

# A Model Theoretic Semantics for Ontology Versioning

Jeff Heflin and Zhengxiang Pan

Department of Computer Science and Engineering, Lehigh University  
19 Memorial Dr. West, Bethlehem, PA 18015, U.S.A.  
{heflin,zhp2}@cse.lehigh.edu

**Abstract.** We show that the Semantic Web needs a formal semantics for the various kinds of links between ontologies and other documents. We provide a model theoretic semantics that takes into account ontology extension and ontology versioning. Since the Web is the product of a diverse community, as opposed to a single agent, this semantics accommodates different viewpoints by having different entailment relations for different ontology perspectives. We discuss how this theory can be practically applied to RDF and OWL and provide a theorem that shows how to compute perspective-based entailment using existing logical reasoners. We illustrate these concepts using examples and conclude with a discussion of future work.

## 1 Introduction

The Semantic Web (Berners-Lee, Hendler, and Lassila 2001)[1] has been proposed as the key to unlocking the Web's potential. The basic idea is that information is given explicit meaning, so that machines can process it more intelligently. Instead of just creating standard terms for concepts as is done in XML, the Semantic Web also allows users to provide formal definitions for the standard terms they create. Machines can then use inference algorithms to reason about the terms and to perform translations between different sets of terms. It is envisioned that the Semantic Web will enable more intelligent search, electronic personal assistants, more efficient e-commerce, and coordination of heterogeneous embedded systems.

Unfortunately, the Semantic Web currently lacks a strong underlying theory that considers its distributed aspects. To date, the semantics for semantic web languages have looked little different from the traditional semantics of knowledge representation languages. Traditional knowledge bases assume a single consistent point-of-view, but the knowledge of the Semantic Web will be the product of millions of autonomous parties and may represent many different viewpoints. We argue that the Semantic Web is not just AI knowledge representation using an XML syntax, but actually changes the way we should think about knowledge representation. Semantic web knowledge bases must deal with an additional level of abstraction, that of the document or resource that contains assertions and formulas. Furthermore, the semantics of the knowledge base must explicitly account for the different types of links that can exist between documents. Although languages such as RDF and OWL currently give a definitive account of the meaning of any single document, things become more ambiguous when you consider how documents should be combined. In this respect, semantic web systems are in a state

analogous to the early days of semantic nets. A quote from Brachman [3] about links between concepts in early semantic nets seems just as appropriate for “links” between semantic web documents today:

... the meaning of the link was often relegated to “what the code does with it”  
 - neither an appropriate notion of semantics nor a useful guide for figuring out what the link, in fact means.

Without a better understanding of inter-document links on the Semantic Web, we will have serious interoperability problems. This paper will examine the relationship between different types of inter-document links and propose an unambiguous semantics for them. In particular we focus on links that indicate that one document is a revision of another.

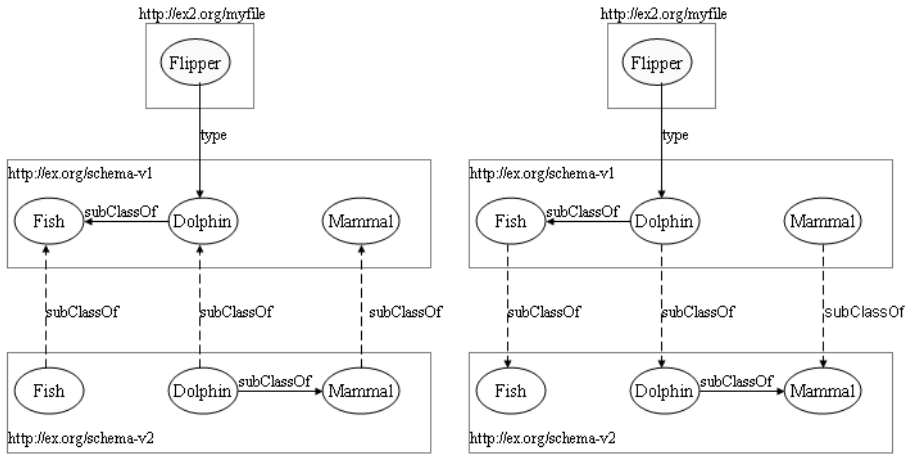
## 2 Ontology Versioning

Ontologies have come to be seen as a critical component of the Semantic Web. An ontology provides a common vocabulary to support the sharing and reuse of knowledge. When two parties agree to use the same ontology, they agree on the meanings for all terms from that ontology and their information can be combined easily. Unfortunately, there is no widely accepted definition of an ontology. We prefer this definition from Guarino [4]: “An ontology is a logical theory accounting for the intended meaning of a formal vocabulary.”

The first author [7] initially described the problems of ontology versioning in distributed environments such as the Web. The Web is a dynamic place, where anyone can instantaneously publish and update information. It is important that this ability is not lost when it is described by a Semantic Web language. People must be able to publish semantic web ontologies as easily as other documents, and they must be allowed to revise these ontologies as well. Ontologies may be changed in order to correct errors, to model new phenomenon, or simply to represent the world in a different way.

When we decide to change an ontology, then we must consider that in a distributed ontology framework such as the one needed by the Semantic Web, there will often be information resources that depend on it. Since the owner of the resource may find these changes undesirable, we should not actually update the original ontology but instead create a new file that represents the new version.

