Ontology Revision as Non-Prioritized Belief Revision

Mauro Mazzieri and Aldo Franco Dragoni

Department of Electronics, Artificial Intelligence and Telecommunications,
Università Politecnica delle Marche, Ancona, Italy
{m.mazzieri,a.f.dragoni}@univpm.it

Abstract. Ontology revision is the process of managing an ontology when a new axiom or fact would render it inconsistent. So far, the AGM approach to belief revision has been adapted to work with ontologies. However, when multiple sources are contributing uncertain knowledge about a static domain, an approach that doesn't give priority to incoming information and allows to recover previously discarded axioms is more suited.
We describe an ontology revision framework that links symbolic and numerical techniques to allow the consistent evolution of an ontology from the contributions of multiple potentially unreliable sources.

Key words: Ontology revision, belief revision, OWL.

1 Introduction

An explicit specification of a conceptualization for a shared domain of discourse is called an ontology[1]. Hence, changes in ontologies are caused by changes either in the domain, or in the conceptualization, or in the defined specification[2].

Ontology evolution[3] is the process of modifying an ontology in response to a change in the domain (first kind of change) or its conceptualization (second kind). The case of change in the domain is analogous to *belief updating*, thus it can be defined as *ontology updating* (more on this on §3). This work deals with changes in the shared conceptualization: a problem analogous to *belief revision*, thus the name *ontology revision*. The third kind of change refers to a change in the way the conceptualization is formally recorded; this type of change is dealt with in the field of *ontology translation*[4].

Current work on ontology evolution is based on the idea of bringing the AGM belief change theory[5, 6] to work within ontology evolution; Flouris' PhD thesis[3] contains both novel contributions and a survey of the field; [7] depicts the state of the art in AGM-based ontology revision.

However, AGM belief revision is not apt to all kind of ontology changes. One of its principles states that incoming information has a priority: it must belong to the new set of beliefs. This principle works well when the new information represent a certain fact: either a realization of the new contingent state of the world, or a correction of a previous error in conceptualization, or a required property of the formalization. The principle can not be accepted when the new information represents a new evidence about the world, supposed to be a fixed static entity, while its description is only partial and uncertain. In particular, it can not be accepted in a distributed environment, where multiple potentially unreliable information sources are present. Not only an information from an external source can not be unconditionally accepted (can you trust everything you hear?); also, there is not always a relation between the arrival order of information and their acceptability.

There are many different possibility to discard the principle of priority to incoming information. Hansson[8] makes a survey of different varieties of non-prioritized belief revision, i.e. belief revision in which the new information has no special priority due to its novelty. The problem is when and how to choose if the new information must be accepted. We will follow an integrated approach, already successfully applied to a juridical domain[9], which deals with old and new information as they were come at the same time. This approach relies both on symbolic and numerical techniques and make use of a new principle, called *principle of recoverability*[10] [1]:

> Any previously held piece of knowledge should belong to the current knowledge space if consistent with it.

To circumscribe the work, we will refer to a specific use case. A team of loosely-coordinated domain experts has the duty to build an ontology for their domain. Each team member contributes to the activity building his conceptualization with an editor. The domain is assumed as a fixed static entity, while the conceptualization is constantly changing during the building and refinement process. The work of each member is shared with the other experts in a peer-to-peer way: each member receives the contribution of the other experts. A supporting software must be able to use an ontology revision mechanism to maintain a consistent local ontology to be visualized and used as the basis for further editing. An example of a work session will be shown in section 5.

In the following we will first summarize in an informal way the syntax and semantic of the OWL ontology language (§2). Then, after an introdution to the problem of ontology revision (§3), we will show the proposed revision procedure (§4), both in its symbolical (§4.1) and numerical (§4.2) steps. Finally in §6 we sketch the future research perspectives.

2 Ontology

The OWL web ontology language[12] is the language used for publishing and sharing ontologies on the World Wide Web. OWL is developed as an extension of the RDF[13] knowledge representation language. The language has two specific subsets: OWL DL and OWL Lite. The complete language is called OWL Full

[1] Introduced as the *store and recover principle*[11] and also known as the *principle of persistence*[9].

to distinguish it from the subsets. The DL in OWL DL stands for "Description Logic"[14], a decidable subset of first order logic used for expressing structured knowledge. OWL DL and OWL Lite are both based on description logic; the former is more expressive, while the latter has better computational properties.

