# The Generation of DAML+OIL

Ian Horrocks
Department of Computer Science,
University of Manchester, UK
horrocks@cs.man.ac.uk

Peter F. Patel-Schneider
Bell Labs Research
Murray Hill, NJ, USA
pfps@research.bell-labs.com

**Abstract**

DAML+OIL is a new description logic developed for use within the DAML project and as a submission to the upcoming W3C semantic web ontology working group. It is closely based on the OIL, but also has strong influences from the existing W3C efforts as well as input from DAML researchers. DAML+OIL pushes very close to the undecidability barrier. Developing effective reasoners for DAML+OIL will stress the capabilities of description logic researchers.

## 1 Introduction

The Ontology Inference Layer (OIL) [4, 5] is a logic for representing ontologies. OIL was designed for use in the semantic web [1], as it walks the line between inadequate expressive power, on one hand, and computational infeasability, on the other; is built on top of XML [8]; and is relatively compatible with the Resource Description Framework (RDF) [6] and RDF Schema (RDFS) [2].

However, OIL was not considered to be quite adequate by some groups. Some people within the World Wide Web Consortium (W3C)[1] felt that OIL was not sufficiently compatible with RDF and RDFS. Some people within the Darpa Agent Modelling Language (DAML)[2] project felt that OIL was not sufficiently expressive for use as an ontology language.

[1]http://www.w3.org/
[2]http://www.daml.org/

To address these concerns a group of researchers from the DAML project, from the OIL project, and from W3C, as well as an outside expert have been working to create a new ontology language. This language is being used within the DAML project to represent ontology information. It will also be submitted to an upcoming W3C working group on web ontologies. The language, christened DAML+OIL,[3] has some interesting features for a description logic, and certainly has had an interesting genesis.

## 2 DAML+OIL

DAML+OIL was designed to have a much closer relationship to RDF and RDFS than does OIL. To this end the syntax of DAML+OIL is RDF and, moreover, is specified as an RDF document, available at

http://www.daml.org/2001/03/daml+oil.daml.

All DAML+OIL constructs are composed of RDF triples, using RDF classes and properties that have special meaning in DAML+OIL.

Unfortunately, syntax specification via RDF is much more verbose than the usual description logic specification, and thus the syntax of DAML+OIL will not fit within this paper. We present a couple of examples of the syntax specification and syntax of DAML+OIL to give a general idea of how DAML+OIL is written.[4]

To create a concept, one creates an RDF resource that is an instance of daml:Class. This is specified as:

```
<rdfs:Class rdf:ID="Class">
  <rdfs:subClassOf
        rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdfs:Class>
```

To create value restrictions, one uses an instance of the daml:Restriction class with a daml:onProperty link to the property and a daml:toClass link to the restriction class. This is specified as:

```
<rdfs:Class rdf:ID="Restriction">
  <rdfs:subClassOf rdf:resource="#Class"/>
</rdfs:Class>

<rdf:Property rdf:ID="onProperty">
```

[3] For current information on DAML+OIL, see http://www.daml.org/. The current version of DAML+OIL, as of the writing of this paper is at http://www.daml.org/2001/03/daml+oil-index.

[4] The exposition here is necessarily abbreviated, and ignores some aspects of DAML+OIL. For full details see the DAML+OIL specification.

```
  <rdfs:domain rdf:resource="#Restriction"/>
  <rdfs:range rdf:resource=
              "http://www.w3.org/1999/02/22-rdf-syntax-ns#Property"/>
</rdf:Property>

<rdf:Property rdf:ID="toClass">
  <rdfs:domain rdf:resource="#Restriction"/>
  <rdfs:range rdf:resource=
              "http://www.w3.org/2000/01/rdf-schema#Class"/>
</rdf:Property>
```

To create the primitive concept, called sf Primitive, that is a subconcept of person and has its friends restricted to doctors, one would write:

```
<daml:Class rdf:ID="Primitive">
  <rdfs:subClassOf rdf:resource="#Person"/>
  <rdfs:subClassOf>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#friend"/>
      <daml:toClass rdf:resource="#Doctor"/>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

To create a defined version of this concept, called Defined, one would write:

```
<daml:Class rdf:ID="Defined">
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Person"/>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#friend"/>
      <daml:toClass rdf:resource="#Doctor"/>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>
```

DAML+OIL has an extensive "concrete" domain, taken from XML Schema datatypes[7]. XML Schema datatypes can be used as DAML+OIL datatypes and in DAML+OIL datatype properties. For example, one can create the property age which has integral values via:

```
<daml:DatatypeProperty rdf:ID="age">
  <rdf:type rdf:resource=
            "http://www.daml.org/2001/03/daml+oil#UniqueProperty"/>
  <rdfs:range rdf:resource=
            "http://www.w3.org/2000/10/XMLSchema#nonNegativeInteger"/>
</daml:DatatypeProperty>
```

DAML+OIL has a large collection of concept constructors and other concept-restricting constructs, for "abstract" concepts, including subsumption, equivalence, and disjointness; union, intersection, and complement; universal and existential restrictions; qualified and unqualified cardinality restrictions; fillers; and one-of. DAML+OIL includes facilities for the most-common kinds of roles, including role hierarchy; domain and range restrictions on roles; transitive roles; and unique and unambiguous roles. DAML+OIL does not have the transitive closure role constructor.

DAML+OIL has statements about individuals, as well as concept and property definitions. To state that an individual is an instance of a concept, one says that the rdf:type of the individual is the concept, as in

