# Supporting Inheritance Mechanisms in Ontology Representation

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

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

**Abstract.** Research in the ontology engineering field is becoming increasingly important, especially in the area of knowledge sharing. Many research efforts aim to reuse and integrate ontologies that have already been developed for different purposes. This gives rise to the need for suitable architectures for knowledge sharing. This paper analyses a specific aspect of knowledge sharing; that is the integration of ontologies in a way such that different inheritance mechanisms within the ontology are supported, and focuses on conflicts due to multiple inheritance. We first illustrate the problems that inheritance can cause within ontologies together with different approaches presented in the literature to deal with multiple inheritance conflicts and then propose a semi-automatic approach to deal with such conflicts.

## 1 Introduction

Ontologies have become increasingly important in sharing and reusing knowledge. In [26] an architecture of multiple shared ontologies for knowledge sharing was presented. In this architecture, resources no longer commit to a single comprehensive ontology but instead are clustered together on the basis of the similarities they show in the way they conceptualise the common domain: each cluster sharing an ontology. Ontology clusters are then organised in a hierarchical fashion thus permitting concepts to be described at different levels of abstraction. Since different siblings can extend their parent cluster concepts in different ways the cluster hierarchy permits the co-existence of heterogeneous (sibling) ontologies. This approach has the advantage of minimising the information loss when performing translations between resources, since they communicate using the least abstract ontology common to them.

The proposed structure of multiple shared ontologies is based on inheritance mechanisms. From studies on inheritance [2], [24] it has emerged that anomalies might arise when dealing with inheritance mechanisms; research efforts in *non-monotonic reasoning* have focused on these anomalies [15], [19], [13]. This paper

R. Dieng and O. Corby (Eds.): EKAW 2000, LNAI 1937, pp. 140–155, 2000.

analyses how inheritance problems can affect ontologies and proposes a methodology to deal, in a semi-automatic fashion, with the conflicts caused by the use of inheritance mechanisms, following the approach proposed by Goldszmidt and Pearl [9].

The proposal is to represent ontologies by an "enriched" frame-based language where the set of the slot's facets has been extended to encompass additional information required for a full understanding of a concept. Understanding a concept involves a number of things. First it involves knowing what can sensibly be said of a thing falling under that concept. This can be represented by associating *attributes* with the concept, and possible values that these attributes can take when applied to things of that type. Thus it is important to know that some birds fly and others do not. A full understanding of a concept involves more than this, however: it is important to know also what is true of a *prototypical* [22] instance of a concept, to know that the prototypical bird flies. There are, however differences in how confident we can be that an arbitrary instance of a concept 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 attribute values change over time: people may have eyes of various colours, but they do not change over time, whereas hair colour does. This dynamic behaviour also forms part of the domain conceptualisation. We believe that this additional information needs to form part of the ontology. In this paper we concentrate on prototypical values, but also mention, in passing, one method of dealing with dynamic values.

Representation of concepts within the ontology should be enriched by information concerning the *degree of strength* associated with some properties and some measures of how is likely that a value is associated to an attribute. We take these measures to be qualitative rather than numeric. This additional information enables us to deal with conflicts and inconsistencies due to inheritance mechanisms, as rules with a higher degree of strength and ranking can be given precedence. Other facets are introduced to represent how the attribute's value can change over time; this is based on the intuition that attributes can change their values either regularly in time or if an event occurs and that these changes contribute to enrich the attribute description.

The remainder of the paper is organised as follows: section 2 describes problems caused by different inheritance mechanisms, while section 3 presents Goldzsmidt and Pearl's nonmonotonic approach which is the theoretical framework we follow to deal with inheritance problems and section 4 highlights the problems arising when supporting multiple inheritance in the ontology representation. Section 5 illustrates the extended knowledge model used to apply the Goldzsmidt and Pearl's approach, while section 6 sketches the framework used to deal with inheritance conflicts and section 7 applies this framework to a classical artificial intelligence example of inheritance problem: the Nixon diamond. Finally, section 8 draws conclusions and presents future work.

## 2 Providing a Motivation for the Additional Facets

Representing knowledge about the world means representing the objects presumed or hypothesised to exist and to be relevant in the world and their relationships. It has been argued that a knowledge representation is a surrogate [5], a stand-in, for what is in the world. Like any surrogate it is not completely *accurate*; it will necessarily contain simplifying assumptions because of the complexity of the natural world. Indeed even restricting to a subset of the natural world is still overwhelmingly complex. In this respect, a knowledge representation is also a set of *ontological commitments*. Ontological commitments determine not only the objects of the world but also what are the features of these objects that are relevant for the knowledge representation task.

Objects correspond to *classes*: all the member of a class share some common properties. Classes represent *concepts* (the terms can be used as synonyms).

