Attribute meta-properties for knowledge sharing

Valentina Tamma and **Trevor J.M. Bench Capon** [1]

Abstract. Formal ontological analysis is a methodology that builds on some philosophical notions in order to guide the process of building ontologies whose structure is correct and little or no tangled.

This paper presents an ontology model that facilitates formal ontological analysis, by providing a set of *metaproperties* which characterise the behaviour of concept properties in a concept definition, while providing a richer semantics of the concept. We describe concepts in terms of their attributes (characterising features) and we also describe the role played by these features in the concept definition, whether they are prototypical or exceptional, whether they are permitted to change over time, and if so, how often this happens, how likely is a concept to show these features, etc. We show that these metaproperties can support a methodology, OntoClean [44] that uses formal ontological analysis to build cleaner taxonomies (which are thus more sharable). The set of metaproperties for attributes we propose can be used to guide in determining which metaproperties for concepts hold for an ontology and therefore can support the use OntoClean.

1 Introduction

Many current applications such as e-commerce or the semantic web rely on the ability of different resources or agents to interoperate with each others and with users. In some cases, interoperation becomes more complex, because agents may have been independently developed, therefore the assumption that agents use the same communication language and the same terminology in a consistent way cannot be made. When dealing with independently developed agents, their interoperability with humans and others depends on the agents' ability to understand them, which leads us directly to ontologies. Ontologies are an explicit, formal specification of a shared conceptualisation, where a 'conceptualisation' refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon, 'explicit' means that the type of concepts used, and the constraints on their use are explicitly defined, 'formal' refers to the fact that the ontology should be machine-readable, and lastly 'shared' reflects the notion that an ontology captures consensual knowledge, that is it is not private to some individual, but accepted by a group [37]. That is ontologies provide a formally defined specification of the meaning of those terms that are used by agents during the interoperation.

Agents can differ in their understanding of the world surrounding them, in their goals, and their capabilities, but they can still interoperate in order to perform a task. The interoperation among agents is the result of reaching an agreement on a shared understanding, mainly obtained by the reconciliation of the differences. This kind of reconciliation might be accomplished by *merging* the ontologies to which the agents involved in the interoperation refer to, that is, by building a single ontology that is the merged version of different agent's ontologies, which often cover similar or overlapping domains [8].

Ontology merging starts with the attempt to find the places in which the source ontologies overlap [24], that is the coalescence of two semantically identical terms in different ontologies so that they can be referred to by the same name in the resulting ontology. This is the only step of the merge process which is relevant to the scope of this article. The coalescence of terms in diverse ontologies has to be accomplished bearing in mind that agent's ontologies might be heterogeneous, and any kind of heterogeneity has to be reconciled in order to share knowledge. Heterogeneity is out of the scope of this article, however we recognise that it can hinder attempts to coalesce terms, especially when it concerns semantics. Ontology or semantic heterogeneity occurs when different ontological assumptions about overlapping domains are made [43].

Any consideration on ontology heterogeneity it is usually done assuming that the ontologies involved in the merging process are either built according to some kind of engineering methodology, such as Methontology [6], or ontology taxonomic structures are validated according to some methodologies such as OntoClean [44]. Both methodologies are aimed to insure that the ontology obtained after applying them is correct, that it does not contain cycles or recursive definitions, and it has a taxonomic structure that is no or little tangled.

Methontology and OntoClean are complementary methodologies in that Methontology provides the guidelines for building or re engineering ontologies, whereas OntoClean can be used either in the validation step (when ontologies are engineered or restructured) or simultaneously with the ontology construction (when ontologies are built from scratch). These two methodologies are currently undergoing an integration process [5] as part of the activities of the OntoWeb special interest group on Enterprise-standards Ontology Environments (SIG's home page: `http://delicias.dia.fi.upm.es/ontoweb/sig-tools/index.html`).

Methodologies to obtain well-built ontologies, however, are not enough to support the semi-automatic coalescence process. In fact in order to recognise whether two concepts (that can be affected by heterogeneity) are similar, we cannot only rely on the the terms denoting them, on the relationships with other terms, and on their descriptions, but we need to have a full understanding of the concepts. As noted by McGuinness [23], an explicit representation of the semantics of terms would be useful to understand whether two concepts are similar. It emerges that the current ontology models are not expressive enough to provide such an explicit representation of the semantics. Even when heavyweight ontologies are considered (that is, concepts described in terms of attributes, linked by relations,

---

[1] Department of Computer Science, University of Liverpool, Chadwick Building, Liverpool L69 7ZF, UK, email: {valli, tbc}@csc.liv.ac.uk

organised into an Is-a relationship and constrained by axioms) their expressiveness does not allow a full account of the semantics of the concepts described.

This paper is organised as follows: Section 2 presents the OntoClean methodology and the notions of formal ontological analysis, while Section 3 introduces our proposal for an ontology model encompassing a set of metaproperties for attributes which are discussed in the following subsections. This ontology model was also presented in [39], in this paper we do not discuss any implementation issues and we concentrate on the metaproperties, clarifying the relationship with the concept metaproperties used in OntoClean and the role attribute's metaproperties play in associating senses to concepts. Section 4 discusses the metaproperties and relates them with two notions (identity and rigidity) of formal ontological analysis and with roles. Then we proceed by presenting in Section 5 and subsections a novel approach to knowledge sharing that we are currently investigating and which motivated the ontology model presented in Section 3. This approach, called *ontology clustering*, is thought of being more suited to open evironments in which agents interoperate with each others. We Finally, Section 6 draws conclusions and in Section 7 we describe future work.

2 The philosophical notions of Identity, Unity, Essence, and Dependence

OntoClean [44] is a methodology to perform a *formal ontological analysis* on taxonomies in order to to verify which formal metaproperties hold, thus making clear and explicit the modelling assumptions made while designing the ontologies. The clarification and explication of the modelling assumptions is a necessary step to perform in order to evaluate ontologies, it permits knowledge engineers to detect and reconcile ontological conflicts that may affect one or more ontologies. Ontological conflicts may become apparent when two ontologies are compared in order to coalesce term, and they reveal cases of ontological heterogeneity. For example two well known ontologies, present the following conflict: one models Physical Object as subconcept of Amount of matter wheres the other models Amount of matter as subconcept of Physical object, this is a case of ontology heterogeneity due to different modellings of the concepts. Ontologial conflicts need to be detected and resolved if terms are to be coalesced.

