1-1-2007

Ontology revision on the semantic web: integration of belief revision theory

S. H. Kang
University of Wollongong

S. Lau
University of Wollongong, simlau@uow.edu.au

Abstract
Ontology is used to define terms and relations on the Semantic Web to form well-structured semantics of Web resources. Ontology revision refers to the process of updating ontology to ensure changes are made in a consistent manner. Belief revision theory deals with approaches to ensure consistency in the belief sets is maintained when beliefs need to be revised. This paper discusses the integration of belief revision theory to the ontology reengineering method as a means to ensure consistency in ontology revision.

Publication Details
This paper was originally published as: Kang, SH & Lau, SK, Ontology revision on the semantic web: integration of belief revision theory, 40th Annual Hawaii International Conference on System Sciences (HICSS 2007), Waikoloa, Hawaii, USA, January 2007, 61. Copyright 2007 IEEE.

Ontology Revision on the Semantic Web:

Integration of belief revision theory

Seung Hwan Kang
University of Wollongong
3k33@uow.edu.au

Sim Kim Lau
University of Wollongong

Abstract

Ontology is used to define terms and relations on the Semantic Web to form well-structured semantics of Web resources. Ontology revision refers to the process of updating ontology to ensure changes are made in a consistent manner. Belief revision theory deals with approaches to ensure consistency in the belief sets is maintained when beliefs need to be revised. This paper discusses the integration of belief revision theory to the ontology reengineering method as a means to ensure consistency in ontology revision.

1. Introduction

The proliferation of the World Wide Web (WWW) has resulted in a highly heterogeneous and distributed information-seeking and information-distribution environment. As a result, a more structured approach to facilitate machine-enabled searching and querying capabilities is required. Thus the Semantic Web has been developed to support machine-processable global information exchange. In the Semantic Web, ontology deals with relationships and descriptions of web resources by providing a way to define meanings, structures and semantics of web resources. Each web resource now has a more meaningful identification to allow relationships to be linked and thus improved on information searching and querying. This way semantically rich and descriptive information with any web resources can be associated and referenced to allow automated machine processing (W3C 2005).

Ontology is defined as an explicit specification of a conceptualization (Gruber 1993 p.2). It enables knowledge sharing and reuse by allowing software agents to share descriptions and relationships of terms and concepts of a particular domain within the community of practice. Ontological commitment thus enables software agents to communicate and function through formal definitions of terms. Due to the increasing importance of the role of the WWW as knowledge provider, many organizations need to ensure their web resources are kept up-to-date and be able to be referenced without ambiguity. However constant changes of business dynamics and application requirements mean effective mechanisms that can handle ontology inter-operability and multiple ontologies are required. In general, ontology can evolve as a result of changes in domain, conceptualization and specification (Klein and Fensel 2001). One may argue that conceptualization in ontology should be well planned and defined in the designing phase of any web-based systems, however software agents in machine-processable environment is capable of learning to gain new knowledge through the process of information seeking. When learning occurs, the knowledge gained can lead to changes of conceptualization thus resulting in the needs to revise ontology. Example of question that can be asked when such situation arises includes whether reference to a concept should remain valid in the ontology if partial change of relationship is detected.

In this research, we propose an ontology revision framework based on the Alchourrón, Gärdenfors and Makinson (AGM) model of belief revision theory (Gärdenfors 1992, Gärdenfors and Rott 1995). The proposed framework focuses on revising components in ontology through three operators of expansion, contraction and revision. The belief revision theory deals with approaches of changing belief through the process of revising a knowledge base to ensure revision does not cause inconsistency after changes are applied (Segal 1994). It provides a means to ensure new information applied as a result of learning does not result in contradiction of conceptualizations and specifications with the existing system or knowledge base (Gärdenfors 19 , Gärdenfors 1994).

The aim of this research is to investigate the feasibility of integrating the AGM model to the ontological reengineering method proposed by Gómez-Pérez et al. (2004). Firstly the paper will discuss ontology and belief revision theory. Then we will discuss the proposed ontology revision method which integrates the belief revision theory to the ontology revision framework. Then illustrations of the implementation of the proposed approach will be presented. The paper concludes with future research direction.

2. Ontology

According to McCarthy and Hayes (1969), for any computer program to function intelligently it must have a general representation of the world in which its input can be interpreted. Similarly, in order for software agents to function autonomously and intelligently in the distributed heterogeneous environment such as the WWW, agents must know or be able to interpret the meaning of terms referenced in order to prudently communicate and perform tasks either autonomously or in respond to user request. This is only achievable if they can communicate through sharing a commonly agreed term of reference over the Semantic Web. Therefore ontology has been proposed as a way of representing the semantics of web resources and enabling it to be used by web applications and software agents (W3C 2005).

