Supporting Uncertainty and Inconsistency in Semantic Web Applications

Neli P. Zlatareva
Department of Computer Science,
Central Connecticut State University
1615 Stanley Street, New Britain, CT 06050, USA
zlatareva@ccsu.edu

Abstract

Ensuring the consistency and completeness of Semantic Web ontologies is practically impossible, because of their scale and highly dynamic nature. Many web applications, therefore, must deal with vague, incomplete and even inconsistent knowledge. Rules were shown to be very effective in processing such knowledge, and future web services are expected to depend heavily on them. RuleML, which is the earliest effort to define a normalized markup for representing and exchanging rules on the web, is currently limited to Horn rules. Significant research efforts are underway to extend RuleML with more flexible representation and reasoning capabilities. This paper presents an extension of the current rule format intended to accommodate uncertain and/or inconsistent knowledge, and shows how one truth maintenance logic can be adapted and extended to support such rules.

Introduction

Semantic Web is envisioned to extend and dramatically improve current web services by creating and supporting a universal medium for information exchange accessible to people and machines alike. Current web, although a huge success, is nothing more than just a collection of human-readable pages which people, helped by web browsers, navigate to find, share and combine information. Often this process is slow, tedious, and inefficient. Consider, for example, business-to-consumer e-commerce. To buy a product, a consumer visits several online shops to compare (manually) their prices, special offers, etc. with no guarantee that the "best deal" will be found because of the limited search involved. Compare this ad-hoc approach to the following scenario. The customer enters desired product specifications and leaves the rest of the search to his computer. The latter autonomously navigates through online retailers offering the product, collects and evaluates offers, and returns one or several best offers to the customer for final decision. To implement this scenario, we must ensure the following:

1. The domain navigated by the computer is represented in a machine-understandable form.
2. An inference engine able to process incomplete, uncertain and possibly inconsistent knowledge in order to adequately match customer specifications to product descriptions found on the web is in place.

In the past decade, a lot of research coordinated by the World Wide Web Consortium (W3C) (http://www.w3.org/) was devoted to addressing the first challenge. A number of ontology languages for the Semantic Web were introduced; RDF, RDFS and OWL are among the best known ones. RDF and RDFS target simple typed ontologies where data is completely and consistently specified. OWL, which was recently recommended by W3C as a standard web ontology language, provides more expressive representation, but its inference capabilities are limited to satisfiability, subsumption, equivalence and disjointness. Composition of properties cannot be expressed in OWL, thus making it inadequate to tackle the second challenge above.

As the size of ontologies and the complexity of applications grow, uncertainty, incompleteness and inconsistency are becoming common properties of ontological knowledge. Rules were shown to be very effective in processing such knowledge, and future Semantic Web services are expected to depend heavily on them. The RuleML initiative (http://www.ruleml.org/) is the earliest effort to define a normalized markup for representing and exchanging rules on the Semantic Web. Current RuleML is limited to Horn rules, but significant research efforts are underway to extend it with more flexible representation and reasoning capabilities (Lukasiewicz 2008; Damasio et al. 2007). Most of this research is based on non-classical logics (probabilistic, fuzzy, possibilistic, etc.) which are primarily concerned with uncertainty and incompleteness of knowledge, assuming its consistency. Semantic Web, being an open and highly dynamic environment, will inevitably contain inconsistencies. The importance of their adequate processing in order to maintain the validity of ontological knowledge has been widely acknowledged by the Semantic Web community, but little work has been done so far to develop techniques and tools for reasoning with inconsistent knowledge ((Huang, van Harmelen, and ten Teije 2006) and (Schobach and Corner 2003) are notable exceptions).