However, this can lead to problems. Consider the example in Fig.1. The original version of the ontology incorrectly states that all Dolphins are Fish. There is another web page that states that Flipper is a Dolphin. Later, the ontology is corrected to say all Dolphins are Mammals. In RDF and OWL, there are no semantics associated with versioning, so we have to attempt to approximate them using existing language constructs. On the left of Fig.1, we could consider making each version2 class be a subclass of a corresponding version1 class. This means version2 Dolphin will not be version2 Fish, but also means that Flipper will not be a member of any version2 classes. In other words, version1 data is lost in version2. Alternatively, on the right of Fig. 1, we could consider making each version1 class be a subclass of a corresponding version2 class, this means Flipper will be a version2 Dolphin and a version2 Mammal, but will also



**Fig. 1.** Ontology revision example. On the left: version 2 classes are subclasses of version 1 classes. On the right: version 1 classes are subclasses of version 2 classes.

be a version2 Fish. If the Fish and Mammal classes are disjoint, this will lead to an inconsistent ontology.

An additional complexity deals with the intended meanings of the ontology. Consider the case that in version 1, “Dolphin” actually meant “Dolphin Fish.” Perhaps because this term was confusing users, we decide to change it to mean a kind of porpoise, which is the more common usage. The resulting ontology would be indistinguishable from the one described in the preceding paragraph. Yet, the implications are very different. In the first case, we were correcting an error in our definitions and would like to retroactively apply this correction to existing information resources. In the second case, we have decided to change our concept of “Dolphin”, and consequently changed our definition to correspond to the new meaning. In this case, it would be a mistake to retroactively apply the new definitions to old resources. That is no instance of version1 Dolphin should be considered instance of version2 Dolphin. Note that there is no syntactic distinction between these two cases.

The first author [6] has shown how to resolve such problems by allowing ontology authors to specify backwards-compatibility. Essentially an ontology version is backward compatible with a prior version if it contains all of the terms from the earlier ontology and the terms are supposed to mean the same thing in both ontologies.

Since backwards-compatibility depends on knowledge of the intended models of an ontology, it cannot be computed automatically, instead it must be specified by an ontology’s author. This is driven by the fact that ontologies only specify a theory partially, and that the intended meaning of a term may change even though the ontology’s theory remains the same. Since the ontology can only restrict unintended models, there is no way to formally describe the intended models of an ontology. Furthermore, we suggest that an ontology may not correctly describe the real world, thus it may be missing models that were intended, may include models that are undesirable.

In fact, OWL has already taken ontology versioning into account. For example, `owl:backwardCompatibleWith` and `owl:incompatibleWith` are dedicated to specifying the compatibility. However, the Web Ontology Group felt that versioning was not understood well enough to provide semantics for those features when the language was designed. Associating formal semantics to those language constructs can help OWL support ontology versioning.

This paper builds on the first author's previous work and provides a model theoretic semantics for ontology versioning, particularly the concept of backward compatibility. Meanwhile, these inter-document links on the Semantic Web are established: extension, prior Version, backward compatible version and commitment to an ontology by a resource.

### 3 Preliminary Definitions

We now provide formal definitions to describe our model of the Semantic Web. We need to have structured representations of the data associated with information resources, and these resources must be able to commit to ontologies which provide formal definitions of terms. We will describe this by defining ontologies and resources using a logical language and by providing a model theoretic semantics for these structures. These definitions improve upon those given by the first author [6].

Let  $D$  be the domain of discourse, i.e., the set of objects that will be described. Let  $R$  be a subset of  $D$  that is the set of information resources, such as web pages, databases, and sensors.

There are many candidate representation languages for an ontology. In order to maintain the generality of this theory, we will use first-order logic as the representation language. Due to the expressivity of FOL, this theory will still apply to the languages commonly used for semantic web systems, such as description logics, as well as proposed Horn logic extensions. We will assume that we have a first-order language  $\mathcal{L}^S$  with a set of non-logical symbols  $S$ . The predicate symbols of  $S$  are  $S_P \subset S$ , the variable symbols are  $S_X \subset S$ , and the constant symbols are  $S_C \subset S$ . For simplicity, we will not discuss function symbols, since an  $n$ -ary function symbol can be represented by a  $n+1$ -ary predicate. The well-formed formulas of  $\mathcal{L}^S$  are defined in the usual recursive way. We will use  $L(V)$  to refer to the infinite set of well-formed first-order logic formulas that can be constructed using only the non-logical symbols  $V$ . We define interpretation and satisfaction in the standard way.

Now, we define the concept of semantic web space, which is a collection of ontologies and information resources.

**Definition 1.** A semantic web space  $\mathcal{W}$  is two-tuple  $\langle \mathcal{O}, R \rangle$ , where  $\mathcal{O}$  is a set of ontologies and  $R$  is a set of resources. We assume that  $\mathcal{W}$  is based on a logical language  $\mathcal{L}^S$ .

In the next two subsections, we formally define ontology and resource.

### 3.1 Ontology Definitions

We begin by formally defining an ontology that supports versioning.

