Semantic Web Ontologies (continued)
Expressing, Querying, Building

CS 431 – April 6, 2005

Carl Lagoze – Cornell University

Acknowledgements for various slides and ideas

- Ian Horrocks (Manchester U.K.)
- Eric Miller (W3C)
- Dieter Fensel (Berlin)
- Volker Haarslev (Montreal)
- Alun Preece (Aberdeen)

Putting the building blocks together

- XML
  - Syntax for document markup
- URIs
  - Universal naming syntax
- Namespaces
  - Concept space naming
- RDF
  - Sentence construction
- RDFs
  - Primitive Vocabulary building
- Description Logic
  - Formal basis for ontology models
- OWL
  - Expression of logic via RDF sentences

Web Ontology Language (OWL)

- W3C Web Ontology Working Group (WebOnt)
- Follow on to DAML, OIL efforts
- W3C Recommendation
- Vocabulary extension of RDF

Species of OWL

- *OWL Lite*
  - Good for classification hierarchies with simple constraints (e.g., thesauri)
  - Reasoning is computational simple and efficient
- *OWL DL*
  - Computationally complete and decidable (computation in finite time)
  - Correspondence to *description logics* (decidable fragment of first-order logic)
- *OWL Full*
  - Maximum expressiveness
  - No computational guarantees (probably never will be)

- Each language is extension of simpler predecessor

Namespaces and OWL

```
<rdf:RDF
    xmlns     ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
    xmlns:vin ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
    xml:base  ="http://www.w3.org/TR/2004/REC-owl-guide-20040210/wine#"
    xmlns:food="http://www.w3.org/TR/2004/REC-owl-guide-20040210/food#"
    xmlns:owl ="http://www.w3.org/2002/07/owl#"
    xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:xsd ="http://www.w3.org/2001/XMLSchema#">
```

OWL Class Definition

```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>
```

```
<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>
  ...
</owl:Class>
```

Why owl:class vs. rdfs:class

- Rdfs:class is "class of all classes"
- In DL class can not be treated as individuals (undecidable)
- Thus owl:class, which is expressed as rdfs:subclass of rdfs:class
  - No problem for standard rdf processors since an owl:class "is a" rdfs:class

- Note: there are other times you want to treat class of individuals
  - Class drinkable liquids has instances wine, beer, ….
  - Class wine has instances merlot, chardonnay, zinfandel, …

OWL class building operations

- disjointWith
  - No vegetarians are carnivores
- sameClassAs (equivalence)
- Enumerations (on instances)
  - The Ivy League is Cornell, Harvard, Yale, ….
- Boolean set semantics (on classes)
  - Union (logical disjunction)
    - Class *parent* is union of *mother, father*
  - Intersection (logical conjunction of class with properties)
    - Class *WhiteWine* is conjunction of things of class *wine* and have property *white*
  - complimentOf (logical negation)
    - Class *vegetarian* is disjunct of class *carnivore*

OWL Properties

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>
```

OWL property building operations & restrictions

- Transitive Property
  - P(x,y) and P(y,z) -> P(x,z)
- SymmetricProperty
  - P(x,y) iff P(y,x)
- Functional Property
  - P(x,y) and P(x,z) -> y=z
- inverseOf
  - P1(x,y) iff P2(y,x)
- InverseFunctional Property
  - P(y,x) and P(z,x) -> y=z
- Cardinality
  - Only 0 or 1 in lite and full

OWL DataTypes

- Full use of XML schema data type definitions
- Examples
  - Define a type age that must be a non-negative integer
  - Define a type clothing size that is an enumeration “small” “medium” “large”

OWL Instance Creation

- Create individual objects filling in slot/attribute/property definitions

```
<Person ref:ID="William Arms">
    <rdfs:label>Bill</rdfs:label>
    <age><xsd:integer rdf:value="57"/></age>
    <shoesize><xsd:decimal rdf:value="10.5"/></shoesize>
</Person>
```

OWL Lite Summary

Schema constructs

Class (i.e. owl:Class)
rdf:Property
rdfs:subClassOf
rdfs:subPropertyOf
rdfs:domain
rdfs:range
Individual

Property characteristics

inverseOf
TransitiveProperty
FunctionalProperty
InverseFunctionalProperty
SymmetricProperty

Equality constructs

equivalentClass
equivalentProperty
sameIndividualAs
differentFrom
allDifferent

Cardinality

minCardinality (0 or 1)
maxCardinality (0 or 1)
Cardinality (0 or 1)

Class intersection

intersectionOf

Headers

imports
priorVersion
backwardCompatibleWith
incompatibleWith

Property type restrictions

allValuesFrom
someValuesFrom

RDF datatyping

OWL DL and Full Summary

Class axioms	Class expressions
oneOf	equivalentClass
disjointWith	rdfs:subClassOf
	unionOf
	intersectionOf
	complementOf
Property fillers	**Arbirtary cardinality**
hasValue	minCardinality
	maxCardinality
	Cardinality

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

Language Comparison

	DTD	XSD	RDF(S)	OWL
Bounded lists ("X is known to have exactly 5 children")				X
Cardinality constraints (Kleene operators)	X	X		X
Class expressions (unionOf, complementOf)				X
Data types		X		X
Enumerations	X	X		X
Equivalence (properties, classes, instances)				X
Formal semantics (model-theoretic & axiomatic)				X
Inheritance			X	X
Inference (transitivity, inverse)				X
Qualified contraints ("all children are of type person"				X
Reification			X	X

Protégé and RACER – tools for building, manipulating and reasoning over ontologies

- Protégé - http://protege.stanford.edu/
  - Use the 3.x version
  - Multiple plug-ins are available
- Protégé OWL plug-in
  - http://protege.stanford.edu/plugins/owl/
- Other semantic web related plug-ins
  - http://protege.cim3.net/cgi-bin/wiki.pl?ProtegePluginsLibraryByTopic#nid349
- Racer
  - Description Logic based reasoning engine
  - Server-based
  - Integrates with Protégé-OWL
- Pizza Ontology
  - http://www.cs.cornell.edu/courses/cs431/2005sp/examples/OWL/pizza_20041007.owl

Protégé and OWL Concepts

- Classes and sub-classes
  - Disjoint classes (remember the open-world assumption)
  - Multiple inheritance
- Properties
  - Functional, inverse functional, transitive, symmetric
  - Domains and ranges
- Property restrictions
  - Quantifier
    - Existential
    - Universal
      - Closure axioms
        - » Remember the open world assumption
    - Cardinality
  - Covering axioms
    - Remember the open-world assumption

Storing and querying RDF-based models

- Persistent storage implementations
  - Jena 2 - http://www.hpl.hp.com/semweb/jena2.htm
    - Relational databases (mysql , postgres, oracle)
  - Kowari – http://www.kowari.org
    - Mapped files
  - Sesame - http://www.openrdf.org/
    - Relational databases (mysql, postgres, oracle)
- Query languages
  - RDQL (Kowari, Jena)
  - SPARQL
    - W3C working draft
    - http://www.w3.org/TR/rdf-sparql-query/

RDQL-by-example

- RDF source
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-db-3.rdf
- Queries
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q1
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q2
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q3
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q4
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q5
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q6
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q7
  - http://webpub.cs.cornell.edu/courses/cs431/2005sp/examples/RDQL/vc-q8