In this paper, we argue that ensuring the consistency of large Semantic Web ontologies "composed of concepts which are to some extend valid in a domain, relations that hold to some degree of certainty, and rules that apply only in some cases" (Davis et al. 2006) is practically impossible; inconsistencies will be an intrinsic part of ontological knowledge and must be treated as first class citizens. That is, the inference procedure must be able to identify and maintain inconsistencies preserving at the same time the validity of the inference and the meaningfulness of derived knowledge. We show how the truth maintenance logic introduced in (Popchev, Zlatareva, and Mircheva 1990) can be adapted and extended to carry out this task. But first, we discuss some of the limitations of the current RuleML syntax to justify the need for more expressive rule representation format.

RuleML limitations: a practical way to address them

Consider the following example inspired by (Antoniou and van Harmelen 2007).

"Carlos is looking for an apartment of at least 45 sq m with at least 2 bedrooms. Carlos is willing to pay $300 for a centrally located 45 sq m apartment, but $250 for a similar one in the suburbs. He will pay extra for a larger apartment or an apartment with a garden. Carlos does not want to pay more than $400, but if the apartment is centrally located, offers a swimming pool, and have other desired features, he may consider a higher rent. If the apartment is on the third floor or higher, the building must have an elevator. Pets must be allowed, because Carlos cannot leave his dog behind. Given a choice, the price will be Carlos' first priority, but amenities (swimming pool, garden) will also play a role in Carlos' final decision."

Some of Carlos' requirements are quite vague (location, swimming pool, garden), while others are very specific and firm (pet friendliness, number of bedrooms, size). Firm requirements can be easily expressed in RuleML. For example, "Carlos is looking for an apartment of at least 45 sq m" can be represented as follows:

```
<Implies>
  <Body>
    <And>
      <Atom>
      <Rel>size</Rel>
      <Var>Apartment</Var>
      <Var>X</Var>
    </Atom>
      <Atom>
      <Rel>GreaterThan</Rel>
      <Var>X</Var>
      <Ind>45 sq m</Ind>
    </Atom>
    </And>
  </Body>
  <Head>
      <Atom>
      <Rel>consider</Rel>
      <Var>Apartment</Var>
    </Atom>
  </Head>
</Implies>
```

Representing statements, such as "an apartment may be OK with or without a garden", however, is not straightforward without enforcing some change in the meaning of the statement. What we need in order to adequately match such "relaxed" statements to rule premises, is to allow for "relaxed" premises as well. For example, to say that Carlos will consider an apartment which costs less than $400 despite of its location or availability of a garden, the following rule will do:

If A: size > 45 and bedroom > 1 and pets-allowed
IN SPITE garden or location = Central
Then A: consider

According to this rule, A will be considered if Carlos' firm requirements (size > 45, bedroom > 1, pets-allowed) are met. But if the apartment is centrally located and/or has a garden, it will be "even better".

To represent and process rules of this type, we need more than a simple Horn-style syntax and classical forward or backward chaining. Non-monotonic logics will also not work in this case, because their non-monotonic premises have a very different semantics – they serve as exceptions to the rule's conclusion. However, various possibilistic logics (probabilistic, fuzzy, etc.) intended to handle uncertain knowledge do have means to represent rules with "relaxed" (uncertain) premises by associating a "degree of uncertainly" with each premise and employing "combining functions" to maintain the uncertainty during the reasoning process. There are two main problems with these logics, though: (i) combining functions are application-dependent, and (ii) it is difficult to provide a reasonable interpretation for numerical values of uncertainty. Truth maintenance logics address these difficulties by providing explicit justifications for derived statements instead of numerical values.

Next, we briefly review one truth maintenance logic (Popchev, Zlatareva, and Mircheva 1990) (we shall refer to it as *TM-logic* for brevity), which was originally introduced as an alternative to Doyle's non-monotonic Truth Maintenance System (Doyle 1979) in order to support inconsistency maintenance, rather than inconsistency resolution. We argue that the TM-logic is a good candidate for implementing the Logic and Proof layer of the Semantic Web because of its expressiveness, computational efficiency, and ability to explicitly maintain inconsistencies during the reasoning process. The latter, as pointed out in (Huang, van Harmelen, and ten Teije 2006), is the better alternative for Semantic Web applications, where resolving inconsistencies may be an impossible task.

