A conceptual model to facilitate knowledge sharing in multi-agent systems

Valentina Tamma & Trevor Bench-Capon
Agent ART group,
Department of Computer Science
Chadwick Building, Peach Street
Liverpool, L69 7ZF, UK
{valli, tbc}@csc.liv.ac.uk

ABSTRACT

This paper presents and motivates an extended ontology knowledge model which represents semantic information about concepts explicitly. This knowledge model results from enriching the standard 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 knowledge engineers 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.

This enriched semantics can prove useful to describe what is known by agents in a multi-agent systems, as it facilitates the use of reasoning mechanisms on the knowledge that instantiate the ontology. These mechanisms can be used to solve ambiguities that can arise when heterogeneous agents have to interoperate in order to perform a task.

1. INTRODUCTION

Advances in the Internet have made it possible to access huge amounts of diverse information from different places all over the world. This possibility has stimulated a growing demand for understanding how to integrate multiple and heterogeneous knowledge sources in order to provide added value. The complexity of this task is quite high, chiefly because of the heterogeneity of the knowledge sources and, to a limited extent, of their size.

One knowledge engineering paradigm that has proved to be useful for dealing with the integration of heterogeneous knowledge is based on a multi-agent system architecture, where human and software agents interoperate and so cooperate within common application areas. Agents in a multi-agent system are characterised by abstraction, interoperability, modularity and dynamism. These qualities are particularly useful in that they can help to promote open systems which are typically dynamic, unpredictable and highly heterogeneous [14], as is the Internet. In these types of application domains, the interoperability offered by the multi-agent system approach is required because the individual components that interact with agents are not known a priori. Additionally, this paradigm provides robustness and flexibility of the interfaces between both the agents that exist within the Internet and between agents and software systems, this is essential since the interfaces cannot be anticipated at design time.

Within a multi-agent system, agents are characterised by different "views of the world" that are explicitly defined by *ontologies*, that is views of what the agent knows to be the concepts describing application domain which is associated with the agent together with their relationships and constraints [3]. The interoperability typical of multi-agent systems is achieved through the reconciliation of these views of the world by a commitment to common ontologies that permit agents to interoperate and cooperate while maintaining their autonomy.

In open systems, agents are associated with knowledge sources which are diverse in nature and have been developed for different purposes. Knowledge sources embedded in a dynamic environment can join and leave the system at any time. From the ontologies perspective dealing with open systems implies that ontologies are often the efforts of many domain experts and are designed and maintained independently in distributed environments. In such a situation interoperation between agents is based on the reconciliation of their heterogeneous views, which is accomplished by merging or integrating the diverse ontologies associated with the agents composing the system [27]. The merging and integration of diverse ontologies has to be accomplished bearing in mind that since agents are highly heterogeneous, they are likely to be incapable to fully understand each other, therefore both syntactic and semantic inconsistencies can arise and thus need to be reconciled.

Agent's ability to represent domain knowledge in a consistent manner has to be complemented by some reasoning capability. According to Wooldridge and Jennings, [31] an agent architecture is *one that contains an explicitly represented, symbolic model of the world. and in which decisions (for example about what action to perform) are made via logical (or at least pseudo-logical) reasoning, based on pattern matching and symbolic manipulation.* Therefore ontologies in multi-agent systems require a high degree of expressive power to support the application of reasoning techniques that result in sophisticated inferences such as those used in negotiation, which is motivated by the requirement for agents to solve problems arising from their interdependence upon one another. [19]

Designing multi-agent systems to deal with the sharing of heterogeneous knowledge sources gives rise to the requirement for ontologies that can be easily integrated and provide a base for applying reasoning mechanisms, highlighting 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 [4, 16, 28].

This paper presents and motivates a knowledge model for ontologies which extends the usual set of facets in the OKBC frame-base model [2] to encompass more semantic information concerning the concept, to give of a precise characterisation of the concept's properties and expected ambiguities: these include 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 presents the motivations for adding semantics to the conceptual model, section 3 presents the enriched knowledge model while in section 4 the model is discussed with respect to the motivations. Section 5 discusses the representation of roles using the knowledge model and section 6 provides an example of concept description using the knowledge model. Finally, in section 7 conclusions are drawn and future research directions are illustrated in section 8.

