A knowledge model to support inconsistency management when reasoning with shared knowledge

Valentina A.M. Tamma & Trevor J.M. Bench-Capon
Department of Computer Science
University of Liverpool
Chadwick Building
Peach Street
L69 7ZF Liverpool UK

Abstract

This paper presents and motivates an extended ontology knowledge model which represents explicitly semantic information about concepts. This knowledge model results from enriching the usual conceptual model with semantic information which precisely characterises the concept's properties and expected ambiguities, including which properties are prototypical of a concept and which are exceptional, the behaviour of properties over time and the degree of applicability of properties to subconcepts. This enriched conceptual model permits a precise characterisation of what is represented by class membership mechanisms and helps a knowledge engineer to determine, in a straightforward manner, the meta-properties holding for a concept. Meta-properties are recognised to be the main tool for a formal ontological analysis that allows building ontologies with a clean and untangled taxonomic structure. Moreover, this enriched semantics facilitates reasoning mechanisms that can be used in order to solve ambiguities that can arise when ontologies are integrated.

1 Introduction

In the last decade ontologies have moved out of the research environment and have become widely used in many expert system applications not only to support the representation of knowledge but also complex inferences and retrieval. [McGuinness, 2000]. The extensive application of ontologies to broader areas has affected the notion of what ontologies are: they now range from light-weight ontologies, that is taxonomies of non-faceted concepts to more sophisticated ones where not only concepts but also their properties and relationships are represented.

The size of ontologies has also increased dramatically, and it is not so unusual to have ontologies with thousands of concepts. Such huge ontologies are sometimes the efforts of many domain experts and are designed and maintained in distributed environments. For this reasons research efforts are now devoted to merging and integrating diverse ontologies [Pinto *et al.*, 1999].

Lastly, the growing use of ontologies in expert systems requires that ontologies provide a ground for the application of reasoning techniques that result in sophisticated inferences such as those used to check and maintain consistency in knowledge bases.

The interest in designing ontologies that can be easily integrated and provide a base for applying reasoning mechanisms has stressed the importance of suitable conceptual models for ontologies. Indeed, it has been made a point that the sharing of ontologies depends heavily on a precise semantic representation of the concepts and their properties [Fridman Noy and Musen, 1999; McGuinness, 2000; Tamma and Bench-Capon, 2000].

This paper presents and motivates a knowledge model for ontologies which extends the usual set of facets in the OKBC frame-base model [Chaudhri *et al.*, 1998] to encompass more semantic information concerning the concept, which consists of a precise characterisation of the concept's properties and expected ambiguities, including which properties are prototypical of a concept and which are exceptional, the behaviour of the property over time and the degree of applicability of properties to subconcepts. This enriched knowledge model aims to provide enough semantic information to deal with problems of semantic inconsistency that arise when reasoning with integrated ontologies.

The paper is organised as follows: section 2 and subsections presents the motivations for adding semantics to the conceptual model, section 3 presents the knowledge model applying the conceptual model while in section 4 the model with respect to the motivations is discussed. An example of concept description using the knowledge model is given in section 5 and finally section 6 draws conclusions.

2 Encompassing semantics in the conceptual model

The motivation for enriching semantically the ontology conceptual model draws on three distinct arguments that are analysed in the reminder of this section.

2.1 Nature of ontologies

The first argument is based on the nature of ontologies. It has been argued that an ontology is ”*an explicit specification of a conceptualisation*” [Gruber, 1993]. In other words an ontology *explicitly* defines the type of concepts used to describe the abstract model of a phenomenon and the constraints on their use. [Studer *et al.*, 1998]. An ontology is an *a priori* account of the objects that are in a domain and the relationships modelling the structure of the world seen from a particular perspective. In order to provide such an account one has to understand the concepts that are in the domain, and this involves a number of things. First it involves knowing what can sensibly be said of a thing falling under a concept. This can be represented by describing concepts in terms of their properties, and by giving a full characterisation of these properties. Thus, when describing the concept *Bird* it is important to distinguish that some birds fly and others do not. A full understanding of a concept involves more than this, however: it is important to recognise which properties are *prototypical* [Rosch, 1975] for the class membership and, more importantly, which are the permitted exceptions. There are, however differences in how confident we can be that an arbitrary member of a class conforms to the prototype: it is a very rare mammal that lays eggs, whereas many types of well known birds do not fly. Understanding a concept also involves understanding how and which properties change over time. This dynamic behaviour also forms part of the domain conceptualisation and can help to identify the *meta-properties* holding for the concept.

