Characterising Concept's Properties in Ontologies

Valentina A.M. Tamma and Trevor J.M. Bench-Capon

Department of Computer Science, The University of Liverpool
Liverpool L69 7ZF, United Kingdom
{valli,tbc}@csc.liv.ac.uk

Abstract. This paper presents and motivates an extended ontology conceptual model which represents explicitly semantic information about concepts. This model results from explicitly representing 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. Moreover, this enriched semantics facilitates the development of reasoning mechanisms on the state of affairs that instantiate the ontologies. Such reasoning mechanisms can be used in order to solve ambiguities that can arise when ontologies are integrated and one needs to reason with the integrated knowledge.

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. [McG00]. 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.
More and more often ontologies are 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 [PGPM99]. 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

F. Esposito (Ed.): AI*IA 2001, LNAI 2175, pp. 189–200, 2001.

sharing of ontologies depends heavily on a precise semantic representation of the concepts and their properties [FM99, McG00, TBC00].
This paper presents and motivates an extension to the classic conceptual model for ontologies, which describes entities in the domain by a set of concepts defined in terms of some exhibited properties or *attributes*, and the relationships connecting these concepts. The enriched ontology model presented in this paper proposes to encompass additional 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 conceptual 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 an OKBC-based [CFF$^+$98] knowledge model instantiating the proposed conceptual model while section 4 discusses the model. An example of concept description using the proposed 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 Integrating Diverse Ontologies

The first 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 [GP98, GP99]. Another approach [GW00] concentrates on providing means to perform an ontological analysis which gives prospects for better taxonomies. It is based 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* [GW00].
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 [VJBCS98]

and the ontology environments currently available try to deal with this inconsistencies, such as SMART [FM99] and CHIMAERA [MFRW00]. 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 structure of the concept definitions. 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 require a deeper knowledge on the domain [MFRW00, TBC00]. 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.

2.2 Reasoning with Ontologies

The second 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* [BLC96]. 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.

2.3 Nature of Ontologies

The last argument is based on the nature of ontologies. An ontology *explicitly* defines the type of concepts used to describe the abstract model of a phenomenon and the constraints on their use [SBF98]. It 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.
It has been argued that such information is not, strictly *ontological* but it is more of *epistemic* nature (Guarino, personal communication). From a philosophical

viewpoint an ontology is an *a priori* description of what constitues *necessary truth* in any possible world [Kri80]. Such a formal standing on ontologies permits to add a meta-level of description to ontologies and thus to reason about *meta-properties* [GW00]. We believe that in order to be able to share and reuse ontologies and to reason with the knowledge expressed in ontologies, the formal meta-level of the description should be complemented by a richer concept description, more oriented to the knowledge sharing task. If we consider the different ways in which the term *ontology* has been used in artificial intelligence, we obtain a spectrum where formal ontologies are at one end, while something close to knowledge bases are at the other end of the spectrum. Our view on ontology is somewhere in the middle: ontologies should provide enough information to enable knowledge engineers to have a full understanding of a concept *as it is in the actual world*, but should also enable knowledge engineers to perform a formal ontological analysis. For this reason, we believe in ontologies that provide an a priori account of necessary truth on all the possible worlds but also some information on the *actual world and all the worlds accessible from it.*

A full understanding of a concept involves more than this, however: it is important to recognise which properties are *prototypical* [Ros75] 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.

3 A Knowledge Model Representing the Enriched Conceptual Model

In this section we illustrate a frame-based model which results by representing the elements of the conceptual model in terms of the frame paradigm. We have chosen to extend a frame-based, OKBC-like [CFF$^+$98] knowledge model, since the frame-based paradigm applied to ontologies is thought of being easy to use because closer to the human way of conceptualise, and providing a rich expressive power (a discussion on frame-based languages for ontologies can be found in [LM01]).