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"* [8]. 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. [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. It involves knowing what can be sensibly 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* [20] 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.

From the multi-agent system perspective, we wish to provide a better characterisation, and thus understanding of the concepts that are known to an agent. Understanding which concepts are associated with an agent and the properties holding for each concept becomes extremely important when agents need to agree on one or more common shared ontologies, where each shared concept is obtained as reconciliation of the local views. Describing concepts by characterising the behaviour of their properties allow inconsistencies while integrating and reasoning that have to be dealt with, as illustrated is the next two subsections.

2.2 Integrating diverse ontologies

The second argument concerns the integration of the diverse agent views, which is accomplished by integrating the ontologies associated with the agents.

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. That is, let us suppose that the concept C is present in n ontologies $O_1, O_2, \cdots, O_n$, although described by different properties. That is each ontology $O_i, i = 1, \cdots, n$ defines a concept $C_i, i = 1, \cdots, n$ such that $C_1 \approx C_2 \approx \cdots \approx C_n$ (where $\approx$ denotes that the concepts are *overlapping*). Each concept $C_i, i = 1, \cdots, n$ is described in terms of a set of properties $P_i^C, i = 1, \cdots, n$. The result of the integration of the n ontologies is another ontology defining the concept $C_{integrated}$ which is defined in terms of $\bigcup_{i=1}^{n} P_i^C$, where all the P_i^C have to be distinguished.

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 focusing on providing engineering principles to build ontologies, for example [6, 7]. Another approach [11, 12] 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* [13].

When the domain knowledge associated with different agents needs to be integrated, inconsistencies can become evident. Many types of ontological inconsistencies have been defined in the literature, for instance in [30] and there are ontology environments currently available that try to deal with these inconsistencies, such as SMART [4] 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. For example, a concept C can be defined in two different ontologies O_1 and O_2 in terms of an attribute A that is specified as tak-

ing values in two different domains D_1 in O_1 and D_2 in O_2, where $D_1 \subseteq D_2$. 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, 28]. 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.
We have already mentioned that one of the important problems to be solved when building agents is the *representation/reasoning problem*, [31] that is:

> how to symbolically represent information about real world entities and processes, and how to get agents to reason with this information in time for the result to be useful.

From the ontology perspective the reasoning aspect of the *representation/reasoning problem* involves the ability of reasoning with the knowledge obtained by integrating or merging diverse ontologies. Indeed, when 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* [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. Furthermore, if the ontologies one wants to reason with have been developed at different times and for diverse purposes, it is likely that problem of *implicit inconsistencies* will arise. This kind of problem is quite similar to the semantic inconsistencies that have been defined in section 2.2. Such a problem has been first identified in the inheritance literature [18] where Morgenstern distinguishes *explicit* from the *implicit inconsistencies* ones. Explicit inconsistencies arise when two concepts C_i and C_j are described in terms of explicitly conflicting properties, that is in terms of the same attribute which is associated with conflicting values V and $\neg V$. Implicit inconsistencies arise when the properties are described by different attributes but with opposite meanings. Morgenstern [18] has modified the (notorious) Touretzky's Nixon diamond [29] to show an example of implicit inconsistencies. Let us consider:

- `Nixon`$\rightarrow$ `Republican` ;

- `Nixon`$\rightarrow$ `Quaker` ;

- `Quaker`$\rightarrow$ `Pacifist` ;

- `Republican`$\rightarrow$ `Hawk` ;

The two concepts `Quaker` and `Republican` are described by two attributes *Pacifist* and *Hawk* that have different names but are semantically related (one is the opposite of the other), as they both describe someone's attitude towards going to war. 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 always be made automatically and might need to be validated by the system user or by some other agent.

