An enriched knowledge model for formal ontological analysis

Valentina A.M. Tamma & Trevor Bench-Capon
Department of Computer Science,
University of Liverpool
Liverpool L69 7ZF, UK
{valli, tbc}@csc.liv.ac.uk

Abstract — This paper presents and motivates an extended ontology knowledge model which explicitly represents semantic information about concepts. This knowledge model is grounded on the meta-properties of formal ontological analysis and it 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 the development of reasoning mechanisms on the state of affairs that instantiates 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.

Categories & Descriptors — I.2.0 [Artificial Intelligence: General]: *Philosophical foundations.*

General Terms — Ontology modelling.

Keywords — Ontologies, conceptual models, knowledge sharing.

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 [16].

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 ontologies where not only concepts but also their properties and relationships are represented.

Ontologies with thousands of concepts are not so unusual and are sometimes the efforts of

---

FOIS'01, October 17-19, 2001, Ogunquit, Maine, USA.

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 [18].

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 [3, 16, 27].

This paper presents and motivates the traditional conceptual model for ontologies that describes entities (concepts in the domain conceptualisation) in terms of characteristic features called *attributes* and the relationships between these concept. Our proposal is to extend this model by encompassing additional semantics which permits the precise characterisation of the concept's properties, by explicitly representing prototypical and exceptional properties, how attributes behave over time and the degree of applicability of properties to subconcepts.

In order to provide an example of how knowledge can be modelled by using the conceptual model presented in this paper, we have instantiated the conceptual model in an OKBC-like ontology model [2].

The additional information provided by this model supports the formal ontological analysis by Guarino and Welty [11], which permits the building of ontologies that have a cleaner taxonomic structure and so gives better prospects for maintenance and integration. Such a formal ontological analysis is usually demanding 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.

Moreover, the enriched conceptual model provides enough semantic information to deal with the 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 conclusions are drawn in section 6.

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.

Ontologies *explicitly* define the type of concepts used to describe the abstract model of a phenomenon and the constraints on their use [26].

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.

It can be argued at this point that this kind of knowledge is not *ontological* but has *epistemic* nature [28]. If we consider ontologies from a pure philosophical perspective, they are an *a priori* description of what constitutes *necessary truth* in all possible worlds [14]. Such a formal stance on ontologies makes it possible to add to ontologies a meta-level of description and thus to reason about *meta-properties* [10]. We believe that in order for ontologies to be really effective in sharing and reusing knowledge and to reason with the instantiation of the knowledge represented in ontologies, the formal meta-level of the description should be complemented by a richer concept description, which is more oriented to knowledge sharing purposes. If we consider the different ways in which the term *ontology* has been defined and used in artificial intelligence, we obtain a spectrum where formal ontologies are at one end of the spectrum, while something close to knowledge bases is at the other end. Our view on ontologies is somewhere in the middle of this spectrum: ontologies should provide sufficient information to enable knowledge engineers to have a full understanding of a concept *as it is in the domain* (that is in the real world), but should also enable knowledge engineers to perform a formal ontological analysis. For this reason, we believe that ontologies should be compatible with an *a priori* account of necessary truth in 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* [20] for the class membership and, more importantly, which are the permitted exceptions (in the actual world). 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. However, there are cases, as highlighted in [9, 10], in which recognising overlapping concepts is not sufficient to guarantee that a suitable generalising concept (expressing the integrated viewpoints) can be found.

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 using engineering principles to build ontologies, for example [5, 6]. Another approach [9, 10] concentrates on providing means to perform an ontological analysis, which gives prospects for better taxonomies. This analysis 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* [10, 11].

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 [29] and the ontology environments currently available try to deal with this inconsistencies, such as SMART [3] and CHIMAERA [17]. 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 [17, 27]. Adding semantics to the concept descriptions can be beneficial in solving this latter type of conflict, because a richer concept description provides more information with which 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 axioms in a logical theory, since an ontology is only *providing the means for describing explicitly the conceptualisation behind the knowledge represented in a knowledge base* [1]. Thus, in a concept 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 model which results from representing the elements of the conceptual model in terms of the frame paradigm. We have chosen a frame-based paradigm, namely OKBC [2] since this paradigm applied to ontologies is thought to be easy to use because closer to the human way of conceptualise, and provides a rich expressive power (a thorough discussion on why frame-based languages are suitable for ontologies can be found in [15]).