The set of relevant concepts and the relationships holding between them form the *conceptualisation* [8] used to represent the world. When selecting a conceptualisation of the world some decisions have to be made in order to establish what concepts to describe and how to describe them. Concepts are identified by sets of attribute-value pairs, where the attributes are those deemed important for the knowledge representation task and the values associated with them permit us to distinguish one concept from another. Usually in the conceptualisation are also represented properties that are *generally* true for that concept, that is the conceptualisation usually describes a prototypical member of that class [22]. This gives rise to an important issue in knowledge representation: properties that are true for a class prototype are not necessarily true for all members of the class represented by the concept. Examples of such cases are frequent in everyday life; *Almost all* mammals give birth to live young, but three highly unusual mammals (monotreme) do not. Analogously, the ability of birds to fly is a property that is *generally* true; it is a property describing the prototypical bird. This type of information on the descriptive strength of properties should be encompassed in the conceptualisation of the domain and thus in the ontologies derived from it.

An ontology is *"an explicit specification of a conceptualisation"* according to Gruber [10]. That is, the conceptualisation refers to an abstract model of some phenomenon in the world by identifying the concepts that are relevant to that phenomenon; in the ontology, the type of concepts used to describe the phenomenon and the constraints on their use are *explicitly* defined [23]. Ontology representations should include ways to represent how generally a property is shared among the members of a class. The ability of ontologies to distinguish not only between *hard* statements like "Elephants are animals" and *soft* ones like "Birds fly", but also between degrees of "softness", is crucial for reasoning about the knowledge represented in the ontology. This reasoning can prove helpful in dealing with problems arising from of the hierarchical organisation of concepts in ontologies. Concepts in ontologies are hierarchically organised through an IS-A relationship, with a partial order relation that is the ontology's main structure and that is further enriched by attributes, and by relationships or functions relating concepts. The IS-A relationship introduces also the powerful notion of *inheritance of properties*. Properties are shared by concepts either in their original form or modified in order to give the inheriting class, known as *subclass*, a more restrictive definition than that provided by the parent concept. Furthermore other properties can be added to form more specialised concepts.

Anomalies arising from inheritance mechanisms have been illustrated in the literature ([2] and [24]), where a distinction is made between *single inheritance* and *multiple inheritance*. The former permits a concept to inherit properties from one parent only and can cause *default conflicts* while the latter permits a concept to inherit properties from more than one parent and can cause inconsistencies in inherited attribute values. In [3], default values are defined as a way to deduce information about a concept if the information is consistent with what is already known about the concept. Reasoning about defaults can became extremely problematic when only *strict inheritance* is allowed, that is when the IS-A link amounts to logical implication or set inclusion. Then, more specific information cannot overrule information obtained from more general classes thus causing wrong conclusions to be inferred. A *defeasible* approach [24] permits the more specific information to overrule the more general one“ thus solving the conflict.

Other kind of conflicts can arise when multiple inheritance is supported and conflicting information is inherited from two or more general concepts. In this case a choice has to be made about which value has to be associated with the attribute. This choice can be made by knowledge engineers, or the value can be (semi) automatically provided by determining the property's degree of "softness". The same inconsistency problems caused by supporting multiple inheritance can be encountered when trying to integrate ontologies developed for different purposes (the word *integration* is used here to summarise all the possible meanings that the term takes in the ontological engineering field, and that are illustrated in [20]). In fact, with ontology integration an attempt is made to relate concepts in different ontologies. Concepts to relate can be described by the same attributes, but inconsistent values may be associated with them. So, when integrating ontologies, a crucial issue is to choose, among the inconsistent candidates, the value to associate with an attribute. An example of problems encountered while trying to integrate two or more ontologies can be found in [7] and [6]. Once again, the choice is made by the knowledge engineers performing the integration who can be assisted by some tool that (semi) automatically chooses the most promising attribute's value among a set of candidates.

Before proceeding with the discussion, we would like to clarify a point: most of the classical example of default inconsistencies, such as Tweety the penguin or the Nixon diamond concern *instances* instead of concepts. However, all the considerations that have been made for the instances still hold true also for classes, as we can semantically overload the IS-A relationship with the meaning of *Instance-of relation*. We are aware that such an attitude has been strongly criticised in the literature [27], but we deem that such a difference can be disregarded when considering multiple inheritance.

It is interesting to note that the conflicts do not only arise when the IS-A relationship is explicitly stated; in fact conflicts as the Tweety triangle can as well occur in cases of feature inheritance. The Tweety triangle can be easily reformulated as *Concept*: ***Bird***, *Feature*: **Flier** =" yes" etc.; in this case also the conflict for Tweety arises.

Other kind of conflicts can arise when the knowledge representation system allows multiple inheritance and conflicting information is inherited from two or more concepts. The typical example of such a situation is the *Nixon diamond*. In this case, however, we are not able to infer any conclusion, not even the wrong one.