2.2 Integrating diverse ontologies

The second argument concerns the integration of ontologies. Integrating ontologies involves identifying overlapping concepts and creating a new concept, usually by generalising the overlapping ones, that has all the properties of the originals and so can be easily mapped into each of them. Newly created concepts inherit properties, usually in the form of attributes, from each of the overlapping ones.

One of the key points for integrating diverse ontologies is providing methodologies for building ontologies whose taxonomic structure is clean and untangled in order to facilitate the understanding, comparison and integration of concepts. Several efforts are focussing on providing engineering principles to build ontologies, for example [Gómez-Pérez, 1998; 1999]. Another approach [Guarino and Welty, 2000a; 2000b] concentrates on providing means to perform an ontological analysis which gives prospects for better taxonomies. This analysis is based on on a rigorous analysis of the *ontological meta-properties* of taxonomic nodes, which are based on the philosophical notions of *unity, identity, rigidity* and *dependence* [Guarino and Welty, 2000c].

When the knowledge encompassed in ontologies built for different purposes needs to be integrated inconsistencies can become evident. Many types of ontological inconsistencies have been defined in the literature, for instance in [Visser *et al.*, 1998] and the ontology environments currently available try to deal with this inconsistencies, such as SMART [Fridman Noy and Musen, 1999] and CHIMAERA [McGuinness *et al.*, 2000]. Here we broadly classify inconsistencies in ontologies into two types: structural and semantic. We define structural inconsistencies as those that arise because of differences in the properties that describe a concept. Structural inconsistencies can be detected and resolved automatically with limited intervention from the domain expert. Semantic inconsistencies are caused by the knowledge content of diverse ontologies which differs both in semantics and in level of granularity of the representation. They affect those attributes that are actually representing concept features and not relations with other concepts. Semantic inconsistencies require a deeper knowledge on the domain. Examples of semantic inconsistencies can be found in [McGuinness *et al.*, 2000; Tamma and Bench-Capon, 2000]. Adding semantics to the concept descriptions can be beneficial in solving this latter type of conflict, because a richer concept description provides more scope to resolve possible inconsistencies.

2.3 Reasoning with ontologies

The last argument to support the addition of semantics to ontology conceptual models turns on the need to reason with the knowledge expressed in the ontologies. Indeed, when different ontologies are integrated, new concepts are created from the definitions of the existing ones. In such a case conflicts can arise when conflicting information is inherited from two or more general concepts and one tries to reason with these concepts. Inheriting conflicting properties in ontologies is not as problematic as inheriting conflicting rules in knowledge bases, since an ontology is only *providing the means for describing explicitly the conceptualisation behind the knowledge represented in a knowledge base* [Bernaras *et al.*, 1996]. Thus, in a concept's description conflicting properties can coexist. However, when one needs to reason with the knowledge in the ontology, conflicting properties can hinder the reasoning process. In this case extra semantic information on the properties, such as the extent to which the property applies to the members of the class, can be used to derive which property is more likely to apply to the situation at hand. Of course, such sophisticated assumptions cannot be made automatically and are left to knowledge engineers who are assisted in this delicate task by a system presenting them with the most likely options.

