normally put off topic, underline its importance and determine if the available technology is well-suited to provide semantic relations with the level of granularity that we claim problem-solving OBLR need. Furthermore, we aim at covering some of the blanks left in our current research regarding the representation of relations and their semantics, as well as its possible application(s) to the construction of OBLR.

The rest of the paper is organized as follows. In section 2, the limitations of SE techniques in the representation of relations are described. In section 3, the overstress of concepts in detriment of relation representation in AI and taxonomy structuring methodologies is shown. In section 4, the current initiatives to provide relations with explicit semantics are presented. In Section 5, control and verification is introduced as a meaningful operation based on the intrinsic semantics of relations. In section 6, a methodology for the design and development of OBLR is proposed. Finally, in section 7, some conclusions and future work are outlined.

Semantic Relations in Software Engineering

Semantic relations (relationships in the SE vocabulary) are a key component of vital design artefacts such as Entity-Relationship (E-R) models and object-class diagrams. However, they only capture a limited set of relationships, leaving much of the domain's relationship structure out of the design (Yoo et al., 2004).

In the E-R model (Chen, 1976) the model of the problem is represented by identifying its entities, properties and relationships. In this model, relationships are classified among entities as binary, n-ary, or recursive. Nonetheless, although they are depicted on the E-R diagram, the amount of information they convey is rather limited, that is, the model itself only provides minimal information describing the relationships (i.e. mainly the cardinality among entities).

In object-class diagrams (OCD) (Booch, 1986; Rumbaugh, 1994), objects are organized by their similarities into classes that describe a set of objects having the same attributes and behavior patterns. In these models, relationships among classes can be classified under three basic categories denoting: a) generalization-specialization; b) whole-part/aggregation and c) an association among otherwise unrelated classes. In OCD, generalization and aggregation types of relationships are strongly defined among classes. Nevertheless, all other types of relationships that exist in the problem domain are lumped into the association category and depicted by a name connecting the classes. These names only indicate that a dependency exists but do not explicitly indicate how. Thus, association relationships are identified more implicitly than explicitly. Furthermore, the distinction between aggregation and association is often a matter of state rather than a difference on semantics (Steimann et al, 2003).

In the UML (Booch et al., 1999) relationships suffer from the same underspecification. The UML categorizes relationships under four categories: association (denoting a meronymic relationship), dependency, generalization-specialization and realization. As with OCD relationship classification, generalization and aggregation types of relationships are strongly defined among classes. However, for dependency and realization, only a label is used to indicate that these relationships exist. Thus, as in OCD, they are identified more implicitly than explicitly.

Although we have described just a few semantic data models here, several others suffer from the same problem (Catania, 2004): namely minimal information describing relationships. As can be seen, there is a put-off of relationships in favor of object or entity representation. However, as it will be shown next, this is not inherent to SE; it is also a pervasive phenomenon in AI.

Semantic Relations in Artificial Intelligence

In AI, we find that knowledge representation formalisms (e.g. DAML, OWL, etc.) are intended to describe the terminology of a domain in terms of classes/concepts

describing sets of individuals and properties/roles relating these (Kashyap & Borgida, 2003). However, although in the above formalisms is possible to make statements about a set of concepts, such as to declaratively specify that two classes are disjoint; analogous declarative statements are not possible for relations. This also comprises the assignment of properties, while concepts are assigned with as many properties as needed, the same level of precision cannot be applied to semantic relations.

Furthermore, the modeling effort is done through the construction of subsumption hierarchies among the defined classes and properties, that is, taxonomic reasoning is restricted to a generalization-based one. This is partially caused by the lack of conclusive mechanisms to reason along other types of relations (Schulz et al., 1999), and because, generalization was initially regarded as the primary mechanism for mastering the complexity of domains. Nevertheless, its use for simplification through the omission of detail comes at a high cost (Steimann et al., 2003).

Consequently, there is a subsumption overload that has led to the misuse and confusion of semantic relations due to the lack of analysis to: a) distinguish between the different relations to be used in the representation of a domain and b) precise the semantics of these relations. For instance, without such an analysis what we have are resources that have very general or under-specified relations that cannot be adequately interpreted (Nirenburg et al., 2005), resources whose relations are subject to multiple interpretations (Kashyap & Borgida, 2003), resources where the semantics of the relations are not enough fine-grained as to allow to differentiate between two relations that are close in meaning but are not the same (e.g. the is-a and hypernym relations (Hirst, 2004)), and improperly structured taxonomies (Welty & Guarino, 2001; Bachimont et al., 2002).