## 3 Reasoning with Conflicts

Both inheritance with exceptions and multiple inheritance default conflicts have been widely investigated in the literature concerning *inheritance networks*. Several approaches have tried to infer a reasonable conclusion (if not the right one) from conflicting premises. Horty in [12] divides theories of inheritance into *direct* and *translational* theories. Direct theories are those where the properties and the features of the inheritance networks (such as consistency) along with the set of conclusions that can be inferred from the premises are analysed and characterised in terms of the networks formalism itself. Examples of direct theories can be found in [24], [13], and [12].

Translational theories are those where the meaning of an inheritance network is specified in terms of some type of logical language, either classical first order logic, or some nonmonotonic logic such as Circumscription [15], or Default Logic [21]. This section focuses mainly on direct approaches. Among the direct approaches here we mention the approach by Pearl [19] and lately by Goldszmidt and Pearl [9] as being particularly relevant for dealing with multiple inheritance within ontologies. The main idea of this approach is that knowledge from an inheritance network can be associated with a probability expressing the degree of (dis)belief associated with that bit of knowledge. This measure of the degree of belief permits the approach to handle more complex default interactions (such as inheritance with exceptions) correctly, as pointed out in [1].

More formally, given the language $L$ of the inheritance network (that for Pearl is the language of propositional formulas), every sentence in $L$ corresponds to a set of possible *worlds*, where a world is a conjunction of all the properties describing a typical individual in the domain. As some worlds are definitely more typical than others it is necessary to express the differences between all the possible worlds. This is obtained by weighing every world by assigning it a probability $\varepsilon$, which defines a probability distribution $P$ over $L$. All the inheritance rules such as $Elephant(x) \rightarrow Animal(x)$ impose restriction conditions on P in the form of extreme conditional probability infinitesimally close either to 0 or to 1, where the closer to 1 the probability, the higher the number of subclasses (and eventually individuals) inheriting the property. So, if we consider the inheritance rule *Mammal(x)* $\rightarrow$ *Gives-birth-to-live-young(x)*, this means

*P(Gives-birth-to-live-young(x)*$|$*Mammal(x))* $\geq 1 - \varepsilon$ that for $\varepsilon$ arbitrarily small is close to 1, meaning that if all is known is that $x$ is a mammal, than $x$ almost certainly inherits the property of giving birth to live young.

However, the full precision provided from this framework is not necessary for taking decisions on inheriting conflicting default values. Under this assumption Goldszmidt and Pearl measure the degree of belief not in the continuous interval $[0, 1]$ but rather on a logarithmic scale and they consider beliefs that map into two different values as being of different order of magnitude.

Let $P(\omega)$ be the probability distribution defined over a set $\Omega$ of possible worlds; if we write the probability $P(\omega)$ as a polynomial in $\varepsilon$ (that is $P_\varepsilon(\omega) = 1 - c_1\varepsilon$, or $\varepsilon^2 - c_2\varepsilon^4$ and so on) then the ranking function $\kappa(\omega)$ is defined as the power of the most significant $\varepsilon$-term in $P_\varepsilon(\omega)$. That is the ranking $P(\omega)$ is expressed as some power of a parameter $\varepsilon$ which plays only the role of linking the defaults together. Letting $\varepsilon \to 0$ means that the defaults tend to be certain.

The ranking $\kappa$ permits to reason about both "hard" and "soft" statements; "Birds fly" is, for instance, a soft one because it is **typically** true for most of the subclasses of the class Birds. The rank $\kappa$ roughly corresponds to linguistic quantifiers such as *believable*, *unlikely*, *very rare* etc. In fact for $\kappa(\phi) = 0$ it means that both $\phi$ and $\neg\phi$ are equally possible, for $\kappa(\phi) = 1$ it means that $\neg\phi$ is believed, for $\kappa(\phi) = 2$ it means that $\neg\phi$ is ***strongly*** believed, for $\kappa(\phi) = 3$ it means that $\neg\phi$ is ***very strongly*** believed and so on.

An inference system (*Z-system*) based on the ranking of probabilities has been developed in [9]. The Z-system is able to draw plausible conclusions in most of the cases by a technique known as *z-entailment*, [9] which guarantees that conclusions in inheritance rules will receive high probabilities whenever the premises receive sufficiently high probabilities. This system can compute the priorities of inheritance rules and provides also consistency checks. Unfortunately, one of the main drawbacks of the z-entailment is that it cannot sanction the inheritance property from classes to subclasses with exceptions. This happens because the z-entailment labels all the classes with exceptions as exceptional in all respects, so that they become unable to inherit any of the properties that are typical of their parent class. To overcome this drawback the authors introduce the capabilities for a Z-system to handle variable-strengths thus allowing some defaults to be stated "more strongly" than others.