There are different definitions of ontology in the literature, from philosophy to artificial intelligence. In philosophy, ontology is the theory of being. In artificial intelligence, ontology is "an explicit specification of conceptualization" (Gruber 1993 p.2). Borst (1997 p.12) has slightly modified the definition to "a formal specification of a shared conceptualization". Fundamentally, ontology encourages sharing of meaning of terms and concepts in the community of practice to achieve clear understanding of a particular domain (Gómez-Pérez 1999, Nodine and Fowler 2005). As a general rule, ontology deals with describing and distinguishing, providing descriptive analysis and classification of concepts and facts. In the Semantic Web viewpoint, ontology is developed as a way to define the meaning, structure the terms and present semantics of web resources (Heflin and Hendler 2001, Hendler 2001). A formal structure of a web resource can be considered as a set of named relations or schemas and information semantics captured in this structure (Stuckenschmidt 2003).

Our proposition is that even though ontology can be carefully designed and developed, ontology may still need to be revised over time as a result of new knowledge gained. Heflin and Hendler (2000) define ontology revision as a change of components in ontology, which can involve addition and/or removal of categories, relations, and/or axioms. To handle changes in ontologies, ontology versioning and ontology library have been proposed (Ding and Fensel 2001, Klein et al. 2002). The ontology versioning system allows comparability issues to be taking into consideration when new knowledge is added to the system over time. The ontology library system manages, adapts and standardises collections of ontologies. However, the use of these approaches does not present a way to consistently revise ontology.

We propose to handle ontology revision based on the belief revision theory. The components in the ontology represent the beliefs in the systems. Through learning, definitions of conceptualization and/or relationships between components of concepts may need to be revised to reflect the changes. This is similar to the changes of knowledge in the belief sets.

3. Belief Revision Theory

From historical viewpoint of belief revision, there are two belief revision theories: foundation theory and coherent theory. The foundation theory of belief revision models the dynamics of epistemic states by keeping track of justifications for, and logical structure of beliefs (Doyle 1979). Whereas, the coherence theory of belief revision highlights semantics in a form of logically consistent structure (Gärdenfors and Rott 1995). Its rationale is that all justification of beliefs relies on coherence within a belief system. It is a holistic view in which the basic of justification in a systematic network of beliefs can be justified via coherence that offers an idea for other justified beliefs. This research is based on the coherence theory, in particular, the AGM model of the coherence theory (Gärdenfors 1992, Gärdenfors and Rott 1995).

Let a belief set K be represented by a set of sentences in the logical language L, which contains the standard logical connectives: negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), implication ($\rightarrow$), and two truth values of truth (T) and falsity ($\perp$). In a consistent belief set K, there are three possible epistemic states (accepted, rejected or unknown) towards a logical sentence *p*. The idea of truth is its coherence in the belief set, which means the truth of *p* depends on coherence between *p* and other beliefs in the coherent set. A set of sentences is a belief set K if and only if (i) ⊥ is not a logical consequence of the sentences in *K,* and (ii) if $K \vdash q$, then $q \in K$. Accepting *p* in *K* refers to accepting a proposition *p* in an epistemic state, that is there is no doubt that *p* is true in *K*. Rejecting *p* means negation of *p* $(\neg p)$ is *true* in *K*; *p* is *unknown* means both accepting *p* and $\neg p$ are not possible because it results in inconsistency in *K*. The set of accepted sentences in *K* should be logically consistent so that it is possible to draw consequences of what is accepted.

There are three types of belief changes in the coherence theory: expansion, contraction and revision. Firstly, expansion occurs through learning of new information. A sentence *A* can be changed from the state of *unknown* to that of *accepted* during the expansion operation. The belief set that results from expansion of *K* by a sentence *A* is denoted by $K+_A$. Secondly, revision refers to the need to revise the belief set when by introducing a new concept it results in contradiction between the new and existing concepts in the belief set. In this case, the resultant belief set from revision of *K* by a sentence *A* is denoted by $K*_A$. Finally, contraction refers to retracting one or more sentences from the belief set to ensure the resulting belief set is closed under logical consequences. The belief set that results from contraction of *K* by a sentence *A* is denoted by K^-_A. Tables 1, 2 and 3 show the postulates that should be satisfied to meet the requirements of expansion, revision and contraction operations respectively.

Table 1 Postulates of expansion function.