In this model 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 provided by OKBC has been extended 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, where we distinguish *epistemic nature* facets from *ontological nature* ones, and are discussed in the next section:

– **Defining Values**: 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 it is more likely to associate with the slot **Defining Values** either the whole domain (when no decision at all can be made on the attribute's value) or a subset of the domain (when a concept is defined by means of inheritance from a parent, thus the concept inherits the slot's filler from its parent but specialises it by identifying a subset of the domain characterising the parent concept), that is **Defining Values** = `Domain` or **Defining Values** = `Subdomain` $\subset$ `Domain.`For example, when describing a generic concept such as *Person* in terms of the attribute *Age*, we can associate with this slot the **Defining Values**=[0, 120], expressing the fact that a person's age can range between 0 and 120. In such a case [0, 120] coincides with **Domain**. The concept is too generic in order to associate a single value with the slot *Age*. If, then, we define the concept *Teenager* as subconcept of *Person*, this inherits from *Person* the slot *Age*, but the child concept is qualified by associating with this slot the **Defining Values**=[11, 18] which is a subset of [0, 120]. This is an ontological facet;
– **Value descriptor**: The possible filler for this facet are *Prototypical, Inherited, Distinguishing, Value*. 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 credibility expressed by the facet **Modality**. 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. If this facet is set to *Value* this means that the value is neither prototypical, nor inherited or distinguishing. Note that inherited and distinguishing values are incompatible in the same concept description, that is a value is either inherited or distinguishing, but cannot be both. On the other hand a value can be prototypical and inherited. Distinguishing values become inherited for subclasses of the class. This is an ontological facet;
– **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 they are 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. This property is epistemic;
– **Modality**: 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*. The value *None* associated

with this facet tantamounts to negation. 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. This facet is epistemic;

– **Change frequency**: Its possible values are: *Regular, Once only, Volatile, Never*. This facet describes how often an attribute's value changes. If the information is set equal to *Regular* it means that the change process is continuous, for instance the age of a person can be modelled as changing regularly. If the facet is set equal to *Once only* it means that only one change over time is possible, while if the facet is set equal to *Never* it indicates that the value is set only once and then it cannot change again, for example a person's date of birth once set cannot change again, and finally *Volatile* means that the change process is discrete and can be repeated, that is the attribute's value can change more than once, for example people can change job more than once. This property is epistemic;
– **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. This facet is epistemic. 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 permits the characterisation of concepts by providing means to understand and detect also the *meta properties* holding for a concept. By adopting the proposed conceptual model knowledge engineers are assisted in performing the ontological analysis which is usually demanding to perform, and they are forced 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 [Sea83], it can never be fully specified. Many assumptions about meaning and context are usually made when dealing with real situations [Ros99]. 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. 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* [KS86]. 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.*

Since most of the forms of reasoning for continuous properties require discrete approximations, we transform any regular occurrence of time in form of an event, by representing the event triggering the change in property as 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)).*

Events in this representation are always *point events*, and we consider *durational events* (events which have a duration) as being a collection of *point events* in which the state of the pair attribute-value as determined by the value of V_j, holds as long as the event lasts. The duration is determined by the definition of an *event* in *Event Calculus*, where for each event is given an initial and a final time point. We realise that this representation oversimplify the dynamic of process changes and we aim to investigate a more sophisticated change representation as future work.

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 change, which is based on the common sense notion that an individual may remain the same while showing different properties at different times [GW00]. 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 [GCG94] 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 only one of these systems of possible worlds, however characterising a property as rigid even if only with respect to time gives a better angle on the *necessary* and *sufficient* conditions for the class membership.