**Definition 2.** *Given a logical language  $\mathcal{L}^S$ , an ontology  $O$  in  $\mathcal{O}$  is a five-tuple  $\langle V, A, E, P, B \rangle$ , where*

1.  $V \subseteq S$  (the vocabulary  $V$  is a subset of the non-logical symbols of  $\mathcal{L}^S$ )
2.  $V \supset S_X$  and  $V \supset S_C$  ( $V$  contains all of the variable and constant symbols of  $\mathcal{L}^S$ )
3.  $A \subseteq L(S)$  (the axioms  $A$  are a subset of the well-formed formulas of  $\mathcal{L}^S$ ).
4.  $E \subset \mathcal{O}$  is the set of ontologies extended by  $O$ .
5.  $P \subset \mathcal{O}$  is the set of prior versions of the ontology, such that for all  $O_i \in P$  with  $O_i = \langle V_i, A_i, E_i, P_i, B_i \rangle$ ,  $P \supset P_i$ .
6.  $B \subset P$  is the set of ontologies that  $O$  is backwards compatible with. For each  $B_i \in B$ ,  $V \supseteq V_i$ .

As a result of this definition, an ontology defines a logical language that is a subset of the language  $\mathcal{L}^S$ , and defines a core set of axioms for this language. Note, the second condition is for convenience in defining well-formed formulas below.

There are three basic kinds of “links” between ontologies. An ontology can extend another, which means that it adds new vocabulary and or axioms. The set  $E$  is the set of ontologies extended by  $O$ . Ontologies also have a set  $P$  of prior ontologies and set  $B$  of prior ontologies that they are backward compatible with. Note that the prior versions  $P$  of an ontology need only be a superset of the prior versions of those ontologies in  $P$ . Thus, versioning need not be a total order.

It is important to note that this definition requires that an ontology include the vocabulary of any ontology with which it is backward compatible. This is in order to guarantee that formulas that were well-formed with respect to one version are also well-formed with respect to the backward-compatible version.

Note that backwards-compatibility does not require that the revision contains a superset of the axioms specified by the original version. This allows us to remove axioms that are no longer correct or that would be better expressed in another ontology.

For convenience, we define the concept of an ancestor ontology. An ancestor of an ontology is an ontology extended either directly or indirectly by it.<sup>1</sup> If  $O_2$  is an ancestor of  $O_1$ , we write  $O_2 \in \text{anc}(O_1)$ . The formal definition of an ancestor is:

**Definition 3 (Ancestor function).** *Given two ontologies  $O_1 = \langle V_1, A_1, E_1, P_1, B_1 \rangle$  and  $O_2 = \langle V_2, A_2, E_2, P_2, B_2 \rangle$ ,  $O_2 \in \text{anc}(O_1)$  iff  $O_2 \in E_1$  or there exists an  $O_i = \langle V_i, A_i, E_i, P_i, B_i \rangle$  such that  $O_i \in E_1$  and  $O_2 \in \text{anc}(O_i)$ .*

Since the ontology defines a language, we can talk about well-formed formulas with respect to an ontology. Essentially, a formula is well-formed with respect to an ontology if it is a well-formed formula of the logic that only uses the ontology’s vocabulary. First, we must identify the vocabulary accessible to an ontology. Ancestor ontologies play a role in this. Since an ontology should have access to all symbols defined in its ancestors, a formula of that ontology should still be well-formed if it uses symbols from the ancestor ontologies.

<sup>1</sup> Extension is sometimes referred to as inclusion or importing. The semantics of our usage are clarified in Definition 7

**Definition 4.** *The vocabulary closure of an ontology is given by a function  $vclose : \mathcal{O} \rightarrow \mathcal{P}(S)^2$  such that for each ontology  $O = \langle V, A, E, P, B \rangle$ ,  $vclose(O) = V \cup \bigcup_{\{j | O_j \in \text{anc}(O)\}} V_j$  (where  $O_j = \langle V_j, A_j, E_j, P_j, B_j \rangle$ ).*

Using the vocabulary closure, we can define well-formedness.

**Definition 5.** *A formula  $\phi$  is well-formed with respect to an ontology  $O$  iff  $\phi \in L(vclose(O))$ .*

We also need the definition of a well-formed ontology. We must consider two factors. First, we must consider whether or not a well-formed ontology can extend itself, either directly or indirectly. We will remain agnostic on the issue and not place any restrictions on cycles for the extension relation here. We only insist that all of an ontology's ancestor ontologies be well-formed. Second, we must ensure that no resource or ontology can commit to (or extend) two different versions of the same ontology.

**Definition 6.** *An ontology  $O = \langle V, A, E, P, B \rangle$  is well-formed iff:*

1.  *$A$  is well-formed with respect to  $O$*
2. *for each  $O_i \in \text{anc}(O)$ ,  $O_i$  is well-formed*
3. *there does not exist any ontology  $O_i \in \text{anc}(O)$ , where  $O_i = \langle V_i, A_i, E_i, P_i, B_i \rangle$ , such that there is an  $O_j \in \text{anc}(O)$  and  $O_j \in P_i$ .*

We will now provide meaning for ontologies by defining models of ontologies. Recall that in logic, a model of a theory is an interpretation that satisfies every formula in the theory. Thus, we must first determine the structure of our interpretations. One possibility is to use interpretations of  $\mathcal{L}^V$ , the language formed using only the non-logical symbols in  $V$ . However, this will limit our ability to compare interpretations of different ontologies, so we will instead use interpretations for  $\mathcal{L}^S$ . As such, each interpretation contains mappings for any predicate symbol in any ontology, and thus a single interpretation could be a model of two ontologies with distinct vocabularies.

For ontology extension to have its intuitive meaning, all models of an ontology should also be models of every ontology extended by it.

