The Implementation of the Semantic Web

Ian Horrocks

horrocks@cs.man.ac.uk

University of Manchester

Manchester, UK

Talk Outline

- The Semantic Web
- Web Ontology Languages
- DAML+OIL Language
- Reasoning with DAML+OIL
  - myGrid
  - OilEd Demo
- Research Challenges

Summary

- **Semantic Web** aims to make web resources accessible to automated processes
- **Ontologies** will play key role by providing vocabulary for semantic markup
- **DAML+OIL** is an ontology language designed for the web
  - Exploits existing standards: XML, RDF(S)
  - Adds KR idioms from object oriented and frame systems
  - Formal rigor of Description Logic
  - Facilitates provision of **reasoning support**
  - Set to become W3C standard (OWL) & already being widely adopted
- **Challenges** remain
  - Reasoning with full language
  - (Convincing) demonstration(s) of scalability
  - Development of (high quality) tools and infrastructure

The Semantic Web

The Semantic Web Vision

- Web made possible through established **standards**
  - **TCP/IP** for transporting bits down a wire
  - **HTTP & HTML** for transporting and rendering hyperlinked text
- **Applications** able to exploit this common infrastructure
  - Result is the WWW as we know it
- **1st generation** web mostly handwritten HTML pages
- **2nd generation** (current) web often machine generated/active
- Both intended for direct human processing/interaction
- In **next generation** web, **resources** should be more accessible to automated processes
  - To be achieved via **semantic markup**
  - **Metadata** annotations that describe content/function
- Coincides with Tim Berners-Lee’s vision of a **Semantic Web**

Ontologies

- Semantic markup must be **meaningful** to automated processes
- Ontologies will play a key role
  - Source of **precisely defined** terms (vocabulary)
  - Can be **shared** across applications (and humans)
- Ontology typically consists of:
  - **Hierarchical** description of important **concepts** in domain
  - Descriptions of the **properties** of each concept
- Degree of formality can be quite variable (NL–logic)
- Increased formality and regularity facilitates machine understanding
- Ontologies can be used, e.g.:
  - To facilitate buyer–seller communication in **e-commerce**
  - In semantic based **search**
  - To provide richer **service descriptions** that can be more flexibly interpreted by intelligent agents

Web Ontology Languages

Web Languages

- Web languages already extended to facilitate **content description**
  - XML Schema (XMLS)
  - RDF and RDF Schema (RDFS)
- RDFS recognisable as an **ontology language**
  - Classes and properties
  - Range and domain
  - Sub/super-classes (and properties)
- But RDFS not a suitable foundation for Semantic Web
  - **Too weak** to describe resources in sufficient detail
- Requirements for web ontology language:
  - **Compatible** with existing Web standards (XML, RDF, RDFS)
  - **Easy to understand** and use (based on common KR idioms)
  - **Formally specified** and of “adequate” expressive power
  - Possible to provide **automated reasoning** support

History: OIL and DAML-ONT

- Two languages developed to satisfy above requirements
  - **OIL**: developed by group of (largely) European researchers (several from OntoKnowledge project)
  - **DAML-ONT**: developed by group of (largely) US researchers (in DARPA DAML programme)
- Efforts merged to produce DAML+OIL
  - Development was overseen by **joint EU/US committee**
  - Now **submitted to W3C** as basis for standardisation
  - **WebOnt working group** developing language standard
  - New standard to be called **OWL** (Ontology Web Language)
  - OWL will be **very** similar to DAML+OIL

DAML+OIL

- DAML+OIL **layered** on top of RDFS
  - RDFS based **syntax**
  - **Inherits** RDFS ontological primitives (subclass, range, domain)
  - Adds **much** richer set of primitives (transitivity, cardinality, . . .)
- DAML+OIL designed to describe **structure** of domain (**schema**)
  - **Object oriented**: classes (concepts) and properties (roles)
  - DAML+OIL ontology consists of set of **axioms** asserting characteristics of classes and properties
  - E.g., Person is **kind of** Animal whose parents are Persons
- RDF used for class/property membership assertions (**data**)
  - E.g., John is an **instance of** Person; $\langle \text{John}, \text{Mary} \rangle$ is an instance of parent

DAML+OIL Language

Foundations

- DAML+OIL equivalent to very expressive **Description Logic**
  - But don't tell anyone!
- More precisely, DAML+OIL is (extension of) $\mathcal{SHIQ}$ DL
- DAML+OIL benefits from many years of DL research
  - Well defined **semantics**
  - **Formal properties** well understood (complexity, decidability)
  - Known **reasoning algorithms**
  - **Implemented systems** (highly optimised)
- DAML+OIL classes can be names (URI's) or **expressions**
  - Various **constructors** provided for building class expressions
- **Expressive power** determined by
  - Kinds of constructor provided
  - Kinds of axiom allowed