3 Extended knowledge model

In this section we extend a frame-based [Minsky, 1992] knowledge model. This is a result of the enriched conceptual model where properties are characterised with respect to their behaviour in the concept description. The knowledge model is based on *classes*, *slots*, and *facets*. *Classes* correspond to concepts and are collections of objects sharing the same properties, hierarchically organised into a multiple inheritance hierarchy, linked by *IS-A* links. Classes are described in terms of *slots*, or attributes, that can either be sets or single values. A slot is described by a name, a domain, a value type and by a set of additional constraints, here called *facets*. Facets can contain the documentation for a slot, constrain the value type or the cardinality of a slot, and provide further information concerning the slot and the way in which the slot is to be inherited by the subclasses. The set of facets has been extended from that provided by OKBC [Chaudhri *et al.*, 1998] in order to encompass descriptions of the attribute and its behaviour in the concept description and changes over time. The facets we use are listed below and discussed in the next section:

- **Value**: It associates a value $v \in$ **Domain** with an attribute in order to represent a property. However, when the concept that is defined is very high in the hierarchy (so high that any conclusion as to the attribute's value is not possible), then either **Value** = `Domain` or **Value** = `Subdomain` $\subset$ `Domain`;
- **Type of value**: The possible fillers for this facet are *Prototypical, Inherited, Distinguishing*. An attribute's value is a *Prototypical* one if the value is true for any prototypical instance of the concept, but exceptions are permitted with a degree of softness expressed by the facet **Ranking**. An attribute's value can be *Inherited* from some super concept or it can be a *Distinguishing* value, that is a value that differentiates among siblings. Note that distinguishing values become inherited values for subclasses of the class;
- **Exceptions**: It can be either a single value or a subset of the domain. It indicates those values that are permitted in the concept description because in the domain, but deemed exceptional from a common sense viewpoint. The exceptional values are not only those which differ from the prototypical ones but also any value which is possible but highly unlikely;
- **Ranking**: An integer describing the degree of confidence of the fact that the attribute takes the value specified in the facet **Value**. It describe the class membership condition. The possible values are 1: *All*, 2: *Almost all*, 3: *Most*, 4: *Possible*, 5: *A Few*, 6: *Almost none*, 7: *None*. For example, in the description of the concept *Bird* the slot *Ability to Fly* takes value `Yes` with *Ranking* 3, since not all birds fly;
- **Change frequency**: Its possible values are: *Regular, Once only, Volatile*. This facet describes how often an attribute's value changes. If the information is set equal to *Regular* it means that the process is continuous, for instance the age of a person can be modelled as changing regularly; if set equal to *Once only* it indicates that only one change is possible, for example a person's date of birth changes only once, and finally *Volatile* indicates that the attribute's value can change more than once, for example people can change job more than once;
- **Event**: Describes conditions under which the value changes. It is the set $\{((E_j, S_j, V_j), R_j) | j = 1, \cdots, m\}$ where E_j is an event, S_j is the state of the pair attribute-value associated with a property, V_j defines the event validity and R_j denotes whether the change is reversible or not. The semantics of this facet is explained in the section below.

4 Relating the extended knowledge model to the motivations

The knowledge model presented in the previous section is motivated by the the problems described in section 2. It is based on an enriched semantics that aims to provide a better understanding of the concepts and their properties by characterising their behaviour.

Concept properties are to be considered on three levels: *instance level*, *class-membership level* and *meta level*. Properties at *instance level* are those exhibited by all the instances of a concept. They might specialise properties at *class-membership level*, which instead describe properties holding for the class. Properties at *meta level* have been mainly described in philosophy, such as *identity, unity, rigidity* and *dependency*. The proposed model permits the characterisation of concepts on the three distinct property levels, thus also considering the meta level which is the basis for the ontological analysis illustrated in [Guarino and Welty, 2000b]. Such an enriched model helps to characterise the meta properties holding for the concepts, thus providing knowledge engineers with an aid to perform the ontological analysis which is usually demanding to perform.

Furthermore, the enriched knowledge model forces knowledge engineers to make ontological commitments explicit. Indeed, real situations are information-rich complete events whose context is so rich that, as it has been argued by Searle [Searle, 1983], it can never be fully specified. Many assumptions about meaning and context are usually made when dealing with real situations [Rosch, 1999]. These assumptions are rarely formalised when real situations are represented in natural language but they have to be formalised in an ontology since they are part ontological commitments that have to be made explicit. Enriching the semantics of the attribute descriptions with things such as the behaviour of attributes over time or how properties are shared by the subclasses makes some of the more important assumptions explicit.