OntoClean is strongly based on the philosophical notions of *identity, unity, essence (rigidity),* and *dependence*. The attribute metaproperties we present in this paper are related to these notions, and we discuss them below.

Identity: Identity is the logical relation of numerical sameness, in which a thing stands only to itself. Based on the idea that everything is what it is and not anything else, philosophy has tried for a long time to identify the criteria which allow a thing to be identified for what it is even when it is cognised in two different forms, by two different descriptions and/or at two different times [45, 15]. This comprises both aspects of finding constitutive criteria (which features a thing must have in order to be what it is), and of finding re-identification criteria (which feature a thing has to have in order to be recognised as such by a cognitive agent). These are distinct, although equally important aspects of identity.In fact, while identity is not affected by the context and is based on the the intrinsic features of an object, whereas re-identification is affected by context and it is based on features that are external to the object. For example, an identity criterion for people is to have matching fingerprints, so two people are the same if they have the same fingerprints. Fingerprints are intrinsic to the individual, they are not assigned by an external agent. A re-identification criterion might depend on the role played by the object: one can be a student and an employee at the same time, and is re-identified as student by the student id, whereas is re-identified as employee by an employee number.

Although the problem of *identifying* what features an entity should have in order to be what it is and recognised as such has been central to philosophy, it did not have the same impact in conceptual modelling and more generally AI. The ability to identify individuals is central to the modelling process, more precisely, it is not the mere problem of identifying an entity in the world that is central to the ontological representation of the world, but the ability to *re-identify an entity in all its possible forms*, or more formally *re-identification in all possible worlds.* [2] That is, the problem is related to distinguishing a specific instance of a concept from its siblings on the basis of certain *characteristic properties* which are unique and intrinsic to *that instance* in its whole. Intrinsic properties correspond to the modelling primitive *attributes*. Extrinsic properties represent relations between classes, thus corresponding to the modelling primitive *relationship*.

This notion is, of course inherently time dependent, since time gives rise to a particular system of possible worlds where it is highly likely that the same instance of a concept exhibits different features [3]. This problem is known as *identity through change*: an instance of a concept may remain the same while exhibiting different properties at different instants of time. Therefore it becomes important to understand which features or properties can change and which cannot [44], and also the situations that can trigger such changes. If we reformulate the identity problem as *re-identification* we realise that re-identification is also affected by time; how can we re-identify the same instance at different instant of times? We face the re-identification problem in everyday life; we are able to recognise the features that permits us to distinguish an instance from the others, and when intrinsic features are not available, we 'attach' artificial features, that permit us to establish identity. One example is the *Student ID*, which is assigned to university students, in order to identify students univocally.

Unity: the notion of *unity* is often included in a more generalised notion of identity, although these two notions are different. While identity aims to characterise what is unique for an entity of the world when considered as a whole, the goal of unity is that of *distinguishing the* parts *of an instance from the rest of the world by means of a* unifying relation *that binds them together (not involving anything else)* [44]. For example, the question 'Is this my car?' represents a problem of identity, whereas the question 'Is the steering wheel part of my car?' is a problem of unity. Also the notion of unity is affected by the notion of time; for example, can the parts of an instance be different at different instants of time?

Essence: The notion of *essence* is strictly related to the notion of *necessity* [16]. An *essential property* is a property that is necessary for an object, that is, a property that is true in every possible world [22]. Based on the notion of *essence*, Guarino and colleagues [14] have introduced the notion of *rigidity*. A rigid property is a

---

[2] Some philosophers, e.g. Lewis [21, page 39 ff], hold that there is no such thing as trans-world identity, although objects in one world can have *counterparts* in other worlds.

[3] Here the counterpart theory does not hold, and so identity through time is always accepted.

property that is necessary to all instances in any instant of time, that is a property ϕ such that: $\Box(\forall x,\ t\phi(x,\ t) \rightarrow \Box\forall t'\ \phi(x,\ t'))$. For this formula, and in the remainder of this paper, we use the modal notions of *necessity* $\Box$ and *possibility* $\diamond$ quantified over possible worlds (in Kripke's semantics [18]), meaning that the extension of predicates concerns what exists in any possible world. We use these operators according to the following meanings: $\Box\ \phi$ means that ϕ holds in *all* possible worlds $\diamond\ \phi$ means that ϕ is possible, i.e. that ϕ holds in *at least* one possible world.

Rigidity strictly depends on the notions of *time* and *modality* [38]; this point is further elaborated in Section 4.2. It is important,however, not to confuse modal necessity with temporal permanence. Modal necessity means that the property is true in every possible world. Time is undoubtedly one partition of these worlds, but temporal permanence means that the property is true in that world (time), with no information concerning the other possible worlds, and this might happen by pure chance.

Dependence: In OntoClean [44], the notion of dependence is considered related to concept properties. In this context, dependence permits us to distinguish between *extrinsic* and *intrinsic* properties based on whether they depend on objects other than the one they are ascribed to or not.

In order to establish whether these metaproperties hold, OntoClean is supported by a description logic based system that can help knowledge engineers to assign the metaproperties to concepts and to verify the taxonomic structure on the grounds of the modelling methodology. In this paper we focus our attention on the process of assigning the metaproperties. OntoClean guides knowledge engineers in this process by asking them to answer some questions such as "Does the property carry identity". Knowledge engineers can answer yes, no or unsure, in this latter case more specific questions can be asked, such as "Are instances of the property countable?".

The OntoClean methodology depends on the knowledge engineers understanding of the ontologies to analyse and can thus be problematic if used to evaluate independently designed ontologies. Moreover, OntoClean does not take into account the structure of concept definitions, as it does not consider the characteristic features (or *attributes*) that might have been used to define concepts.

This work proposes an enriched ontology model whose aim is to complement the OntoClean methodology, by providing an additional way to determine metaproperties to concepts. In our proposal we describe concepts in terms of their characterising properties, which are in turn described not only in terms of their structural features (such as range, domain, cardinality etc.), but also in terms of their metaproperties, which describe the contribution given by these properties to the concept definition. We describe the enriched ontology model and the metaproperties for attributes in the next sections.

