Editor: Steffen Staab
University of Koblenz-Landau
staab@uni-koblenz.de

Uncertainty and the Semantic Web

Giorgos Stoilos, Nikos Simou, Giorgos Stamou, and Stefanos Kollias,
National Technical University of Athens

The Semantic Web must handle information from applications that have special knowledge representation needs (such as multimedia-processing, geospatial, and situation-awareness applications) and that face uncertain, imprecise knowledge. More precisely, some applications deal with random information and events, others deal with imprecise and fuzzy knowledge, and still others deal with missing or distorted information—resulting in uncertainty. For example, in applications involving sensor readings, such measurements usually come with degrees of evidence; in applications like multimedia processing, object recognition might come with degrees of truth.

To deal with uncertainty in the Semantic Web and its applications, many researchers have proposed extending OWL and the Description Logic (DL) formalisms with special mathematical frameworks. Researchers have proposed probabilistic,[1] possibilistic,[2] and fuzzy extensions,[3–5] among others. Researchers have studied fuzzy extensions most extensively, providing impressive results on semantics, reasoning algorithms, and implementations. Building on these results, we've created a fuzzy extension to OWL called Fuzzy OWL. Fuzzy OWL can capture imprecise and vague knowledge—for example, we can say that Athens is hot to a degree 0.8 rather than saying that Athens is either hot or not. Moreover, our reasoning platform, Fuzzy Reasoning Engine (FiRE), lets Fuzzy OWL capture and reason about such knowledge (see www.image.ece.ntua.gr/~nsimou).

Uncertainty representation

Many research communities have exploited Semantic Web technologies to build interoperable applications. Consider multimedia databases—many multimedia documents (such as images, video, and sound records) reside in huge databases of production companies, museums, and TV channels. For these documents to be available in a semantically rich manner, they must be (semi)automatically processed and annotated with the aid of knowledge representation languages. But processing and describing multimedia documents involves a lot of uncertain information. For example, one meaningful query could be, "Get all pictures that illustrate mountains with tall trees and a blue sky with few clouds." Such a query involves many vague concepts, including "tall," "blue," and "few."

The problem with proposed uncertainty extensions to Semantic Web languages is that uncertainty comes in many flavors, so there couldn't be just one global extension. For uncertainty resulting from incomplete or distorted knowledge, you might assign possibility degrees to the possible alternatives. For uncertainty resulting from our inability to precisely define concepts, you might assign degrees of truth. (Although fuzziness isn't a type of uncertainty, we make that assumption here to simplify our presentation.) And for uncertainty resulting from several conflicting alternatives, you might assign degrees of probability. To this extent, extensions to OWL and DLs feature different mathematical and logical properties. For example, both probabilistic and possibilistic logics aren't truth functional, but fuzzy logic is.

Fuzzy OWL

As with OWL, Fuzzy OWL's building blocks are classes, properties, and individuals. Although in crisp (not fuzzy) OWL, these entities represent (crisp) sets of objects, in our case they're fuzzy classes and properties. More precisely, a fuzzy class A is seen as a fuzzy set over a universe of discourse X. This is defined by a membership function $A : X \rightarrow [0, 1]$, which given an object $x \in X$ returns the degree that x belongs to A. On the other hand, a fuzzy property R is interpreted as a fuzzy relation over the set $X \times X$, defined by the function $R : X \times X \rightarrow [0, 1]$. For example, the class **Tall** is the fuzzy set of tall people and **Tall(George) = 0.8** says that George is tall to a degree 0.8.

Like OWL, Fuzzy OWL lets you specify intentional knowledge. For example, we could define the class **TennisBall** as something that's yellow and round with a white stripe. We could also define the class **Yellow** as something that's **Green** and **Red**, and ¬**Blue** (that is, **Not Blue**) or the class **White**. Such definitions involve several fuzzy classes. Consider a segment (that is, any region of the image) whose red, green, and blue components each have the value 255 in the RGB

1541-1672/06/$20.00 © 2006 IEEE
Published by the IEEE Computer Society

color model—that is, a white segment. What about a segment whose components have the values 255, 240, and 236? It's better to specify the degree to which the segment is White rather than strictly classifying it as White or not. We could make similar arguments for other concepts such as Yellow and Round.

