OWL: the Web Ontology Language

Alun Preece
http://www.csd.abdn.ac.uk/~apreece/foaf.rdf

OWL: what?

- Core of the World Wide Web Consortium’s Semantic Web activity
- In various senses a successor to previous work on “Web-friendly” knowledge modelling languages
  - RDF & RDF Schema
  - DAML-ONT
  - OIL / DAML+OIL
- W3C’s Web Ontology Working Group are a “who’s who” of the knowledge representation field
- Last Call Working Drafts issued in late March - closed on May 9 2003; final recommendation will then follow
**OWL: why?**

- **Semantic Web apps:**
  - portal Websites & intranets (information architecture)
  - multimedia digital libraries (rich metadata)
  - agents & Web services (interoperability, automation)
  - design documentation (complex, interlinked)

- **Capabilities:**
  - ontology sharing, evolution, interoperability
  - inconsistency detection
  - expressivity vs scalability
  - standards compliance

---

**Semantic Web architecture**

- Data
- Rules
- Logic
- Ontology vocabulary
- RDF + rdfscheme
- XML + NS + xsmschema
- Unicode
- URI
- Self-desc. doc.
- Digital Signature
- Trust
- Proof
XML, RDF & OWL

- **XML**: universal syntax
- **XML Schema**: defines structure of XML docs
- **RDF**: datamodel for resource objects
- **RDF Schema**: basic vocabulary for defining RDF classes & properties, and hierarchies of each
- **OWL**: extended vocab for defining classes & properties, including
  - cardinality (e.g. minCardinality 1)
  - equality (e.g. equivalentClass)
  - relationships between classes (e.g. disjointWith)
  - characteristics of properties (e.g. FunctionalProperty)

OWL sublanguages ("species")

- **OWL Lite**
  - "RDF-and-a-half"
  - Mainly intended for class hierarchies & simple constraints (cardinality 0 or 1, equality, ...)
- **OWL DL**
  - Description Logic theoretical properties
  - Intended where completeness & decidability are an issue
- **OWL Full**
  - Max expressivity; no computational guarantees
  - Supports "Web-scale" & "Web-style" KR&R
OWL sublanguages cont’d

- Every legal OWL Lite ontology is a legal OWL DL ontology
- Every legal OWL DL ontology is a legal OWL Full ontology
- Every valid OWL Lite conclusion is a valid OWL DL conclusion
- Every valid OWL DL conclusion is a valid OWL Full conclusion

The converse in each case does not hold

OWL Lite: essentials

<table>
<thead>
<tr>
<th>Schema constructs</th>
<th>Equality constructs</th>
<th>Headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class (i.e. owl:Class)</td>
<td>equivalentClass</td>
<td>imports</td>
</tr>
<tr>
<td>rdf:Property</td>
<td>equivalentProperty</td>
<td>priorVersion</td>
</tr>
<tr>
<td>rdfs:subClassOf</td>
<td>sameIndividualAs</td>
<td>backwardCompatibleWith</td>
</tr>
<tr>
<td>rdfs:subPropertyOf</td>
<td>differentFrom</td>
<td>incompatibleWith</td>
</tr>
<tr>
<td>rdfs:domain</td>
<td>allDifferent</td>
<td></td>
</tr>
<tr>
<td>rdfs:range</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Property characteristics</td>
<td></td>
<td></td>
</tr>
<tr>
<td>inverseOf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TransitiveProperty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FunctionalProperty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>InverseFunctionalProperty</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SymmetricProperty</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cardinality</td>
<td></td>
</tr>
<tr>
<td></td>
<td>minCardinality (0 or 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>maxCardinality (0 or 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cardinality (0 or 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Class Intersection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>intersectionOf</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RDF datatyping</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Class axioms
- oneOf
- disjointWith

### Class expressions
- equivalentClass
- rdfs:subClassOf
- unionOf
- intersectionOf
- complementOf

### Property fillers
- hasValue

### When is a Class not a Class?
- **Answer:** in OWL Lite & OWL DL, when it’s an Individual - DL restrictions (apparently) do not permit Classes to be treated as Individuals
- So, no “Class, an Individual class, being the Class of all Classes” (as in RDF)
- So, rdfs:Class cannot be used in OWL Lite or OWL DL
- owl:Class is defined as rdfs:subClassOf rdfs:Class
- (But, in OWL Full, they coincide!)
- Note that this means an RDF-processing agent can still use a lot of OWL, because it understands the triple: owl:Class rdfs:subClassOf rdfs:Class
Defining an owl:Class (I)