3 Enriched ontology model

The ontology model we propose comprises *concepts*, *attributes*, *relations*, and *instances*. We do not consider here axioms. Concepts represent the entities of the domain and the tasks we want to model in the ontology. Concepts are described in terms of defining properties, which are represented by associating an *attribute* with either a single value or a set of values. Concepts are organised into an Is-a hierarchy, so that a concept attributes and their values are inherited by subconcepts. Multiple inheritance is permitted, so attributes and their values can be inherited from multiple parents. The values associated with an attribute can be restricted in order to provide a better definition of a concept [19].

Attributes are described in terms of their structural characteristics, such as the concepts that they are defining, their allowed values, the type of the values (string, integer, etc.), and the maximum and minimum values (if attributes are numeric). Attributes are also described in term of the following metaproperties:

- *Attribute's behaviour over time*: The metaproperties *Mutability*, *Mutability Frequency*, *Event Mutability* and *Reversible Mutability* provide a better description of attributes by characterising their behaviour over time, that is, whether they are allowed to change their value during the concept lifetime (*Mutability*) and how often the change occurs *Mutability Frequency*), whether the change is reversible (*Reversible Mutability*), and what triggers change (*Event Mutability*);
- *Modality*: this meta-property is a qualitative description of the degree of inheritability of a concept property by its subconcepts;
- *Prototypes* and *Exceptions*: the metaproperties *Prototypical* and *Exceptional* aim to describe properties that are prototypical for a concept, that is the properties that obtain for the *prototypical* (from a cognitive viewpoint, according to Rosch [30]) instances of a concept. Exceptions are those properties which can be ascribed to a concept although being highly unusual;
- *Inheritance* and *Distinction*: *inherited* metaproperties regard those properties that hold because inherited from an ancestor concept, they may be overruled in the more specific concept in order to accommodate inheritance with exceptions. *Distinguishing* are those properties that permit us to distinguish among siblings of a same concept. In other words a distinguishing property ϕ is a property such that $\diamond\exists x\ \phi(x)\ \wedge\ \diamond\exists x\ \neg\phi(x)$, that is there is possibly something for which the property ϕ holds, and there is possibly something for which the property does not hold, and these are neither tautological nor vacuous [44]. Distinguishing properties might cause disjoint concepts in the ontology's taxonomic structure.

These metaproperties provide means to distinguish between *necessary* and *sufficient* conditions for class membership. Indeed, the modality meta-property and those describing the behaviour over time permit the identification of essential (or rigid) properties and necessary properties are those that are essential to all instances of a concept. Prototypical properties are good candidates to identify sufficient conditions, as discussed in Section 3.3.

Relations between concepts are supported by the model as are instances. Finally, the ontology model supports roles. Concepts are also used to represent *roles*, which can be thought of describing the *part played* by a concept in a context, (a more complete discussion on roles is postponed to Section 4.3). Roles are described in terms of their context, and the formal role relationship holds, that is, roles are related to concepts by a 'Role-of' relations.

This ontology model enriches the traditional model proposed initially by Gruber [12], in that it permits the characterisation of a concept properties. From this viewpoint it should be more expressive. The solution of adding information characterising concept properties is a controversial one. Although we do realise that often it is not true that 'more is better', this work claims that an ontology model which include this type of property's characterisation might be helpful to deal with ontology heterogeneity problems in two ways. On the one hand the model complements the set of formal ontological properties proposed in [44], and can guide in assigning these to concepts in a way which depends on concept definitions in terms of attributes. This might result particularly useful when knowledge engineers need to assign formal properties to ontologies they have not designed.

On the other hand, this conceptual model for ontologies facilitates a better understanding of the concept semantics. Currently ontology merge is performed by hand based on the expertise of the knowledge engineers and on the ontology documentation. Even in this case the ontology model we propose can prove useful by providing a characterisation of the properties, which can help to identify semantically related terms. The following subsections describe all the metaproperties for attributes but Inheritance and Distinction (which are trivial) more in detail:

3.1 Behaviour over time

The metaproperties which model the behaviour of the attributes over time are:

- *Mutability*, which models the liability of a concept property to change, a property is mutable if it can change during the concept lifetime;
- *Mutability Frequency*, which models the frequency with which a property can change in a concept description;
- *Event Mutability*, which models the reasons why a property may change; *Reversible Mutability*, which models reversible changes of the property.

These metaproperties 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 corresponds to modelling the changes in properties that are permitted in a concept description without changing the essence of the concept. Describing the behaviour over time also involves distinguishing properties whose change is *reversible* from those whose change is *irreversible*.

Property changes over time are caused either by the natural passage 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. In [39] we chose *Event Calculus* [17] to accommodate the representation of changes. 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* [35]. Processes can be described in terms of their start and end points and 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* to be 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 a property's mutability frequency is *regular* (that is it changes regularly), then the process is continuous, if it is *volatile* the process is discrete, and if it changes *once only* in the concept lifetime, then the process is considered discrete and the triggering event is set equal to *time-point=T*.

Any regular occurrence over 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 ontology model we present here, continuous properties are thought of as discrete properties where the event triggering the change in property is the passing of time from the instant t to the instant t'. Events are always thought of as *point events*, and we consider *durational events* (events which have a duration) as being a collection of *point events* in which the property whose mutability is modelled by the set of metaproperties hold as long as the event lasts.

3.2 Modality: Weighing the validity of attributes' properties

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 [18]. The term can be extended to qualitatively measure 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 work, by denoting the degree of confidence that we can associate with finding a certain world with the meta-property *modality*. This notion is analogous to the *rankings* defined by Goldszmidt and Pearl [10]: '*Each world is ranked by a non-negative integer κ 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 meta-property is important for reasoning with statements that have different degrees of credibility. Indeed there is a difference in asserting facts such as 'Cats are pets' and 'All felines are pets', 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 truth*.

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 merge. In such a case, mismatches 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. In case of conflict the property's degree of credibility can be used to apply some forms of non monotonic reasoning or belief revision. For example, we could rank the possible alternatives on the grounds of the degree of credibility following an approach similar to the one presented in [10].

3.3 Prototypes, exceptions, and concepts

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 known exceptions. In this way, we partially take the cognitive view of prototypes and graded structures, which is also reflected by the information modelled in the meta-property *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 [30]. 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 the property of giving birth to live young. Prototypes depend on the context (that is on the specific domain that is conceptualised); there is no universal prototype but there are several prototypes depending on the context, therefore a prototype for the category *mammal* could be *cat* if the context taken is that of *animals that can play the role of pets* but it is *lion* if the assumed context is *animals that can play the role of circus animals*. In the ontology model presented above the context can be partially described by the roles applicable to the concept for which prototypical and exceptional properties are modelled. By providing this example we do not mean that any member of the category *animals that can play the role of pets* could be a prototype, but just that prototypes vary if we vary the perspective we are taking on the domain. Therefore there is no unique prototype for the category *animal* but a number of prototypes, depending on how people conceptualise the domain, and this implies also contextual information, for example what is the role played by animals.