Suppose that we apply an image processing algorithm over an image and that, for a particular segment o_1, it returns the values red 235, green 240, and blue 30. The algorithm then looks at the fuzzy sets that define the fuzzy concepts Red, Green, and Blue and specifies the degree to which o_1 belongs to them. The domain expert defines these fuzzy sets. For example, the fuzzy set for Red can look like that in figure 1 (that is, o_1 is red to a degree 0.8). Similarly, the fuzzy sets of Green and Blue could be defined by the same function as that of Red. So, o_1 is green to a degree 0.85 and blue to a degree 0.05. Suppose also that after some processing, the algorithm estimates that o_1 contains part of another segment, o_2, to a degree 0.75. We can represent this with the following RDF/XML syntax in Fuzzy OWL's DL language.

```
<rdf:Description rdf:about="o1">
        <rdf:type rdf:resource="Red" owlx:ineq
        Type="≥" owlx:degree="0.8"/>
        <hasPart rdf:resource="o2" owlx:ineq
        Type="≥" owlx:degree="0.75"/>
</rdf:Description>
```

Fuzzy OWL uses crisp OWL's syntax for class and property axioms and definitions. It minimally extends the syntax of OWL facts to fuzzy facts and extends crisp OWL's semantics. Fuzzy OWL semantics are based on membership functions and fuzzy set theoretic operations, which provide meaning to conjunction, disjunction, negation, and logical implication. For example, consider the class Yellow that we defined earlier. Suppose that we use the Gödel conjunction (intersection) given by $t(a, b) = \min(a, b)$ and the Lukasiewicz negation given by $c(a) = 1 - a$. From the fuzzy facts we gave earlier and the definition of Yellow, we can deduce that o_1 is yellow to a degree $\text{Yellow}(o_1) = \min(0.8, \min(0.85, 1 - 0.05)) = \min(0.8, 0.85) = 0.8$.

Because of the mathematical properties of the fuzzy set theoretic operations, if we restrict our attention to the extreme limits of 0 and 1, the Fuzzy OWL DL semantics coincide with those of the crisp OWL DL. So, we usually refer to Fuzzy OWL as a *sound extension* of OWL.

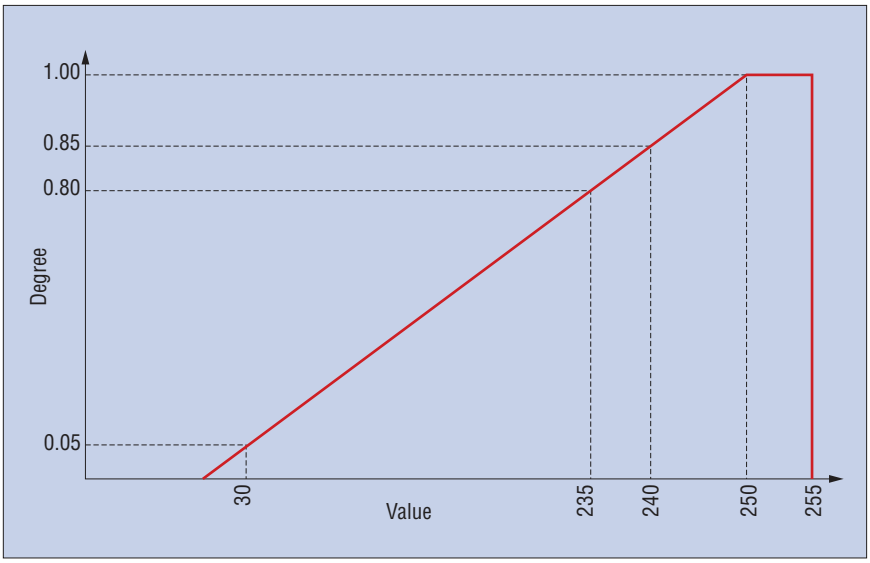

Figure 1. Definition of the fuzzy set *Red*.