**Definition 7.** *Let  $O = \langle V, A, E, P, B \rangle$  be an ontology and  $\mathcal{I}$  be an interpretation of  $\mathcal{L}^S$ . Then  $\mathcal{I}$  is a model of  $O$  iff  $\mathcal{I}$  satisfies every formula in  $A$  and  $\mathcal{I}$  is a model of every ontology in  $E$ .*

Thus an ontology attempts to describe a set of possibilities by using axioms to limit its models. Some subset of these models are those intended by the ontology, and are called the intended models of the ontology. Note that unlike a first-order logic theory, an ontology can have many intended models because it can be used to describe many different states of affairs [4]. The axioms of an ontology limit its models by restricting the relations that directly or indirectly correspond to predicates in its vocabulary, while allowing models that provide for any possibility for predicate symbols in other domains.

---

<sup>2</sup> We use  $\mathcal{P}$  to refer to the powerset.

### 3.2 Resource Definitions

**Definition 8 (Knowledge Function).** Let  $K : R \rightarrow \mathcal{P}(L(S))$  be a function that maps each information resource into a set of well-formed formulas.

We call  $K$  the *knowledge function* because it provides a set of formulas for a resource. For example, this could be the first order logic expressions that correspond to the RDF syntax of the resource.

Now we need to associate an ontology with each resource.

**Definition 9 (Commitment Function).** Let  $R$  be the set of information resources and  $\mathcal{O}$  be the set of ontologies. The commitment function  $C : R \rightarrow \mathcal{O}$  maps resources to ontologies.

We call this the *commitment function* because it returns the ontology that a particular resource commits to. When a resource commits to an ontology, it agrees to the meanings ascribed to the symbols by that ontology.

The vocabulary that a well-formed resource may use is limited by the ontology to which it commits.

**Definition 10.** An information resource  $r \in R$  is well-formed iff

1. there exists an  $O \in \mathcal{O}$  such that  $C(r) = O$
2.  $O$  is well-formed
3.  $K(r)$  is well-formed with respect to  $O$ .

We now wish to define the semantics of a resource. When a resource commits to an ontology, it has agreed to the terminology and definitions of the ontology. Thus every interpretation that is a model of resource must also be a model of the ontology for that resource.

**Definition 11.** Let  $r$  be a resource where  $C(r) = O$  and  $\mathcal{I}$  be an interpretation of  $\mathcal{L}^S$ .  $\mathcal{I}$  is a model of  $r$  iff  $\mathcal{I}$  is a model of  $O$  and  $\mathcal{I}$  satisfies every formula in  $K(r)$ .

A possible limitation imposed by the commitment function is that it only allows each resource to commit to a single ontology. However, a virtual ontology can be created that represents multiple ontologies committed to by a single resource. If we assume that committing to two ontologies means that the vocabulary from the resource can come from either ontology and that its models must be models of both ontologies, then committing to two ontologies  $O_1 = \langle V_1, A_1, E_1, P_1, B_1 \rangle$  and  $O_2 = \langle V_2, A_2, E_2, P_2, B_2 \rangle$  is equivalent to committing to the ontology  $O_{union} = \langle \emptyset, \emptyset, \{O_1, O_2\}, \emptyset, \emptyset \rangle$ . The well-formed formulas of  $O_{union}$  are equivalent to well-formed formulas of the union of  $O_1$  and  $O_2$ , and the models of  $O_{union}$  are precisely the intersection of the models of  $O_1$  and  $O_2$ .

In section 4, we will provide a semantics for ontology extension and versioning in terms of ontology perspectives.

### 3.3 Relationship to RDF(S) and OWL

Since OWL is essentially a description logic, all OWL axioms can be expressed in  $\mathcal{L}^S$ . Unfortunately, OWL does not distinguish between resources and ontologies; instead all OWL documents are ontologies.<sup>3</sup> We reconcile this situation by introducing a rule:

<sup>3</sup> This decision was not endorsed by all members of the working group that created OWL



any OWL document that contains a description of a class or a property is an ontology; otherwise, the document is a resource.

After having made this distinction, some language constructs can be associated to the ontology definition (Definition 2). From an OWL DL ontology, we can construct an ontology  $O = \langle V, A, E, P, B \rangle$

1. for each symbol  $s$  in `rdf:ID`, `rdf:about` and `rdf:resource`,  $s \in V$
2. for each axiom  $a$  in class axioms and property axioms,  $a \in A$
3. for each  $O_i$  in triple  $\langle O \text{ owl : imports } O_i \rangle$ ,  $O_i \in E$
4. for each  $O_i$  in triple  $\langle O \text{ owl : priorVersion } O_i \rangle$ ,  $O_i \in P$
5. for each  $O_i$  in triple  $\langle O \text{ owl : backwardCompatibleWith } O_i \rangle$ ,  $O_i \in P$  and  $O_i \in B$ .

Note if the subject of `owl:imports` is a resource document in our definition, then  $\langle r \text{ owl : imports } O \rangle \rightarrow C(r) = O$ . In the case that a resource document imports multiple ontologies,  $C(r) = O_{union}$ .  $O_{union}$  is a virtual ontology where

$$O_{union} = \langle \emptyset, \emptyset, \{O_i | O_i \in \langle r \text{ owl : imports } O_i \rangle\}, \emptyset, \emptyset \rangle.$$

RDF and RDFS have a model theoretic semantics [5], but do not have any explicit support for versioning, nor do they have explicit semantics for inter-document links. One possible way to assign such a semantics is to treat each RDF schema as an ontology and each reference to a namespace as an implicit commitment or extension relationship.