Ontologies typically presuppose context and this feature is a major source of difficulty when merging them, since information about context is not always made explicit.

Prototypes are also quite important in that they provide a frame of reference for linguistic quantifiers such as *tall*, *short*, *old*, etc. These quantifiers are usually defined or at least related to the prototypical instance of the concept which is being described, and indeed their definition changes if we change the point of reference.

Therefore including the notions of prototypes and exceptions permits us to provide a frame of reference for defining these qualifiers with respect to *a specific concept*. 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 becomes a natural candidate for a necessary condition. Prototypes, therefore, permit the identification of 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 membership criteria in that it permits us to determine what are the properties that, although rarely holding for that concept, are still possible properties describing the cognitive category.

Prototypes and exceptions can prove useful in dealing with conflicts arising from ontology merging. When no specific information is made available about a concept and it inherits conflicting properties, then we can assume that the prototypical properties hold for it.

4 Discussion

The ontology model presented in previous section could be implemented in any kind of ontology representation formalisms. In [39] we presented an implementation of the ontology model above in a frame-based representation formalism, therefore attributes were described by associating values to slots, and their structural description and metaproperties were modelled by the slot's facets.

By adding the metaproperties to the ontology model, we provide an explicit representation of the attributes' behaviour over time, their prototypicality and exceptionality, and their degree of applicability to subconcepts. This explicit representation may be used to support and complement the OntoClean methodology [44], in that they can help in determining which metaproperties hold for concepts, as we will illustrate in remainder of this section.

Furthermore, the enriched ontology model we propose forces knowledge engineers to make ontological commitments explicit, that is the agreement on the meaning of the terms used to describe a domain [13]. Knowledge sharing is possible only if the ontological commitment of the different agents is made explicit. Real situations are information-rich events, whose context is so rich that, as it has been argued by Searle [32], it can never be fully specified. When dealing with real situations one makes many assumptions about meaning and context [31], and these are rarely formalised. But when dealing with ontologies these assumptions must be formalised 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 subconcepts makes some important assumptions explicit.

The enriched semantics is essential to reconcile cases of ontology heterogeneity. By adding information on the attributes we are also aiming to measure the similarity between concepts more precisely and to disambiguate between concepts that *seem* similar while they are not.

A possible drawback of enriching the ontology model is that knowledge engineers are required a deeper analysis of a domain. We realise that it makes the process of building an ontology even more time consuming but we believe that a more precise ontological characterisation of the domain at least balances the increased complexity of the task. Indeed, in order to include the attribute's metaproperties to the ontology model, knowledge engineers need to have a full understanding not only of the concept they are describing, but also of the context in which the concept is used. Arguably, they need such knowledge if they are to perform the modelling task thoroughly.

The evaluation of the cost to pay for this enriched expressiveness and of the kind of reasoning inferences permitted by this model are strictly dependent on the domain and the task at hand. We can imagine that the automatic coalescence of terms might require more sophisticated inferences whose cost we cannot evaluate *a priori*. In some other cases, the simple matching between properties' charactersiations might help in establishing or ruling out the possiblity of semantic relatedness. For example, two concepts are described by the same properties but with different characterisations, this might indicate that the concepts have been conceptualised differently.

4.1 Identity

The idea of modelling the permitted changes for a property is strictly related to the philosophical notion of *identity*. The metaproperties modelling the behaviour over time are, thus, relevant for establishing the *identity* of concept descriptions [44], since the proposed ontology model addresses the problem of modelling identity when time is involved, namely *identity through change*, which is based on the common sense notion that an individual may remain the same while showing different properties at different times [16]. 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.

Prototypical and exceptional properties and the modality metaproperties describing how the property is inherited in the hierarchy can all contribute to determine what are the necessary and sufficient conditions for class membership. Necessary and sufficient conditions are ultimately the conditions that permit us to define the properties constitutive of identity and to distinguish them from those that permit re-identification.

In order to find suitable identity criteria (which permit to identify a concept), knowledge engineer should look at *essential property*, that is those properties which hold for an individual in every possible circumstance in which the individual exists. It is important to note that essential properties should also be intrinsic if they have to be used to determine identity.

Also inheritance and distinction contribute to identify identity conditions, in that identity conditions have to be looked for among distinguishing properties.

4.2 Rigidity

Identity through change is also relevant to determine *rigidity*. In Section 2 a *rigid property* is defined as *a property that is essential to* all *its instances*.

In [38] we have related the notion of *rigidity* to those of *time* and *modality*; and, by including in our ontology model a meta-property *modality* and that concerning the behaviour over time, we can precisely identify rigidity in the subset of the set of 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 maximal descriptions obtainable using the vocabulary defined by the ontology [26]. By characterising the rigidity of a property in this subset of possible worlds we aim to provide knowledge engineers the means to reach a better understanding of the *necessary* and *sufficient* conditions for the class membership. However, this does not mean that the rigidity of a property depends on any account of whether the property is used to determine class membership or not. That is, the final aim is to try to separate the properties constitutive of identity from those that permit re-identification. Under the assumption of restricting the discourse to this set of possible worlds, *rigid properties* are those properties which are inherited by all subconcepts, and thus which have a certain degree of belief associated with the meta-property *modality* and that cannot change in time.

It is important to note that, although in [39] we have modelled this information as a facet which can take value in the set {*All, Almost all, Most, Possible, A Few, Almost none, None*}, the choice of such a set is totally arbitrary, and it was meant to be such. Knowledge engineers should be able to associate with this meta-property either a probability value, if they know the probability with which the property is inherited by subconcepts, or a degree of belief (such as a κ-value, as in [10], which depends on a ϵ whose value can be changed according to the knowledge available, thus causing the κ function to change), if the probability function is not available.

4.3 Roles dependence on identity and rigidity

Rigidity 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 definition of role that makes use of the formal metaproperties and includes also the definition given by Sowa [34] is provided by Guarino and Welty. In [44] they define a role as: ‘ *the 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.* $\Box(\forall x,\ t\phi(x,\ t) \rightarrow \Diamond\exists\, t'\ \neg\phi(x,t'))$...