FiRE: A fuzzy reasoning engine

FiRE is a prototype implementation of a reasoning algorithm for an expressive fuzzy DL language, f_{KD}-*SHIN*.[6] This algorithm builds on previous results about reasoning with fuzzy DLs by extending f_{KD}-*ALC*'s reasoning algorithm[5] to handle most of OWL's features. Figure 2 illustrates the FiRE platform's GUI. FiRE consists of three components: the editor panel, inference services panel, and output tabs.

Editor panel

This panel lets us open and edit a knowledge base or create a new one from scratch. FiRE uses the same syntax as the RACER DL reasoning engine (www.sts.tu-harburg.de/~r.f.moeller/racer) to encode captured knowledge. In FiRE, we had to slightly extend RACER's syntax to support fuzzy facts.

Using the keyword equivalent gives us definitions for the fuzzy concepts TennisBall, White, and Yellow (see figure 2). Using the keywords instance and related as well as an inequality type and a membership degree, we can define fuzzy facts. In addition to the fuzzy facts we already introduced, we've specified that segment o_2 is red to a degree 0.6, green to a degree 0.5, and blue to a degree 0.9 and that its shape represents a stripe to a degree 0.8. Furthermore, we've extended our knowledge about segment o_1, which is round to a degree 0.6.

Inference services panel

The FiRE platform supports three types of inferences. The first type involves checking a fuzzy knowledge base's consistency. To provide reasoning support for Fuzzy OWL, we've reduced a Fuzzy OWL ontology to a fuzzy DL knowledge base.[4]

The latter two focus more on querying given knowledge to derive new implied knowledge. In the first case, Fuzzy OWL supports the entailment of a fuzzy fact. For example, one useful query to our fuzzy knowledge would be to ask if segment o_1 is yellow to a degree greater or equal than 0.8. The user can input this query in the inference services panel using RACER syntax. For our fuzzy knowledge, the answer is positive. Another fuzzy fact that our knowledge entails is that o_1 is a tennis ball to a degree greater or equal than 0.5.

The second query-related inference service is the subsumption between two fuzzy concepts. For example, we can query whether the concept Yellow is a subconcept of the conjunction of the fuzzy concepts Red, Green, and ¬Blue, which is obviously true. RACER syntax specifies subsumption with the keyword implies.

Output tabs

FiRE uses several output tabs to provide information about the fuzzy knowledge base. Figure 2 shows the model output tab, which returns the model (fuzzy interpretation) that satisfies the concept, role, and instance axioms the fuzzy knowledge base specified. It also shows a model of the fuzzy knowledge base after we've checked its con-