In order to introduce the problem of ontology revision and to make the work self-contained, we will give here an informal definition of an ontology language syntax and semantics, roughly correspondant to OWL Lite. The full formal semantics and syntact can be found in [15].

2.1 Syntax

The basic building blocks of an ontology are *classes*, *individuals* and *properties*. A class is related to a set of individuals, called class extension. Properties can be either data-valued, relating individuals to values, or individual-valued, relating individuals to other individuals.

An *ontology* is a set of class axioms, property axioms and facts.

There are two kinds of *class axioms*. A class can be defined as either exactly equivalent to the conjunction of a set of superclasses, or as a subclass of the conjunction of a set of superclasses. A superclass can be either another class, or an anonymous class specified giving constraints on properties.

The allowed restrictions on property values are:

- all the values must be instances of a class (or from a datatype, in the case of data-valued properties);
- some of the values must be instances of a class (or from a datatype, in the case of data-valued properties);
- the cardinality must be at least (or at most, or equal to) either 0 or 1.

Property axioms are used to define properties. A property can be given a super-property, allowing the construction of a property hierarchy. Properties can also be given domains and ranges.

Data-valued properties can be specified as partial functional, i.e. with at most a value. Individual-valued properties can be specified to be functional, inverse-functional, symmetric, transitive, or the inverse of another property.

Finally, a *fact* states that an individual belongs to a class or that an individual's property has a certain value.

2.2 Semantics

An OWL *interpretations* defines:

- a class as a collection of individuals,
- a datatype as a set of literal values,
- a data-valued property as a relation from individuals to literal values,
- an individual-valued property as a relation from individuals to other individuals.

An interpretation I *satisfies* an ontology O if it obeys to all restrictions given by O's axioms and facts.

An ontology O is *consistent* if there is at least an interpretation I which satisfies the ontology.

An ontology O *entails* an ontology O' if each interpretation I which satisfies O also satisfies O'.

3 Ontology Revision

3.1 Belief Revision

Ontology revision has many similarities with belief revision.

Belief revision is the process of rearranging a knowledge base to preserve global consistency while accommodating incoming information. In the AGM theory[5, 6], the belief is formalized as a set of logical statements, (the *belief set*), i.e. a logic theory K described in a formal language L. The belief set is closed under logical consequences. A finite subset B of K such that $K = \mathrm{Th}(B)$ is a *knowledge base* for K. The problem of revision arises when we get a new formula p that makes the knowledge base *inconsistent*. Then, we have to *revise* the knowledge base, retracting some of the beliefs, in order to restore consistency. The revised theory is K^*p. The AGM theory gives three rationality principles affecting K^*p:

Consistency The revised belief K^*p must be consistent.

Minimal change The revision process should alter as little as possible the current belief set.

Priority to incoming information The new information p must belong to the new belief set K^*p.

From these principles eight postulates follow. However, neither the rationality principles nor the postulates univocally define revision.

3.2 Definition of Ontology Revision

Ontology revision is defined as a change of components in ontology[16]. Coeherently with belief revision theory, we define ontology revision as the process of rearranging an ontology to preserve consistency while accomodating changes. Foo[17] presents a summary of issues concening ontology revision from artificial intelligence, philosophy and recursion theory.

Our approach to ontology revision will be based on *belief bases*, a set of sentences not closed under logical consequence, from which a belief set can be derived[18]. Our belief base is an ontology, i.e. a set of axioms and facts. The incoming information is represented as an axiom or a fact, i.e. a TBox or a ABox statement[2]. The problem of revision arises when the new axiom or fact would render the ontology inconsistent.

[2] Another approach, such the one in [19], considers only inconsistences due to objects introduced in the ABox.

The choice to represent changes at the level of single axioms is very fine-grained, but it doesn't forbid to define more complex, higher-level changes[20]. A finer-grade approach, involving the single constraints in class and property axioms, would be problematic as not all combinations are allowed. For example, in OWL Lite, not all properties can have cardinality restrictions placed on them or be specified as functional or inverse-functional[15]. An example of this approach, involving the weakening of the original ontology to accomodate the incoming axiom, is presented in [3].

To choose an ontology revision procedure we have first to understand why an axiom or fact, potentially incompatible with the current ontology, can arrive. We want to point out two different scenarios, demanding a different approach:

– The ontology represents the current state of an evolving world, and the new information reflects a change in the world. The consequent change in the representation of the world is called *updating*.
– We have an incomplete, approximate or erroneous representation of a static world. The new information represent a new evidence regarding this world. The consequent change in the representation is properly called *revision*.