The $system - Z^+$ [9] extends the specification of the inheritance rules by associating with each rule a parameter $\delta$ which expresses the *degree of strength* of the rule. Inheritance rules are now ordered on the grounds of a *priority function* $Z^+$, which is computed as function of both the ranking associated with an inheritance rule and the degree of strength $\delta$; each of them reflects different considerations to be taken into account when drawing conclusions about inheriting properties. The degree of strength $\delta_i$ associated with an inheritance rule $r_i = \phi_i \to \psi_i$ establishes the relative strength with which $\psi_i$ is committed to be accepted in the context of $\phi_i$ while the priority $Z^+(r_i)$ expresses the degree of surprise concerning the finding of a world that violates $r_i$, which includes also the degree of surprise associated with $\phi_i$. Again all the consistency considerations hold also for the $system - Z^+$. Therefore now for each rule $r_i = \phi_i \rightarrow \psi_i$ the $Z^+$ ordering is determined by both the degree of strength and the ranking function. This type of ordering guarantees that features of more specific contexts override conflicting features of a less specific order, thus allowing the well known Tweety the Penguin problem to be solved. Furthermore, whenever the ranking functions associated with the rules do not permit us to distinguish between inheritance rules, because no specificity consideration is made, the $Z^+$ ordering depends on the degree of strength alone, therefore permitting preference of one inheritance rule over the other(s). In this way the system can deal with types of conflicts such as the Nixon diamond.

## 4 Ontologies and Multiple Inheritance

Latest research on inheritance has focused on extending the basic framework of single inheritance without exceptions to inheritance with exceptions and multiple inheritance. However, research on these issues has mainly been confined to academia, giving the impression that problems such as multiple inheritance and inheritance with exceptions are quite rare in real applications [18].

Research in the ontology field has not yet considered any of the problems due to the inheritance of conflicting default values. Indeed many languages to represent ontologies support either multiple inheritance or inheritance with exceptions, but often they do not have any mechanism to deal with the problems caused by these formalisms. Possibly, the problem of handling conflicts has not been regarded as such because ontologies have been usually written from scratch whenever they were needed. This trend in the ontology field has been changing recently, mainly due to research in ontology engineering, which has stressed the importance of building ontologies that are reusable and sharable.

When trying to integrate ontologies developed for different purposes, inconsistencies can arise ([7] and [6]). In fact, one concept can have different parents in different ontologies, and those parents can be described in terms of conflicting attributes. The situation can be even more complicated because inconsistencies can be implicit. Inheritance literature has not been extensively discussed in this context although it is extremely relevant in ontology merging. Indeed, it is likely that ontologies built for different purposes and then merged represent concepts in terms of attributes that are semantically equivalent although with mismatches in the names [25]. Morgenstern [18] has modified the Touretzky's Nixon diamond [24] to show how inconsistencies can be also implicit. The new Nixon's diamond example is shown in figure 1.

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. The proposed framework, illustrated in the next section, deals with such types of inconsistencies.

From the inheritance network viewpoint, ontologies are mixed inheritance networks, where both strict and defeasible paths are allowed, therefore, when

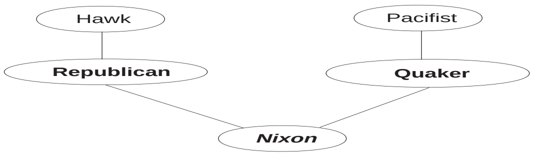

**Fig. 1.** The modified Nixon diamond

trying to reason with the knowledge expressed in the ontology, an inference method that is able to deal with both types of path is needed. Unfortunately most of the ontologies are based on frames representation systems such as the Generic Frame Protocol [4] where no slot's facet is used to distinguish between these paths. This is the reason why we propose to augment the typical facets of a slot by introducing some additional pieces of information which are useful in dealing with default and inheritance problems: the ranking associated with the inheritance rule, a degree of strength associated with the attribute and facets about how the attribute's value can change over time. The ranking expresses the degree of belief which is associated with the inheritance rule expressed by the attribute, that is how surprising is to find out that, for the concept that is being described, the attribute takes a particular value. In our approach the degree of strength is associated with the attribute (and therefore with the inheritance rule) by the knowledge engineers who are either writing or merging the ontologies, as these people should be familiar with the domain and should therefore be able to weight inference rules. The degree of strength not only distinguishes between strict and defeasible links, but can also be used to measure the degree of defeasibility. Moreover, it permits us to establish preferences among defaults when no specificity considerations are available. If we consider the Nixon diamond example, both facts A: "Quakers are pacifists" and B: "Republicans are not pacifists" are absolutely true, but they might be evaluated differently depending on the domain and even on specific circumstances. A degree of strength can be associated with both these rules. Let us assume that the knowledge engineer believes that religious convictions carry more weight than political affiliations, than the degree of strength associated with A, $\delta_A$ is greater then the degree of belief associated with B $\delta_B$. So when the value of the attribute Pacifist is determined for the object Nixon, whenever no specific information on the object is available, the degree of strength makes it infer that Nixon is a pacifist.