(K+1)	For any sentence A and any belief set K, K^+_A is a belief set.	(Closure)
(K+2)	$A \in K^+_A$.	(Success)
(K+3)	$K \subseteq K^+_A$.	(Expansion)
(K+4)	If $A \in K$, then $K^+_A = K$.	(Inclusion 1)
(K+5)	If $K \subseteq H$, then $K^+_A \subseteq H^+_A$.	(Inclusion 2)
(K+6)	For all belief sets K and all sentences *A*, K^+_A is the smallest belief set that satisfies (K+1) – (K+5).	(Representation)

Table 2 Postulates of revision function

(K 1)	For any sentence A and any belief set K, K_A is a belief set.	(Closure)
(K 2)	$A \in K_A$.	(Success)
(K 3)	$K_A \subseteq K^+_A$.	(Expansion 1)
(K 4)	If $\neg A \notin K$, then $K^+_A \subseteq K_A$.	(Expansion 2)
(K 5)	$K_A = K_\perp$ if and only if $\vdash \neg A$.	(Consistency Preservation)
(K 6)	If $\vdash A \leftrightarrow B$, then $K_A = K_B$.	(Extensionality)
(K 7)	$K_{A \wedge B} \subseteq (K_A)^+_B$.	(Conjunction 1)
(K 8)	If $\neg B \notin K_A$, then $(K_A)^+_B \subseteq K_{A \wedge B}$.	(Conjunction 2, Rational Monotony)

Table 3 Postulates of contraction function.

(K-1)	For any sentence A and any belief set K, K^-_A is a belief set	(Closure)
(K-2)	$K^-_A \subseteq K$.	(Inclusion)
(K-3)	If $A \notin K$ then $K^-_A = K$.	(Vacuity)
(K-4)	If $\nvdash A$, then $A \notin K^-_A$.	(Success)
(K-5)	If $A \in K$, then $K \subseteq (K^-_A)^+_A$.	(Recovery)
(K-6)	If $\vdash A \leftrightarrow B$, then $K^-_A = K^-_B$.	(Extensionality)
(K-7)	$K^-_A \cap K^-_B \subseteq K^-_{A \wedge B}$.	(Conjunction 1)
(K-8)	If $A \notin K^-_{A \wedge B}$, then $K^-_{A \wedge B} \subseteq K^-_A$.	(Conjunction 2)

The first postulate of each operation requires the resultant belief set be a consistent belief set. According to Gärdenfors (19 p.49), belief should be retained as much as possible and unnecessary loss of information are to be avoided in the process of belief revision. This is often termed as the criterion of informational economy. In the case of expansion, the postulates (K+4) and (K+5) are referred as the inclusion principle and the postulate (K+6) means we should ensure the resultant belief set is the smallest belief set.

The first six postulates for the revision operator can be viewed as similar to that of the expansion operator. The important aim is to ensure that the revision operation produce a new consistent belief set. More importantly the postulates (K*7) and (K*) are concerned with composite belief revisions that express a revision as a form of expansion.

Finally in the contraction operation, the concept of epistemic entrenchment needs to be considered. The degree of epistemic entrenchment formally represents the relative importance of a sentence in the belief set. This ordering depends on the importance of that knowledge and belief. The basic idea here is that one particular belief can give more valuable information than others in the belief set. In the belief revision theory it is important to first revise sentences that are epistemologically less entrenched (Gärdenfors 19 p.67). In each case when a new belief is considered by a belief revision operator, a ranking for the new belief will be assigned based on its entrenchment ordering. In applying the contraction operator, epistemologically least entrenched sentence is retracted first to allow minimal loss of information. Table 4 shows the postulates of epistemic entrenchment. These postulates express the transitive, dominance, conjunctive, minimality and maximality relationships in the belief sets.

Table 4 Postulates of Epistemic Entrenchment.

(EE1)	For any A, B, and C, if $A \leq B$ and $B \leq C$, then $A \leq C$.	(Transitivity)
(EE2)	For any A and B, if $A \vdash B$, then $A \leq B$.	(Dominance)
(EE3)	For any A and B in K, $A \leq A \wedge B$ or $B \leq A \wedge B$.	(Conjunctiveness)
(EE4)	When $K \neq K_{\perp}$, $A \notin K$ iff $A \leq B$, for all B.	(Minimality)
(EE5)	if $B \leq A$ for all B, then $\vdash A$.	(Maximality)

3.1 Example

Consider a person who initially has the following beliefs (represented by *α, β, γ and δ* in a belief set *K*):

α: All music players are electronic products.
β: The music player displayed in the shop is iPod.
γ: The music player displayed in the shop is an Apple product.
δ: Apple belongs to the electronic industry.

Given the above four sentences, we can infer and add the following new sentence *ε* (where *ε: The music player displayed in the shop is an electronic product*) to *K*. In this case we said that *K* is expanded by *ε*. Now let us imagine that the person learns that the music player displayed in the shop is actually a MP3 player, and not iPod as he originally believed in. Furthermore he also learns that the MP3 player is classified in the category of computer. Thus the sentence *ε* is no longer consistent in his belief set and there is a need to add negation of *ε* ($\neg\varepsilon$) to the belief set. The addition of $\neg\varepsilon$ requires an expansion to be operated on *K*. Let us rename $\neg\varepsilon$ as *ϕ* (where *ϕ: The music player displayed in the shop is not an electronic product*). In this case the resultant belief set is now consists of *α, β, γ, δ, ε and ϕ*.