This last topic has received most of the attention in AI. However, as it will be seen next, the available approaches for taxonomy structuring all focus more on concepts and their properties than on relations and their semantics.

Taxonomy Structuring Methodologies

Although the structure problems of taxonomies and hierarchies represent a serious obstacle in their development, there are only two proposals that actually deal with such problems: The OntoClean (Welty & Guarino, 2001) and Archonte (Bachimont et al., 2002) methodologies. These methodologies are similar but at the same time, as it will be seen below, follow opposite approaches to attain the same goal.

OntoClean is grounded on the philosophical ideas of essentialism (Barrett, 2001), that state that for any specific kind of entity (e.g. a tiger), it is theoretically possible to specify a finite list of properties (e.g. the rigidity, identity and unity metaproperties of OntoClean) all of which any entity must have to belong to a specific group or natural kind (e.g. see the table of properties and the taxonomy of kinds in (Welty & Guarino, 2001)). It is also influenced by

the ideas of psychological essentialism (Medin & Ortony, 1989) that enunciate that the world is divided into essences from which preset associated properties can be inferred (e.g. the metaproperties mentioned above), and that these properties play a key role in our everyday reasoning and categorization tasks by backing-up our inferences about kind membership. Seen this way OntoClean can be understood as a reasoning heuristic and inference system that establishes that the compatibility between the metaproperties of concepts determine if a concept can subsume another and vice versa. Nevertheless, a global theory of reference and categorization, independent of any domain and task, like the one OntoClean provides is not possible. Recent work in cultural psychology has shown systematic cognitive differences between East Asians and Westerners, and some work indicates that this extends to intuitions about philosophical cases (Machery et al., 2004).

Archonte relies on the work of (Rastier et al., 1994) that states that even for well-defined domains, the norms that fix the meaning of a word and of its reference (e.g. its concept) cannot be foreseen, and that the meaning of words is immanent to a given situation and context of usage. Archonte claims to provide concepts with a domain and task-dependent meaning by means of the similarities and differences that a word has with other neighboring units in the same context of usage. In order to do so, it uses a set of principles (Bachimont et al., 2002) to create a taxonomy where the differences and similarities are expressed in natural language. These principles are the following: a) similarity with parent (SWP); b) similarity with siblings (SWS); c) difference with siblings (DWS) and d) difference with parent (DWP). Since these principles are attached to concepts, herein lies the similarity with OntoClean. To properly structure a taxonomy, concepts (not relations) must have a set of (meta) properties that determine if a semantic link can exist between any two concepts. Furthermore, although Archonte claims to be domain and task-dependent, it is clear that it is only domain-dependent. Concepts and relations are obtained by processing a domain-dependent corpus, but the corpus itself is independent of any task, and the concepts are arranged using a set of properties that are also independent of any task.

Given the evidence, we claim that structuring a taxonomy can only be meaningfully accomplished within the scope of a specific domain and task. In addition, this cannot be done relying solely on the properties of concepts, semantic relations must also be taken into account. In the next section, we will try to clarify this last point.

Describing and Refining the Semantics of Relations

In (Vaquero et al., 2007) we do an analysis of “task-neutral” LR and point out the need to provide relations with intrinsic semantics in order to prevent the taxonomic flaws of these resources., and in (Vaquero et al., 2006b,c)

we propose to divide the semantics of relations into algebraic and intrinsic properties and to apply the principles of SE to the development of OBLR. Nonetheless, several things were left out.

First, do SE and AI provide the tools for the level of semantic relation description that is needed for the development of software engineered problem-solving OBLR? Second, do taxonomy structuring methodologies actually deal with the contents of the semantic link around which the backbone taxonomy is constructed? Third, although the algebraic properties of relations can be well understood in our proposal, the intrinsic properties are left unspecified. Hence, what could these intrinsic properties be? What is the meaning they could convey? What would they be useful for?