The enriched semantics is essential to solve the inconsistencies that arise either while integrating diverse ontologies or while reasoning with the integrated knowledge. By adding information on the attributes we are able to better measure the similarity between concepts, to disambiguate between concepts that *seem* similar while they are not, and we have means to infer which property is likely to hold for a concept that inherits inconsistent properties. The remainder of this section describes the additional facets and relates them to the discussion in section 2.

4.1 Behaviour over time

In the knowledge model the facets *Change frequency* and *Event* describe the behaviour of properties over time, which models the changes in properties that are permitted in the concept's description without changing the essence of the concept. The behaviour over time is closely related to establishing the *identity* of concept descriptions [Guarino and Welty, 2000b]. Describing the behaviour over time involves also distinguishing properties whose change is *reversible* from those whose change is *irreversible*.

Property changes over time are caused either by the natural passing of time or are triggered by specific event occurrences. We need, therefore, to use a suitable temporal framework that permits us to reason with time and events. The model chosen to accommodate the representation of the changes is the *Event Calculus* [Kowalski and Sergot, 1986]. Event calculus deals with local event and time periods and provides the ability to reason about change in properties caused by a specific event and also the ability to reason with incomplete information.

We can distinguish *continuous* versus *discrete properties*. *Continuous properties* are those changing regularly over time, such as the age of a person, while *discrete properties* are those characterised by an event which causes the property to change. If the value associated with change frequency is *Regular* then the property is continuous, if it is *Volatile* the property is discrete and if it is *Once only* then the property is considered discrete and the triggering event is set equal to *time-point=T*.

Any regular occurrence of time can be, however, expressed in form of an event, since most of the forms of reasoning for continuous properties require discrete approximations. Therefore in the knowledge model presented in the next section, continuous properties are modelled as discrete properties where the event triggering the change in property is the passing of time from the instant t to the instant t'. Each change of property is represented by a set of quadruples $\{((E_j, S_j, V_j), R_j) | j = 1, \cdots, m\}$ where E_j is an event, S_j is the state of the pair attribute-value associated with a property, V_j defines the event validity while R_j indicates whether the change in properties triggered by the event E_j is reversible or not. The model used to accommodate this representation of the changes adds reversibility to *Event Calculus*, where each triple (E_j, S_j, V_j) is interpreted either as *the concept is in the state* S_j *before the event* E_j *happens* or *the concept is in the state* S_j *after the event* E_j *happens* depending on the value associated with V_j. The interpretation is obtained from the semantics of the event calculus, where the former expression is represented as *Hold(before(*E_j, S_j*))* while the latter as *Hold(after(*E_j, S_j*))*.

The idea of modelling the permitted changes for a property is strictly related to the philosophical notion of *identity*. In particular, the knowledge model addresses the problem of modelling identity when time is involved, namely *identity through changes*, which is based on the common sense notion that an individual may remain the same while showing different properties at different times [Guarino and Welty, 2000a]. The knowledge model we propose explicitly distinguishes the properties that can change from those which cannot, and describes the changes in properties that an individual can be subjected to, while still being recognised as an instance of a certain concept.

The notion of changes through time is also important to establish whether a property is *rigid*. A *rigid property* is defined in [Guarino *et al.*, 1994] as:

> a property that is essential to *all* its instances, i.e. $\forall x \phi(x) \rightarrow \Box \phi(x)$.

The interpretation that is usually given to *rigidity* is that if x is an instance of a concept C than x has to be an instance of C in every possible world. Time is one of these systems of possible worlds and characterising a property as rigid in time gives a better angle on the *necessary* and *sufficient* conditions for the class membership.

4.2 Ranking

Rankings are defined as [Goldszmidt and Pearl, 1996]:

> Each world is ranked by a non-negative integer representing the degree of surprise associated with finding such a world.