## 4 Ontology Perspectives

Now that we have defined models of individual resources and ontologies, we need to return our attention to a set of distributed information resources, as is the case with a semantic web space. What does it mean to be a model of multiple resources? The first author [6] has suggested that there should not be a universal model of all the resources and ontologies on the Web. In fact, it is extremely unlikely that one could even exist. Instead, we must allow for different viewpoints and contexts, which are supported by different ontologies. He defines perspectives which then allow the same set of resources to be viewed from different contexts, using different assumptions and background information. However, This definition of perspectives was somewhat ad-hoc, and did not have a solid theoretical basis. Here we present a model theoretic description of perspectives.

Each perspective will be based on an ontology, hereafter called the basis ontology or base of the perspective. By providing a set of terms and a standard set of axioms, an ontology provides a shared context. Thus, resources that commit to the same ontology have implicitly agreed to share a context. When it makes sense, we also want to maximize integration by including resources that commit to different ontologies. This includes resources that commit to ancestor ontologies and resources that commit to earlier versions of ontologies that the current ontology is backward compatible with.

Given these guidelines, we can define an ontology perspective model of a set of resources that incorporates backward-compatibility. This model cannot be based solely on the models of some subset of the resources. Since one of the problems we want to be able to solve is to correct an ontology that has incorrectly specified its models, we need



to be able to substitute the models for a new version of an ontology where the models of the older version sufficed before. We will start with a supporting definition that allows us to define the model of a resource assuming the ontology it commits to has been replaced with a newer, backward-compatible one.

**Definition 12.** Let  $r$  be an information resource,  $O$  be an ontology such that  $C(r) = O$ ,  $O' = \langle V', A', E', P', B' \rangle$  be an ontology such that  $O \in B'$ . An  $O'$ -updated model of  $r$  is an interpretation  $\mathcal{I}$  such that  $\mathcal{I}$  is a model of  $O'$  and  $\mathcal{I}$  satisfies every formula in  $K(r)$ .

We can now define an ontology perspective model of a semantic web space. Any perspective model must also be an updated model of those resources that commit to ontologies with which the basis ontology is backwards-compatible, and updated models of those resources that commit to ontologies that the base's ancestor ontologies are backwards-compatible with. In essence the following definitions define the semantics of the document links we have discussed by determining the logical consequences of a semantic web space based on the extensions, backward-compatibilities and commitment relations. Note, prior versions play no role, and thus are essentially "semantic-free".

**Definition 13 (Ontology Perspective Model).** An interpretation  $\mathcal{I}$  is an ontology perspective model of a semantic web space  $\mathcal{W} = \langle \mathcal{O}, R \rangle$  based on  $O = \langle V, A, E, P, B \rangle$  (written  $\mathcal{I} \models_O \mathcal{W}$ ) iff:

1.  $\mathcal{I}$  is a model of  $O$
2. for each  $r \in R$  such that  $C(r) = O$  or  $C(r) \in \text{anc}(O)$  then  $\mathcal{I}$  is a model of  $r$ .
3. for each  $r \in R$  such that  $C(r) \in B$  then  $\mathcal{I}$  is an  $O$ -updated model of  $r$
4. for each  $r \in R$  such that  $\exists O_i, O_i = \langle V_i, A_i, E_i, P_i, B_i \rangle \in \text{anc}(O) \wedge C(r) \in B_i$ , then  $\mathcal{I}$  is an  $O_i$ -updated model of  $r$

Note the first part of the definition is needed because there may be no resources in  $R$  that commit to  $O$ , and without this condition,  $O$  would not have any impact on the models of the perspective.

We now turn our attention to how one can reason with ontology perspectives.

**Definition 14 (Ontology Perspective Entailment).** Let  $\mathcal{W} = \langle \mathcal{O}, R \rangle$  be a semantic web space and  $O$  be an ontology. A formula  $\phi$  is a logical consequence of the ontology perspective of  $\mathcal{W}$  based on  $O$  (written  $\mathcal{W} \models_O \phi$ ) iff for every interpretation  $\mathcal{I}$  such that  $\mathcal{I} \models_O \mathcal{W}$ ,  $\mathcal{I} \models \phi$ .

It would also be useful if we can define a first order logic theory that is equivalent to the definition above.

**Definition 15 (Ontology Perspective Theory).** Let  $\mathcal{W} = \langle \mathcal{O}, R \rangle$  be a semantic web space where  $\mathcal{O} = \{O_1, O_2, \dots, O_n\}$  and  $O_i = \langle V_i, A_i, E_i, P_i, B_i \rangle$ . An ontology perspective theory for  $\mathcal{W}$  based on  $O_i$  is:

$$\begin{aligned} \Phi = & A_i \cup \bigcup_{\{j \mid O_j \in \text{anc}(O_i)\}} A_j \cup \bigcup_{\{r \in R \mid C(r) \in B_i\}} K(r) \\ & \cup \bigcup_{\{r \in R \mid C(r) = O_i \vee C(r) \in \text{anc}(O_i)\}} K(r) \\ & \cup \bigcup_{\{r \in R \mid \exists j, O_j \in \text{anc}(O_i) \wedge C(r) \in B_j\}} K(r) \end{aligned}$$

These theories describe a perspective based on a selected ontology. Each theory contains the axioms of its basis ontology, the axioms of its ancestors, the assertions of all resources that commit to the basis ontology or one of its ancestors, and the assertions of all resources that commit to an ontology with which the basis ontology is backward compatible, or one of its ancestors is backward compatible.