Sections 2 and 3 provide an answer for the first two questions. As for the third one, relation element theory (RET) provides an answer. As explained in (Russomanno, 2006), RET is an effort to provide an exhaustive as possible classification (under the form of a taxonomy) of binary semantic relations (here, the reader must notice the resemblance between RET and the relationship classifications of OCD and UML), on the basis of the nature of the relation between a parent or domain concept and a child or range concept. However, although it would be very difficult to derive an exhaustive and universally agreed upon taxonomy of relations, a set of relation elements or relation primitives are proposed that can be used to describe and refine the semantics of a relation between two entities. These elements are the following: Composable, Connected, Functional, Homeomeric, Intangible, Intrinsic, Near, Separable, Structural and Temporal.

A possible application of these primitives would allow countering the polysemy and synonymy of relations within a same knowledge domain or context. For instance, a relation “part-of” relating an entity Engine to another entity Car could be defined synonymously in another ontology as “physicalParts”, but a machine without prior definitions cannot infer that these two relations are identical. Moreover, providing that we have a suitable algebra, there is the possibility of doing plausible inference (Russomanno, 2006) by using the semantic primitives of relations to infer new relation instances between sets of entities.

However, although the use of the aforementioned set of primitives can clarify the underlying semantics of relations, and meaningful operations can be performed with them, it seems that these primitives conflate several properties that should be separately and explicitly represented as part of the semantics of a single relation or they simply ignore these properties despite the fact that, as it will be seen below, these properties stem from the definition of the primitive itself.

Algebraic and Intrinsic Properties of Relations

First, the primitive Connected indicates that the domain element is temporally or physically connected to the range

element either directly or transitively. Here, transitivity is conflated into one single primitive and made inherent to the primitive itself. This is a mistake, because for a given domain and relation, apparent reasoning anomalies appear, proving that for some relations, transitivity is not inherent to the conceptual relation (Hahn et al., 1999).

Second, the primitive Intangible denotes that the relation that links the domain and range elements is hierarchical with regard to ownership or mental inclusion. However, it is well-known that hierarchical relations (e.g. is-a and part-of), can have a set of algebraic properties (e.g. asymmetry, irreflexivity, transitivity, etc.) that are useful to make valid syllogistic inferences. Nonetheless, it is unknown (as it is not explicitly stated) if the primitive comprises any of these properties or others.

Third, the Structural primitive specifies that the domain and range elements have a hierarchical relationship in which the domain element is below the range element in the hierarchy. Basically, what this primitive is telling us is that the range subsumes the domain or vice versa. Nevertheless, as with the Intangible primitive the set of algebraic properties related to hierarchical relations are simply ignored.

In spite of this, (Russomanno, 2006) argues that the scope of these primitives could be restricted to a knowledge domain of interest or to a context within a knowledge domain, in order to avoid the complications that arise when aiming for a universal set of primitives that could describe every relation in every domain. Just like object-oriented semantic data models, RET seeks universality by using a small set of abstractions. However, it is unlikely that with such a limited set of primitives, the semantics of relations could be described for every possible domain and task.

In (Vaquero et al., 2006b) we propose to represent semantic relations in terms of intrinsic and algebraic properties. We achieve a manifold goal by partitioning the semantics of relations this way. First, to separate any property that can be mathematically represented from properties that represent psychological states (i.e. as in the Intangible primitive), material likeness (i.e. as in the Homeomeric primitive), a specific position in a hierarchy (i.e. as in the Structural primitive), etc. Second, to avoid making any property or primitive a general definitional property of a relation. Third, to allow making fine-grained distinctions for each relation independently of any knowledge representation language. Fourth, to introduce a clear model for the representation and interpretation of relations.

Nevertheless, it has to be clearly stated that we do not aim for an exhaustive classification of relations nor do we propose a universal set of primitives through which any relation can be represented. We just propose to represent relations with well-defined defined semantics up to the granularity needed by the ontology developer. Moreover, based on these ideas, in (Vaquero et al. 2006a) we introduce the notion of control and verification of semantic relations as part of the construction of OBLR for

educational purposes. We will try to clarify this concept in the following section.

Control and Verification of Semantic Relations

Of the many difficulties in building a useful knowledge-based system (KBS), verification is one of the greatest challenges, and as we automate even more and more tasks the need for verification becomes even more crucial (Hicks, 2003).