Context-dependent rules: basic definitions and notation

To represent and process rules with "relaxed" premises we need a logic that is:

- Expressive enough to accommodate uncertain, incomplete, and inconsistent statements.
- Flexible enough to "adapt" to the current domain specifications by implementing appropriate belief revision whenever necessary.
- Computationally efficient to process large scale web applications.

We show next that the TM-logic is a good candidate for the job. We outline some of its representation and reasoning properties, and discuss how they can be adapted to the needs of our intended application (for more details on the TM-logic, see (Popchev, Zlatareva, and Mircheva 1990)).

The TM-logic addresses the first of the above requirements by defining a family of closely related languages intended to capture different domain concepts:

1. L, which consists of statements (literals) A, ¬A, B, ¬B, etc.

2. L_e, which consists of "endorsed formulas" (e-formulas) explicitly defining contexts where elements from L hold. The general form of e-formulas is $A^{LV}: (T_1,...,T_n)(P_1,...,P_m)$. Here $A \in L$ and $\neg A \notin \{T_1,...,T_n\}$; LV is the logical value of A which can be T (logically true), T* (evidentially true), or U (uncertain, depends on the context); $T_1,...,T_n$, are firm arguments (T-premises) for A, while $P_1,...,P_m$ represent additional evidence for A related to its truth value.

3. $L^+ = L \cup \{C_A, C_B, \ldots\}$, where $C_A = \{A, \neg A\}$, $C_B = \{B, \neg B\}$, etc.

4. $L_e^+ = L_e \cup \{C_A^U: (A, \neg A)(C_A), C_B^U: (B, \neg B) (C_B)$, etc.$\}$, where $\{A^T: (T_1,..,T_n)(\,), \neg A^T: (T_p,..,T_s)(\,)\}$, $\{B^T: (T_1,...,T_n)(\,), \neg B^T: (T_p,...,T_s)(\,)\}$, etc. $\subseteq L_e$

The TM-logic employs two types of inference rules:

- Firm (monotonic) rules or *T-rules*. These have the form $(T_1,...,T_n)(\,) \rightarrow A^T$, and require that all T-premises match logically or evidentially true e-formulas for the rule to fire.

- Plausible rules or *P-rules*. These have the form $(T_1,...,T_n)(P_1,...,P_m) \rightarrow A^U$. The truth value of conclusion A depends on the context where it is derived. The ***minimal context*** is defined by the rule's T-premises. The truth value of the conclusion increases with the number of satisfied P-premises.

To illustrate the notion of the ***context***, consider the following rule:

(size > 45, bedroom > 1, pets-allowed)
(garden, location = Central) ➔ consider^U

The conclusion, *consider*, can be derived in the following three contexts:

1. All T-premises hold, but none of the P-premises hold. This defines the ***minimal context***, and therefore the degree of certainty associated with the conclusion is nominal.

2. All T-premises and all P-premises hold. This defines the ***maximal context***, and therefore the degree of certainty associated with the conclusion is the highest.

3. All T-premises and some of the P-premises hold (either A has a garden, or A is centrally located). T-premises, along with the satisfied P-premises, define the ***context*** in which the conclusion holds, and its degree of certainty depends on the number of satisfied P-premises.

To formalize the idea of "context-dependent inference", the following ***duplicate rules*** are created for each P-rule:

- $(T_1,...,T_n, P_1,...,P_m)(\,) \rightarrow A^{T*}$. This rule, called the *T-duplicate* of the original P-rule, captures the case where all relevant evidence for A holds. It is important to note that T-duplicates are not logically equivalent to T-rules, because they may not define the complete evidence for A.

- For any $\{i_1,...,i_k\} \subseteq \{1,...,m\}$, $(T_1,...,T_n, P_{i1},..., P_{ik})(\{P_1,..., P_m\} \setminus \{P_{i1},...,P_{ik}\}) \rightarrow A^U$. These are called *P-duplicates* of the original P-rule, and they define all possible contexts in which A holds with different degree of certainty.