Although the degree of strength is decided by the knowledge engineer, it can be affected by specific events that can change the status of an attribute. The intuition behind this is that nonmonotonicity is either time dependent or event dependent, meaning that the value of an attribute can change regularly in time or it can change if a particular event occurs. Therefore, in case of conflicting default values the choice among the possible values should be made by taking into account the regularity in time or the occurrence of one of the modifying events. Going back to the Nixon example, one of the events that can change the status of the attribute Pacifist is the declaration of a war: as president of the United States, although maybe personally inclined to be pacifist, Nixon would tend to protect the interests of his country in the event of a conflict, and so in such a scenario he would not act as a pacifist. This can also mean that until a war is not declared we can assume that the degree of strength associated with the religious conviction is stronger than the one associated with the political conviction, but in case of war this would be no longer true.

Finally, it is interesting to note that many problems with multiple inheritance could be solved by a more careful design of ontologies as pointed out by Guarino [11]. This is due to the fact that in many cases the IS-A relationship is used to represent many other specialised links such as reduction of sense, overgeneralisation, confusion of senses, clash of senses, and sometimes some kind of type-to-role links.

## 5 The Extended Knowledge Model

So far, all the efforts to deal with inconsistencies have been performed by hand by a knowledge engineer who is expert in the domain that is being described, and who can thus associate the correct default value with an attribute. Choosing between several conflicting defaults requires an extremely rich semantics. For this reason performing the choice automatically is quite unrealistic, but a more realistic possibility is a semi-automatic approach, where an inferential system presents the knowledge engineer with a list of sound alternatives (according to the inference process), but leaves the actual choice to the knowledge engineer.

The model of knowledge used to represent the ontology plays a crucial role in the framework proposed in this paper, as it provides the elements necessary to apply the Goldszmidt and Pearl inference process. The proposed knowledge model is frame-based [17]. Our model is based on *classes*, *slots*, and *facets*. *Classes* correspond to concepts and are collections of objects sharing the same properties, hierarchically organised into a multiple inheritance hierarchy, linked by *IS-A* links. Classes are described in terms of *slots*, or attributes, that can either be sets or single values. A slot is described by a name, a domain, a value type and by a set of additional constraints, here called *facets*. Facets can contain the documentation for a slot, constrain the value type or the cardinality of a slot, and provide further information concerning the slot and the way in which the slot is to be inherited by the subclasses. Our framework suggests the introduction of a set of facets that describes in detail the attribute and its behaviour in the concept description to accommodate different inheritance mechanisms, both within and between ontologies, and changes over time. This additional information is to be used in case of inconsistencies as a guide towards the most reasonable and informed suggestion to be presented to the domain expert, who will than validate such suggestion. The facets we introduce are:

- **Value**: There are three possibilities:
  - If the concept that is being defined is very high in the hierarchy (so high that any distinction based on the attribute's value is not possible), then **Value** is equal to **Domain**;
  - If the concept is still general, but it is possible to determine that it can have different attribute values for its children then **Value** is set equal to **Sub-domain** $\subset$ **Domain**;
  - If the concept is defined in terms of a specific value for an attribute then **Value** is set $v \in$ **Domain**.

  For the *third case* only, further information about the type of value (see next item) or the degree of strength (see item below) can be added;
- **Type of value**: {*Necessary, Prototypical, Inherited, Distinguishing*}. An attribute's value is a *Necessary* one if the value is true for all concept's children. It describes necessary conditions in the concept's description. An attribute's value is a *Prototypical* one if the value is generally true for any children of the concept that is being defined, that is the value is generally true for any prototypical instance of the concept, but exceptions are permitted with a degree of softness expressed by the facet *Ranking*. An attribute's value can be *Inherited* from some super concept or it can be a *Distinguishing* value, that is a value that differentiates among siblings;
- **Degree of strength**: a number describing how relevant is, in the concept's description, the property represented by the attribute. For example, to reason about birds ability to fly, the attribute *species* is more relevant than the attribute *feather colour*. In merging ontologies this facet represents the weight associated with the inheritance rule corresponding to the attribute;
- **Ranking**: an integer describing the probability ranking associated to the fact that the attribute takes the value specified in the facet **Value**. The possible values for this facet are 1: *All*, 2: *Almost all*, 3: *Most*, 4: *Possible*, 5: *A Few*, 6: *Almost none*, 7: *None*. So, to represent the soft statement *Birds fly* we could describe the concept *Bird* by a slot, **Fly** that takes value *Yes* with *Ranking* equal to "Most";
- **Change frequency**: {*Regular*, *Once only*, *Volatile*}. This facet describes how often an attribute's value changes. If the information is set equal to *Regular* it means that the value changes at regular time intervals; if set equal to *Once only* it indicates that only one change is possible, and finally *Volatile* indicates that the attribute's value can change more than once. If the change frequency is *Regular* then the time interval is specified otherwise the event causing the attribute to change is specified;
- **Time interval**: This information can either be empty (if the change frequency is not *Regular*) or it contains the time interval between two changes;
- **Event**: This facet is either empty (if the change frequency is *Regular* and the time interval is set) or it is the set of events $\boldsymbol{E}$ that causes a change in the attribute's value. The logical theory chosen to reason about events is the

