



Reviewing the Design of DAML+OIL: An Ontology Language for the Semantic Web



Ian Horrocks

University of Manchester
Manchester, UK
horrocks@cs.man.ac.uk

Peter F. Patel-Schneider

Bell Labs Research
Murray Hill, NJ, U.S.A.
pfs@research.bell-labs.com

Frank van Harmelen

Vrije Universiteit
Amsterdam, the Netherlands
Frank.van.Harmelen@cs.vu.nl

Influences and Constraints

The Semantic Web

- The Semantic Web layer cake
- RDF and RDF Schema

Ontologies

- Definitions of terms, e.g., Student

Description Logics (DLs)

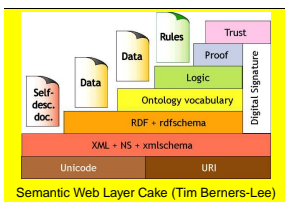
- Formal semantics
- Practical inference
- OIL—a predecessor of DAML+OIL
 - A description logic for the web
 - Frame-oriented syntax in an XML dialect
- DAML-ONT—another predecessor of DAML+OIL

XML Schema Datatypes

- The source of datatypes for DAML+OIL.

The RDF Vision of the Semantic Web

- All syntax is RDF syntax, i.e., n-triples or convertible to n-triples.
- All meaning is compatible with RDF meaning.



Compromises and Problems

DAML+OIL pushes the boundary of practical inference.

- Inference is difficult, because of expressive power requested by user communities.
- E.g., sets of individuals + invertible properties.
- E.g., transitivity and number restrictions.

DAML+OIL allows untyped literals, e.g., "10", which is either a string or an integer.

- Needed because RDF permits untyped literals.
- Makes for a more complex semantics.
- Interacts poorly with some XML Schema datatypes.

DAML+OIL separates objects, like John, from datatype values, like integers.

- To help with practical and decidable inference.
- But can't talk about some kinds of (poorly-typed) relationships
 - e.g., an author that is either a person or a string (the name of the author).

DAML+OIL has two different semantics, model-theoretic and axiomatic.

- To satisfy two communities.
- The two semantics are different in some cases.

DAML+OIL syntax is RDF n-triples.

- Mandated by the view of RDF as the language for the Semantic Web.
- Triples are not good syntax carriers for complex constructs.
- What to do with malformed knowledge bases?

```

.r1 rdf:type daml:Restriction .
.r1 daml:onProperty :friend .
.r2 daml:toClass :Student .
.r1 daml:hasValue :Susan .
  
```

- The structure of a knowledge base is not maintained in the triples,
 - e.g., definitions vs. information about a class—both turn into just triples.

The new semantics for RDF is not compatible with the DAML+OIL model theory.

- DAML+OIL model theory is specified from RDF triples, not from the RDF semantics.
 - DAML+OIL thus uses RDF syntax, but not RDF semantics.
 - Would like DAML+OIL semantics to be an extension of the RDF semantics.

Extending the new RDF semantics leads to paradoxes.

- To have desirable inferences, need to have a built-in theory of restrictions.
- Theory ends up with unusual, paradoxical restrictions, e.g.,

```

.r1 daml:onProperty rdf:type .
.r1 daml:maxCardinality "0" .
.r1 daml:toClassQ :2 .
.r2 daml:oneOf :3 .
.r3 daml:first :1 .
.r3 daml:rest daml:nil .
  
```

- Similar to Russell's paradox—the set of all sets that don't contain themselves.

- Ends up destroying the semantics.

- Possible Solutions:

1. Remove some DAML+OIL constructs, e.g., restrictions, from RDF resources, violating semantic extension vision.
2. Make DAML+OIL syntax an extension of RDF syntax, violating same-syntax vision.
3. Limit DAML+OIL inferences, losing, e.g., commutativity of conjunction.

Moving to OWL

OWL is the W3C web ontology language, and a successor for DAML+OIL.

- OWL thus should integrate well with other W3C recommendations.
- OWL thus should not change too much from DAML+OIL.