A pair <E-set, R-set>, where E-set is a set of e-formulas, and R-set is a set of T- and P-rules, and T- and P-duplicates defines a *TM-theory*.

If $A^T : (T_1,...,T_i)\ (\ ) \in$ E-set and $\neg A^T : (T_j,...,T_k)\ (\ ) \in$ E-set, then the set $\{A^T : (T_1,...,T_i)\ (\ ),\ \neg A^T : (T_j,...,T_k)\ (\ )\}$ defines a ***contradiction*** and requires $C_A^U: (A, \neg A)\ (C_A) \in L_e^+$ to be added to the current E-set to represent it.

Contradictions are handled by the TM-logic the same way as other e-formulas. However, to ensure the soundness of the inference relation and the meaningfulness of generated conclusions, the following revisions of current R- and E-sets take place anytime a new contradiction is detected.

Revision of the R-set For each $C_X^U: (X, \neg X)\ (C_X) \in$ E-set, the following transformations take place:

- $(X, T_1,...,T_i)(\ ) \rightarrow A^T \Rightarrow (X, T_1,...,T_i)(C_X) \rightarrow A^U$
- $(\neg X, T_j,...,T_n)(\ ) \rightarrow A^T \Rightarrow (\neg X, T_j,...,T_n)(C_X) \rightarrow A^U$
- $(X, T_1,...,T_i)(P_1,...,P_m) \rightarrow A^U \Rightarrow$
  $\Rightarrow (X, T_1,...,T_i)(P_1,...,P_m, C_X) \rightarrow A^U$
- $(\neg X, T_j,...,T_n)(P_1,...,P_m) \rightarrow A^U \Rightarrow$
  $\Rightarrow (\neg X, T_j,...,T_n)(P_1,...,P_m, C_X) \rightarrow A^U$

Notice that uncertainty associated the conclusions of revised rules increases. Moreover, these conclusions cannot be further matched to T-premises of other rules, which blocks the latter from firing.

Revision of the E-set For each $C_X^U: (X, \neg X)\ (C_X) \in$ E-set, the following transformations take place:

- $A^T : (X, T_1,...,T_i)\ (\ ) \Rightarrow A^U : (X, T_1,...,T_i)\ (C_X)$
- $A^T : (\neg X, T_j,...,T_n)\ (\ ) \Rightarrow A^U : (\neg X, T_j,...,T_n)\ (C_X)$
- $A^U : (X, T_1,...,T_i)\ (P_1,...,P_m) \Rightarrow$
  $\Rightarrow A^U : (X, T_1,...,T_i)\ (P_1,...,P_m, C_X)$
- $A^U : (\neg X, T_j,...,T_n)\ (P_1,...,P_m) \Rightarrow$
  $\Rightarrow A^U : (\neg X, T_j,...,T_n)\ (P_1,...,P_m, C_X)$

Given a TM-theory, <*E-set$_0$*, *R-set$_0$*>, the inference process is carried out as follows:

1. *E-set$_0$* is checked for contradiction, and if such are found do:
   a) For each contradiction, augment *E-set$_0$* with that contradiction matching e-formula from L_e^+.
   b) Revise R-set$_0$ and E-set$_0$ as described.
2. The new *E-set* is computed by augmenting the current one with the conclusions of all applicable rules from the current *R-set*.
3. Newly derived *E-set* is checked for new contradictions, and if such are found do:
   a) For each contradiction, augment *E-set* with contradiction's matching e-formula from L_e^+.
   b) Revise the current *R-* and *E-sets.* Go to 2.
4. If the current *E-set* does not contain new contradictions, and no more rules from the current *R-set* can fire, stop.

It is easy to see that this process will always terminate, resulting in a unique stable extension, *SE*, provided that initial *E-* and *R-sets* are finite.