As far as semantic relations are concerned, control and verification entails that for a given domain and task, a set of conditions must be established to test whether two concepts can be linked by a given relation. Nevertheless, this can only be enforced but through the use of relations with well-defined semantics highly dependent on the domain and task.

The goal is to properly structure an ontology by using a set of properties that can act as domain constraints, without resorting to the metaphysical universality or the text-dependent generality of the available taxonomy structuring methods, as they do not take into account the semantics of relations nor the task for which they are being built.

Consequently, a complete set of relations must be identified and documented early at the development process by doing an analysis of the domain. Moreover, for each relation, its set of intrinsic and algebraic properties must be established as well.

For example, if a semantic relation has an intrinsic property that states that both parent and child must be made of the same stuff, then the system could ask meaningful questions to the OBLR builder in order to keep the consistency inside a domain. Thus, if we were to build an ontology for the domain of pastry, then, in PieSlice “PieceOf” Pie the relation “PieceOf” should ensure (providing that it has a property that states so) that both PieSlice and Pie have the same stuff-like nature. The same could apply to any other intrinsic property. For instance, we could have the property separable as part of a relation “ComponentOf” in the domain of cars. Hence, if we want to add Wheel “ComponentOf” Car, then, the system should ask if Wheel can really be separated from Car and exist independently of it and vice versa. Algebraic properties of relations are also subject to this kind of control in order to enable or disable role propagations and concept specialization on demand, for each relation, as part of the ontology engineering process (Hahn et al., 1999).

Finally, in our work, the notion of control and verification is not an isolated one. We included it as part of a methodological framework for the development of OBLR that will be described next.

A Methodology for the Development of OBLR

Linguistic resources and ontologies for diverse NLP applications have been extensively studied (Sáenz &

Vaquero, 2002). However, there are no references on how these information systems have been developed or upgraded along their life. Moreover, although tools for managing diverse OBLR have been described (Sáenz & Vaquero, 2002), there is no declared SE approach for their development.

This shows that weak attention has been paid on topics about development methodologies for building the software systems which manage LR. Consequently, we claim that the SE methodology subject is necessary in order to develop, reuse and integrate the diverse available LR. Mainly, because a more or less automated incorporation of different OBLR into a common information system, perhaps distributed requires compatible software architectures and sound data management from the different databases to be integrated.

Under this view, we understand OBLR as information systems which are composed of a database core and an application layer which allows the user and applications to interact with the lexical data. Having a database core instead of other file related approaches comes from well-known issues in the DB community (Sáenz & Vaquero, 2002). In particular, we need integrity constraints for maintaining consistency when modifying data. As for the application layer, it should be understood as possibly containing user interfaces. When considering these two components we propose to isolate data from applications, so that all consistency checking is encapsulated into the database core.

Moreover, we claim that both components should be developed following known SE methods. Nonetheless, it is more likely to find these methods applied to the application layer, but, in general, we do not find them applied to the modeling of OBLR. Consequently, we propose the use of a methodology based on relational database technology described in (Sáenz & Vaquero, 2002) in order to build OBLR with a sound and simple structure. We also propose the inclusion of control and verification as part of the methodology, in order to have a controlled way for building domain and task specific OBLR where the intrinsic semantics of relations can be represented, controlled and easily interpreted.

As can be seen, SE plays a major role in our approach, and its methodological advantages are many. Nonetheless, one of its biggest advantages is that it allows us to achieve reusability. This is an important word in SE and AI, but it has different meanings depending on the field.

In SE, reusability stands for the repetitive usage of any part of a software system (Pfleeger, 2001): the documentation, the design, the requirements, test cases, etc. In AI, claiming that a construct (e.g., an ontology, LR, etc.) is reusable entails that it can be used to express knowledge for tasks other than the ones for which it was designed (Bouaud, 1994).

Since we aim at developing problem-solving OBLR, it is important to establish where we stand over this issue. Following (Bouaud, 1994; Bachimont, 2000), we believe that the task for which an application is developed fixes a

particular point of view on the ontology, and the reusability of this resource for another system (i.e., another task) seems difficult. For instance, in non formal domains like medicine, where knowledge is rather descriptive than formal, this point of view is likely dependent on the application. Moreover, we claim that the definition of an ontology is not the characterization or the determination of primitives that already exist in a domain, but the modeling or construction of primitives for the resolution of a problem. Hence, the OBLR constructed with the methodology will be reusable in the SE sense and valid only in the ambit of a given domain and task.