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.* $\forall x\Box(\phi(x) \rightarrow \exists y\psi(y) \wedge \neg P(y,x) \wedge \neg C(y,x))$’, where $P(y,x)$ denotes that y is a *part* of x while $C(y,x)$ denotes that y is a *constituent* of 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.

Roles may be ‘naturally’ determined when social context is taken into account, and the social context determines the way in which a role is acquired and relinquished. For example, the role of `President of the country` is relinquished differently depending on the context provided by the country. So, for example, in Italy the role may be acquired and relinquished only once in the lifetime of an individual, whereas if the country is the United Sates, the role can be acquired and relinquished twice, because a president can be re-elected. Social conventions may also determine that once a role is acquired it cannot be relinquished at all. For example, the role `Priest` in a catholic context is relinquished only with the death of the person playing the 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.

In [36] Steimann compares the different characteristics that have been associated in the literature with roles. From this comparison it emerges that the notion of role is inherently temporal, indeed roles are acquired and relinquished dependent on either time or a specific event. For example the object *person* acquires the role *teenager* if the person is between 13 and 19 years old, whereas a person becomes *student* when they enroll for a degree course. Moreover, from the list of features in [36] it derives 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. Indeed, what distinguishes a role from a concept, in the modelling process, is that a role holds during a specific span of time in which some property holds. For example, the role ‘Student’ is applicable only if the property of being registered to a university holds. Therefore, the metaproperties that model the behaviour over time permits the representation of the acquisition and relinquishment of a role.

For the aforementioned reasons, ways of representing roles must be supported by some kind of explicit representation of time and events. Indeed the proposed model provides a way to model roles as fluents; moreover, by modelling the reason for which a property change, we provide knowledge engineers the ability to model the events that constrain the acquisition or the relinquishment of a role.

5 A novel proposal to knowledge sharing

We have illustrated and discussed a ontology model which is enriched with metaproperties providing a better characterisation of attribute. This characterisation is meant to help in disambiguating heterogeneous concepts when merging ontologies, since we assume that *two concepts can be matched if*:

- their description comprises attributes with matching names (synonyms, the name of an attribute is included into the other, etc.);
- candidate matching attributes are described by matching structural definitions (range of the attribute, cardinality, etc.);

- candidate matching attributes show the same behaviour in modelling the concept, that is, the same metaproperties hold for the attributes.

Matching similar concepts plays a pivotal role in those approaches to knowledge sharing which rely on shared ontologies in order to perform the translation between concepts in heterogeneous ontologies. Usually, knowledge sharing is obtained by creating one shared ontologies to which all the agents commit. However, such an approach has been compared to imposing a standard and suffers from the same drawbacks [42]. In this paper we propose a novel architecture to knowledge sharing, which is thought to be more scalable and maintainable, and thus offers more support to the Semantic Web paradigm we have discussed in the Section 1.

In contrast to an approach in which all resources share one body of knowledge here we propose to locate shared knowledge in multiple but smaller shared ontologies. This approach is referred to as ontology-based resource clustering, or shortly, ontology clustering [33]. Resources no longer commit to one comprehensive ontology but they are clustered together on the basis of the similarities they show in the way they conceptualise the common domain. Thus, we have not one, but multiple shared ontologies aggregated into clusters.

Each cluster can be thought of as a micro-theory shared by all the agents that conform to that cluster. Each micro-theory is in turn generalised and they are all eventually generalised by the top-level ontology which is a standard upper ontology like the *Upper-Cyc* [20], so as to obtain a structure that is able to reconcile different types of heterogeneity. We discuss here the feasibility of building such a structure, and in particular, we have investigated the different similarity measures that can be used in order to build clusters of ontologies.

This approach is analogous to *modularisation* in software engineering and is thought of having the same advantages, which are:

- **Modularity/separability**: Each cluster is like a module in software engineering and represents a specific aspect of the domain;
- **Composability**: Different clusters are composed by generalising the concepts that are common to them. This is the first step to permit heterogeneous resources to communicate;
- **Scalability**: The addition of a new resource to the architecture requires only the production of the mapping rules between the ontology associated to the new resource and the cluster to which this resource belongs;
- **Impact of change minimisation**: If a concept description needs to be changed only the mapping rules between the updated ontology and the cluster to which this ontology belongs need to be rewritten;
- **Division of ontology authoring efforts**: Ontologies composing a cluster do not need to be authored by the same people as long as their concepts can be mapped into the concepts of the cluster.
- **Accommodation of diverse formalisations**: A cluster can be comprised of ontologies representing different formalisations of the same domain, such as different temporal ontologies.

This approach has not been tested yet, therefore we can only foresee some disadvantages:

- There is no methodology which permit to build the structure of ontology clusters;
- Complexity of the first order clustering problem from the machine learning viewpoint;
- Lack of semantic-sensitive similarity measure to use to assess the similarity among concepts;
- Lack of tools that can support the building of the ontology clusters.

5.1 Ontology clusters

Ontology clustering is based on the similarities between the concepts known to different resources, where each resource represents a different aspect of the domain knowledge. We assume that the ontologies modelling the resources are consistent, non-redundant, and well structured. We also assume that the ontologies have been built with a methodology including a formal evaluation step, such as Methontology [11]. We also assume that the ontologies are specified by using a language that conforms to the ontology model described above.

Since our resources need to communicate in a sensible fashion they are all supposed to be familiar with some high level concepts. We group these concepts in an ontology rooted at the top of the hierarchy of ontologies. As it describes concepts that are specific to the domain and tasks at hand we refer to this ontology as the application ontology (following Van Heijst and colleagues, [41]. These concepts are reusable within the application but not necessarily outside the application. The concept definitions in the application ontology are chosen from an existing top-level ontology, which in our case is WordNet [25]. The application ontology thus contains a relevant subset of WordNet concepts. For each concept one or more senses are selected, depending on the domain. If some resources share concepts that are not shared by other resources then this leads to the creation of two (or more) sibling ontologies. Each sibling is a consistent extension of its parent ontology, but heterogeneous with respect to its peers. We do not pose any restriction to the types of heterogeneity that can affect the ontologies.