However this resultant belief set consists of inconsistent sentences of *α* and *ϕ*. Therefore the person needs to revise his belief set to allow all sentences in *K* to be consistent. In this case the belief set is revised by adding a new sentence *α′* (where *α′: All music players except the one displayed in the shop are electronic products*) to *K*. Therefore the resultant belief set will now consist of sentences: *α, β, γ, δ, ε, ϕ* and *α′*.

At this point, it is found that the resultant belief set still contains inconsistent sentences *α* and *ϕ*. Therefore we need to retract one of these sentences from the belief set. In determining which sentence to retract, we have to make the decision based on the principle of epistemic entrenchment to resolve which sentence holds more valuable information. It is found that *ϕ* holds more valuable information because it identifies the item itself as not an electronic item, thus *ϕ* is considered as a more entrenched sentence compared to *α*. Thus *α* is retracted and the resultant belief set consists of sentences: *β, γ, δ, ε, ϕ* and *α′*.

4. Proposed Ontology Revision Framework

In this research the belief revision theory is integrated to the ontological reengineering method proposed by Gómez-Pérez et. al. (2004). The method consists of three phases: reverse engineering, restructuring and forward engineering. The first phase, the reverse engineering, derives the ontology conceptual model from its implementation code. This phase analyses an existing ontology to identify its components and their relations to create a conceptual model as a representation of ontology at a higher level of abstraction. The second phase of restructuring, evaluates the conceptual model of ontology. The third phase of forward engineering transforms the new conceptual model to the new ontology. We propose to integrate the belief revision theory to the restructuring phase of the ontological reengineering method. This way we aim to consistently revise the ontology when changes occur. Figure 1 shows the proposed ontology revision method in which the three belief revision operators of expansion, contraction, and revision are embedded in the restructuring phase, which we will call it the *revise* phase.

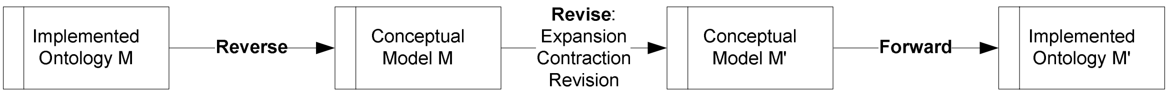

Figure 1 Proposed Ontology Revision Framework

In Figure 1, the first phase of *reverse* phase is to derive the ontology conceptual model from its implemented ontology *M*. Here, the concept hierarchy is used to present the parent-child relation to illustrate the conceptual relationship of different concepts in the conceptual model. The second phase of *revise* phase revises the initial conceptual model *M* to that of a new one which we now call *M'*. In our approach, this revision is achieved using the expansion, revision or contraction operator. Epistemic entrenchment is applied in this phase by ranking the concepts in the conceptual model. In a simple term, if α is consistent with the model *m*, its rank is validated for consistency with the rest of the entrenchment. Figures 2, 3 and 4 show the pseudocodes for expansion, contraction and revision operators respectively.

Let *expand*(*m*, α) denotes the expansion of an ontology *M* by a concept α, where *m* is the model of ontology *M*. When new concept α is expanded, α is tested for logical consistency with the current concepts stored in the ontology on the basis that *m* meets the requirements of the postulates. The expansion of α is accepted if and only if it is consistent with the existing ones. As each concept is assigned with an epistemic entrenchment ranking, after the expansion the epistemic ordering of the sentences in the ontology will be reviewed to ensure it remains consistent after the expansion process. The general rule used is if the ranking of the existing concept is greater than and equal to that of the new concept, no expansion is made, otherwise update it to the new rank.

```
expand(m, α)
  IF rank(m, ¬α) THEN
    return(m)
  ELSE
    oldrank = rank(m, α)
    IF oldrank >= newrank THEN
      return (m)
    ELSE
        m′ = update(m, α, newrank)
        IF prove(m′ , β) THEN // β ∈ m′
          FOR each β
            if (β > oldrank)
              m′ = remove(m′ , β)
          ENDFOR
      ENDIF
    ENDIF
  ENDIF
  return(m′)
END
```

Figure 2 Expansion pseudocode.

Let *contract(m, α)* denotes the contraction of an ontology *M* by a concept α, which is no longer valid in the model of ontology *M*. Similar to the expansion operator, the contraction operator must meet the requirement of the postulates as stated in Table 3. In the case of contraction, as it does not add any new concept to the model *m*, the ranking of the original concept will remain as the same entrenchment as they previously had. In addition, if there is any existing child-concept that logically entails from the parent concept, then the child concept will be tested for logical consistency with the parent concept.

```
contract(m, α)
  IF rank(m, ¬α) THEN
    return(m)
  ELSE
    oldrank = rank(m, α)
    FOR each β
      IF prove(m′ , α ∨ β) THEN
        IF (β > oldrank) THEN
          m′ = remove(m′ , β)
        ENDIF
        IF prove(m′ , α → β, oldrank)
          newrank = oldrank + 1
          m′ = update(m, α, newrank)
        ENDIF
      ENDIF
    ENDFOR
  ENDIF
  return(m′)
END
```

Figure 3 Contraction pseudocode.