**Theorem 1.** *Let  $\mathcal{W} = \langle \mathcal{O}, R \rangle$  be a semantic web space,  $O = \langle V, A, E, P, B \rangle$  be an ontology, and  $\Phi$  be a ontology perspective theory for  $\mathcal{W}$  based on  $O$ . Then  $\Phi \models \phi$  iff  $\mathcal{W} \models_O \phi$ .*

**PROOF.** (Sketch) We can show that the sets of models are equivalent. The models of Definition 13, part 1 are exactly the models of  $A_i \cup \bigcup_{\{j \mid O_j \in \text{anc}(O_i)\}} A_j$  (from Definition 7). From Definition 11 we can conclude that the models of both part 1 and 2 are exactly the models of  $A_i \cup \bigcup_{\{j \mid O_j \in \text{anc}(O_i)\}} A_j \cup \bigcup_{\{r \in R \mid C(r) = O_i \vee C(r) \in \text{anc}(O_i)\}} K(r)$

From Definition 12 we can conclude that models of part 3 are  $A_i \cup \bigcup_{\{r \in R \mid C(r) \in B_i\}} K(r)$ . Finally, from the same definition, we can conclude that the models of part 4 are exactly the models of  $\bigcup_{\{j \mid O_j \in \text{anc}(O_i)\}} A_j \cup \bigcup_{\{r \in R \mid \exists j, O_j \in \text{anc}(O_i) \wedge C(r) \in B_j\}} K(r)$ . The union of these axioms corresponds to the conjunction of the conditions, and when simplified is equivalent to the theory specified in Definition 15.

The implication of Definition 15 is that we do not need special purpose reasoners to perform ontology perspective reasoning. Since the entailments of an ontology perspective theory are equivalent to ontology perspective entailment (by Theorem 1), you can create the corresponding FOL theory and use an FOL theorem prover to reasoner about ontology perspective entailments. Furthermore, if we restrict the logic used in defining the resources and ontologies, then we should be able to use the more efficient reasoners that correspond to these restricted logics.

## 5 Example

We will reconsider the “Dolphin” example from Figure 1 using our definitions. In Figure 2, Ontology  $O_A$  is the original ontology which contains an incorrect axiom. Ontology  $O'_A$  is a backward compatible revision of  $O_A$  (as indicated by the fifth element in the tuple) that replaces the offending axiom with a correct one.

What are the logical consequences of  $r$  in Figure 2? That depends on what perspective we choose. The ontology perspective theory of  $\mathcal{W}$  based on  $O_A$  is simply

$$\Phi_{O_A} = \{Dolphin(flipper), Dolphin(x) \rightarrow Fish(x)\}$$

It is clear that  $\Phi_{O_A} \models Fish(flipper)$ , and thus  $\mathcal{W} \models_{O_A} Fish(flipper)$ . On the other hand, the perspective theory of  $\mathcal{W}$  based on  $O'_A$  is:

$$\Phi_{O'_A} = \{Dolphin(flipper), Dolphin(x) \rightarrow Mammal(x)\}$$

Consequently,  $\mathcal{W} \models_{O'_A} Mammal(flipper)$ .

As shown here, semantic web spaces essentially have multiple consequence relations, where each corresponds to the perspective of a particular ontology or version of an ontology. This allows users to decide whether or not they want to reason in terms of the old version of the ontology or the newer version.

$$\begin{aligned}
O_A &= \langle \{Fish, Mammal, Dolphin\}, \\
&\quad \{Dolphin(x) \rightarrow Fish(x)\}, \\
&\quad \emptyset, \\
&\quad \emptyset, \\
&\quad \emptyset \rangle \\
O'_A &= \langle \{Fish, Mammal, Dolphin\}, \\
&\quad \{Dolphin(x) \rightarrow Mammal(x)\}, \\
&\quad \emptyset, \\
&\quad \{O_A\}, \\
&\quad \{O_A\} \rangle \\
K(r) &= \{Dolphin(flipper)\} \\
C(r) &= O_A \\
\mathcal{W} &= \langle \{O_A, O'_A\}, \{r\} \rangle
\end{aligned}$$

**Fig. 2.** Ontology revision using the formal model.

Furthermore, the effects are limited to relevant resources. Consider the example where in  $O_A$ , *Dolphin* actually means “dolphin fish”, but in  $O'_A$  it is decided to give it the new meaning “a kind of porpoise.” In this case,  $O'_A$  is not backward-compatible with  $O_A$  because it has different intended interpretations. This is expressed by changing the last element of the  $O'_A$  tuple to  $\emptyset$ . In this case,  $\Phi_{O_A}$  is as above, but  $\Phi_{O'_A}$  is:

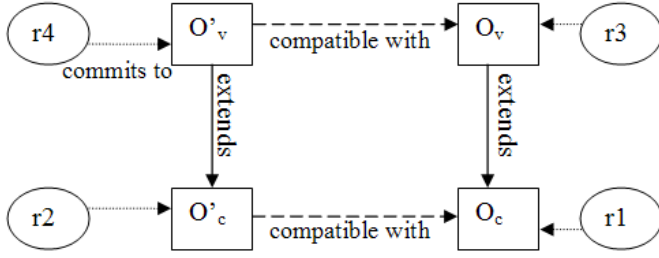
$$\Phi_{O'_A} = \{Dolphin(x) \rightarrow Mammal(x)\}$$

Note, that in this case  $Dolphin(flipper) \notin \Phi_{O'_A}$  because of the definition of ontology perspective models. Consequently,  $\mathcal{W} \not\models_{O'_A} Mammal(flipper)$

Now we turn to a more complex example that combines the extension and versioning (Figure 3 and 4).