DAML+OIL Class Constructors

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

- XMLS **datatypes** as well as classes
- Arbitrarily complex **nesting** of constructors
  - E.g., Person $\sqcap$ $\forall$hasChild.(Doctor $\sqcup$ $\exists$hasChild.Doctor)

RDFS Syntax

```
<daml:Class>
  <daml:intersectionOf rdf:parseType="daml:collection">
    <daml:Class rdf:about="#Person"/>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasChild"/>
      <daml:toClass>
        <daml:unionOf rdf:parseType="daml:collection">
          <daml:Class rdf:about="#Doctor"/>
          <daml:Restriction>
            <daml:onProperty rdf:resource="#hasChild"/>
            <daml:hasClass rdf:resource="#Doctor"/>
          </daml:Restriction>
        </daml:unionOf>
      </daml:toClass>
    </daml:Restriction>
  </daml:intersectionOf>
</daml:Class>
```

DAML+OIL Axioms

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
sameIndividualAs	$\{x_1\} \equiv \{x_2\}$	$\{$President_Bush$\} \equiv \{$G_W_Bush$\}$
disjointWith	$C_1 \sqsubseteq \neg C_2$	Male $\sqsubseteq \neg$Female
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	$\top \sqsubseteq \leqslant 1 P$	$\top \sqsubseteq \leqslant$1hasMother
unambiguousProperty	$\top \sqsubseteq \leqslant 1 P^-$	$\top \sqsubseteq \leqslant$1isMotherOf$^-$

☞ Axioms (mostly) **reducible to subClass/PropertyOf**

XML Datatypes in DAML+OIL

- DAML+OIL supports the full range of **XML Schema** datatypes
  - Primitive (e.g., decimal) and derived (e.g., integer sub-range)
- Clean **separation** between “object” classes and datatypes
  - Disjoint interpretation domains: $\text{John}^{\mathcal{I}} \neq (\text{int } 5)^{\mathcal{I}}$
  - Object properties disjoint from datatype properties
- Philosophical reasons:
  - Datatypes structured by **built-in predicates**
  - Not appropriate to form new datatypes using ontology language
- Practical reasons:
  - Ontology language remains **simple and compact**
  - **Semantic integrity** of ontology language not compromised
  - **Implementability** not compromised—can use hybrid reasoner
- In practice, DAML+OIL implementations can choose to support **subset** of XML Schema datatypes.

Reasoning with DAML+OIL

Reasoning

- ☞ Why do we want it?
  - Semantic Web aims at “machine understanding”
  - **Understanding** closely related to **reasoning**
- ☞ What can we do with it?
  - **Design and maintenance** of ontologies
    - Check class consistency and compute class hierarchy
    - Particularly important with large ontologies/multiple authors
  - **Integration** of ontologies
    - Assert inter-ontology relationships
    - Reasoner computes integrated class hierarchy/consistency
  - **Querying** class and instance data w.r.t. ontologies
    - Determine if set of facts are consistent w.r.t. ontologies
    - Determine if individuals are instances of ontology classes
    - Retrieve individuals/tuples satisfying a query expression
    - Check if one class subsumes (is more general than) another w.r.t. ontology
    - . . .

Why Decidable Reasoning?

- ☞ DAML+OIL constructors/axioms restricted so reasoning is **decidable**
- ☞ Consistent with Semantic Web's **layered architecture**
  - XML provides syntax **transport layer**
  - RDF(S) provides basic **relational language** and simple ontological primitives
  - DAML+OIL provides powerful but still decidable **ontology language**
  - Further layers (e.g., **rules**) will extend DAML+OIL
  - **Extensions** will almost certainly be undecidable
- ☞ Facilitates provision of **reasoning services**
  - Known “practical” **algorithms**
  - Several implemented **systems**
  - Evidence of **empirical tractability**
- ☞ Understanding dependent on **reliable & consistent** reasoning