Let *revise(m, α)* denotes the revision of an ontology *M* revising a concept α. In this case, the revision operator is performed in terms of the contraction and expansion operations as shown by the pseudocode.

```
revise(m, α, newrank)
  return(expand(contract(m, ¬α),
    α, newrank))
END
```

Figure 4 Revision pseudocode.

As a final point, the forward phase transforms the new ontology conceptual model M' to the new implemented ontology. It is a process of transforming high-level abstraction to the physical implementation of ontology using some specific ontological language such as the OWL (Web Ontology Language).

5. Illustrations of Implementation

In our implementation, we have developed the ontology using the Protégé ontology editor (Noy and McGuinness 2001). The Jena 2 ontology API (Application Program Interface) provides a collection of toolkits to build a hierarchy of concepts as well as to manipulate ontologies in the OWL (HPL 2002). To model the implemented ontology, a particular OWL model is created with in-memory storage model using the Jena API. The three revision operations of expansion, contraction and revision are implemented using the Jena API.

Consider a scenario of an online purchase of a digital camera by a buyer agent in the e-commerce environment. This scenario assumes the buyer agent has accessed to ontology that stores and describes the conceptual idea of electronic products, camera, manufacturers of electronic products and so on. We have used the concept hierarchy of parent-child relationship to show the conceptual relationship of different concepts stored in the ontology. Figure 5 shows two concepts of manufacturer and camera in two ontologies in which the buyer agent has accessed to. The left column of Figure 5 shows an ontology that describes *Sony_Style (Sony Australia) is a manufacturer*, *Samsung is a manufacturer,* and *camera is an electronic product*. The right column in the same diagram shows a second ontology *N* which indicate *Dell, Sony* and *LG are manufacturers* and *digital camera is a computer*. We have used two different names (*Sony_Style* and *Sony*) to represent the concept of Sony in the two ontologies.

Now let us consider the request that triggers from a purchase order to the buyer agent is to buy a digital camera manufactured by *LG*. Based on the current information stored in ontology *M* in which the buyer agent has accessed to, it only contains the conceptual model of the camera as *Sony_Style and Samsung are manufacturers* and *camera is an electronic product*. In this instance, ontology *M* does not indicate relationship of *LG* as a manufacturer of the camera and there is no conceptual description of digital camera. Thus the buyer agent will not be able to process the purchase order unless the buyer agent learns new concepts such that *LG is also a manufacturer* and *digital camera is also a type of camera that is described as belongs to the category of computer*. For purpose of illustration let us assume that the buyer agent obtain these new information from ontology *N* and thus wishes to update its ontology by including this new knowledge in its ontology *M.*

Firstly, the new concepts need to be validated by determining whether it is a member of ontology *M*. Here we use arbitrary rank to assign epistemic entrenchment. In this example, *Sony_Style* is assigned a rank of 1 and *Samsung* a rank of 2 in ontology *M.* Similarly, let assume that *Dell* is assigned a rank of 1, *Sony* a rank of 2 and *LG* a rank of 3 in ontology *N*. To expand *LG* into ontology *M*, we first ensure that it is consistent with the existing concept, i.e., *LG* is also a subclass of *Manufacturer*. When it is found to be the case, then *LG* is expanded in ontology *M*. The bottom part of the screen shot in Figure 6 shows the result of the new conceptual model for ontology *M* after *LG* is expanded in *M*.

Next we consider an illustration to remove inconsistencies using the contraction operation. In this example, we will contract the concept of *Electronics* and its associated sub-concept of C*amera*. In this instance if the concept of *Electronics* is retracted, then the concept of *Camera* will also be removed. The bottom part of the screen shot in Figure 7 shows the result of ontology *M* after the contraction operation.

Finally we consider the revision operation. Let us consider adding the concept of *Sony* from ontology *N* to ontology *M*. In our example *Sony* in ontology *N* is assigned a ranking of 2. Compared to the same concept (*Sony_Style*) in ontology *M* (which has been arbitrary assigned a ranking of 1) it therefore has a higher value of epistemic ranking. In this case, the revision operator will first contract the concept of *Sony_Style* in model *M* and then expand the concept of *Sony* from model *N*. Again, the bottom part of the screen shot in Figure   shows the result of the revised ontology *M* after the revision operation.