In our scenario, a loosely-coupled group of peers are incrementally building an ontology for a fixed domain. Thus, the world is not supposed to change, while the world's description is constantly evolving as the participants add, refine or retract classes and properties definitions. This scenario is that of a *revision* process and need to be handled within a framework possessing some specific requisites. The need for those requisites already appeared in a juridical scenario (incremental building of a proof in court[9]) and in distributed multi-agent belief revision[21].

Ability to reject incoming axioms. A belief revision system for a multi-source environment should drop the rationality principle of "priority to the incoming information", which is not acceptable since the sources are asynchronous and there is no strict correlation between the chronological sequence of information and their credibility or importance[11].

The ability to recover previously discarded axioms. Each domain expert should be able to recover previously discarded pieces of the the ontology if new axioms redeem them. This should be done not only when the new axioms directly support previously rejected axioms, but also when they indirectly support them by disclaiming the axioms that caused their ostracism.

For these reasons we adapt to ontology revision a belief revision framework that replace the priority to incoming information with the *principle of recoverability*[10]. The rationale for this principle is that, if an axiom was part of the ontology in the past, and it would be consistent with the current ontology, then it should be part of the ontology again.

4 Revision procedure

Belief revision has been approached both as a qualitative syntactic process and as a numerical mathematical issue. Our distributed ontology revision system links symbolic and numerical techniques. Computationally, the ontology revision consists of two steps acting on the axioms of the ontology, and three steps working with numerical weights.

Each peer stores his knowledge about the domain in at least two repositories[10]:

1. A *background repository* KB. This is the set of all axioms and facts available to reasoning; it contains both the axioms and facts written by the contributor and received from other contributors. It may be inconsistent.
2. A *working ontology* $B \subseteq KB$, which is the maximally consistent, currently preferred ontology that should be used for reasoning or further editing.

Given an incoming contribution p (an axiom or a fact) from a source, the evolution process consists of the following steps:

1. detection of minimally unsatisfiable subsets of $KB \cup \{p\}$, called *nogoods*;
2. generation of the maximally satisfiable subsets of $KB \cup \{p\}$, called *goods*;
3. revision of the credibility weights of axioms in $KB \cup \{p\}$;
4. choice of a preferred maximally consistent subset of $KB \cup \{p\}$ as the new working ontology B';
5. recalculation of "a posteriori" reliability of sources.

4.1 Symbolic steps

Step 1 and 2 are symbolical ATMS-style operations[22]. We define a *nogood* as a minimally inconsistent subset of KB. Dually, we define a *good* as a maximally consistent subset of KB.

Nogood detection can be demanded to a reasoner, such as Racer[23], FaCT[24], Pellet[25]. The set of goods and nogoods are dual: if we remove from KB exactly one element for each nogood, what remains is a good[26]. So, once an inference engine finds out some nogoods, it is possible to use a set-covering algorithm, such as the one introduced by Reiter for model-based diagnosis[27], to find out the goods. This algorithm has already been succesfully used for belief revision[21].

An interesting property that the inference engine does not need to calculate the collection of all nogoods (i.e. *minimally* inconsistent subsets of KB), but just a collection of inconsistent subsets of KB, which is much easier.

4.2 Numerical steps

The numerical approach to ontology revision deal with the ontology as a set of weighted axioms. Weights usually are reals between 0 and 1, representing explicitly the credibility of the axioms.

The numbers represent uncertainty caused by the not complete reliability of the team members [3]. As the reliability of the source is strongly related to the credibility of the information, it is necessary to deal with couples ⟨source, axiom⟩[28].

The numerical steps of the revision procedure are step 3–5.

Step 3 of the ontology revision process uses the belief function formalism, as the one used by Shafer and Srivastava for auditing[29]. From the reliability value of each source (a propability that the source gives correct information), the credibility of the goods is determined by the Dempster rule of combination. Thus, ontology revision consists in the reassignment of credibility to axioms in the light of the incoming axiom. The credibility ordering reflects the collaborative building of the ontology: the reliability and the number of different contributors affect the credibility of the axiom and the converse.

The recalculation of credibility values involves all the collected axioms in KB. The incoming axiom p is confronted not just with the current ontology B, but with all KB, so that the weight of axioms in $KB \cup \{p\}$ are reviewed in a broader and less prejudicial basis.