Basic Inference Problems

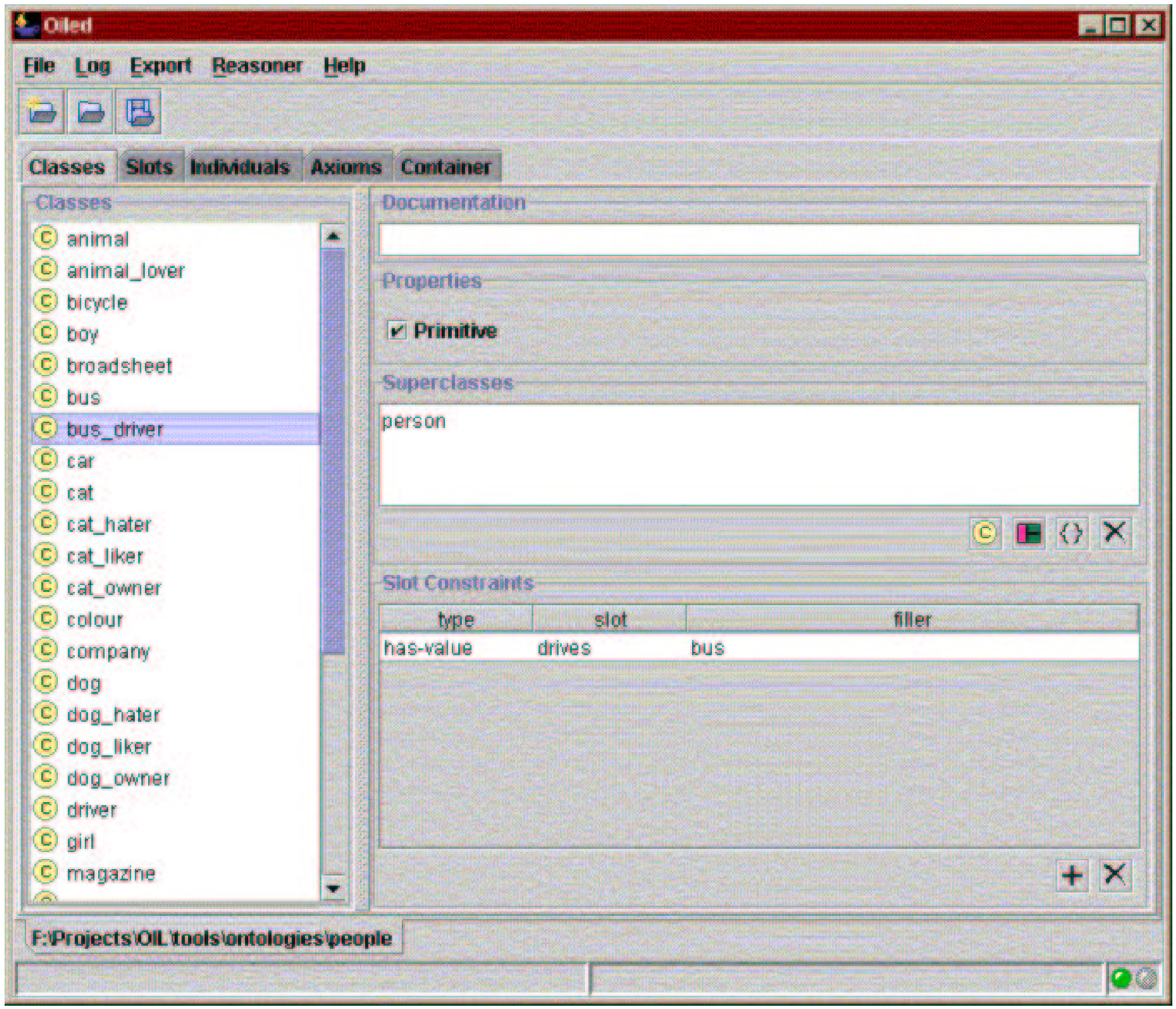
- **Consistency** — check if knowledge is meaningful
  - Is $\mathcal{O}$ consistent? There exists some model $\mathcal{I}$ of $\mathcal{O}$
  - Is C consistent? $C^{\mathcal{I}} \neq \emptyset$ in some model $\mathcal{I}$ of $\mathcal{O}$
- **Subsumption** — structure knowledge, compute taxonomy
  - $C \sqsubseteq_{\mathcal{O}} D$? $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$
- **Equivalence** — check if two classes denote same set of instances
  - $C \equiv_{\mathcal{O}} D$? $C^{\mathcal{I}} = D^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$
- **Instantiation** — check if individual i instance of class C
  - $i \in_{\mathcal{O}} C$? $i \in C^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$
- **Retrieval** — retrieve set of individuals that instantiate C
  - set of i s.t. $i \in C^{\mathcal{I}}$ in all models $\mathcal{I}$ of $\mathcal{O}$
- Problems all **reducible** to consistency (satisfiability):
  - $C \sqsubseteq_{\mathcal{O}} D$ iff $C \sqcap \neg D$ not consistent w.r.t. $\mathcal{O}$
  - $i \in_{\mathcal{O}} C$ iff $\mathcal{O} \cup \{i \in \neg C\}$ is **not** consistent

Reasoning Support for Grid Services: myGrid

myGrid uses DAML+OIL to **describe** and **discover** services, e.g.:

- Find a service that takes a protein and gives its function(s)
  - Both service and requirements descriptions use terms from **service ontology** (based on **DAML-S** ontology)
  - Reasoning (subsumption) used to match services with requirements
- Find another service that displays proteins based on their functions
  - Descriptions/ontology restrict types of inputs and outputs so services can be linked
- Services then linked and enacted to perform required function
  - Generate function based display of given protein
- Reasoning also used in **design and maintenance** of service ontology

Reasoning Support for Ontology Design: OilEd



Description Logic Reasoning

Highly Optimised Implementation

- Naive implementation $\longrightarrow$ effective non-termination
- Modern systems include **MANY** optimisations
- Optimised **classification** (compute partial ordering)
  - Use enhanced traversal (exploit information from previous tests)
  - Use structural information to select classification order
- Optimised **subsumption** testing (search for models)
  - Normalisation and simplification of concepts
  - Absorption (simplification) of general axioms
  - Davis-Putnam style semantic branching search
  - Dependency directed backtracking
  - Caching of satisfiability results and (partial) models
  - Heuristic ordering of propositional and modal expansion
  - . . .

Research and Implementation Challenges

Challenges

- **Increased expressive power**
  - Existing DL systems implement (at most) $\mathcal{SHIQ}$
  - DAML+OIL extends $\mathcal{SHIQ}$ with datatypes and nominals
- **Scalability**
  - Very large KBs
  - Reasoning with (very large numbers of) individuals
- **Other reasoning tasks**
  - Querying
  - Matching
  - Least common subsumer
  - ...
- **Tools and Infrastructure**
  - Support for large scale ontological engineering and deployment

Summary

- **Semantic Web** aims to make web resources accessible to automated processes
- **Ontologies** will play key role by providing vocabulary for semantic markup
- **DAML+OIL** is an ontology language designed for the web
  - Exploits existing standards: XML, RDF(S)
  - Adds KR idioms from object oriented and frame systems
  - Formal rigor of Description Logic
  - Facilitates provision of **reasoning support**
  - Set to become W3C standard (OWL) & already being widely adopted
- **Challenges** remain
  - Reasoning with full language
  - (Convincing) demonstration(s) of scalability
  - Development of (high quality) tools and infrastructure

Acknowledgements

- Members of the OIL and DAML+OIL development teams, in particular Dieter Fensel and Frank van Harmelen (Amsterdam) and Peter Patel-Schneider (Bell Labs)
- Franz Baader, Uli Satler and Stefan Tobies (Dresden — formerly Aachen)
- Members of the Information Management, Medical Informatics, Formal Methods and Artificial Intelligence Groups at the University of Manchester

Resources

Slides from this talk

http://www.cs.man.ac.uk/~horrocks/Slides/planet-prn.p

FaCT system (open source)

http://www.cs.man.ac.uk/FaCT/

OilEd (open source)

http://oiled.man.ac.uk/

OIL

http://www.ontoknowledge.org/oil/

DAML+OIL

http://www.w3c.org/Submission/2001/12/

W3C Web-Ontology (WebOnt) working group (OWL)

http://www.w3.org/2001/sw/WebOnt/

Select Bibliography

I. Horrocks. DAML+OIL: a reason-able web ontology language. In *Proc. of EDBT 2002*, number 2287 in Lecture Notes in Computer Science, pages 2–13. Springer-Verlag, Mar. 2002.

I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. Reviewing the design of DAML+OIL: An ontology language for the semantic web. In *Proc. of AAAI 2002*, 2002. To appear.

I. Horrocks and S. Tessaris. Querying the semantic web: a formal approach. In I. Horrocks and J. Hendler, editors, *Proc. of the 2002 International Semantic Web Conference (ISWC 2002)*, number 2342 in Lecture Notes in Computer Science. Springer-Verlag, 2002.

C. Lutz. *The Complexity of Reasoning with Concrete Domains*. PhD thesis, Teaching and Research Area for Theoretical Computer Science, RWTH Aachen, 2001.

Select Bibliography

I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(\mathbf{D})$ description logic. In B. Nebel, editor, *Proc. of IJCAI-01*, pages 199–204. Morgan Kaufmann, 2001.

F. Baader, S. Brandt, and R. Küsters. Matching under side conditions in description logics. In B. Nebel, editor, *Proc. of IJCAI-01*, pages 213–218, Seattle, Washington, 2001. Morgan Kaufmann.

A. Borgida, E. Franconi, and I. Horrocks. Explaining $\mathcal{ALC}$ subsumption. In *Proc. of ECAI 2000*, pages 209–213. IOS Press, 2000.

D. Calvanese, G. De Giacomo, M. Lenzerini, D. Nardi, and R. Rosati. A principled approach to data integration and reconciliation in data warehousing. In *Proceedings of the International Workshop on Design and Management of Data Warehouses (DWDM'99)*, 1999.