The Semantic Web can be used to locate documents for people or to answer specific questions based on the content of the Web. These uses represent the document retrieval and knowledge base views of the Web. The knowledge base view uses the logical definition of queries: a query is a formula with existentially quantified variables, whose answers are a set of bindings for the variables that make the formula true with respect to the knowledge base. But what is a knowledge base in the context of the Semantic Web? In order to resolve a number of problems faced by the Semantic Web we have extensively discussed means of subdividing it. Theoretically, each of these perspectives represents a single model of the world, and could be considered a knowledge base. Thus, the answer to a semantic web query must be relative to a specific perspective.

Consider the set of ontologies and resources presented in Figure 4. There are four ontology perspective theories generated from this semantic web space:  $\Phi_{O_C}$ ,  $\Phi_{O'_C}$ ,  $\Phi_{O_V}$  and  $\Phi_{O'_V}$ . Based on Definition 15, different ontologies and resources appear in each perspective theory. As shown in Fig. 5,  $\Phi_{O_C}$  includes the axioms from  $O_C$  and the knowledge from  $r_1$ .  $\Phi_{O_V}$  includes the axioms from  $O_V$ , and because  $O_C$  is an ancestor of  $O_V$ , those of  $O_C$ . It also includes the resources  $r_1$  and  $r_3$ , which commit to these ontologies. On the other hand,  $\Phi_{O'_C}$  includes axioms only from  $O'_C$ , and the resources  $r_1$  and  $r_2$ . Note that  $\Phi_{O'_C}$  does not include axioms in  $O_C$ , because the theory in Definition



**Fig. 3.** Ontology extension and revision example.

$$\begin{aligned}
 O_C &= \langle \{Car, Convertible\}, \\
 &\quad \{Convertible(x) \rightarrow Car(x)\}, \\
 &\quad \emptyset, \\
 &\quad \emptyset, \\
 &\quad \emptyset \rangle \\
 O'_C &= \langle \{Car, Convertible, SUV\}, \\
 &\quad \{Convertible(x) \rightarrow Car(x), \\
 &\quad SUV(x) \rightarrow Car(x)\}, \\
 &\quad \emptyset, \\
 &\quad \{O_C\}, \\
 &\quad \{O_C\} \rangle \\
 O_V &= \langle \{Vehicle, Motorcycle\}, \\
 &\quad \{Motorcycle(x) \rightarrow Vehicle(x) \\
 &\quad O_C : Car(x) \rightarrow Vehicle(x)\}, \\
 &\quad \{O_C\}, \\
 &\quad \emptyset, \\
 &\quad \emptyset \rangle \\
 O'_V &= \langle \{Vehicle, Motorcycle, Automobile\}, \\
 &\quad \{Motorcycle(x) \rightarrow Vehicle(x), \\
 &\quad Automobile(x) \rightarrow Vehicle(x), \\
 &\quad Automobile(x) \leftrightarrow O'_C : Car(x)\}, \\
 &\quad \{O'_C\}, \\
 &\quad \{O_V\}, \\
 &\quad \{O_V\} \rangle \\
 C(r_1) &= O_C \\
 K(r_1) &= \{Car(beetle), Convertible(mustang)\} \\
 C(r_2) &= O'_C \\
 K(r_2) &= \{Car(jetta), Convertible(z3), SUV(yukon)\} \\
 C(r_3) &= O_V \\
 K(r_3) &= \{Vehicle(humvee), Motorcycle(yamaha)\} \\
 C(r_4) &= O'_V \\
 K(r_4) &= \{Vehicle(bradley), Motorcycle(harley), Automobile(s600)\}
 \end{aligned}$$

**Fig. 4.** Example ontologies and resources.

Theories	Perspective			
	$\Phi_{O_C}$	$\Phi_{O'_C}$	$\Phi_{O_V}$	$\Phi_{O'_V}$
<i>Axioms</i>	$A_{O_C}$	$A_{O'_C}$	$A_{O_C}, A_{O_V}$	$A_{O'_C}, A_{O'_V}$
<i>Knowledge</i>	$K(r_1)$	$K(r_1), K(r_2)$	$K(r_1), K(r_3)$	$K(r_1), K(r_2), K(r_3), K(r_4)$

**Fig. 5.** Theories based on different perspectives.

15 does not contain the prior version's axioms.  $\Phi_{O'_V}$  includes axioms from  $O'_V$  and  $O'_C$ , and the resources  $r_1, r_2, r_3$  and  $r_4$ .