We have borrowed the term to denote the degree of surprise in finding a world where the property P holding for a concept C does not hold for one of its subconcepts C'. The additional semantics encompassed in this facet is important to reason with statements that have different degrees of truth. Indeed there is a difference in asserting facts such as ”Mammals give birth to live young” and ”Bird fly”, the former is generally more believable than the latter, for which many more counterexamples can be found. The ability to distinguish facts whose truth holds with different degrees of strength is related to finding facts that are true in every possible world and therefore constitute *necessary truth*. The concept of necessary truth brings us back to establishing whether a property is rigid or not, in fact it can be assumed that the value associated with the *Ranking* facet together with the temporal information on the changes permitted for the property lead us to determine whether the property described by the slot is a rigid one. Rigid properties have often been interpreted as *essential* properties (*i.e.*, a property holding for an individual in every possible circumstance in which the individual exists), however, we have to note that a property might be essential to a member of a class without being essential for membership in that class. For example, being odd is an essential property of the number 5, but it is not essential for membership in the class of prime numbers.

The ability to evaluate the degree of truth of a property in a concept description is also related to the problem of reasoning with ontologies obtained by integration. In such a case, as mentioned in section 2.3 inconsistencies can arise if a concepts inherits conflicting properties. In order to be able to reason with these conflicts some assumptions have to be made, concerning on how likely it is that a certain property holds; the facet *Ranking* models this information by modelling a qualitative evaluation of how subclasses inherit the property. This estimate represents the common sense knowledge expressed by linguistic quantifiers such as *All, Almost all, Few, etc.*.

In case of conflicts the property's degree of truth can be used to rank the possible alternatives following an approach similar to the non-monotonic reasoning one developed by [Goldszmidt and Pearl, 1996]: in case of more conflicting properties holding for a concept description, properties are ordered according to the degree of truth, that is a property holding for all the subclasses is considered to have a higher rank than one holding for few of the concept subclasses. This ordering of the conflicting properties needs to be validated by the knowledge engineer, however, it reflects the common sense assumption that, when no specific information is known, people assume that the most likely property holds for a concept.

4.3 Prototypes and exceptions

In order to get a full understanding of a concept it is not sufficient to list the set of properties generally recognised as describing a typical instance of the concept but we need to consider the expected exceptions. Here we partially take the cognitive view of prototypes and graded structures, which is also reflected by the information modelled in the facet *Rank-*

ing. In this view all cognitive categories show gradients of membership which describe how well a particular subclass fits people's idea or image of the category to which the subclass belong [Rosch, 1975]. Prototypes are the subconcepts which best represent a category, while exceptions are those which are considered exceptional although still belong to the category. In other words all the sufficient conditions for class membership hold for prototypes. For example, let us consider the biological category *mammal*: a *monotreme* (a mammal who does not give birth to live young) is an example of an exception with respect to this attribute. Prototypes depend on the context; there is no universal prototype but there are several prototypes depending on the context, therefore a prototype for the category *mammal* could be *cat* if the context taken is that of *pets* but it is *lion* if the assumed context is *circus animal*. Ontologies typically presuppose context and this feature is a major source of difficulty when merging them.

For the purpose of building ontologies, distinguishing the prototypical properties from those describing exceptions increases the expressive power of the description. Such distinctions do not aim at establishing default values but rather to guarantee the ability to reason with incomplete or conflicting concept descriptions.

The ability to distinguish between prototypes and exceptions helps to determine which properties are necessary and sufficient conditions for concept membership. In fact a property which is prototypical and that is also inherited by all the subconcepts (that is it has the facet *Ranking* set to *All*) becomes a natural candidate for a necessary condition. Prototypes, therefore, describe the subconcepts that best fit the cognitive category represented by the concept *in the specific context given by the ontology*. On the other hand, by describing which properties are exceptional, we provide a better description of the class membership criteria in that it permits to determine what are the properties that, although rarely hold for that concept, are still possible properties describing the cognitive category. Here, the term *exceptional* is used to indicate something that differs from what is normally thought to be a feature of the cognitive category and not only what differs from the prototype.

Also the information on prototype and exceptions can prove useful in dealing with inconsistencies arising from ontology integration. When no specific information is made available on a concept and it inherits conflicting properties, then we can assume that the prototypical properties hold for it.