OWL has a larger design group and user community.

- Pressure for even more expressive power.
- More pressure for practical inference.
- Pressure for alternative different syntaxes, e.g., UML.

OWL should address DAML+OIL problems.

- Implementations to show practicality of inference.
- A better datatype solution, compatible with XML Schema and RDF datatyping.
- A way of describing the object / value split.
- A single, definitive semantics.
- A way of thinking about syntax that avoids the problems of triples.
- A way of building on the new RDF semantics.
- A solution to the paradoxes caused by the RDF vision.

A Simple DAML+OIL Example

In Description Logic syntax:

```
John ∈ Student ∩ ∃friend.Susan ∩ ∀friend.Student
```

In the XML syntax for RDF (RDF/XML):

```

<rdf:Description rdf:ID="John">
  <rdf:type>
    <daml:intersectionOf rdf:parsetype="Collection">
      <daml:Class rdf:about="#Student" />
      <daml:Restriction>
        <daml:onProperty rdf:resource="#friend" />
        <daml:hasValue rdf:resource="#Susan" />
      </daml:Restriction>
      <daml:Restriction>
        <daml:toClass rdf:resource="#Student" />
        <daml:onProperty rdf:resource="#friend" />
      </daml:Restriction>
    </daml:intersectionOf>
  </rdf:type>
</rdf:Description>
  
```

In RDF n-triples:

```

:John rdf:type :d .
.r13 daml:rest daml:nil .
.r1 daml:hasValue :Susan .
.r11 daml:First :Student .
.r13 daml:first :r2 .
.r12 daml:onProperty :friend .
.r1 rdf:type daml:Restriction .
.r12 daml:first :r1 .
.r2 rdf:type daml:Restriction .
.r12 daml:rest :r13 .
.r1 daml:onProperty :friend .
.r1 daml:intersectionOf :r11 .
.r11 daml:rest :r12 .
.r2 daml:toClass :Student .
  
```

DAML+OIL Class Constructors (in DL syntax)

Constructor	DL Syntax	Example
intersectionOf	$C_1 \sqcap \dots \sqcap C_n$	Human \sqcap Male
unionOf	$C_1 \sqcup \dots \sqcup C_n$	Doctor \sqcup Lawyer
complementOf	$\neg C$	\neg Male
oneOf	$\{x_1 \dots x_n\}$	{john, mary}
toClass	$\forall P.C$	\forall hasChild.Doctor
hasClass	$\exists P.C$	\exists hasChild.Lawyer
hasValue	$\exists P.\{x\}$	\exists citizenOf.{USA}
minCardinalityQ	$\geq n.P.C$	≥ 2 hasChild.Lawyer
maxCardinalityQ	$\leq n.P.C$	≤ 1 hasChild.Male
cardinalityQ	$= n.P.C$	$= 1$ hasParent.Female

DAML+OIL Axioms (in DL syntax)

Axiom	DL Syntax	Example
subClassOf	$C_1 \sqsubseteq C_2$	Human \sqsubseteq Animal \sqcap Biped
sameClassAs	$C_1 \equiv C_2$	Man \equiv Human \sqcap Male
subPropertyOf	$P_1 \sqsubseteq P_2$	hasDaughter \sqsubseteq hasChild
samePropertyAs	$P_1 \equiv P_2$	cost \equiv price
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$ Female
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	{President_Bush} \equiv {G_W_Bush}
differentIndividualFrom	$\{x_1\} \sqsubseteq \neg \{x_2\}$	{john} $\sqsubseteq \neg$ {peter}
inverseOf	$P_1 \equiv P_2^-$	hasChild \equiv hasParent
transitiveProperty	$P^+ \sqsubseteq P$	ancestor ⁺ \sqsubseteq ancestor
uniqueProperty	$\top \sqsubseteq \leq 1 P$	$\top \sqsubseteq \leq 1$ hasMother
unambiguousProperty	$\top \sqsubseteq \leq 1 P^-$	$\top \sqsubseteq \leq 1$ isMotherOf