As a result, the answer to any particular query depends on which perspective it is issued against. For example, the answers to  $\text{Car}(x)$  based on  $O_C$ 's perspective will be {beetle, mustang}. The answers to same query based on  $O'_C$ 's perspective will be {jetta, z3, yukon, beetle, mustang}. The answers to  $\text{Vehicle}(x)$  based on  $O_V$ 's perspective will be {humvee, yamaha, beetle, mustang}, while the answers to that query based on  $O'_V$ 's perspective will be {bradley, harley, s600, humvee, yamaha, jetta, z3, yukon, beetle, mustang}.

## 6 Related Work

Prior to the concept of the Semantic Web, there was little work related to ontology versioning. Perhaps this is because most ontology systems were centralized, and in such cases versioning is less of an issue. One exception is CONCORDIA [10], which provides a model of change with applications in medical taxonomies. However, this model is limited to simple taxonomic ontologies and does not have a formal semantics.

Klein and Fensel [8] were the first to compare ontology versioning to database schema versioning [11]. They proposed that both prospective use (viewing data from the point of view of a newer ontology) and retrospective use (viewing data from the point of view of an older ontology) of data should be considered. However, Klein and Fensel do not describe a formal semantics.

Stuckenschmidt and Klein [12] provide a formal definition for modular ontologies and consider the impact of change in it. However, their approach involves physical inclusion of extended ontologies and requires that changes be propagated through the network. This approach is unlikely to scale in large, distributed systems. Furthermore, they do not allow for resources to be reasoned with using different perspectives, as is described here.

Bouquet et al. [2] have also argued that it often does not make sense to have a global interpretation of a set of on semantic web ontologies. Their solution involves creating local interpretations and providing bridge rules between different contexts. By contrast, our paper shows that ontologies can provide the benefits of contexts too. Furthermore, Bouquet et al. do not consider the impact of versioning.

An orthogonal problem to the one described in this paper is how to determine the correspondences and differences between two different versions of an ontology. PROMPT-DIFF [9] is an example algorithm intended for that purpose.

## 7 Conclusion

We have discussed the problem of ontology versioning and in particular the need for a formal semantics of the links that can exist between ontologies. We mathematically described the components of the Semantic Web and provided a model theoretic semantics for these components. We also discussed how this semantics relates to RDF(S) and OWL. We then showed how Heflin's perspectives [6] could also be defined using model theoretic semantics. This provided a formal account for three kinds of links between semantic web documents: commitment to an ontology by a resource, extension of one ontology by another, and backward-compatibility of an ontology with a prior version. We described a simple approach for reasoning with perspectives and finally gave examples to illustrate their utility.

There are a number of directions for future work. First, we intend to look at other kinds of versioning. For example, we will consider a model theoretic semantic for retrospective as well as prospective use of data. The incompatible prior versions will also be considered. A possible approach would be introducing deprecation, which is similar to the retired concept in CONCORDIA [10]. Second, although our theory presents a good logical account of reasoning with distributed ontologies, it has some practical limitations. Ontology perspectives divide the information resources into subsets that were likely to have the same viewpoint, but still cannot guarantee consistency. Since perspectives could be very large, containing millions of resources, it is a shame if a single inconsistency trivialized the whole thing. An important problem is to find a theory that allows useful reasoning in the face of these inconsistencies. Third, we will look at how a model theoretic approach can be used as a basis for beginning to formalize semantic notions of trust.

**Acknowledgement.** This work was supported by the Defense Advanced Research Projects Agency under a subcontract from ISX Corporation and by the National Science Foundation under Grant No. IIS-0346963.

## References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, May 2001.
2. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. C-OWL: Contextualizing ontologies. In *Proc. of the 2003 Int'l Semantic Web Conf. (ISWC 2003)*, LNCS 2870, pages 164–179. Springer, 2003.
3. R. Brachman. What IS-A is and isn't: An analysis of taxonomic links in semantic networks. *IEEE Computer*, 16(10):30–36, October 1983.
4. N. Guarino. Formal ontology and information systems. In *Proceedings of Formal Ontology and Information Systems*, Trento, Italy, June 1998. IOS Press.
5. P. Hayes. RDF semantics. Proposed Recommendation, December 2003. <http://www.w3.org/TR/2003/PR-rdf-schema-20031215/>.
6. J. Heflin. *Towards the Semantic Web: Knowledge Representation in a Dynamic, Distributed Environment*. PhD thesis, University of Maryland, 2001.

7. Jeff Heflin and James Hendler. Dynamic ontologies on the Web. In *Proc. of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, pages 443–449, Menlo Park, CA, 2000. AAAI/MIT Press.
8. M. Klein and D. Fensel. Ontology versioning for the semantic web. In *Proc. of the 1st Int'l Semantic Web Working Symposium (SWWS)*, pages 75–91, 2001.
9. N. Noy and M. Musen. PROMPTDIFF: A fixed-point algorithm for comparing ontology versions. In *Proc. of the Eighteenth National Conference on Artificial Intelligence (AAAI-2002)*, Menlo Park, CA, 2002. AAAI/MIT Press.
10. D. E. Oliver, Y. Shahar, M. A. Musen, and E. H. Shortliffe. Representation of change in controlled medical terminologies. *Artificial Intelligence in Medicine*, 15(1):53–76, 1999.
11. J. Roddick. A survey of schema versioning issues for database systems. *Information and Software Technology*, 37(7):383–393, 1995.
12. H. Stuckenschmidt and M. Klein. Integrity and change in modular ontologies. In *Proc. IJCAI'03*, Acapulco, Mexico, 2003.