A cluster is referred to as *a group of consistent ontologies (possibly one) in our structure and is described by an ontology which is shared by those composing the cluster*. Both ontology clusters and ontologies within each cluster are organised in a hierarchical fashion where each sibling cluster specialises the concepts that are in its parent cluster. However, while multiple inheritance is permitted within the ontologies, it is not permitted between ontologies, therefore the structure of clusters is a tree. In this structure, the lower level clusters have more precise concept definitions than the higher levels, making the latter more abstract.

Clusters are linked by *restriction or overriding* relations, that is concepts in one parent ontology are inherited by its children cluster, but overriding is permitted [42]. The link between the resources and the local ontologies, on the other hand, is different, and is a *mapping relation* as defined in [42], that is a function preserving the semantics.

Figure 1 illustrates an example of this structure, where Local Ont. are the local ontologies.

Since different siblings can extend their parent cluster concepts in different ways the cluster hierarchy permits the co-existence of heterogeneous (sibling) ontologies. Figure 1 illustrates this particular structure, where $Local\ Ont._1$, $Local\ Ont._2$, $Local\ Ont._3$, and $Local\ Ont._4$ are the local ontologies, $Shared_{12}$ is the ontology shared by the local ontologies 1 and 2. Analogously $Shared_{34}$ is the ontology shared by the local ontologies 3 and 4. $Shared_{1234}$ indicates the ontology shared by the two below that is $Shared_{12}$ and $Shared_{34}$, and in this example is the application ontology itself, here denoted by *Application Ontology*. If some ontologies share concepts that are not shared by other ontologies then there is a reason to create a new cluster. A new ontology cluster here is a child ontology that defines certain new concepts using the concepts

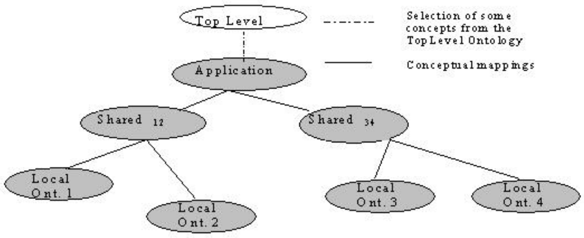

Figure 1. The hierarchy of multiple shared ontologies

already contained in its parent ontology. Ultimately, ontologies are likely to have concepts that are not shared with any other ontology. In our ontology structure, we then create a separate, domain-specific ontology as sub ontology of the cluster in which the ontology resides. We refer to these ontologies as local ontologies. The local ontologies are the leaf nodes of our ontology hierarchy. In each of the ontologies in the structure, concepts are described in terms of attributes and inheritance relations holding in the ontology's structure. Concepts are hierarchically organised and the inheritance (with exceptions) allows the passing down of information through the hierarchy. Multiple inheritance is only permitted within the ontologies.

Concepts are expressed in terms of *inherited* and *distinguishing* attributes. To the set of inherited attributes other attributes are added to distinguish the specific concept from the more general one. These attributes describe the characteristic differences between a concept and its siblings. The distinguishing attributes are used to map concepts from a source ontology into a target ontology preserving the meaning of the concept.

5.2 Towards the semi-automatic construction of ontology clusters

The structure of ontology clusters introduced in Section 5.1 builds on the ability of identifying similar concepts in different ontologies. Identifying which concepts are similar and assessing the degree of semantic similarity between them are, thus, two essential steps in the process of building ontology clusters. However, assessing the similarity between concepts in diverse ontologies is not a trivial task because of the heterogeneity that can affect concepts and their descriptions.

The problem of assessing semantic similarity has received much attention in the artificial intelligence field [27], [3]. In these efforts, 'semantic similarity' refers to a form of semantic relatedness using a network representation. In particular, Rada and colleagues [28] suggest that similarity in semantic networks can be assessed solely on the basis of the IS-A taxonomy, without considering other types of links. One of the easiest way to evaluate semantic similarity in taxonomies is to measure the distance between the nodes corresponding to the items being compared, that is the shorter the path between the nodes, the more similar they are. This way of assessing semantic similarity might be useful for semantic networks, however has the major drawback of computing the semantic distance between concepts which have a common ancestor, and thus it is not suitable for assessing the similarity of heterogeneous local ontologies that have to be clustered. Moreover, this method does not fully exploit the structure of the concept representation, since it does not take into account the concept description in terms of attributes, relationships, etc. thus making it more sensitive to synonym and homonym heterogeneity.

In fact, only few efforts are addressing the problem of facilitating the (semi) automatic reconciliation of different ontologies, and they have been mainly developed for merging different ontologies. Reconciling different ontologies involves finding all the concepts in the ontologies which are similar to one another, determine what the similarities are, and either change the source ontologies to remove the overlaps or record a mapping between the sources for future reference [9]. Similarity in these efforts is mainly lexical and not semantic. Most systems for ontology merging rely on dictionaries to determine synonyms, common substrings in the concept names, and concepts whose documentation share many unusual words. They do not take into account the internal structure of concept representation and the structure of the ontology.

The ontology merging environment Chimaera [24] partially considers the ontology structure in that it assess similarity between concepts also on the grounds of the subclass-superclass relationship and the attributes attached to the concept. Anchor-PROMPT [9] reconciles ontologies by finding *matching terms*, that is, terms from different source ontologies that represent similar concepts. Anchor-PROMPT assess both lexical and semantic matches exploiting the content and structure of the source ontologies (names of classes and slots, subclasses, superclasses domains and ranges of slot values, etc.), and the user's actions in merging the ontologies. However, the method used in Anchor-PROMPT is based on the assumption that if the ontologies to be merged cover the same domain, the terms with the same name are likely to represent the same concepts. Such an assumption is a good rule of thumb, but does not take into account cases of heterogeneity among the source ontologies. In fact, similar concepts might have different names, and be described by attributes with different names. Moreover, the hierarchical structure of the source ontologies might be different, thus a certain subclass-superclass relationship holding in one source ontology might not hold in the others.

The ontology model we have presented has been inspired by a particular approach to assess semantic similarity [29], where the authors propose a method for assessing semantic similarity which takes into account the differences in the level of explicitness and formalisation of the source ontologies specifications. This method does not require an *a priori* shared ontology, and thus makes it suitable for building the ontology clusters. The similarity between concepts in different sources ontologies is assessed by a matching process over synonym sets (thus accounting for lexical similarity), semantic neighborhood, and distinguishing features. The use of distinguishing features to assess similarity enables the authors not only to handle binary similarity measures, typical of lexical similarity (two terms are either similar or not), but also to consider gradients of similarity. This is based on the assumption that, in order for concepts to be considered similar, they should present some common features. By assessing similarity on the grounds of the distinguishing and common features, this method accounts for those problem of synonym terms heterogeneity that can affect both concepts and attributes.