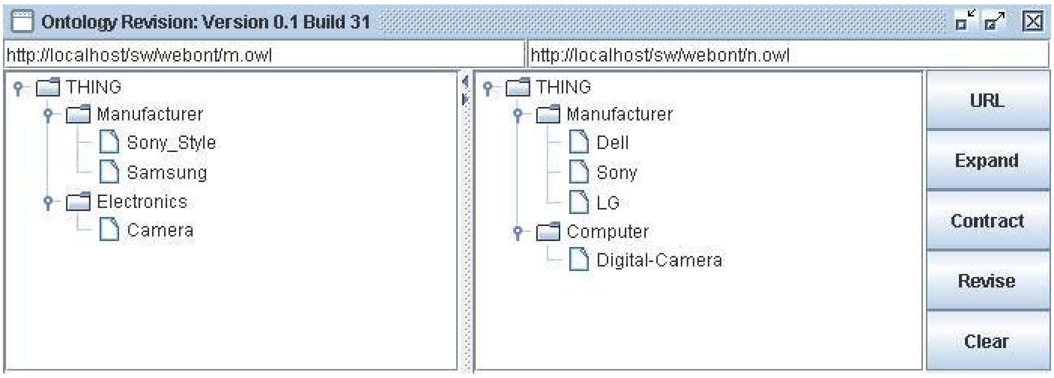

Figure 5 A sample modeled ontology *M* and *N*.

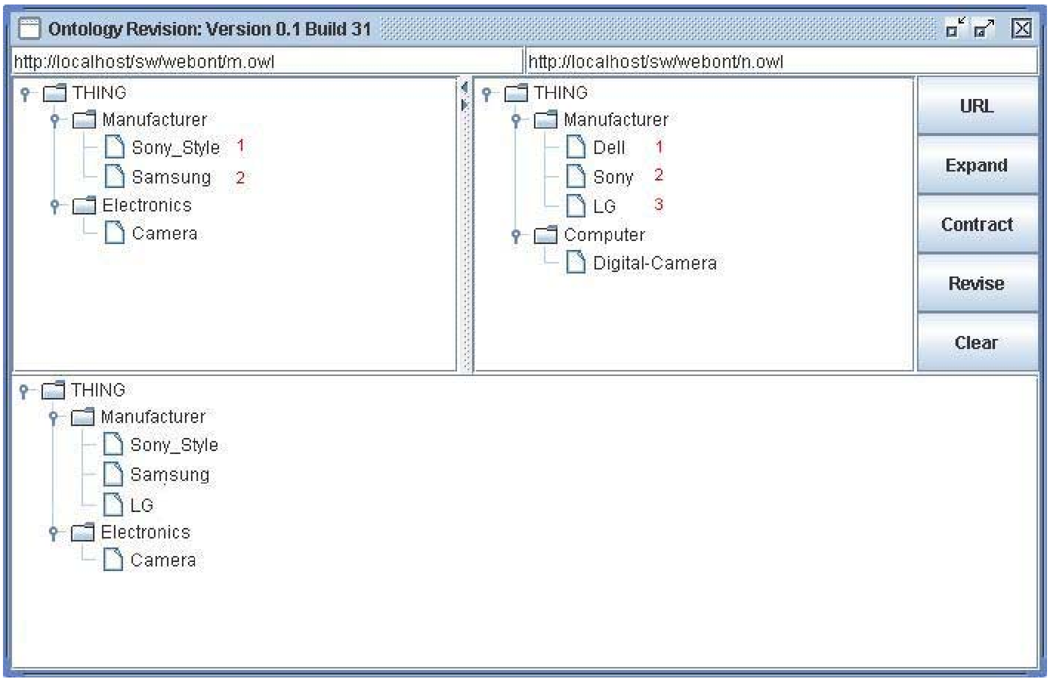

Figure 6 The result of expansion from ontology *N* to *M*.

