Semantic Relations: Issues and Proposals

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 a 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 an 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.

The need to provide the relations used in OBLR with definite semantics has been pointed out by (Vaqueiro et al., 2006c). Moreover, the same authors propose the development of OBLR with such relation semantics following a software engineering approach in (Vaqueiro et al. 2006d). Nonetheless, although they propose a set of ideas for the representation of relations and their semantics, the results of their research stem mainly from an analysis of “task-neutral” LR (Vaqueiro et al., 2006a).

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 problem-solving OBLR need. Furthermore, we aim at covering some of the blanks in the 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 over stress of concepts in detriment of relation representation in AI and ontology 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
Although relationships are a key component of vital design artifacts such as Entity-Relationship (E-R) models and object-class diagrams, they only capture a limited set of relations (relationships in the SE vocabulary), leaving much of the domain’s relationship structure out of the design (Yoo et al., 2004).

E-R modeling (Chen, 1976) one of the best known semantic data modeling approaches, represents the conceptual schema or model of the problem domain by identifying its entities, properties, and relationships. The E-R model depicts the data model of the system so as to map the E-R diagram directly into a specific database management system (DBMS). 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-oriented (OO) modeling (Booch, 1986; Rumbaugh, 1994) object-class diagrams are used to represent the problem domain. In these diagrams, objects are organized by their similarities into classes that describe a set of objects that have the same attributes and behavior patterns. Hence, in this approach, the focus is on objects and their interactions via messages (Yoo et al., 2004). The relationships between classes in OO models can be categorized as denoting: a) Generalization-Specialization; b) Whole-Part/Aggregation and c) an association among otherwise unrelated classes.
Similar to E-R modeling, OO modeling provides a strong representation of entities represented as classes and their attributes. Inheritance 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. The same applies to the UML (Booch et al., 1999) where only a label is used to indicate to indicate that an association, dependency, or realize relationship exists but does not explicitly indicate how.

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, most of these hierarchies have been developed relying essentially on the intuition of their developers (Burgun & Bodenreider, 2001; Oltramari et al., 2005), with no clear understanding of the semantics of the relations involved (Kashyap & Borgida, 2003) and without control and verification tools (Martin, 2003). This 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 with relations that are very general or under-specified and cannot be adequately interpreted (Nirenburg et al., 2005) or that have 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 (Weltz & 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 (Weltz & 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 (Weltz & 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 state 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 tree-shaped 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
Vaquero et al. in (Vaquero et al., 2006a) 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. In subsequent articles, Vaquero et al. (Vaquero et al., 2006c,d) 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 back-bone taxonomy is constructed? Third, although the algebraic properties of relations can be well understood in their 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, 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, Homeomerous, 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 un-known (as it is not explicitly stated) if the primitive comprises any of these properties or others.

Finally there is the Structural primitive. This 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 addition, although (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 relations in every domain, it is unlikely that with such a limited set of primitives, the semantics of relations could be described for every possible domain and task.
Vaquero et al. in (Vaquero et al., 2006c) propose to represent semantic relations in terms of intrinsic and algebraic properties. They 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 Homeomerous 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. Figure 1 illustrates this proposal by introducing an E-R model for an OBLR. Compare this model (in terms of relation representation) to the one in (Moreno & Pérez, 2000), used to represent the entire Mikrokosmos ontology as a relational database.

Furthermore, Vaquero et al in (Vaquero et al. 2006b) introduce the idea of control and verification of semantic relations as part of the construction of OBLR for educational purposes. We will try to clarify this notion in the next 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 domain and range as well as 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 (provid-ing 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).

In addition, although we mentioned above that SE could play a major role in the elicitation and representation of semantic relations, its utility for the construction of OBLR, as it will be shown next, goes beyond semantic relations.
A Methodology for the Development of OBLR

Linguistic resources and ontologies for diverse NLP applications have been extensively studied (Sáenz & Vaquero, 2005). However, there are no references on how these information systems have been developed an upgraded along their life. Moreover, although tools for managing diverse OBLR have been described (Sáenz & Vaquero, 2005), 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, 2005). 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.

Furthermore, 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 as described in (Sáenz & Vaquero, 2005,) in order to build OBLR with a sound and simple structure. In addition, we 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.

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. Consequently, the state-of-the-art in relation representation in both SE and AI needs to be extended.

In SE, semantic modeling techniques have an emphasis on identifying main system components but loosely identify how components are related and interrelated. These semantic models offer the modeler a small set of the fundamental abstractions needed to identify the relationship structure of the application domain, but none of the existing techniques explicitly helps the analyst in determining and documenting the relation structure of the application domain. There is only one proposal that aims at providing a SE methodological approach for relation discovery and documentation: the relationship analysis (RA) methodology developed by (Yoo et al., 2004).

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., 2006a; Kashyap & Borgida, 2003), few initiatives (Vaquero et al., 2006c; 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) that (Hicks, 2003) states that is needed in KBS. Furthermore, their enforcement through the properties of relations would allow the proper structuring of 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 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).

Nevertheless, this implies: a) the development of tools to build OBLR for particular application domains with a specific set of relations; b) inserting the algebraic and intrinsic properties of relations to regulate the authoring process as part of the mechanisms of control and verification and c) the comparison of the results of this process with other ones reached with the same methodology using previously developed tools considering a unique relation (subsumption) in the same domain. The original methodology was used to develop a tool for the construction of an ontology-based Computer Science dictionary with just one implicit relationship between the nodes of the ontology (Vaquero et al. 2005). We intend to carry on the aforementioned tasks by creating an OBLR for a sub-domain of the dictionary (i.e. relational databases) considering the is-a and part-of relations. We are in the stage of defining the intrinsic properties of these relations for this particular sub-domain.
References