*Event Calculus* [14], and the information **Event=**$e_i$ is interpreted as one of the following Event calculus expressions:

1. *Hold(before(*$e_i$*, P))* that is, the property P holds *BEFORE* the event $e_i$;
2. *Hold(after(*$e_i$*, P))* that is, the property P holds *AFTER* the event $e_i$;

where the interpretation is decided on the information **Event Validity** (see below). For each event $e_i \in \boldsymbol{E}$ we specify also the *Event Property* and the *Event Validity* facets as follows:

- **Event Property**: $\{V\}$. This facet describes the value taken by the attribute before or after the event E. If this bit of information is empty it means that the event E causes a change in the attribute's value that cannot be specified, possibly because the value can be identified only by considering the instances of the concept;
- **Event Validity**: {*Before, After*}. It states whether the property $V$ specified in the item above holds before the event E or after the event E.

The above facets describe how crucial the slot is in characterising a class, and what conditions determine a change in the value of the slot for that class. These changing conditions are used to query the knowledge engineer while solving default inconsistencies to try to associate with a slot a value as close to the true one as possible. These facets could also be used by knowledge engineers to learn more about the attribute they are dealing with.

## 6 The Framework to Deal with Inheritance Conflicts

When dealing with heterogeneous resources, mismatches in the names of concepts and attributes might occur [25], [6]. The first step of our framework consists of resolving name mismatches following [7]. This is necessary to avoid cases of implicit inconsistencies, where attributes describing two parent concepts are denoted with different names, while describing the same property. Then the attempt to relate the concepts in the ontologies composing the structure can begin.

In the remainder we present the steps composing this framework, explaining how a system can resolve inconsistencies when trying to build multiple shared ontologies. Ontologies are assumed to be represented by the knowledge model illustrated above. The knowledge engineer **KE** interacts with the system in several steps:

- The first step of our framework consists of scanning both the class names and the slot names in all the ontologies to find possible synonyms. Synonyms are evaluated intensionally, selecting them on the basis of a general thesaurus such as WordNet [16]. For the attributes, however, also an extensional check is performed, by checking the similarity in the attribute's domains.
- Once the name mismatches are resolved, the system proceeds both bottom-up and top down trying to relate classes. When it finds two or more classes

that are suitable parents for the class the system is handling, then a consistency check is performed, according to the technique by Goldszmidt and Pearl [9];

- If an inconsistency is detected then the *priority functions* (see section 3) for the inheritance rules are computed on the grounds of both the *rankings* of probabilities and the *degrees of strength*. These facets permit to solve both default conflicts and inconsistencies due to either multiple inheritance or to the integration of diverse ontologies. The slot's facets encompassing information about the events that can cause the attribute to change are taken into account too, as this information is presented to the **KE** who is requested to validate the events. The system should have now everything necessary to compute the priority function: if so it proceeds to the next step, otherwise if either the ranking or the degree of strength are missing, the systems asks the **KE** to insert them. The value of the inserted facet is decided on the grounds of the information regarding the attribute's changes over time.
- After all the priority functions are computed and ordered, the system presents the **KE** with the slot's value with the best scores.
- The **KE** decides whether to accept the system suggestion or to ask the system to present the list of possible choices in rank order.

## 7 Applying the Framework to the Nixon Diamond Problem

To explain more clearly how the proposed approach works let us consider the following example, which is an extension of the Nixon diamond. Let us suppose that we need to model the beliefs of the US population from two different viewpoints: political affiliations and religious convictions. The two ontologies describing these viewpoints are partially illustrated in figure 2. These different viewpoints do not always contrast in the process of taking decisions because one of them often prevails, depending on the matter: political affiliations usually determine people's

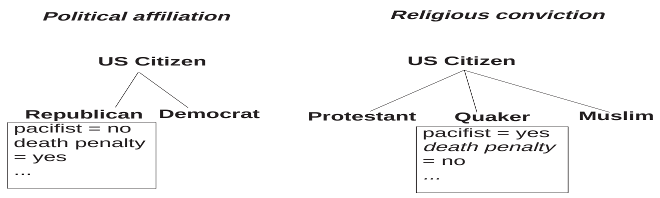

**Fig. 2.** Sections of the two ontologies modeling the beliefs of the US population

positions on issues such as welfare and economics whereas religious convictions affect moral issues. However there are some controversial issues that have also a strong moral component, therefore both viewpoints contribute to the process of decision making. In such cases the two viewpoints can either agree or contrast so in this latter case a choice is necessary.