In [29] the authors argue that from an analysis of different feature-based models for semantic similarity has emerged the necessity to account for the context dependence of the relative importance of distinguishing features and asymmetric characteristic of similarity assessments.

The method proposed by Rodríguez and Egenhofer is based on Tversky [40] matching process, which produces a similarity value that depends on both common and different characteristic. In order to take into account common and distinguishing features into the matching process, the usual ontology model is extended to include also an explicit specification of the features. By features the authors collectively mean the set of *functions, parts* and *attributes*. *Functions* represent the intended purpose of the instances of the concept they describe. For example the function of a university is to educate. *Parts* are the structural element of a concept, and they do not necessarily coincide with those expressing the *part-of* relationship, while *attributes* correspond to additional characteristics of a concept that are not considered to be neither parts nor functions.

It could be argued that enriching the concept structure by distinguishing between parts, functions and attributes can give rise to the articulation of new types of mismatches associated with the classifications of features. However, the authors claim that the advantages of enriching the concept structure, namely a matching process that compares corresponding characteristics of concepts, and the ability to distinguish different aspects of the context, modelled by the features, overweights the possible disadvantages deriving from a higher number of mismatches.

We believe that Rodríguez and Egenhofer approach to assess semantic similarity rises an important issue, which is that, in order to be able to have a better assessment of semantic similarity (that gives also gradients of similarity and not only a binary function) it is necessary to provide a richer description of the structure of the concepts in the source ontologies. However, we believe that the distinguishing features proposed in [29] overlap with the semantics already modelled by some relationships, such as *part-of*.

6 Conclusions

Sharing ontologies independently developed is a burning issue that needs to be solved. This paper presents a set of metaproperties describing concept characteristic features (attributes) that can be used to support both the process of building correct ontologies (by complementing and supporting the formal ontological analysis performed by the OntoClean methodology [44]) and the disambiguation of cases of ontology heterogeneity. Formal ontological analysis is usually demanding to perform and we believe that the set of metaproperties for attributes we propose can support knowledge engineers in determining the metaproperties holding for the concepts by forcing them to make the ontological commitments explicit.

The metaproperties we propose, namely Mutability, Mutability Frequency, Reversible Mutability, Event Mutability, Modality, Prototypicality, Exceptionality, Inheritance and Distinction encompass semantic information aiming to characterise the behaviour of properties in the concept description. We have argued that such a precise characterisation might help to disambiguate among concepts that only seem similar, and in turn can support mappings across the structure of multiple shared ontologies that we have devised as alternative to the current approaches to knowledge sharing. We claim that this characterisation of the concept properties is also very important in order to provide a precise specification of the semantics of the concepts. Such characterisation is essential if we want to perform a formal ontological analysis, in which knowledge engineers can precisely determine which formal tools they can use in order to build an ontology which has a taxonomy that is clean and not very tangled. The novelty of this characterisation is that it explicitly represents the behaviour of attributes over time by describing the permitted changes in a property that describe a concept. It also explicitly represents the class membership mechanism by associating with each attribute (represented in a 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. By providing this explicit characterisation, we are asking knowledge engineers to make more hidden assumptions explicit, thus providing a better understanding not only of the domain in general, but also of the role a concept plays in describing a specific domain.

This paper has also presented a structure of multiple shared ontologies for knowledge sharing. Although this is still on going research, we believe that such a structure has advantages over the others especially if considered in the context of an open environment such as the Internet. We believe that this kind of modularisation is the key to applications where intelligent agents (whose knowledge is represented by ontologies) interoperate dynamically, by agreeing on the vocabulary (and shared knowledge) which is closer to the conceptualisations of *only those agents which are involved in the interoparation* and not of all agents that can be potentially involved. We realise that we have not investigated in sufficient detail the issues related to building such structure in an efficient and cost effective manner, and the relationships existing within and between the ontologies composing the structure (both topics are future research directions that we will consider, see next section); but we think that we have laid the basis for future research.

7 Future work

Future research on ontology clusters concerns the relationships between and within ontologies, which need to be clarified with respect to previous work presented in the literature. Two candidate sets of relations have been identified, these are Borst's *ontology projections*: include and extend, include and specialise, include and map [2]; and Visser and Cui's *ontology relations*: subset/superset, extension, restriction, mapping [42]. Another issue emerging from this research is how knowledge sources (or agents), reach consensus on which cluster in the structure of multiple shared ontologies they have to join in order to achieve interoperation. This kind of consensus should be based on suitable similarity measure, that take into account the semantics of the concepts involved, and the semantics of their properties. There are no similarity functions of this type, that we are aware of, and it would be interesting to investigate complex similarity measures, such as those for symbolic objects [4]. We are particularly interested in investigating similarity functions that make use of the extra semantics provided by the conceptual metamodel, in a way analogous to the similarity measure presented in [29]. These kind of similarity functions usually provide a measure of the degree of similarity among different concepts, and not just a binary measure that indicates whether two concepts are similar or not.

From the viewpoint of the ontology conceptual metamodel, future work include understanding the kind of inferences and the reasoning mechanisms that are supported by the additional semantics included in the ontology metamodel. In order to support complex reasoning inferences, we will consider the implementation of the metamodel in some description logic's based language, which should provide the capabilities to perform the inferences. This model is also quite demanding to use, future work should concentrate also on identifying the kinds of applications that can benefit from the expressive power provided by this model.

In order to test the effectiveness of the conceptual metamodel, we are planning to include the metaproperties in tools to build ontologies such as WebOde [1] or Protégé [7].

Acknowledgements

The authors wish to thank Asunción Gómez-Pé. The PhD presented in this paper was funded by BT plc.

REFERENCES

[1] J.C. Arpírez, O. Corcho, M. Fernández-López, and A. Gómez-Pérez, 'WebODE: A scalable workbench for ontological engineering', in *Proceedings of the First International Conference on Knowledge Capture, K-CAP 2001*. ACM-Sigmod, (2001).

[2] P. Borst, *Construction of Engineering ontologies for knowledge sharing and reuse*, Ph.D. dissertation, Centre for Telematica and Information Technology, University of Twente, 1997.