```
<rdf:Description rdf:ID="peter">
  <rdf:type rdf:about="Person"/>
</rdf:Description>
```

To state that an individual is related to another individual via a property, on directly uses an RDF triple, as in

```
<rdf:Description rdf:about="peter">
  <friend rdf:about="ian"/>
</rdf:Description>
```

or

```
<rdf:Description rdf:about="peter">
  <age>13</age>
</rdf:Description>
```

Usually, a collection of information about an individual is given as RDF shorthand, as in

```
<Person rdf:ID='peter'>
  <friend rdf:resource="ian"/>
  <age rdf:value="13"/>
</Person>
```

which is in all ways equivalent to the above two statements.

There are two semantic specifications for DAML+OIL.

The first is a relatively standard model-theoretic semantics, which is available at http://www.daml.org/2001/03/model-theoretic-semantics.html. This semantics provides a mapping from a subset of RDF triples to the usual extensions. The only unusual aspects of this semantics have to do with the RDF triple syntax and the XML Schema datatype concrete domains.

The second semantic specification for DAML+OIL is an axiomatization of RDF, RDF Schema, and DAML+OIL in KIF [3].

# 3 Unusual Features of DAML+OIL

The description of DAML+OIL above shows that DAML+OIL is a description logic in RDF clothes. Taken as a description logic, DAML+OIL has a number of unusual features.

Some of these features come from the RDF-based syntax for DAML+OIL. The syntax makes it awkward to write DAML+OIL directly. We expect that users will not write DAML+OIL directly, instead using an special-purpose ontology editor. This will hide the syntax and allow users a more-intuitive way of creating DAML+OIL knowledge bases.

Another unusual feature of DAML+OIL that comes from its RDF syntax is that there is no sense of a main definition of a class or individual. There is just a collection of triples in a knowledge base, some of which are about a particular class or individual. This is not really a problem for the logic, as the logic doesn't care how the knowledge base is organized. However, some people have complained that DAML+OIL will lose organizational information that they are used to seeing in description logic knowledge bases.

The "concrete" portion of DAML+OIL is taken from XML Schema datatypes. Thus DAML+OIL implicitly has a large collection of primitive datatypes, such as integers, as well as constructors for other datatypes including restricted datatypes, such as non-negative integers, and union datatypes, taken from XML Schema datatypes.

DAML+OIL has a full assertional (A-box) component, but, as well, allows individuals to occur in concept definitions. This extends DAML+OIL beyond the usual description logic expressive power, and will make reasoners for DAML+OIL difficult to construct. This feature was strongly requested by DAML participants, so it had to be incorporated into DAML+OIL.

DAML+OIL is unusual in that it has two semantics, the more-usual model-theoretic semantics and an axiomatization. Participants in the development of DAML+OIL were dissatisfied with only having the model-theoretic semantics, so both semantics are considered to be part of the definition of DAML+OIL.

The terminology used within the description language community was not acceptable to many people outside it. The terminology in DAML+OIL thus does not correspond completely with description logic usage. For example, DAML+OIL uses the RDF and RDFS terminology "resource", "class", and "property" instead of the description logic terminology "entity", "concept", and "relationship".

Even where there was no existing terminology in RDF and RDF, the description logic names were considered unacceptable. Thus DAML+OIL uses "data value" and "datatype" instead of "concrete object" and "concrete class", and "object class" instead of "abstract class".

# 4 Summary

The effort of creating DAML+OIL has (already) been quite long and arduous. As with any result of a large group, there are lots of compromises in DAML+OIL. However, we feel that the result has been worth the effort, as there is now a description logic that has a very good change of being used by a large community. To us, DAML+OIL *is* a description logic, albeit a description logic with a rather unusual syntax.

It is now up to the description logic community to ensure that DAML+OIL has no (large) problems, and to work hard to produce reasoners for DAML+OIL.

# References


[1] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, May 2001.

[2] Resource description framework (RDF) schema specification 1.0. http://www.w3.org/TR/rdf-schema, March 2000.

[3] Richard Fikes and Deborah L. McGuinness. An axiomatic semantics for RDF, RDF-S, and DAML+OIL. http://www.daml.org/2001/03/axiomatic-semantics.html, March 2001.

[4] I. Horrocks, D. Fensel, J. Broekstar, S. Decker, M. Erdmann, C. Goble, F. van Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta. The ontology inference layer OIL. http://www.ontoknowledge.org/oil/TR/oil.long.html, 2000.

[5] Ian Horrocks, Peter Patel-Schneider, Frank van Harmelen, and Dieter Fensel. OIL: A good ontology language for the semantic web. In *Tenth International World Wide Web Conference*, May 2001.

[6] Resource description framework (RDF) model and syntax specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, February 1999.

[7] XML schema part 2: Datatypes. http://www.w3.org/TR/xmlschema-2/, March 2001.

[8] Extensible markup language (XML) 1.0 (second edition). http://www.w3.org/TR/REC-xml, 2000.