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 has been extended from that provided by OKBC [2] 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:

Class type: The facet *class type* has been added to the OKBC ones to specify whether the class that is being defined is a concept or a role. This facet can take two possible values: `concept` and `role,` which are used to change the meaning of some of the frame facets.

Slot value label: The value associated with the facet *slot value label* of a slot S is one or more elements from the list of possible fillers: {`Value, Prototypical, Inherited, Distinguishing`}.

If the slot value is labelled as `Value` it means that the value is neither prototypical, nor inherited or distinguishing but peculiar to the thing itself. If the value associated with the facet slot value label of slot S `Prototypical` it means that the value associated with the slot S is true for any prototypical instance of the class, but exceptions are permitted with a degree of credibility expressed by the facet *modality*. If the value associated with the facet is `Inherited` this means that the value associated with S has been inherited from some super class. If the slot value is labelled through the facet *slot value label* as `Distinguishing` this means that it is a value that differentiates among siblings with a common super class.

It should be noted 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 values* for subclasses of the class.

Exceptions: The facet *exceptions* of slot S specifies which values of those associated with the slot S are to be considered as exceptional, that is 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. The value that this facet can take is therefore a value or a subset of the values associated with the slot S.

Modality: The facet *modality* of slot S of frame F denotes the degree of confidence in the fact that the slot takes one or more specified values. It describes the class membership conditions. The value associated with this facet is a nonnegative integer between 1 and 7. Each of this number is associated with a specific meaning. The possible fillers for this facet are: {1="All"; 2="Almost all"; 3="Most"; 4="Possible"; 5="A Few"; 6="Almost none"; 7= "None"}.

In particular the value "None" associated with this facet is tantamount 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;

Change frequency: The facet *change frequency* of slot S specifies whether and how often the value of slot S changes during the lifetime of the concept which is represented by the frame F that is being described. The value associated with this facet is one of the following possible values: {"Regular", "Once only", "Volatile", "Never"}.

If the value of the facet is "Regular" it denotes that the change process is continuous, for instance the age of a person can be modelled as changing regularly. If the facet value is equal to "Once only" it means that only one change over time for the value of slot S is possible. If the value of the facet is "Never" it specifies that the value of the slot S 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 at irregular intervals, that is the attribute's value can change more than once, for example people can change job more than once.

Event: The *event* facet of slot S specifies the conditions under which the values associated with slot S change. Values of this facet are the quadruples $(((E_j, S_j, V_j), R_j)|j=1, \dots, m)$ 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.

If the class describes a role, that is the facet *class type* is associated with the value "role", then the facet *event* defines the conditions regulating the acquisition and the relinquishment of a role.

Documentation: This is not strictly speaking a facet of slot S, but a string that is added to document the choices made by the knowledge engineers while associating values with the facets describing the slots. It should give an account of information such as why the ranking has been set to a specific value or what is the context associated with a prototype (see below the discussion concerning prototypes). It is added to keep track of the process leading to the modelling decisions.

4. Relating the extended knowledge model to the motivations

The knowledge model presented in the previous section is motivated by 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.

Properties can be distinguished into *instance properties, class properties* and *meta-properties*. *Instance properties* are those exhibited by all the instances of a concept[*]. They might specialise *class properties*, which instead describe properties holding for the class. *Meta-properties* have been mainly described in philosophy, such as *identity, unity, rigidity* and

---

[*] A subclass of class is considered an instance of that class; we make no distinction here between individuals and subclasses.

dependency. The proposed model permits the characterisation of all these types of properties, thus also includes in the concept description means to derive the meta-properties, which are the basis for the ontological analysis illustrated in [10]. Knowledge engineers making use of the proposed knowledge model are invited to provide more details concerning the concepts with respect to the ones they would have if they were using a traditional OKBC-like knowledge model; they are thus guided in performing 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 [22], it can never be fully specified. Many assumptions about meaning and context are usually made when dealing with real situations [21]. 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 of the 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 measure better 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 *fluents* over time, where the term *fluent* is borrowed from situation calculus to denote a property of the world that can change over time. Modelling the behaviour of fluents over time corresponds to model 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 [10].

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* [13]. 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.

Changes of properties can be modelled as *processes* [24]. Processes can be described in terms of their starting and ending points and of the changes that happen in between. We can distinguish between *continuous* and *discrete change*, the former describing incremental changes that take place continuously while the latter describe changes occurring in discrete steps called *events*. Analogously we can define *continuous properties* as those changing regularly over time, such as the age of a person, versus *discrete properties* which are characterised by an event which causes the property to change. If the value associated with change frequency is *Regular* then the process is continuous, if it is *Volatile* the process is discrete and if it is *Once only* the process is considered discrete and the triggering event is set equal to *time-point=T*.