Step 4 is the selection of a new ontology B'. The new ontology is the maximally consistent subset of $KB \cup \{p\}$ with the greater credibility. Since the incoming information causes a recalculation of all the credibility values, and the selected ontology is maximal, it is possible to rescue axioms from KB.

Even when the new contribution is compatible with the working ontology (meaning that $B \cup \{p\}$ is satisfiable), not necessarily $B' = KB \cup \{p\}$, since the global revision of numerical weight in step 3 may yield a totally different choice of ontology in step 4. A previously rejected set of axioms r can be rescued if p support r against a previously accepted set q.

In general, even when the new ontology B' is syntactically equal to the previous B, meaning that p has been rejected, B' may have a different credibility distribution (assignment of weights) from B. The incoming contribution p might be rejected even when a new ontology B', different from B, is selected, but $B' \cup \{p\}$ is still unsatisfiable.

Step 5 uses Bayesian conditioning to determine the probability that a source give correct contributions, gives the new accepted ontology B'. The main point is that a reliable source can not give false informations, while an unreliable source may occasionally give correct contributions.

As an alteration of the credibility of an axiom might result in the perturbation of the credibility of all the axioms from the same source, thus causing a completely different ontology to be selected at the next step.

[3] Even the contribution from the agent self can be considered not completely reliable, as this depends of the relative trust a contributor has on his work compared to trust on other experts' works.

5 Examples

We will show two examples, showing the symbolical and numerical steps respectively. In both, we suppose that a group of domain experts are working on an ontology of birds.

5.1 Symbolic Example

The initial knowledge base KB of one of those experts is made of the axiom $Bird \sqsubseteq Fly$ and the fact $Bird(Tweety)$, where $Bird$ and Fly are classes (the class of individuals that are birds, and the class of individuals that can fly, respectively), while Tweety is an individual. The knowledge base is consistent, so the initial ontology is $B = KB$.

The expert receive from a colleague (let's call him source 1) the axiom $\neg Fly(Tweety)$. Now $KB = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ \neg Fly(Tweety)\}$ is unsatisfiable. If we would adopt the AGM principle of Priority to Incoming Information, the new working ontology would be chosen among

1. $B_1 = \{Bird \sqsubseteq Fly,\ \neg Fly(Tweety)\}$
2. $B_2 = \{Bird(Tweety),\ \neg Fly(Tweety)\}$

If we adopt the Principle of Recoverability instead, we have a third candidate working copy,

3. $B_3 = B = \{Bird \sqsubseteq Fly,\ Bird(Tweety)\}$

Next, another expert (let's call him source 2) affirms $Fly(Tweety)$.

If we use the AGM principles, the new working ontology would be, respectively,

1. $B_{1'} = \{Bird \sqsubseteq Fly,\ Fly(Tweety)\}$, if B_1 was chosen after the input from source 1,
2. $B_{2'} = \{Bird(Tweety),\ Fly(Tweety)\}$, if B_2 was chosen.

If we allow the rejection of the new contribution, after the arrival of the axiom from source 2, we can:

1. Reject the new axiom. Our working copy remain the same as after step 1.
2. Accept the new axiom.
   (a) $B_1 = \{Bird \sqsubseteq Fly,\ \neg Fly(Tweety)\}$. We recover $Bird(Tweety)$, so $B_{1''} = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ Fly(Tweety)\}$.
   (b) $B_2 = \{Bird(Tweety),\ \neg Fly(Tweety)\}$. We recover $Bird \sqsubseteq Fly$, so $B_{2''} = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ Fly(Tweety)\}$.
   (c) $B_3 = \{Bird \sqsubseteq Fly,\ Bird(Tweety)\}$. This is a simple expansion, so $B_{3''} = \{Bird \sqsubseteq Fly,\ Bird(Tweety),\ Fly(Tweety)\}$.

The example show that, if we consider the axiom $Fly(Tweety)$ more credible than $\neg Fly(Tweety)$, our final working ontology would be the same, independently from the choice made at the first step.

5.2 Numerical Example

The initial knowledge base KB of one expert is made of the axiom $Bird \sqsubseteq Fly$ and the fact $Bird(Tweety)$.

The expert receives from source 1 the axiom $\neg Fly(Tweety)$ and chooses as the new working ontology $B_2 = \{Bird(Tweety), \neg Fly(Tweety)\}$.