[3] A.M. Collins and E.F. Loftus, 'A spreading-activation theory of semantic processing', *Psychological Review*, **82**, 407–425, (1975).

[4] F. Esposito, D. Malerba, V. Tamma, and H.-H. Bock, 'Classical resemblance measures', in *Analysis of Symbolic data. Exploratory methods for extracting statistical information from complex data*, eds., H.-H. Bock and E. Diday, volume 15 of *Studies in Classification, Data Analysis, and Knowledge Organisation*, 139–152, Springer-Verlag, Berlin, (2000).

[5] M. Fernández-López, A. Gómez-Pérez, and N. Guarino, 'The methontology & ontoClean merge', Technical report, OntoWeb special interest group on Enterprise-standards Ontology Environments, (2001).

[6] M. Fernández-López, A. Gómez-Pérez, A. Pazos-Sierra, and J. Pazos-Sierra, 'Building a chemical ontology using METHONTOLOGY and the ontology design environment', *IEEE Intelligent Systems and their applications*, **January/February**, 37–46, (1999).

[7] N. Fridman Noy, R. W. Fergerson, and M. A. Musen, 'The knowledge model of protege-2000: Combining interoperability and flexibility', in *Proceedings of the 12th EKAW Conference*, ed., R. Dieng, volume LNAI 1937, pp. 17–32, Berlin, (2000). Springer Verlag.

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

[9] N. Fridman Noy and M.A. Musen, 'Anchor-PROMPT: Using non-local context for semantic matching', in *Proceedings of the IJCAI'01 workshop on ontologies and information sharing*, eds., A. Gómez-Pérez, M. Gruninger, H. Stuchenschmidt, and M. Uschold, (2001). http://www.semantic-translation.com/IJCAIwp/.

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

[11] A. Gómez-Pérez, 'Knowledge sharing and reuse', in *The Handbook of Applied Expert Systems*, ed., J. Liebowitz, 10.1–10.36, CRC Press LLC, Boca Raton, FL, (1998).

[12] T. R. Gruber, 'A translation approach to portable ontology specifications', *Knowledge Acquisition*, **5**(2), 199–220, (1993).

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

[14] 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)*, pp. 270–280, San Mateo, CA, (1994). Morgan Kaufmann.

[15] E. Hirsch, *The concept of identity*, Oxford University Press, New York, 1982.

[16] I. Kant, *Critique of pure reason*, St. Martin's press, New York, 1965. Translation by N. Kemp Smith from *Kritik der reinen Vernunft*, 1787.

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

[18] S.A. Kripke, *Naming and necessity*, Harvard University Press, Cambridge, Massachusetts, USA, 1980.

[19] O. Lassila and D. McGuinness, 'The role of frame-based representation on the semantic web', *Electronic Transactions on Artificial Intelligence (ETAI) Journal: area The Semantic Web*, **To appear**, (2001).

[20] D.B. Lenat, 'Cyc: a large-scale investment in knowledge infrastructure', *Communications of the ACM*, **38**(11), 33–38, (November 1995).

[21] D.K. Lewis, *Counterfactuals*, Blackwell, Oxford, 1993.

[22] E.J. Lowe, *Kinds of being. A study of individuation, identity and the logic of sortal terms*, Basil Blackwell, Oxford, UK, 1989.

[23] D.L. McGuinness, 'Conceptual modelling for distributed ontology environments', in *Proceedings of the Eighth International Conference on Conceptual Structures Logical, Linguistic, and Computational Issues (ICCS 2000)*, eds., B. Ganter and G.W. Mineau, volume LNAI 1867, (2000).

[24] D.L. McGuinness, R.E. Fikes, J. Rice, and S. Wilder, 'An environment for merging and testing large ontologies', in *Principles of Knowledge Representation and Reasoning. Proceedings of the seventh international conference (KR'2000)*, eds., A.G. Cohn, F. Giunchiglia, and B. Selman, pp. 483–493, San Francisco, CA, (2000). Morgan Kaufmann.

[25] G.A. Miller, 'Nouns in WordNet: a lexical inheritance system', *International Journal of Lexicography*, **3**(4), 245–264, (1990).

[26] A. Plantiga, *The nature of necessity*, Clarendon Library of logic and philosophy. Clarendon Press, New York, 1989.

[27] M.R. Quillian, 'Semantic memory', in *Semantic Information Processing*, ed., Marvin Minsky, 227–270, MIT Press, Cambridge, Massachusetts, (1968).

[28] R. Rada, H. Mili, E. Bicknell, and M. Blettner, 'Development and application of a metric on semantic nets', *IEEE Transactions on Systems, Man, and Cybernetics*, **19**(1), 17–30, (1989).

[29] M.A. Rodriguez and M.J. Egenhofer, 'Determining semantic similarity among entity classes from different ontologies', *IEEE transactions on knowledge and data engineering*, (2002). in press.

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

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

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

[33] M.J.R. Shave, 'Ontological structures for knowledge sharing', *The new review of information networking*, **3**, 125–133, (1997).

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

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

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

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

[38] V.A.M. Tamma and T.J.M. Bench-Capon, 'An enriched knowledge model for formal ontological analysis', in *Proceedings of the international conference on formal ontology and information systems (FOIS'01)*, eds., C. Welty and B. Smith, New York, (2001). ACM press.

[39] V.A.M. Tamma and T.J.M. Bench-Capon, 'An ontology model to facilitate knowledge sharing in multi-agent systems', *Knowledge Engineering Review*, **To appear**, (2002).

[40] A. Tversky, 'Features of similarity', *Psychological Review*, **84**(4), 327–372, (1977).

[41] G. van Heijst, A.Th. Schreiber, and B.J. Wielinga, 'Using explicit ontologies in kbs development', *International Journal of Human-Computer Studies*, **45**, 184–292, (1997).

[42] P. Visser and Z. Cui, 'On accepting heterogeneous ontologies in distributed architectures', in *Proceedings of the ECAI'98 workshop on Applications of Ontologies and Problem-solving methods, Brighton, UK*, eds., P. Borst, V.R. Benjamins, A. Farquhar, and A. Gómez-Pérez, pp. 112–119, (1998).

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

[44] C. Welty and N. Guarino, 'Supporting ontological analysis of taxonomical relationships', *Data and knowledge engineering*, **39**(1), 51–74, (2001).

[45] D. Wiggins, *Identity and Spatio-Temporal continuity*, Basil Blackwell, Oxford, 1967.