3. EXTENDED KNOWLEDGE MODEL

In this section we extend the OKBC knowledge model [**?**]. This 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 of 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.
In the following small example, that will be used throughout the paper to illustrate the knowledge model here provided, we start by describing a concept using the basic information provided by a frame-based knowledge model. 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*.
Classes are denoted by the label **c**, slots by the label **s** and facets by the label **f**. We could describe the concept as:

c: `Circulatorysystem;`
  s: `Bloodpressure`
    f: `Domain`: [(0,0)-(300,200)];
    f: `Value`: [(90,60)-(130,85)];

where, for example, the value [(90,60)-(130,85)] means that usually the minimum systolic pressure is 90 and the minimum diastolic pressure is 60 while the maximum systolic pressure is 130 and the maximum diastolic pressure is 85.
In the extended knowledge model that we propose 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:

- **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 *Prototypical* if the value is true for any prototypical instance or 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 those which differ from the prototypical ones but 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 there are many types of birds that do not fly. Associating a degree of confidence with a pair *(Attribute, Value)* is also an arbitrary process that depends on the way in which the knowledge engineers writing the ontology perceive the domain. By giving 7 possibilities to fill the slot ranking we aim to provide knowledge engineers with the possibility to express with more detail their perception of the domain;

- **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 process is continuous (see section below), 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. If the slot is set equal to *Never* it means that the value associated with the attribute cannot change, and finally *Volatile* indicates that the attribute's value can change more than once, for example people a person's blood pressure can change several times, both because of the aging process and because of specific events such as chock or diseases;

- **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;

- **Documentation**: This is not strictly speaking a facet, but a string that is add to document the choices made by the knowledge engineers while filling 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 the problems described in section 3. 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 [12]. Such an enriched model helps to characterise and identify the meta properties holding for the concepts, thus providing knowledge engineers developing the ontologies 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 [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 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 5.

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 [12]. Describing the behaviour over time involves also distinguishing proper-

ties 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* [15]. 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 changes*, 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* 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 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 previous 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 [11]. 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 [10] 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 can be seen as 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 [5]:

> 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 credibility. 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 credibility 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), but, we 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 credibility of a property in a concept description is also related to the problem of enabling agents to reasoning with ontologies obtained through 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 credibility can be used to rank the possible alternatives following an approach similar to the non-monotonic reasoning one developed by [5]: in case of more conflicting properties holding for a concept description, properties are ordered according to the degree of credibility, that is according to the the filler associated with the *Ranking* facet weighted by the *Degree of strength*.

Therefore, a property holding for all the subclasses is considered to have a higher rank than one holding for few of the concept subclasses, but this ordering is adjusted by the relevance, as perceived by the knowledge engineer, of the property in the concept's description (*Degree of strength*). For example, to reason about birds ability to fly, the attribute *species* is more relevant than the attribute *feather colour*. When reasoning with diverse ontologies, the *Degree of strength* represents the weight associated with the inheritance rule corresponding to the attribute.
Although this ordering of the conflicting properties needs to be validated by the user, it reflects the common sense assumption that, when no specific information is known, people assume that the most likely property holds for a concept. Here we assume that the agents reflect the common sense reasoning that is typically human.

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 *Ranking*. In this view all cognitive categories show gradients of membership which describe how well a particular subclass fits the standard idea or image of the category to which the subclass belongs [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 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 for multi-agent systems, 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. PROSPECTS FOR SUPPORTING ROLES

The notion of *role* is central to any modelling activities as much 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 introduced in section 3. However, the extended semantics provided by the knowledge model presented above gives good prospects for supporting roles. In this section we provide some preliminary consideration and relate the additional facets with the main features of the role notion.
Despite its importance, highlighted in the literature [9, 23], few modelling languages permit the distinction between a *concept* and the *roles* it can play in the knowledge model. This difficulty is partially due to the lack of a single definition for *role*.
A definition of role that makes use of the formal meta-properties and includes also the definition given by Sowa [23] is provided by Guarino and Welty. In [11] 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. $\forall x \phi(x) \rightarrow \neg\Box\phi(x)$... A property ϕ is *(externally) dependent* on a property ψ if, for all its instances x, necessarily some instance of ψ must exist, which is not a part nor a constituent of x, i.e. $\forall x \Box(\phi(x) \rightarrow \exists y \psi(y) \wedge \neg P(y,x) \wedge \neg C(y,x))$.