Now suppose source 1 sends us the axiom $\neg Bird(Tweety)$. If we reject this axiom, probably now our confidence in source 1 will be lower, as the credibility of the information affects the reliability of its source[30].

A change on the credibility of an axiom provided by a source yields corresponding changes in the credibility of the other axioms provided by the same source, even if they are not logically related with each other. As a consequence of this perturbation, a completely different working ontology might be chosen, in the previous example B_3 instead of B_2, thus rejecting the previously accepted axiom from source 1. Since all the collected axioms are retained and their weights can change, the new selection might reconsider some previously discarded axiom, whether the incoming contribution is accepted or not.

Probably, the last come contribution decreases the credibility of the axioms it would render unsatisfiable, even in the case it has been rejected. The same when we receive an axiom which already belongs to the working ontology: it is not the case that nothing happened, as AGM fourth postulate of expansion would suggest[6, p. 49], since we are now, in general, more sure about the correctness of the axiom.

6 Conclusions and Future works

When a group of peer tries to capture in an ontology a static domain, but their domain's knowledge is only partial and potentially unreliable, not all the contribution can be taken as unconditionally useful. It is necessary to use an ontology revision procedure that allows to discard the incoming information, if there is no reason to consider it more reliable than other conflicting contributions, and to rescue previously discarded axioms, if they are now compatible with the current selected ontology.

In general, at each step there will be more than a consistent subset of the ontology with maximal size (i.e., a good). There is the need of a rational criteria to choose a good as the new working ontology. If we keep track of axioms' sources and give to each peer a a-priori reliability value, we can use the belief-function formalism to estimate the reliability of each good and bayesian conditioning to evaluate a new a-posteriori reliability value for the sources.

This work is just the beginning of an analysis of ontology revision process for a distributed environment. Current research work involves the following subjects.

Collaborative Ontology Revision At the end of the work each expert has its own version of the domain ontology. To extract the final result of the collective work of the group of interacting experts, a voting mechanism is needed. The integration of the different conceptualizations must not be performed by an external supervisor, but it can be done by the group itself.

Ontology distribution To allow the distribution of individual fragments of the ontology, it must be possible to partition it and then reconstruct it preserving meaning.

For RDF, this bring to the definition of the *minimal self-contained graph*[31] as the finer decomposition of a graph that would preserve meaning. This minimal set consists in a statement and, recursively, all statements involving a blank node already in the set.

Given the OWL RDF/XML syntax's use of blank nodes to build complex definitions, a similar concept can be applied to OWL. This decomposition allows the distribution of the ontology between peers, as in the scenario introduced in section 1.

User interaction A software supporting the collaborative building of an ontology must be able to use an ontology revision mechanism to maintain a consistent working ontology. Where inconsistencies arise and there is no other available ranking, the choice among different maximally consistent subsets can only be done by the user.

However, there are other times during the work when an user intervention would be useful. Why don't allow the user to explicitly mark a part of the ontology as unreliable, not necessarily causing its deletion from the current working set, but determining a change in the distribution of reliability among the sources?

Explicit reliability judgments by an human agent must be taken into account when the system build a credibility ranking among the available sources.

Strong time-Independence Even if the new information has no priority for his novelty, a complete independence of axiom's weights from contributions' arrival time is not guaranteed. This, given the asynchronous setting, would be a desirable feature of the system.

Ontology versioning Ontology versioning is defined as the ability to handle an evolving ontology by creating and managing different variants of it[2]. A common requirement between ontology versioning and the present ontology revision framework is the ability to work with different versions of the ontology and to recover previous parts of it. Thus the revision process for ontology revision can be at some extent applied to ontology versioning.

References

1. Gruber, T.R.: A translation approach to portable ontology specifications. Knowledge Acquisition **5**(2) (1993) 199–220
2. Klein, M., D.Fensel: Ontology versioning on the semantic web. In: Proc. of the Int. Semantic Web Working Symposium (SWWS). (2001) 75–91