Finally, the original methodology has been used to develop several tools for the construction of an ontology-based dictionaries (Sáenz & Vaquero, 2002; Vaquero et al. 2005), with just one implicit semantic relation.

Conclusions and Future Work

Semantic relations are an important part in the construction of the model of a problem domain in SE and AI. However, they have been put off in favor of concepts or entities.

In SE, semantic data models have an emphasis on entity representation. Relationship representation is done through a small set of fundamental abstractions and when a relationship falls outside their scope it is loosely defined and represented by a tag, or lumped under a category that does not account for its semantics.

In AI, the level of semantic precision in terms of properties and attributes, as well as the operations that can be performed on semantic relations is surprisingly low. AI does not provide any tools for the description and refinement of semantic links up to the granularity needed for the task or problem at hand. Moreover, although the under-specification of semantic links has been pointed out for main OBLR (Vaquero et al., 2007; Kashyap & Borgida, 2003), few initiatives (Vaquero et al., 2006b; Russomanno, 2006) exist that aim at providing not only a richer semantics to these links but also to propose specific operations that can be performed on them.

From the operations that can be performed on semantic relations, we are interested in control and verification, as we believe it would allow the kind verification (at least for semantic relations in OBLR) argued by (Hicks, 2003). Furthermore, their enforcement through the properties of relations would allow to properly structure an ontology. We claim that this structuring is dependent not only on the domain but also on the task for which the OBLR is being built; as it is the context of the task (a specific application) the one that allows fixing the pertinent meaning features of concepts and relations.

Finally, we propose the inclusion of control and verification of semantic relations, as part of a methodology based on relational databases for the construction of OBLR and their interfaces. By doing this we go, qualitatively, further than any other efforts that have used relational databases to build a LR with a Mikrokosmos-like

philosophy and structure (Viegas et al. 1999) for specific applications (Moreno & Pérez, 2000; Cabré et al. 2004).

We are currently in the stage of enhancing a previously developed tool constructed with the methodology (Vaquero et al., 2005) in order to include the ideas mentioned here, and defining for the domain of pneumonia a set of relations and their intrinsic and algebraic properties. Our final goal is to develop an OBLR for the domain of pneumonia that will be used as part of a summarization and categorization system.