In this example we consider two ontologies, one modeling the political affiliations of US citizens and the other the religious convictions: the two ontologies need to be merged to use this knowledge in order to take decisions about public interest issues that can be considered from both a political and a moral viewpoint.

In merging the two ontologies the following inheritance rules hold for the class "Nixon":

$r_1$: *"quakers are pacifists with strength* $\delta_1$*",* $q \xrightarrow{\delta_1} p$
$r_2$: *"republicans are non pacifists with strength* $\delta_2$*",* $r \xrightarrow{\delta_2} \neg p$
$r_3$: *"quakers are against death penalty with strength* $\delta_3$*",* $q \xrightarrow{\delta_3} \neg d$
$r_4$: *"republicans support death penalty with strength* $\delta_4$*",* $r \xrightarrow{\delta_4} d$

Let us suppose we want to use the knowledge in these two ontologies to infer what would be the position of Nixon in two different situations: going to war and voting on the death penalty. These are decisions that might be taken on the grounds of both political and religious beliefs. Therefore we try to apply the algorithm sketched in the previous section to merge the two ontologies in this two cases. Let us start from the situation in which Nixon has to decide whether the US should go to war. In both these examples we are not concerned with problems due to name mismatches, so we assume that the first step of the procedure is executed successfully. Then the system attempts to relate classes; it finds the class Nixon in both ontologies and with a different parent in each ontology, so the system considers the class Nixon as child of both the class Quaker and Republican, thus inheriting attributes from both of them. At this point the system detects an inconsistency, therefore it tries to resolve it by considering the rankings of probability and the degrees of strength associated with rules $r_1$ and $\mathrm{r}_2$.

In such a case, as also pointed out by Goldszmidt and Pearl [9], the $Z^+$ system is not able to decide which rule to prefer on the grounds of the ordering alone, because the priority functions associated to the rules by the $Z^+$ system are: $Z^+(r_1) = \delta_1$ and $Z^+(r_2) = \delta_2$. In fact in this case the decision to prefer one rule over the other does not depend on specificity considerations but rather on the weight that is associated with each inheritance rule and that *depends on the task at hand*. In problems such as the Nixon diamond it is likely to find that the degree of strength associated with the inheritance rule is is left as choice to knowledge engineers. Knowledge engineers use their knowledge of the domain to assign a value with the degree of strength for each inheritance rule. However, the facets concerning the events causing the attribute's value to change can provide additional information to the process of making a decision. In this specific case the event causing the attribute *Pacifist* to take value *No* for any child of the concept *Republican* in the "Political Affiliation" ontology is the threat of a war against the USA, that is in terms of event logic *Hold(after(War-Against-USA, Pacifist=No))*. So, when the knowledge engineers merging the two ontologies decide the values of the degrees of strength, the choice is made on the grounds of the available information. Since it is in the "Political Affiliation" ontology that the attribute which is being handled is described as changing its value if a war occurs, then this inheritance rule prevails. So, knowledge engineers set $\delta_2 > \delta_1$. The system returns the $Z^+$ ordering $r_2, r_1$, thus solving the conflict by preferring the rule *republicans are non pacifists* over the rule *quakers are pacifists*.

In the other situation, that is deciding over death penalty, the algorithm works pretty much in the same way. In this case the class "Nixon" inherits both the rules $r_3$ and $r_4$, thus the system detects an inconsistency. In this case no event is specified as able to change either attribute's values: in general the position taken on the death penalty is a fixed opinion. However, for this example the probability of finding that a quaker is against death penalty is higher then the probability of finding that a republican is against it, since it is *always* true that a quaker does not approve death penalty whereas it is only *likely* that a republican approves it. This difference is reflected by the $Z^+$ ordering of the two rules, which is: $Z^+(r_3) > Z^+(r_4)$. Moreover, if knowledge engineers wish to encompass the information that, in case of death penalty, Nixon's religious conviction carry more weight than Nixon's political affiliation, they might set $\delta_3 > \delta_4$. The system returns in any case the $Z^+$ ordering of the rules, which is $r_3, r_4$, thus solving the conflict by preferring the rule *quakers are against death penalty*, as considerations on the degree of belief prevail in this case.