The inclusion of prototypes in the knowledge model provides the grounds for the semi-automatic maintenance and evolution of ontologies by applying techniques developed in other fields such as machine learning .

5 A modelling example

We now provide an example to illustrate how the previously described knowledge model can be used for modelling a concept in the ontology. The example is taken from the medical domain and we have chosen to model the concept of *blood pressure*. Blood pressure is represented here as an ordered pair (s, d) where *s* is the value of the *systolic pressure* while *d* is the value of the *diastolic pressure*. In modelling the concept of blood pressure we take into account that both the systolic and diastolic pressure can range between a minimum and a maximum value but that some values are more likely to be registered than others. Within the likely values we then distinguish the *prototypical* values, which are those registered for a healthy individual whose age is over 18, and the *exceptional* ones, which are those registered for people with pathologies such as hypertension or hypotension. The prototypical values are those considered normal, but they can change and we describe also the permitted changes and what events can trigger such changes. Prototypical pressure values usually change with age, but they can be altered depending on some specific events such as shock and haemorrhage (causing hypotension) or thrombosis and embolism (causing hypertension). Also conditions such as pregnancy can alter the normal readings.

Classes are denoted by the label **c**, slots by the label **s** and facets by the label **f**. Irreversible changes are denoted by I while reversible property changes are denoted by R.

c: `Circulatory system;`
s: `Blood pressure`
f: `Domain`: [(0,0)-(300,200)];
f: `Value`: [(90,60)-(130,85)];
f: `Type of value`: prototypical;
f: `Exceptions`: [(0,0)-(89,59)] $\cup$ [(131,86)-(300,200)];
f: `Ranking`: 3;
f: `Change frequency`: Volatile;
f: `Event`: (Age=60,[(0,0)-(89,59)] $\cup$
$\cup$ [(131,86)-(300,200)],after, I);
f: `Event`: (haemorrhage,[(0,0)-(89,59)],after, R);
f: `Event`: (shock,[(0,0)-(89,59)],after, R);
f: `Event`: (thrombosis,[(131,86)-(300,200)],after,R);
f: `Event`: (embolism,[(131,86)-(300,200)],after,R);
f: `Event`: (pregnancy,[(0,0)-(89,59)] $\cup$
$\cup$ [(131,86)-(300,200)],after,R);

6 Conclusions

This paper has presented a knowledge model that extends the usual ontology frame-based model such as OKBC by explicitly representing additional information on the slot properties. This knowledge model results from a conceptual model which encompasses semantic information aiming to characterise the behaviour of properties in the concept description. We have motivated this enriched conceptual model by identifying three main categories of problems that require additional semantics in order to be solved.

The novelty of this extended knowledge model is that it explicitly represents the behaviour of attributes over time by describing the permitted changes in a property that are permitted for members of the concept. It also explicitly represents the class membership mechanism by associating with each slot a qualitative quantifier representing how properties are inherited by subconcepts. Finally, the model does not only describe the prototypical properties holding for a concept but also the exceptional ones.

We have also related the extended knowledge model to the formal ontological analysis by Guarino and Welty [Guarino and Welty, 2000b] which permits to build ontologies that have a cleaner taxonomic structure and so gives better prospects for maintenance and integration. Such a formal ontological analysis is usually difficult to perform and we believe our knowledge model can help knowledge engineers to determine the meta-properties holding for the concept by forcing them to make the ontological commitments explicit.

A possible drawback of this approach is the high number of facets that need to filled when building ontology. We realise that this can make building an ontology from scratch even more time consuming but we believe that the outcomes in terms of better understanding of the concept and the role it plays in a context together with the guidance in determining the meta-properties at least balances the increased complexity of the task.

References

[Bernaras *et al.*, 1996] A. Bernaras, I. Laresgoiti, and J. Corera. Building and reusing ontologies for electrical network applications. In *Proceedings of the 12th European Conference on Artificial Intelligence (ECAI)*, pages 298–302, 1996.

[Chaudhri *et al.*, 1998] V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp, and J.P. Rice. OKBC: A programmatic foundation for knowledge base interoperability. In *Proceedings of the Fifteenth American Conference on Artificial Intelligence (AAAI-98)*, pages 600–607, Madison, Wisconsin, 1998. AAAI Press/The MIT Press.