References

- Bachimont, B. Engagement sémantique et engagement ontologique: conception et réalisation d'ontologies en Ingénierie des connaissances. 2000. In J. Charlet, M. Zacklad, G. Kassel & D. Bourigault (Eds.). Ingénierie des connaissances, évolutions récentes et nouveaux défis.
- Bachimont, B., Isaac, A., Troncy, R. 2002. Semantic Commitment for Designing Ontologies: A Proposal. Lecture Notes in Computer Science (LNCS) 2473, pp. 114-119.
- Barrett, H.C. 2001. On the Functional Origins of Essentialism. *Mind and Society*, 3, Vol. 2, pp. 1-30.
- Booch, G. 1986. Object-Oriented Development. *IEEE Transactions on Software Engineering* 12(12)
- Booch, G., Rumbaugh, J & Jacobson, I. 1999. *The Unified Modeling Language User Guide*. Addison Wesley.
- Bouaud, J. et al. 1994. Acquisition and Structuring of an Ontology within Conceptual Graphs. In Proc. of the ICCS'94 Workshop on Knowledge Acquisition using Conceptual Graph Theory.
- Catania, J. 2004. Relationship Analysis: Improving The Systems Analysis Process, Ph.D. Dissertation, New Jersey Institute of Technology.
- Cabré, M., Feliu, J., Vivaldi, J. 2002. Towards an Ontology for a Human Genome Knowledge Base. In Proc. of the 3rd International Conference on Language Resources and Evaluation (LREC), pp. 1885-1890.
- Chen, P. The Entity-Relationship Model – Toward a Unified View of Data. 1976. *ACM Transactions on Database Systems* 1(1)
- Hahn, U., Schulz, S., Romacker, M. 1999. Part-whole Reasoning: A Case Study in Medical Ontology Engineering. *Intelligent Systems and their applications*, Vol. 14(5), pp.59-67, IEEE Computer Society.
- 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.
- Kashyap, V. and Borgida, A. 2003. Representing the UMLS[®] Semantic Network using OWL. *Lecture Notes in Computer Science (LNCS)* 2870, pp. 1-16.
- Machery, E. et al. 2004. *Semantics Cross-Cultural Style*. *Cognition* 92, pp.B1-B12, Elsevier.
- Medin, D. and Ortony, A. 1989. Psychological Essentialism. In S. Vosniadou & A. Ortony (eds.), *Similarity and Analogical Reasoning*. New York: Cambridge University Press, pp. 179-195.
- Moreno, A. and Pérez, C. 2000. Reusing the Mikrokosmos Ontology for Concept-Based Multilingual Terminology Databases. In Proc. of the 2nd International Conference on Language Resources and Evaluation (LREC), pp. 1061-1067.
- Nirenburg, S. et al. 2005. *Ontological Semantic Text Processing in the Biomedical Domain*. Working Paper #03-05, Institute for Language and Information Technologies, University of Maryland.
- Pfleeger, S. 2001. *Software Engineering: Theory and Practice*, Second Edition. Prentice Hall.
- Rastier, F., Cavazza, M., Abeillé, A. 1994. *Sémantique pour l'analyse. De la linguistique à l'informatique*, Masson, Paris.
- Rumbaugh, J. 1991. *Object-Oriented Modeling and Design*. Prentice-Hall, New Jersey.
- Russomanno, D. 2006. A Plausible Inference Prototype for the Semantic Web. *Journal of Intelligent Information Systems*, Vol. 26(3), pp.227-246.
- Sáenz, F. and Vaquero, A. 2002. Towards a development methodology for managing linguistic Knowledge Bases. *Research and development in intelligent systems XIX*, pp. 453-466, Springer.
- Schulz et al. 1999. From Knowledge Import to Knowledge Finishing: Automatic Acquisition and Semiautomatic Refinement of Medical Knowledge. In Proc. of the 12th Banff Knowledge Acquisition for Knowledge-Based Systems Workshop - KAW'99.
- Steimann, F., Gößner, J. and Mück, T. 2003. On the Key Role of Composition in Object-Oriented Modeling. *Lecture Notes in Computer Science (LNCS)* 2863, pp. 106-120.
- Vaquero, A. Alvarez, F. and Sáenz, F. 2005. Learning Linguistic Concepts through the Construction of Dictionaries with a DAG shaped Taxonomy. *New Trends and Technologies in Computer-Aided Learning for Computer-Aided Design*. Springer, pp. 91-106.
- Vaquero, A., Álvarez, F., Sáenz, F. 2006a. Control and Verification of Relations in the Creation of Ontology-Based Electronic Dictionaries for Language Learning. 8th International Symposium on Information and Communication Technologies Applied to Education (SIIE 2006), pp. 166-173.
- Vaquero, A., Sáenz, F. Álvarez, F., Buenaga, M. 2006b. Conceptual Design for Domain and Task Specific Ontology-Based Linguistic Resources. *Lecture Notes in Computer Science (LNCS)* 4275, pp. 658-668.
- Vaquero, A. Sáenz, F. Álvarez, F. and Buenaga, M. 2006c. Thinking Precedes Action: Using Software Engineering for the Development of a Terminology Database to Improve Access to Biomedical Documentation. *Lecture Notes in Computer Science (LNCS)* 4345, pp. 207-218.
- Vaquero, A. Álvarez, F. and Sáenz, F. 2007. A Review of Common Problems in Linguistic Resources and a New Way to Represent Ontological Relationships, *Electronic Journal of the Argentine Society for Informatics and Operations Research (EJSADIO)*, Vol. 7(1), ISSN 1514 - 6774, pp. 1-11.
- Viegas, E., Mahesh, K., Nirenburg S., and Beale, S. 1999. *Semantics in Action*. In: P. Saint-Dizier (ed.), *Predicative Forms in Natural Language and in Lexical Knowledge Bases*. Dordrecht-Boston: Kluwer, pp. 1-33.
- Welty, C. and Guarino, N. 2001. Supporting Ontological Analysis of Taxonomic Relationships. *Data and Knowledge Engineering* vol. 39(1). Elsevier.
- Yoo, J., Catania, J. and Ravi, P. 2004. Relationship Analysis in Requirements Engineering. *Requirements Engineering*, Vol. 9(4), pp. 238-247.