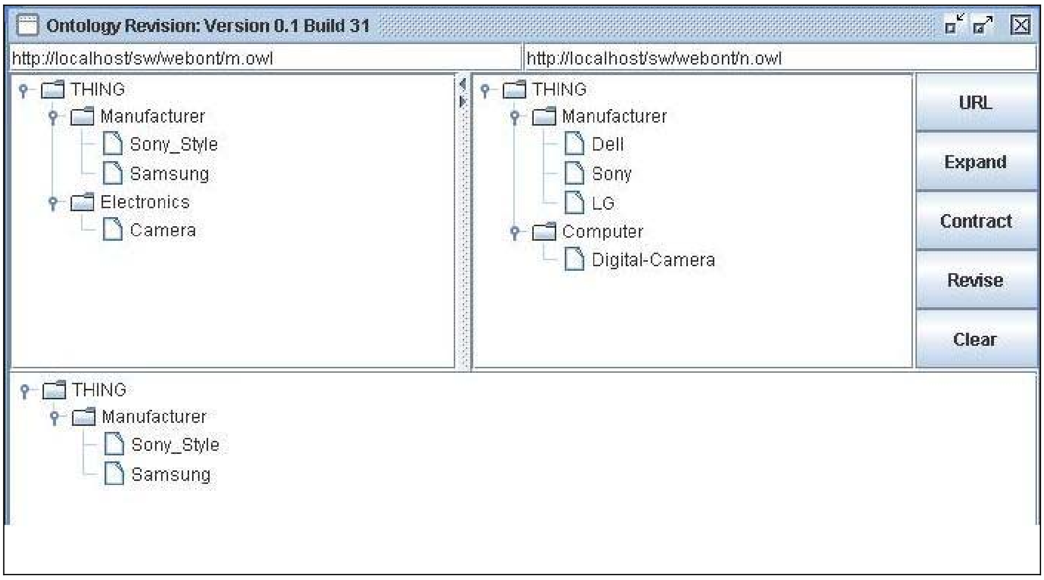

Figure 7 The example of contraction from ontology *M*.

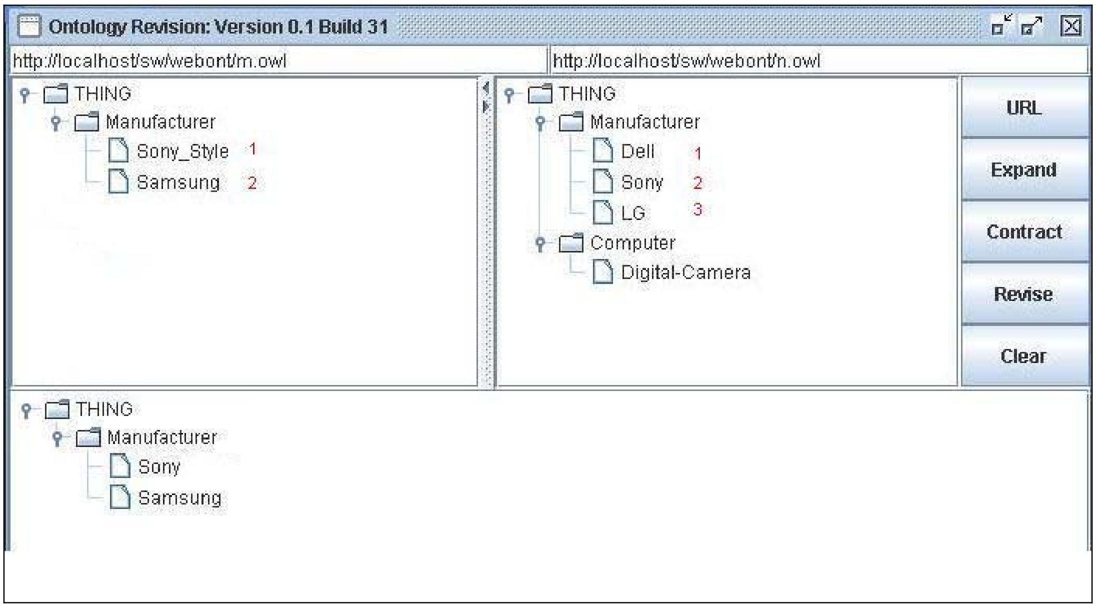

Figure 8 The example of revision from ontology *N* to *M*.

There are some computational limitations. We have selected several ontologies which are available online during the early stage of the development. For example, we have tried to use eCl@ss, however we encountered problems in loading. The eCl@ss describes products and service with more than 25,000 categories (Hepp 2006). In our design stage, we have tried to follow WORDNET-like style of ontologies. However, WORDNET has evolved in a way that it becomes too complex to compute the integrated knowledge. To demonstrate our proposed framework, thus, we have chosen to use simple ontologies to overcome computational inadequacies. A computational tool that can be used by a software agent to perform monitoring and controlling the user queries is required. This is essential to relax comprehensive computational lexicon of general language so that information resource involved the user query can be processed to relax computational challenges.

6. Conclusion and Future Research Direction

This paper has described an innovative idea of using the belief revision theory to revise ontology with the aim to ensure consistency is maintained after ontology is revised. The proposed approach is derived based on the ontological reengineering method. Several examples have been used to illustrate the implementation of this approach.

In this paper we have demonstrated the implementation based on fairly simplistic examples. We are currently implementing the proposed framework in an online buying e-commerce environment to demonstrate the practicality of this approach. In particular we are investigating a way to support ontology revision based on multiple ontologies, for example three or more ontologies, and to investigate a framework that can support more complex relations to provide additional information such as intersectionOf, unionOf, complementOf and others. One of the possible problems which we can foresee is the issue of computational complexity when revision is performed on multiple ontologies. In particular, large ontologies which may have significant computational overhead, thus the issue of efficient computational method needs to be investigated too.