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] Steimann presents a list of the features that have been associated in the literature with roles. Some of these features are conflicting and, as pointed out, no integrating definition has been made available. However, from the different definitions available, it can be derived 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 19 years old, whereas a person becomes *student* when they enroll 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 time, simultaneously, and the sequence in which roles may be acquired and relinquished can be subjected to restrictions.

For the aforementioned reasons ways of representing roles must be supported by some kind of time and event explicit representation. 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 attributes behaviour 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.

6. A MODELLING EXAMPLE

We are now ready to complete the example by modelling the concept *blood pressure* with the enriched knowledge model presented above. 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: `Circulatorysystem;`
  s: `Bloodpressure`
    f: `Domain`: [(0,0)-(300,200)];
    f: `Value`: [(90,60)-(130,85)];
    f: `Typeofvalue`: prototypical;
    f: `Exceptions`: [(0,0)-(89,59)] ∪ [(131,86)-(300,200)];
    f: `Ranking`: 3;
    f: `Changefrequency`: 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);

7. CONCLUSIONS

This paper has presented an ontology model that supports knowledge sharing in multi-agent systems where the agents can be heterogeneous. The proposed model 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 can arise in heterogeneous multi-agent systems and that can hinder the communication between agents and we have shown that these problems require additional semantics in order to be dealt with.

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 [12] 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.

8. FUTURE WORK

The extension of the knowledge model with with additional semantics opens several new research directions. Firstly, the role representation needs to be formalised in the knowledge model in order to represent also the roles hierarchical organisation [25].

We also plan to use the semantics encompassed in the knowledge model to assist knowledge engineers in the tasks of merging and reasoning with diverse ontologies. To reach this goal we intent to introduce some form of temporal reasoning based on the event logics that is used extend the facets.

The description of attributes in terms of prototypical values gives us the possibility of exploring the application of machine learning techniques to dynamically extend ontologies.

Acknowledgement

The PhD research presented in this paper was funded by BT plc. The authors are grateful to Ray Paton for providing the example. This paper is supported by HP.

9. REFERENCES

[1] 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.

[2] V. Chaudhri, A. Farquhar, R. Fikes, P. Karp, and J. 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.

[3] S. Falasconi, G. Lanzola, and M. Stefanelli. Using ontologies in multi-agent systems. In *Proceedings of Tenth Knowledge Acquisition for Knowledge-Based Systems Workshop (KAW)*, 1996.

[4] N. Fridman Noy and M. 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.

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

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

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

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

[9] N. Guarino. Concepts, attributes and aritrary relations. *Data and Knowledge Engineering*, 8:249–261, 1992.

[10] 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.

[11] 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.

[12] 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)*, Amsterdam, 2000. IOS Press.

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

[14] N. Jennings. Agent software. In *Proc. UNICOM Seminar on Agent Software*, pages 12–27, 1995.

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

[16] D. 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.

[17] D. McGuinness, R. 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.

[18] L. Morgenstern. Inheritance comes of age: Applying nonmonotonic techniques to problems in industry. *Artificial Intelligence*, 103:1–34, 1998.

[19] S. Parsons, C. Sierra, and N. Jennings. Agent that reason and negotiate by arguing. *Journal of Logic and Computation*, 8(3):261–292, 1998.

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

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

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

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

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

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

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

[27] K. Sycara, M. Klusch, S. Widoff, and J. Lu. Dynamic service matchmaking among agents in open information systems. *ACM SIGMOD Record. Special Issue on semantic interoperability in global information systems*, 1998.

[28] V. Tamma and T. 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.

[29] D. Touretzky. *The Mayhematics of Inheritance Systems*. Morgan Kaufmann, 1986.

[30] P. Visser, D. Jones, T. Bench-Capon, and M. 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.

[31] M. Wooldridge and N. Jennings. Intelligent agent: Theory and practice. *Knowledge engineering review*, 1995.