Carlos example continued

To illustrate the applicability of the TM-logic to our intended application, consider the following set of rules derived from Carlos' specification:

Rule1: If A: size > 45, bedroom > 1, pets-allowed
Then A: consider

Rule2: If A: consider, price < = 250, floor < 3
Then A: make-offer

Rule3: If A: consider, price < = 300
IN SPITE garden, lift, location = Central
Then A: make-offer

Rule4: If A: consider, price < = 400, location = Central
IN SPITE garden, lift
Then A: make-offer

Rule5: If A: consider, price > 400
Then A: stop

Rule6: If A: location = Central, swimming-pool
Then A: ¬stop

Rule7: If A: consider, floor < 3, ¬stop
IN SPITE garden
Then A: make-offer

Rule8: If A: consider
IN SPITE ¬floor < 3, ¬lift
Then A: ¬make-offer

Assume further that apartment ads do not come in a predefined format. Some features, such as price, size, number of bedrooms, and pet-friendliness are likely to be mentioned, while others such as location or floor number may be omitted. Here are a few likely apartment ads:

A1: price = 450, bedrooms = 2, size = 50, location = central, pets-allowed, swimming-pool, floor = 2, garden.

A2: price = 280, bedrooms = 3, size = 65, floor = 4, pets-allowed, lift.

A3: price = 330, bedrooms = 2, size = 55, location = central, swimming-pool, lift, garden.

A4: price = 350, bedrooms = 3, size = 55, location = central, pets-allowed, garden, lift.

A5: price = 235, bedrooms = 2, size = 45, pets-allowed, garden, floor = 2.

For brevity, we show below only a small part of the resulting TM-theory:

R-set$_0$ = {… r2[(consider, price<=250, floor<3) ( ) ➔ make-offer^T],
r3[(consider, price<=300) (garden, lift, location=Central) ➔ make-offer^U],
r3T[(consider, price<=300, garden, lift, location=Central) ( ) ➔ make-offer^{T*}],
r3P1[(consider, price<=300, garden) (lift,location=Central) ➔ make-offer^U],
r3P2[(consider, price<=300, garden, lift) (location=Central) ➔ make-offer^U],…}

E-set$_0$ = {A1[price=450, bedrooms=2, size=50, location=Central, garden pets-allowed, swimming-pool, floor = 2],

A2[price=280, bedrooms=3, size=65, floor=4, pets, lift], … }

The stable extension, *SE*, of this theory is the following:

{A1[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
stop^T: (consider, price>400),
¬stop^T: (location=Central, swimming-pool),
$C_{stop,\neg stop}{}^U$: (stop, ¬stop) ($C_{stop,\neg stop}$),
make-offer^U: (consider, floor<3, ¬stop, garden) ($C_{stop,\neg stop}$),
¬make-offer^U: (consider) (¬floor<3, ¬lift)],

A2[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
make-offer^U: (consider, price<300, lift) (garden, location=Central),
¬make-offer^U: (consider) (¬floor<3, ¬lift)],

A4[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
make-offer^{T*}: (consider, price<300, lift, garden, location=Central) ( ),
¬make-offer^U: (consider) (¬floor<3, ¬lift)],

A5[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
make-offer^T: (consider, price<250, floor<3) ( ),
make-offer^U: (consider, price<300, garden) (lift, location=Central),
¬make-offer^U: (consider) (¬floor<3, ¬lift)] }

Identification of admissible conclusions

Notice that the stable extension in Carlos example contains multiple e-formulas associated with the same statement or its negation. Each of these e-formulas describes a specific context with respect to which the truth value of a statement is defined. Consider, for example, A5: make-offer and A5: ¬make-offer. There are two different contexts associated with the former, and one associated with the latter. If we can order these contexts with respect to their "plausibility", we can filter out e-formulas defining less plausible contexts and leave only the most plausible ones. We call such e-formulas ***admissible conclusions***, and they comprise the ***set of admissible conclusions***, *SAC*, of the TM-theory.