4.2 Modality

The term modality is used to express the way in which a statement is true or false, which is related to establish whether a statement constitute a *necessary truth* and to distinguish necessity from possibility [Kri80]. The term can be extended to qualitatively measure the way in which a statement is true by trying to estimate the number of possible world in which such a truth holds. This is the view we take in this paper, by denoting the degree of confidence that we can associate with finding a certain world with the facet *Modality*. This notion is quite similar to the one of *Ranking* as defined by Goldszdmidt and Pearl [GP96]: *Each world is ranked by a non-negative integer representing the degree of surprise associated with finding such a world* .
Here we use the term modality 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 ability to distinguish facts whose truth holds with different degrees of strength is important in order to find which facts are true in every possible world and therefore constitute *necessary truth*. The concept of necessary truth brings us back to the discussion about *rigidity*, in fact it can be assumed that the value associated with the *Modality* 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. Good candidate to be rigid properties are those whose *Modality* facet is equal to *All* and that cannot change in time, that is whose *Change frequency* facet is set to *Never*.
The ability to evaluate the degree of confidence in a property describing a concept is also related to the problem of reasoning with ontologies obtained by integration. In such a case, as mentioned in Section 2.2 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 *Modality* 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 [GP96]: in case of more conflicting properties holding for a concept description, properties might be 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.

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 denote by *prototype* those values that are prototypical for the concept that is being defined; in this way, we partially take the cognitive view of prototypes and graded structures, which is also reflected by the information modelled in the facet *Modality*. 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 [Ros75]. 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 *animals that can play the role of pets* but it is *lion* if the assumed context is *animals that can play the role of circus animals*.
In the model presented above we explicitly describe the context in natural language in the *Documentation* facet, however, the context can be also described by the roles that the concept which is being described is able to play.
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 *Modality* 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.

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: `Defining` verb Values : [(90,60)-(130,85)];
    f: `Value descriptor`: prototypical;
    f: `Exceptions`: [(0,0)-(89,59)] ∪ [(131,86)-(300,200)];
    f: `Modality`: 3;
    f: `Change frequency`: Volatile;
    f: `Event`: (Age=60,[(0,0)-(89,59)] ∪ [(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)] ∪ [(131,86)-(300,200)],after,R);

6 Conclusions

This paper has presented an extended conceptual model for ontologies that encompasses additional semantic information aiming to characterise the behaviour of properties in the concept description. We have motivated this enriched conceptual model by discussing the problems that require additional semantics in order to be solved.

The novelty of this extended conceptual 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 [GW00], which is usually difficult to perform and we believe our 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

[BLC96] 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.

[CFF+98] 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.

[FM99] 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.

[GCG94] 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.

[GP96] M. Goldszmidt and J. Pearl. Qualitative probabilistic for default reasoning, belief revision, and causal modelling. *Artificial Intelligence*, 84(1-2):57–112, 1996.

[GP98] A. Gómez-Pérez. Knowledge sharing and reuse. In J. Liebowitz, editor, *The Handbook of Applied Expert Systems*. CRC Pres LLC, 1998.

[GP99] A. Gómez-Pérez. Ontological engineering: A state of the art. *Expert Update*, 2(3):33–43, Autumn 1999.

[GW00] 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.

[Kri80] S.A. Kripke. *Naming and necessity*. Harvard University Press, 1980.

[KS86] R. Kowalski and M. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.

[LM01] O. Lassila and D. McGuinness. *The Role of Frame-Based Representation on the Semantic Web*, volume Vol. 6(2001), number not yet determined of *Linkping Electronic Articles in Computer and Information Science. ISSN 1401-9841*. 2001.

[McG00] 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.

[MFRW00] 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.

[PGPM99] 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.

[Ros75] E.H. Rosch. Cognitive representations of semantic categories. *Journal of Experimental Psychology: General*, 104:192–233, 1975.

[Ros99] E.H. Rosch. Reclaiming concepts. *Journal of Consciousness Studies*, 6(11-12):61–77, 1999.

[SBF98] R. Studer, V.R. Benjamins, and D. Fensel. Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, 25(1-2):161–197, 1998.

[Sea83] J.R. Searle. *Intentionality*. Cambridge University Press, Cambridge, 1983.

[TBC00] 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.

[VJBCS98] P.R.S. Visser, D.M. Jones, T.J.M. Bench-Capon, and M.J.R. Shave. Assessing heterogeneity by classifying ontology mismatches. In N. Guarino, editor, *Formal Ontology in Information Systems. Proceedings FOIS'98, Trento, Italy*, pages 148–182. IOS Press, 1998.