- **By class identifier:**
  <owl:Class rdf:ID="Lecturer">
    <rdfs:subClassOf rdf:resource="#Person" />
  </owl:Class>

- **By enumeration:**
  <owl:Class rdf:ID="ComputingOfficer">
    <owl:oneOf rdf:parseType="Collection">
      <Academic rdf:about="#nmurray" />
      <Academic rdf:about="#jmartin" />
      <Academic rdf:about="#mritchie" />
    </owl:oneOf>
  </owl:Class>

Defining an owl:Class (II)

- **By property restriction:**
  <owl:Class rdf:ID="Researcher">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#activity" />
        <owl:someValuesFrom rdf:resource="#ResearchArea" />
      </owl:Restriction>
    </rdfs:subClassOf>
  </owl:Class>

- **By intersection/union/complement:**
  <owl:Class rdf:ID="UniversityStaff">
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Lecturer" />
      <owl:Class rdf:about="#Researcher" />
      <owl:Class rdf:about="#ComputingOfficer" />
    </owl:unionOf>
  </owl:Class>
Properties in OWL

- Two types
  - ObjectProperty - relations between instances of classes
  - DatatypeProperty - relates an instance to an rdfs:Literal or XML Schema datatype
    (Both rdfs:subClassOf rdf:Property)

```
<owl:DatatypeProperty rdf:ID="name">
  <rdfs:domain rdf:resource="Person" />
  <rdfs:range rdf:resource=
    "http://www.w3.org/2001/XMLSchema/string" />
</owl:DatatypeProperty>

<owl:ObjectProperty rdf:ID="activity">
  <rdfs:domain rdf:resource="Person" />
  <rdfs:range rdf:resource="ActivityArea" />
</owl:ObjectProperty>
```

Individual axioms (“facts”)

- OWL is not only a language for defining ontologies - it is used to define their instances (Individuals)
- Example:
  ```
  <Lecturer rdf:ID="apreece">
    <name>Alun Preece</name>
    <activity rdf:resource="#AgentsResearch" />
    <activity rdf:resource="#WebTeaching" />
  </Lecturer>

  <ResearchArea rdf:ID="AgentsResearch" />
  <TeachingArea rdf:ID="WebTeaching"/>
  ```
  
  (Notice how individual apreece follows the definition of Lecturer given earlier)
An example:
http://www.csd.abdn.ac.uk/~apreece

RDF about Alun Preece

Email: apreece@csd.abdn.ac.uk
Phone: +44 1224 272291
Fax: +44 1224 273422

Friend-of-a-Friend (FOAF) resource data:
http://www.csd.abdn.ac.uk/~apreece/foaf.rdf

vCard resource data:
http://www.csd.abdn.ac.uk/~apreece/apreece.rdf
Visualising FOAF data

http://jibbering.com/foaf/foafnaut.svg

World Wide FOAF
The FOAF ontology

- FOAF is defined using RDF(S) and OWL
  http://xmlns.com/foaf/0.1/
- OWL’s InverseFunctionalProperty is used to state that particular properties unambiguously identify unique people:
  - mbox
  - homepage
  - weblog
  - dnaChecksum (joke)
- So, in the FOAF model, non-personal email addresses (say, info@conoise.org) can’t be used to ID a person

OWL for ontology alignment

- There are overlaps between the ontologies for
  - FOAF - http://xmlns.com/foaf/0.1/
  - vCard - http://www.w3.org/2001/vcard-rdf/3.0
- OWL can articulate equivalences, for example:

```xml
<rdf:Property
   rdf:about="http://www.w3.org/2001/vcard-rdf/3.0#EMAIL">
   <owl:equivalentProperty
     rdf:resource="http://xmlns.com/foaf/0.1/mbox"/>
</rdf:Property>
```

- An OWL reasoner could use this equivalence to derive a value for some resource’s vcard:EMAIL if it can find a value for foaf:mbox
OWL: implications

- OWL is potentially the most important knowledge representation language we’ve yet seen
  (Hendler claims DAML already is, in terms of numbers of statements asserted)
- It could be the “last word” in KR similar to how HTML came to dominate the field of hypertext markup

Implications:
- If you’re doing KR research, you will need to situate yourself in relation to OWL
- If you’re building KBS, OWL will be your first choice of KRL
- There are enormous challenges ahead in creating effective OWL reasoners/processors