Any regular period 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 $(((E_j, S_j, V_j), R_j)|j=1, \ldots, m)$ 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 E_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 change*, which is based on the fundamental notion that an individual may remain the same while showing different properties at different times [12]. 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 [8] as:

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

where with *essential property* we mean a property holding for an individual in every possible circumstance in which the individual exists.

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.

Here we specifically concentrate on one of these systems of possible worlds, namely that which arises from consideration of time. By characterising the rigidity of a property in this specific world we aim to provide knowledge engineers the means to reach a better understanding on the *necessary* and *sufficient* conditions for the class membership. However, this does not mean that the rigidity of a property depends on any account on whether the property is used to determine class membership or not.

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 constitutes a *necessary truth* and to distinguish necessity from possibility [14].

The term can be extended to measure qualitatively the way in which a statement is true by trying to estimate the number of possible worlds 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 *rankings* as defined by Goldszdmidt and Pearl [4]:

“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 "Birds 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 important in order to find which facts are true in every possible world and therefore constitute *necessary truths*. The concept of necessary truth brings us back to the discussion about *rigidity*, in fact 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 property. In particular, we can exactly determine rigidity in a sub-set of all possible worlds. Indeed, since an ontology defines a vocabulary, we can restrict ourselves to the set of possible worlds which is defined as the set of maximum descriptions obtainable using the vocabulary defined by the ontology [19]. Then, under the assumption of restricting the discourse to this set of possible worlds, *rigid properties* are those whose Modality facet is equal to *All*. We can also add time to the set of possible worlds, and in this case a rigid property is one 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 2.3 inconsistencies can arise if a concept 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 [4]: 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, but, it reflects the common sense assumption that, when no specific information is known, people assume that the most likely property (relative to their experience) holds for a concept.

4.3 Rigidity, roles and the additional facets

Establishing whether rigidity holds for a property is not only central in order to distinguish necessary truth but also to recognise *roles* from concepts. The notion of *role* is as central to any modelling activity as those of *objects* and *relations*. A thorough discussion of roles goes beyond the scope of this paper, and roles are not supported yet in the knowledge model presented above, but the extended semantics provided by the knowledge model presented above gives good prospects for supporting roles.

A definition of role that makes use of the formal meta-properties which includes the definition given by Sowa [21] is provided by Guarino and Welty. In [9] they define a role as:

> "properties expressing the *part played* by one entity in an event, often exemplifying a particular relationship between two or more entities. All roles are *anti-rigid* and *dependent*... A property ϕ is said to be anti-rigid if it is not essential to *all* its instances, i.e. $\forall x \; \phi(x) \rightarrow \neg \phi(x)$... A property φ is *(externally) dependent* on a property φ if, for all its instances x, necessarily some instance of φ must exist, which is not a part nor a constituent of x, i.e. $\forall x \; (\phi(x) \rightarrow \exists y \; \varphi(y) \wedge \neg P(y,x) \wedge \neg C(y,x))$."

In other words a concept is a role if its individuals stand in relation to other individuals, and they can enter or leave the extent of the concept without losing their identity. From this definition it emerges that the ability of recognising whether rigidity holds for some property ϕ is essential in order to distinguish whether ϕ is a role.

In [25] the author compares the different characteristics that have been associated with roles in the literature. From this comparison it emerges that the notion of role is inherently temporal, indeed roles are acquired and relinquished in dependence either of time or of a specific event. For example the object *person* acquires the role *teenager* if the person is between 11 and 18 years old, whereas a person becomes *student* when they enrol for a degree course. Moreover, from the list of features in [25] it emerges that many of the characteristics of roles are time or event related, such as: an object may acquire and abandon roles dynamically, may play different roles simultaneously, or may play the same role several times, simultaneously, and the sequence in which roles may be acquired and relinquished can be subject to restrictions.

For the aforementioned reasons ways of representing roles must be supported by some kind of explicit representation of time and event. We believe that the knowledge model we have presented, although it does not encompass roles yet, provides sufficient semantics to model the dynamic features of roles, thanks to the explicit representation of time intervals which is used to model the behaviour of attributes over time. Furthermore, the ability of modelling events, used to describe the possible causes in the state of an attribute, can be used to model the events that constrain the acquisition or the relinquishment of a role.