7. References

[1] Borst, W. (1997).*Construction of Engineering Ontologies for Knowledge Sharing and Reuse*, Center for Telematics and Information Technology, University of Twente, 243 pages.

[2] Ding, Y. and Fensel, D. (2001) Ontology Library Systems: The key to successful Ontology Re-use, in *Proceedings of International Semantic Web Working Symposium (SWWS) on Ontology and Ontology Maintenance*, 30 Jul. - 1 Aug., Stanford University, California, USA, pp.93-112.

[3] Doyle, J. (1979) A Glimpse of Truth Maintenance, in *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, 20-23 Aug., Tokyo, pp.232-237.

[4] Gärdenfors, P. (19) *Knowledge in Flux: Modeling the Dynamics of Epistemic States*, The MIT Press, London.

[5] Gärdenfors, P. (1992) *Belief Revision*, Cambridge University Press.

[6] Gärdenfors, P. (1994) "How Logic Emerges from the Dynamics of Information", in *Logic and Information Flow*, (Eds.) Eijck, v. J. and Visser, A., MIT Press, pp.49-77.

[7] Gärdenfors, P. and Rott, H. (1995) "Belief Revision", in *Handbook of Logic in Artificial Intelligence and Logic Programming*, (Eds.) Dov, G., Hogger, C. and Robinson, J., Clarendon Press, Oxford, pp.35-132.

[ ] Gómez-Pérez, A. (1999) Applications of Ontologies and Problem-Solving Methods, in *Proceedings of Workshop at Thirteenth Biennial European Conference on Artificial Intelligence*, 1 - 4 Oct., Berlin, Germany.

[9] Gómez-Pérez, A., Fernandez-López, M. and Corcho, O. (2004) *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and Semantic Web*, Springer, London, Berlin, Heidelberg, New York, Hong Kong, Milan, Paris, Tokyo.

[10] Gruber, T. (Ed.) (1993) *Toward Principles for the Design for Ontologies used for Knowledge Sharing*, Kluwer Academic Publishers, Padova, Italy.

[11] Heflin, J. and Hendler, J. (2000) Dynamic Ontologies on the Web, in *Proceedings of the Seventeenth National Conference on Aritificial Intelligence*, 30 Jul. - 3 Aug., Austin, Texas, pp.443-449.

[12] Heflin, J. and Hendler, J. (2001) "A Portrait of the Semantic Web in Action", *IEEE Intelligent Systems*, 16(2), pp.54-59.

[13] Hendler, J. (2001) "Agents and the Semantic Web", *The IEEE Intelligent Systems*, (Mar./Apr.), pp.30-37.

[14] Hepp, M. (2006) "Products and Services Ontologies: A Methodology for Deriving OWL Ontologies from Industrial Categorization Standards", *International Journal on Semantic Web and Information Systems*, 2(1), pp.72-99.

[15] HPL (2002) "*Jena Semantic Web Toolkit*", Hewlett-Packard Company, <http://www.hpl.hp.com/semweb/jena.htm>, (Accessed: 19 Jul. 2002).

[16] Klein, M. and Fensel, D. (2001) Ontology Versioning on the Semantic Web, in *Proceedings of International Semantic Web Working Symposium*, 30 Jul. - 1 Aug., Stanford University, California, USA, pp.75-91.

[17] Klein, M., Fensel, D., Kiryakov, A. and Ognyanov, D. (2002) Ontology Versioning and Change Detection on the Web, in *Proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management*, 1-4 Oct., Sigüenza, Spain, pp.197-212.

[1] McCarthy, J. and Hayes, P. (1969) "Some Philosophical Problems from the Standpoint of Artificial Intelligence", in *Machine Intelligence 4*, (Eds.) Meltzer, B. and Michie, D., pp.463-502.

[19] Nodine, M. and Fowler, J. (2005) "On the Impact of Ontological Commitment", in *Ontologies for Agents: Theory and Experiences*, (Eds.) Tamma, V., Cranefield, S., Finin, T. and Willmott, S., Birkhäuser Verlag, Berlin, pp.19-42.

[20] Noy, N. and McGuinness, D. (2001) *Ontology Development 101: A Guide to Creating Your First Ontology*, SMI-2001-0 0, Stanford Medical Informatics, pages.

[21] Segal, R. (1994) "*Belief Revision*", Department of Computer Science and Engineering, FR-35. University of

Washington, <http://citeseer.nj.nec.com/segal94belief.html>, (Accessed: 12 Feb. 2003).
[22] Stuckenschmidt, H. (2003).*Ontology-Based Information Sharing in Weekly Structured Environments*, Computer Science, Vrije Universiteit, 1 3 pages.
[23] W3C (2005) "*Web Ontology Language (OWL)*", <http://www.w3.org/2004/OWL/>, (Accessed: 23 Apr. 2005).