Semantic Web - OWL

CS 431 – April 7, 2008
Carl Lagoze – Cornell University
Acknowledgements for various slides and ideas

• Ian Horrocks (Manchester U.K.)
• Eric Miller (W3C)
• Dieter Fensel (Berlin)
• Volker Haarslev (Montreal)
OWL Lite Summary

RDF Schema Features:
- Class (Thing, Nothing)
- rdfs:subClassOf
- rdf:Property
- rdfs:subPropertyOf
- rdfs:domain
- rdfs:range
- Individual

(In)Equality:
- equivalentClass
- equivalentProperty
- sameAs
- differentFrom
- AllDifferent
- distinctMembers

Property Characteristics:
- ObjectProperty
- DatatypeProperty
- inverseOf
- TransitiveProperty
- SymmetricProperty
- FunctionalProperty
- inverseFunctionalProperty

Property Restrictions:
- Restriction
- onProperty
- allValuesFrom
- someValuesFrom

Restricted Cardinality:
- minCardinality (only 0 or 1)
- maxCardinality (only 0 or 1)
- cardinality (only 0 or 1)

Header Information:
- Ontology
- imports

Class Intersection:
- intersectionOf

Datatypes
- xsd datatypes

Versioning:
- versionInfo
- priorVersion
- backwardCompatibleWith
- incompatibleWith
- DeprecatedClass
- DeprecatedProperty

Annotation Properties:
- rdfs:label
- rdfs:comment
- rdfs:seeAlso
- rdfs:isDefinedBy
- AnnotationProperty
- OntologyProperty
OWL DL and Full Summary

Class Axioms:
- `oneOf`
- `dataRange`
- `disjointWith`
- `equivalentClass` (applied to class expressions)
- `rdfs:subClassOf` (applied to class expressions)

Boolean Combinations of Class Expressions:
- `unionOf`
- `complementOf`
- `intersectionOf`

Arbitrary Cardinality:
- `minCardinality`
- `maxCardinality`
- `cardinality`

Filler Information:
- `hasValue`
OWL DL vs. OWL-Full

• Same vocabulary
• OWL DL restrictions
  – Type separation
    • Class can not also be an individual or property
    • Property can not also be an individual or class
  – Separation of ObjectProperties and DatatypeProperties
Class/Property Example

```xml
<?xml version="1.0"?>
<rdf:RDF xmlns="http://www.co-ode.org/ontologies/wine/2005/10/18/wine.owl#"
xml:base="http://www.co-ode.org/ontologies/wine/2005/10/18/wine.owl"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:owl="http://www.w3.org/2002/07/owl#">
<owl:Class rdf:ID="PotableLiquid"/>
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="#PotableLiquid"/>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>
</owl:Class>
<owl:DatatypeProperty rdf:ID="color">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema/string"/>
</owl:DatatypeProperty>
<owl:Class rdf:ID="Appellation"/>
<owl:ObjectProperty rdf:ID="hasAppellation">
  <rdfs:domain rdf:resource="#Wine"/>
  <rdfs:range rdf:resource="#Appellation"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="producesWine">
  <rdfs:range rdf:resource="#Wine"/>
  <rdfs:domain rdf:resource="#Appellation"/>
  <owl:inverseOf rdf:resource="#hasAppellation"/>
</owl:ObjectProperty>
</rdf:RDF>
```
Demonstrating RDF, OWL, and Inferences

## Language Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>DTD</th>
<th>XSD</th>
<th>RDF(S)</th>
<th>OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded lists (“X is known to have exactly 5 children”)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Cardinality constraints (Kleene operators)</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Class expressions (unionOf, complementOf)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Data types</td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Enumerations</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Equivalence (properties, classes, instances)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Formal semantics (model-theoretic &amp; axiomatic)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Inheritance</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Inference (transitivity, inverse)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Qualified constraints (“all children are of type person”)</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Reification</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Storing and querying RDF-based models

- **Persistent storage implementations**
    - Relational databases (mysql, postgres, oracle)
    - Mapped files
    - Relational databases (mysql, postgres, oracle)

- **Query languages**
  - RDQL (Mulgara)
  - SPARQL (Jena)
    - W3C recommendation
    - [http://www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/)
RDQL-by-example

- RDF source
  - vc-db-3.rdf
- Queries
  - vc-q1
  - vc-q2
  - vc-q3
  - vc-q4
  - vc-q5
  - vc-q6
  - vc-q7
  - vc-q8
What is an **Ontology**?

- A formal specification of conceptualization shared in a community
- Vocabulary for defining a set of things that exist in a world view
- Formalization allows communication across application systems and extension
- Parallel concepts in other areas:
  - *Domains*: database theory
  - *Types*: AI
  - *Classes*: OO systems
  - *Types/Sorts*: Logic
- Global vs. Domain-specific
XML and RDF are ontologically neutral

- No standard vocabulary just primitives
  - Resource, Class, Property, Statement, etc.
- Compare to classic first order logic
  - Conjunction, disjunction, implication, existential, universal quantifier
Components of an Ontology

• Vocabulary (concepts)
• Structure (attributes of concepts and hierarchy)
• Relationships between concepts
• Logical characteristics of relationships
  – Domain and range restrictions
  – Properties of relations (symmetry, transitivity)
  – Cardinality of relations
  – etc.
Wordnet

• On-line lexical reference system, domain-independent
• >100,000 word meanings organized in a taxonomy with semantic relationships
  – Synonymy, meronymy, hyponymy, hypernymy
• Useful for text retrieval, etc.
• http://www.cogsci.princeton.edu/~wn/online/
• Effort in AI community to accommodate all of human knowledge!!!
• Formalizes concepts with logical axioms specifying constraints on objects and classes
• Associated reasoning tools
• Contents are proprietary but there is OpenCyc
  – http://www.opencyc.org/
So why re-invent ontologies for the Web

- Not re-invention
  - Same underlying formalisms (frames, slots, description logic)
- But new factors
  - Massive scale
    - Tractability
    - Knowledge expressiveness must be limited or reasoning must be incomplete
  - Lack of central control
    - Need for federation
    - Inconsistency, lies, re-interpretations, duplications
    - New facts appear and modify constantly
  - Open world vs. Close world assumptions
    - Contrast to most reasoning systems that assume anything absent from knowledge base is not true
    - Need to maintain monotonicity with tolerance for contradictions
  - Need to build on existing standards
    - URI, XML, RDF
Components of the Semantic Web
Protégé and Pellet– tools for building, manipulating and reasoning over ontologies

• Protégé - [http://protege.stanford.edu/](http://protege.stanford.edu/)
  – Use the 3.x version
  – Multiple plug-ins are available, you should install with all plugins

• Protégé OWL plug-in

• Other semantic web related plug-ins

• Pellet
  – Open Source Description Logic based reasoning engine
  – Server-based
  – Integrates with Protégé-OWL