[Fridman Noy and Musen, 1999] N. Fridman Noy and M.A. Musen. SMART: Automated support for ontology merging and alignment. In *Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW)*, Banff, Canada, 1999.

[Goldszmidt and Pearl, 1996] M. Goldszmidt and J. Pearl. Qualitative probabilistic for default reasoning, belief revision, and causal modelling. *Artificial Intelligence*, 84(1-2):57–112, 1996.

[Gómez-Pérez, 1998] A. Gómez-Pérez. Knowledge sharing and reuse. In J. Liebowitz, editor, *The Handbook of Applied Expert Systems*. CRC Pres LLC, 1998.

[Gómez-Pérez, 1999] A. Gómez-Pérez. Ontological engineering: A state of the art. *Expert Update*, 2(3):33–43, Autumn 1999.

[Gruber, 1993] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.

[Guarino and Welty, 2000a] N. Guarino and C. Welty. A formal ontology of properties. In R. Dieng, editor, *Proceedings of the 12th EKAW Conference*, volume LNAI 1937. Springer Verlag, 2000.

[Guarino and Welty, 2000b] N. Guarino and C. Welty. Identity, unity and individuality: Towards a formal toolkit for ontological analysis. In W. Horn, editor, *Proceedings of the 14th European Conference on Artificial Intelligence (ECAI)*. IOS Press, 2000.

[Guarino and Welty, 2000c] N. Guarino and C. Welty. Towards a methodology for ontology based model engineering. In *Proceedings of the ECOOP 2000 workshop on model engineering*, 2000.

[Guarino *et al.*, 1994] N. Guarino, M. Carrara, and P. Giaretta. An ontology of meta-level-categories. In *Principles of Knowledge representation and reasoning: Proceedings of the fourth international conference (KR94)*. Morgan Kaufmann, 1994.

[Kowalski and Sergot, 1986] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[McGuinness *et al.*, 2000] D.L. McGuinness, R.E. Fikes, J. Rice, and S. Wilder. An environment for merging and testing large ontologies. In *Proceedings of KR-2000. Principles of Knowledge Representation and Reasoning.* Morgan-Kaufman, 2000.

[McGuinness, 2000] D.L. McGuinness. Conceptual modelling for distributed ontology environments. In *Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000)*, 2000.

[Minsky, 1992] M. Minsky. A framework for representing knowledge. In A. Collins and E.H. Smith, editors, *Cognitive Science*, pages 191–215. Morgan Kaufmann, Los Altos, CA, 1992.

[Pinto *et al.*, 1999] H.S. Pinto, A. Gómez-Pérez, and J.P. Martins. Some issues on ontology integration. In V.R. Benjamins, editor, *Proceedings of the IJCAI'99 Workshop on Ontology and Problem-Solving Methods: Lesson learned and Future Trends*, volume 18, pages 7.1–7.11, Amsterdam, 1999. CEUR Publications.

[Rosch, 1975] E.H. Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:192–233, 1975.

[Rosch, 1999] E.H. Rosch. Reclaiming concepts. *Journal of Consciousness Studies*, 6(11-12):61–77, 1999.

[Searle, 1983] J.R. Searle. *Intentionality*. Cambridge University Press, Cambridge, 1983.

[Studer *et al.*, 1998] R. Studer, V.R. Benjamins, and D. Fensel. Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197, 1998.

[Tamma and Bench-Capon, 2000] V.A.M. Tamma and T.J.M Bench-Capon. Supporting inheritance mechanisms in ontology representation. In R. Dieng, editor, *Proceedings of the 12th EKAW Conference*, volume LNAI 1937, pages 140–155. Springer Verlag, 2000.

[Visser *et al.*, 1998] P.R.S. Visser, D.M. Jones, T.J.M. Bench-Capon, and M.J.R. Shave. In N. Guarino, editor, *Formal Ontology in Information Systems. Proceedings FOIS'98, Trento, Italy*, pages 148–182. IOS Press, 1998.