Definition SAC ⊆ SE such that:

a) If {$A^T : (T_1,...,T_i)(\,)$, $A^{T*} : (T_j,...,T_n)(\,)$} ⊆ SE and {$T_j,...,T_n$} ⊆ {$T_1,...,T_i$}, then only $A^T : (T_1,...,T_i)(\,) \in$ SAC.
b) If {$A^{T*} : (T_j,...,T_n)(\,)$, $A^U : (T_1,...,T_i)(P_1,...,P_i)$} ⊆ SE, and {$T_1,...,T_i$} ⊆ {$T_j,...,T_n$}, then only $A^{T*} : (T_1,...,T_i)(\,) \in$ SAC.
c) If {$A^T : (T_1,...,T_i)(\,)$, $\neg A^T : (T_j,...,T_n)(\,)$} ⊆ SE, then {$A^T : (T_1,...,T_i)(\,)$, $\neg A^T : (T_j,...,T_n)(\,)$} ⊆ SAC.
d) If {$A^T : (T_1,...,T_i)(\,)$, $A^U : (T_j,...,T_n)(P_1,...,P_m)$} ⊆ SE and {$T_j,...,T_n$} ⊆ {$T_1,...,T_i$}, then only $A^T : (T_1,...,T_i)(\,) \in$ SAC.
e) If {$A^U : (T_1,...,T_i)(P_1,...,P_i)$, $A^U : (T_j,...,T_n)(P_j,...,P_n)$} ⊆ SE, and {$T_1,...,T_i$} ⊆ {$T_j,...,T_n$}, then only $A^U : (T_j,...,T_n)(P_j,...,P_n) \in$ SAC.
f) If {$\neg A^U : (T_1,...,T_i)(P_1,...,P_i)$, $A^U : (T_j,...,T_n)(P_j,...,P_n)$} ⊆ SE and {$T_1,...,T_i$} ⊆ {$T_j,...,T_n$} and some P_k (k=1,i) = $\neg T_l$ (l = j,n), then only $A^U : (T_j,...,T_n)(P_j,...,P_n) \in$ SAC.
g) If {$\neg A^U : (T_1,...,T_i)(P_1,...,P_m)$, $A^T : (T_j,...,T_n)(\,)$} ⊆ SE, and some P_x (x = 1,m) ∈ {$T_j,...,T_n$}, then only $A^T : (T_j,...,T_n)(\,) \in$ SAC.

To get the final answer to Carlos' query, we proceed in two steps.

Step 1 Compute the set of admissible conclusions associated with each of the possible choices, A1, A2, A4, and A5:

{A1[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
stop^T: (consider, price>400),
¬stop^T: (location=Central, swimming-pool),
$C_{stop,\neg stop}{}^U$: (stop, ¬stop) ($C_{stop,\neg stop}$),
make-offer^U: (consider, floor<3, ¬stop, garden) ($C_{stop,\neg stop}$)],

A2[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
make-offer^U: (consider, price<300, lift) (garden, location=Central)],

A4[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
make-offer^{T*}: (consider, price<300, lift, garden, location=Central) ( )],

A5[consider^T: (size>45, bedroom>1, pets-allowed) ( ),
make-offer^T: (consider, price<250, floor<3) ( )] }

Notice that e-formulas in *SAC* explicate the highest degree of "desirability" for each possible choice, but say nothing about how these choices compare between themselves.

Step 2 Compare the "desirability" of possible choices to identify one or several "best" choices. For that: (i) explicate the support evidence (we call it the ***grounded justification***) for each choice by computing the transitive closure of its T-premises, and (ii) order grounded justifications as defined above, thus further restricting the SAC with respect to the choices themselves. Here are the grounded justifications for all possible choices:

A1[make-offer^U: (size>45, bedroom>1, pets-allowed, floor<3, location=Central, swimming-pool, garden) ($C_{stop,\neg stop}$)]
A2[make-offer^U: (size>45, bedroom>1, pets-allowed, price<300, lift) (garden, location=Central)]
A4[make-offer^{T*}: (size>45, bedroom>1, pets-allowed, price<300, lift, garden, location=Central) ( )]
A5[make-offer^T: (size>45, bedroom>1, pets-allowed, price<250, floor<3) ( )]