The ability to distinguish roles gives also a deeper understanding of the possible contexts in which a concept can be used. Recognising a role can be equivalent to defining a context, and the notion of context is the basis on which prototypes and exceptions are defined.

4.4 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 as well.

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 [20]. Prototypes are the subconcepts which best represent a category, while exceptions are those which are considered exceptional although still belonging 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 that 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 *lion* if the assumed context is *animals that can play the role of circus animals*. In the knowledge model presented above we explicitly describe the context in natural language in the Documentation facet, but, the context can be also described by the roles that the concept which is being described is able to play.

Ontologies typically presuppose some particular 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 which properties, although they 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. We assume to have created a class, Range, which is a range of pairs [(min s, min d) – (max s, max d)].

```
c: Circulatory system;
       s: Blood pressure ;
              f: Value-Type : Range;
              f: Value : [(90,60)-(130,85)];
              f: Slot-Value-Label : 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 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 is the compatible with the formal ontological analysis by Guarino and Welty [10] which permits us 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.

The knowledge model we propose 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.

A possible drawback of this approach is the number of facets that need to be filled when building an ontology. We realise that this can make the process of 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.

7. Acknowledgements

The PhD research presented in this paper was funded by BT plc. This paper is supported by HP Labs. The authors are grateful to Ray Paton for providing the example and to Floriana Grasso, Floriana Esposito and Donato Malerba for their invaluable contributions on the paper. The authors are also very grateful to the anonymous referees for their comments on an earlier version of this paper.

8. References

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

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

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

[4] Goldszmidt, M., and Pearl, J. Qualitative probabilities for default reasoning, belief revision, and causal modelling. *Artificial Intelligence,* 84, 1/2, 57-112, 1996.

[5] Gómez-Pérez, A. Knowledge sharing and reuse, Liebowitz, J. (ed.) *The handbook of Applied Expert Systems*, CRC press LCC, 1998.

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

[7] Guarino, N. Formal ontologies and information systems. Guarino, N. (ed.). *Proceedings of FOIS'98*, IOS Press, 1998.

[8] Guarino, N., Carrara, M., and Giaretta, P. An ontology of meta-level categories. *Principles of knowledge representation and reasoning: Proceedings of the fourth international conference (KR 94),* Morgan-Kaufman, 1994.

[9] Guarino, N., and Welty, C. A formal ontology of properties. Dieng, R. (ed.). *Proceedings of the 12th EKAW conference*, LNAI 1937, 2000.

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

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

[12] Kant, I. *Critique of Pure Reason*. trans. Norman Kemp Smith, St. Martin's Press, New York, 1965.

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

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

[15] Lassila, O., and McGuinness, D. The Role of Frame-Based Representation on the Semantic Web. *ETAI Semantic Web Journal*, To appear, 2001.

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

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

[18] Pinto, H., Gómez-Pérez, A., and Martins, J. Some issues on ontology integration. Gómez-Pérez, A., and Benjamins, V.R. (eds.). *Proceedings of the IJCAI'99 Workshop on Ontologies and problem Solving Methods: Lessons learned and Future Trends*, 18, 7.1-7.11, CEUR Publications, 1999.

[19] Plantiga, A. *The Nature of Necessity*. Clarendon Library of Logic and Philosophy, Clarendon Press, 1989.

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

[21] Rosch, E. Reclaiming concepts. *Journal of Consciousness Studies*, 6, 11/12, 61-77, 1999.

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

[23] Sowa, J. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, 1984.

[24] Sowa, J. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Brooks-Cole Publishing Co, 2000.

[25] Steimann, F. On the representation of roles in object-oriented and conceptual modelling. *Data and Knowledge Engineering*, 35, 83-106, 2000.

[26] Studer, R., Benjamins, V.R., and Fensel, D. Knowledge engineering, principles and methods. *Data and Knowledge Engineering*, 25,1/2, 161-197, 1998.

[27] Tamma, V., and Bench-Capon, T. Supporting inheritance mechanisms in ontology representation. Dieng, R. (ed.). *Proceedings of the 12th EKAW conference*, LNAI 1937, 2000.

[28] Tamma, V., and Bench-Capon, T. Characterising concept's properties in ontologies. To appear in *Esposito, F. (Ed.), Proceedings of the "7° Congresso AI*IA"*, Springer-Verlag, LNAI 2175, 2001.

[29] Visser, P., Jones, D., Bench-Capon, T., and Shave, M., Assessing heterogeneity by classifying ontology mismatches. Guarino, N. (ed.). *Proceedings of FOIS'98*, IOS Press, 1998.