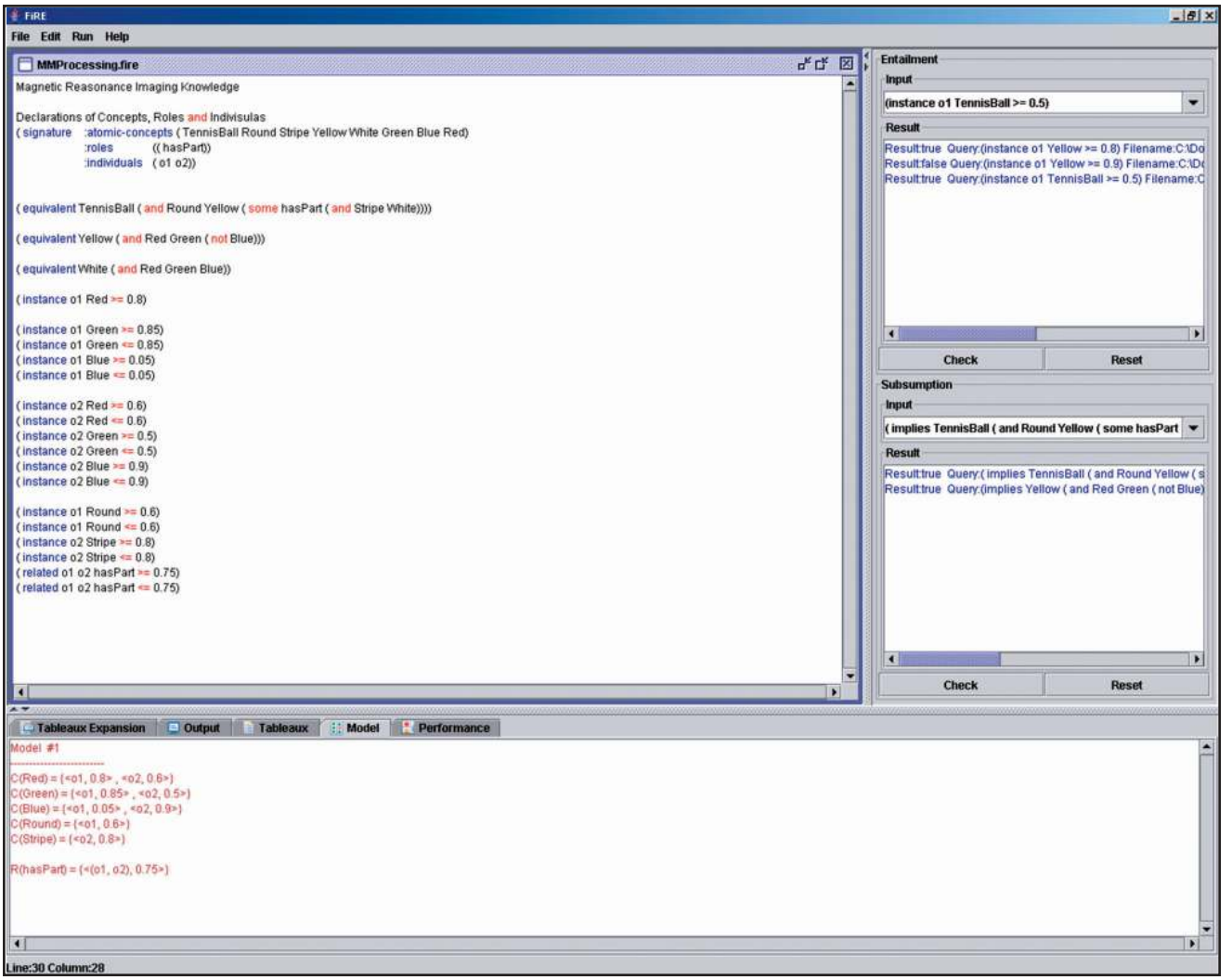

Figure 2. The FiRE user interface consists of an editor panel (upper left), inference services panel (upper right), and output tabs (the bottom).

sistency. We can see that Red is a fuzzy concept with object o_1 belonging to a degree 0.8 and object o_2 belonging to a degree 0.6. Like other fuzzy concepts, the hasPart fuzzy property means that the pair (o_1, o_2) belongs to a degree 0.75. Also important is the tableaux expansion tab, where we can see a trace of the reasoning algorithm's application. In the tableaux tab, we can see the reasoning algorithm's created structure; the output tab, which shows the initial fuzzy knowledge; and the performance tab, which provides view information about PC resource usage.

Uncertainty representations with rules

The rules layer is directly above OWL and the ontology layer in the Semantic Web stack. The W3C working group for rules (www.w3.org/2005/rules) focuses mainly on providing a Rule Interchange Format rather than a single Semantic Web language. Another effort is the Rule Markup Language initiative (www.ruleml.org). RuleML provides a set of markups suitable for representing and interchanging different types of rules.

Many individuals and groups have proposed extending rule systems with mathematical frameworks capable of representing uncertain information. More precisely, approaches exist for probabilistic, possibilistic, and fuzzy rule languages, which aim to capture and handle different types of uncertainties.[7] The fuzzy RuleML Technical Group was established in August 2005 to extend RuleML's functionality and provide ways to interchange uncertainty rule languages (www.image.ece.ntua.gr/FuzzyRuleML). The group's aim is to provide syntactic extensions to RuleML to represent uncertainty logic programming ap-proaches. Similar to ontologies and OWL, these extensions must be as minimal as possible.