## 8 Conclusion and Future Work

This paper has presented a semi-automatic framework to deal with multiple inheritance inconsistencies while integrating ontologies. After analysing the problems that are classically proposed in the multiple inheritance literature, we have presented a formal approach to deal with inconsistencies. This approach has been chosen to deal with inconsistencies in the ontology representation. Inconsistencies in ontologies can be more subtle than the ones in semantic networks because diverse ontologies can use different names for the same concept or attributes, so that some inconsistencies can be implicit.

This framework is based on a knowledge model that extends the usual frame-based model in order to associate with each attribute a degree of strength and other information concerning the behaviour of the attribute. By means of this framework knowledge engineers trying to integrate different ontologies are now provided with a tool that checks the inconsistencies and presents them with a list of suggestions that are evaluated according to a priority function, instead of having to check inconsistencies by hand and resolve them. The final choice is always left to the knowledge engineers, but the system provides them with a set of possible choices and with information concerning how and when the attribute changes.

One crucial issue is the choice of the degree of strength to be associated with a slot. At the moment the choice on the degree of strength for inheritance rules is left to the knowledge engineer, although the possibility of increasing the degree of strength of a slot if an event causing the attribute to change occurs will be investigated. Future work will concentrate on extending the framework by introducing some form of temporal reasoning based on event logics that extend the facets.

## Acknowledgment

This research is conducted as part of a PhD project funded by BT. The authors wish to thank Floriana Grasso and Dean Jones.

## References

[1] R.A. Bourne and S. Parsons. Maximum entropy and variable strength defaults. In T. Dean, editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 50–55. Morgan Kaufmann Publishers Inc., 1999.

[2] R.J. Brachman. On the epistemological status of semantic networks. In R.J. Brachman and H.J. Levesque, editors, *Readings in Knowledge Representation*, pages 191–215. Morgan Kaufmann, Los Altos, CA, 1985.

[3] B. Carpenter. Skeptical and credulous default unification with application to templates and inheritance. In T. Briscoe, A. Copestake, and V. de Paiva, editors, *Default reasoning and Lexical Organization*. Cambridge University Press, 1993.

[4] V.K. Chaudhri, A. Farquhar, R. Fikes, P.D. Karp, and J.P. Rice. The generic frame protocol 2.0. Technical report, Stanford University, 1997.

[5] R. Davis, H. Shrobe, and P. Szolovits. What is a knowledge representation? *AI Magazine*, 14(1):17–33, 1993.

[6] R. Dieng and S. Hug. Comparison of ≪personal ontologies≫ represented trough conceptual graphs. In H. Prade, editor, *Proceedings of the 13th European Conference of Artificial Intelligence (ECAI)*, pages 341–345. John Wiley & Sons, 1998.

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

[8] M.R. Genesereth and N.J. Nillson. *Logical foundations of Artificial Intelligence*. Morgan Kauffman, 1987.

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

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

[11] N. Guarino. The role of identity conditions in ontology design. In V.R. Benjamins, editor, *Proceedings of the IJCAI'99 Workshop on Ontology and Problem-Solving Methods: Lesson learned and Future Trends*, volume 18, Amsterdam, 1999. CEUR Publications.

[12] J.F. Horty. Some direct theories of nonmonotonic inheritance. In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logics in Artificial intelligence and Logic Programming, Vol. 3: Nonmonotonic Reasoning and Uncertain Reasoning*, pages 111–187. Oxford University Press, Oxford, 1994.

[13] J.F. Horty, R.H. Thomason, and D.S. Touretzky. A skeptical theory of inheritance in nonmonotonic semantic networks. *Artificial intelligence*, 42:311–348, 1990.

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

[15] J. McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.

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

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

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

[19] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffman, San Mateo, CA, revised second printing edition, 1988.

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

[21] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

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

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

[24] D.S. Touretzky. *The Mathematics of Inheritance Systems*. Morgan Kaufmann, 1986.

[25] P.R.S. Visser, D.M. Jones, T.J.M. Bench-Capon, and M.J.R. Shave. Assessing heterogeneity by classifying ontology mismatches. In N. Guarino, editor, *Formal Ontology in Information Systems. Proceedings FOIS'98, Trento, Italy*, pages 148–162, Amsterdam, The Netherlands, 1998. IOS Press.

[26] P.R.S. Visser and V.A.M. Tamma. An experience with ontology-based agent clustering. In V.R. Benjamins, editor, *Proceedings of the IJCAI'99 Workshop on Ontology and Problem-Solving Methods: Lesson learned and Future Trends*, volume 18, pages 12.1–12.13, Amsterdam, 1999. CEUR Publications.

[27] W.A. Woods. What's in a link: Foundations for semantic networks. In D. G. Bobrow and A. Collins, editors, *Representation and Understanding*, pages 35–82. Academic Press, New York, 1975.