3. Flouris, G.: On Belief Change and Ontology Evolution. PhD thesis, Dept. of Computer Science, University of Crete (February 2006)
4. Dou, D., McDermott, D., Qi, P.: Ontology translation on the semantic web. In: International Conference on Ontologies, Databases and Applications of Semantics. (2003)
5. Alchourrón, C.E., Gärdenfors, P., Mankinson, D.: On the logic of theory change: partial meet contraction and revision functions. The Journal of Symbolic Logic **50** (1985) 510–530
6. Gärdenfors, P.: Knowledge in Flux: Modeling the Dynamics of Epistemic States. MIT Press, Cambridge, MA (1988)
7. Ribeiro, M.M., Wassermann, R.: First steps towards revising ontologies. In: Proc. of WONRO'2006. (2006)
8. Hansson, S.O.: A survey of non-prioritized belief revision. Erkenntnis **50** (1999) 413–427
9. Dragoni, A.F., Nissan, E.: Salvaging the spirit of the meter-models tradition: A model of belief revision by way of an abstact idealization of responce to incoming evidence delivery during the construction of proof in court. Applied Artificial Intelligence **18** (2004) 277–303
10. Dragoni, A.F.: Belief revision: From theory to practice. The Knowledge Engineering Review **12**(2) (1997)
11. Dragoni, A.F., Mascaretti, F., Puliti, P.: A generalized approach to consistency based belief revision. In Gori, M., Soda, G., eds.: Topics in Artificial Intelligence, Proc. of the 4th Conference of the Italian Association for Artificial Intelligence. Number 992 in LNAI, Florence, Italy, Springer-Verlag (October 11–13 1995) 231–236
12. Bechhofeer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D., Patel-Schneider, P., Stein, L.A.: OWL web ontology language reference. Recommendation, W3C (10 February 2004)
13. Hayes, P.: RDF Semantics. Recommendation, W3C (February 10, 2004)
14. Baader, F., McGuinness, D., Nardi, D., Patel-Schneider, P., eds.: Description Logic Handbook: Theory, implementation and applications. Cambridge University Press (2002)
15. Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL web ontology language semantics and abstract syntax. Recommendation, W3C (February 2004)
16. Heflin, J., Hendler, J.: Dynamic ontologies on the web. In: Proc. of the 17th Nat. Conf. on Artificial Intelligence, Austin, Texas (30 Jul – 3 Aug. 2000) 443–449
17. Foo, N.: Ontology revision. Lecture Notes in Computer Science **954** (1995) 16–??
18. Halaschek-Wiener, C., Katz, Y.: Belief base revision for expressive description logics. In: Proc. of the OWLED'06. (2006)
19. Qi, G., Liu, W., Bell, D.: Knowledge base revision in description logics. In: Proc. of 10th European Conf. on Logics in Artificial Intelligence (JELIA'06). Number 386-398, Springer Verlag (SEP 2006)
20. Haase, P., Stojanovic, L.: Consistent evolution of OWL ontologies. In: Proc. of the 2nd European Semantic Web Conf. (ESWC-05). (2005)
21. Dragoni, A.F., Giorgini, P.: Distributed belief revision. Autonomous Agents and Multi-Agent Systems **6** (2003) 115–143
22. de Kleer, J.: An assumption-based truth maintenance system. Artificial Intelligence (28) (1986) 127–162
23. Haarslev, V., Möller, R.: Racer system description. In: Proc. of the Int. Joint Conf. on Automated Reasoning (IJCAR 2001). Volume 2083 of Lecture Notes in Artificial Intelligence., Springer (2001) 701–705

24. Horrocks, I.: The FaCT system. In de Swart, H., ed.: Proc. of TABLEAUX98. Number 1397 in LNAI, Springer (1998) 307–312
25. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A practical OWL-DL reasoner. Technical Report 2005-68, University of Maryland, Institute for Advanced Computes Studies (UMIACS) (2005)
26. Roos, N.: A logic for reasoning with inconsistent knowledge. Artificial Intelligence **57** (1992) 69–103
27. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence **32** (1987) 57–59
28. Dragoni, A.F.: A model for belief revision in a multi-agent environment. In Demazeau, Y., Werner, E., eds.: Decentralized A.I. Volume 3., North Holland Elsevier Science Publisher (1992) 103–112
29. Shafer, G., Srivastava, R.: The bayesian and belief-function formalisms a general perspective for auditing. In Shafer, G., Pearl, J., eds.: Reading in Uncertain Reasoning. Morgan Kaufmann Publishers (1990)
30. Mazzieri, M., Dragoni, A.F.: On the relation between trust on input and reliability of output. In: Demos and Posters of the 3rd European Semantic Web Conference (ESWC 2006). (2006)
31. Tummarello, G., Morbidoni, C., Puliti, P., Piazza, F.: Signing individual fragments of an RDF graph. In: Proc. of the World Wide Web Conference (WWW2005). (2005)