The technical group's first extension was motivated by fuzzy logic programming approaches. So, fuzzy RuleML proposed a way to specify a membership degree when creating RuleML facts. For example, you could provide a markup for the fuzzy fact Green Eyes(Dora) ≥ 0.8—that is,

```
<Atom>
    <degree><Data>0.8</Data></degree>
    <_opr><Rel>Green_Eyes</Rel></_opr>
    <Ind>Dora</Ind>
</Atom>
```

We don't need to specify the inequality ≥ because traditionally, uncertainty logic programming approaches only consider this inequality. The only additional element that this syntax provides is the element <degree>. Interestingly, the RuleML 0.9 schema specification (www.ruleml.org/0.9) already supports this element. The technical group is investigating ways to further uncertainty logic programming languages, such as probabilistic, possibilistic, and fuzzy approaches. For example, this could help you capture the rule "the probability of getting stuck in traffic before reaching 4th Ave. from the south is at least 0.9" or "if a painting has specific features, the possibility of it being Raphael's is at least 0.8."

In future work, we could extend FiRE in several ways:

- *Extend to other uncertainty formalisms.* Studying and integrating other types of uncertainties is one of our future goals.
- *Extend the DL component's expressiveness*. FiRE supports a fuzzy version of the DL language *SHIN*. *SHIN* is expressive, but algorithms for even more expressive DL languages, such as *SHOIQ*, exist.
- *Extend the fuzzy component's expressiveness.* FiRE supports the Zadeh fuzzy version of *SHIN*, f_{KD}-*SHIN*. Apart from the Zadeh fuzzy operators, several others result in different logical properties.
- *Support data types.* FiRE doesn't yet support data types. Proposals exist for fuzzy data type expressions, such as "about 15," a data type defined as a fuzzy set around the value 15.
- *Support rules.* Previous efforts have mainly focused on providing a rule interchange format rather than a single rule language, but several proposals for a Semantic Web rule language exist. Integrating fuzzy rules with fuzzy DLs is another interesting extension. ■

References

1. Z. Ding and Y. Peng, "A Probabilistic Extension to Ontology Language OWL," *Proc. 37th Ann. Hawaii Int'l Conf. System Sciences* (HICSS 04), IEEE CS Press, 2004, pp. 40111.1.
2. H. Bernhard, "An Alternative Proof Method for Possibilistic Logic and Its Application to Terminological Logics," *Proc. 10th Ann. Conf. Uncertainty in Artificial Intelligence* (UAI 94), Morgan Kaufmann, 1994, pp. 327–335.
3. U. Straccia, "Towards a Fuzzy Description Logic for the Semantic Web," *The Semantic Web: Research and Applications: 2nd European Semantic Web Conf.,* LNCS 3532, Springer, 2005, pp. 167–181.
4. G. Stoilos et al., "Fuzzy OWL: Uncertainty and the Semantic Web," *Proc. Int'l Workshop OWL: Experiences and Directions,* 2005; http://image.ntua.gr/papers/398.pdf.
5. U. Straccia, "Reasoning within Fuzzy Description Logics," *J. Artificial Intelligence Research,* vol. 14, Apr. 2001, pp. 137–166.
6. G. Stoilos et al., "The Fuzzy Description Logic f-SHIN," *Proc. Int'l Workshop Uncertainty Reasoning for the Semantic Web*, 2005, CEUR Electronic Workshop Proc., pp. 67–76; www.image.ntua.gr/papers/385.pdf.
7. C.V. Damásio and L.M. Pereira, "Sorted Monotonic Logic Programs and Their Embeddings," *Proc. 10th Int'l Conf. Information Processing and Management of Uncertainty* (IPMU 04), 2004, pp. 807–814.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.

Giorgos Stoilos is a PhD student in the Image, Video, and Multimedia Laboratory of the National Technical University of Athens. Contact him at gstoil@image.ece.ntua.gr.

Nikos Simou is a PhD student in the Image, Video, and Multimedia Laboratory of the National Technical University of Athens. Contact him at nsimou@image.ece.ntua.gr.

Giorgos Stamou is a research assistant professor at the Institute of Communication and Computer Systems at the National Technical University of Athens. Contact him at gstam@softlab.ece.ntua.gr.

Stefanos Kollias is a professor in the Department of Electrical and Computer Engineering at the National Technical University of Athens. Contact him at stefanos@cs.ntua.gr.