Notice that A5: make-offer^T , satisfies all firm requirements and is derived with the highest degree of certainty, T. However, it does not dominate A4: make-offer^{T*}, because according to our definition (item a) (size>45, bedroom>1, pets-allowed, price<300, lift, garden, location=Central) $\not\subset$ (size>45, bedroom>1, pets-allowed, price<250, floor<3). Therefore, A4 must remain in SAC. On the other hand, A2: make-offer^U is dominated by A4 according to our definition (item b), because (size>45, bedroom>1, pets-allowed, price<300, lift) $\subset$ (size>45, bedroom>1, pets-allowed, price<300, lift, garden, location=Central). Therefore, A2 can be eliminated from the SAC. Finally, A1 which was derived as a possible choice regardless the explicit contradiction originated by the price, should also remain in SAC because according to our definition (item d) (size > 45, bedroom > 1, pets-allowed, floor < 3, location=Central, swimming-pool, garden) $\not\subset$ (size > 45, bedroom > 1, pets-allowed, price < 250, floor < 3).

Here is the final answer to Carlos query:

- **Best choice:** Apartment 5, because size > 45, bedroom > 1, pets-allowed, price < 250, floor < 3.
- **Second best choice:** Apartment 4, because size > 45, bedroom > 1, pets-allowed, price < 300, lift, garden, location = Central.
- **Possible choice:** Apartment 1, because size > 45, bedroom > 1, pets-allowed, location = Central, swimming-pool, garden, but price > 400.

Now, it is up to Carlos to decide which of the presented choices he likes the most.

Conclusion

In this paper, we argued that ensuring the completeness and consistency of large Semantic Web ontologies is practically impossible and therefore uncertainty, incompleteness and inconsistency must be adequately represented and maintained. To address this need, we have proposed an extension to the current RuleML format which makes it possible to define "relaxed" premises. We have shown how such "context-dependent" rules can be handled by one truth maintenance logic, the *TM-logic*, originally introduced in (Popchev, Zlatareva, and Mircheva 1990). We have also discussed how this logic can be adapted and extended to better address the needs of Semantic Web applications. To justify and illustrate our ideas, a detailed example was discussed throughout the paper.

In our future work, we intend to address the integration of the proposed rule representation format into the current Rule Interchange Format recommended by W3C.

Acknowledgements Many thanks to anonymous referees for their thoughtful comments and suggestions.

References

Antoniou, G., and van Harmelen, F. 2007. *A Semantic Web Primer*. MIT Press.

Damasio, C., Pan, J., Stoilos, G., and Straccia, U., 2007. Representing Uncertainty in RuleML. *Fundamenta Informaticae*: 1 – 24, IOS Press.

Davis, J., Studer, R., and Warren, P., 2006. Conclusion and Outlook (page 305). In Davis, Studer, and Warren (eds.) *Semantic Web Technologies: Trends and Research in Ontology-Based Systems.* John Wiley & Sons, Ltd.

Doyle, J., 1979. A Truth Maintenance System. *Artificial Intelligence*, 12: 231 – 272.

Huang, Z., van Harmelen, F., and ten Teije, A., 2006. Reasoning with Inconsistent Ontologies: Framework, Prototype, and Experiment. *Semantic Web Technologies: Trends and Research in Ontology-Based Systems* (eds. John Davies, Rudi Studer, Paul Warren), John Wiley & Sons, Ltd.

Lukasiewicz, T., 2008. Expressive Probabilistic Description Logics. *Artificial Intelligence*, 172, (6-7): 852 -- 888.

Popchev I., Zlatareva, N., and Mircheva, M., 1990. A Truth Maintenance Theory: An Alternative Approach. In *Proc. 9th European Conference on Artificial Intelligence (ECAI'90)*.

Schobach, S., and Corner, R. 2003. Non-standard Reasoning Services for the Debugging of Description Logic Terminologies. In Proc. *International Joint Conferences on Artificial Intelligence'2003*.