Components of the Semantic Web

Problems with RDF/RDFs

Weaknesses in expressivity

- No localized domain and range constraints
  - Can’t say the range of hasChild is person in context of persons and elephants in context of elephants
- No existence/cardinality constraints
  - Can’t say that all instances of persons have a mother that is also a person
  - Can’t say that persons have exactly two biological parents
- No transitive, inverse or symmetric properties
  - Can’t say isPartOf is a transitive property
  - Can’t say isPartOf is inverse of hasPart
  - Can’t say touches is symmetric
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

**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.

**CYC**

- 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/](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
  - 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
  - New facts appear and modify constantly
    - Open world vs. Close world assumptions
  - Need to build on existing standards
    - URI, XML, RDF
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., thesaurs)
  - 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

Relationship between OWL and RDF(s)

- OWL Full is extension of RDF
- OWL Lite and DL extensions of restricted view of RDF
- Every OWL document is an RDF document
- Every RDF document is an OWL Full document
- Only some RDF documents are OWL Lite or OWL DC
- Constraining an RDF document to be OWL Lite or DL
  - Every individual must have class membership (at least owl:thing)
  - URIs for classes, properties, and individuals must be mutually disjoint.

Description Logics

- Fragment of first-order logic designed for logical representation of object-oriented formalisms
- frames/classes/concepts
- sets of objects
- roles/properties
- binary relations on objects
- individuals
- Representation as a collection of statements, with unary and binary predicates that stand for concepts and roles, from which deductions can be made
- High expressivity with decidability and completeness
  - Decidable fragment of FOL

Description Logics Primitives

- Atomic Concept
  - Human
- Atomic Role
  - likes
- Conjunction
  - human intersection male
- Disjunction
  - nice union rich
- Negation
  - not rich
- Existential Restriction
  - exists has-child.Human
- Value Restriction
  - for-all has-child.Blond
- Number Restriction
  - ≥ 2 has-wheels
- Inverse Role
  - has-child, has-parent
- Transitive role
  - has-child

Description Logic - Tboxes

- Terminological knowledge
- Concept Definitions
  - Father is conjunction of Man and has-child.Human
- Axioms
  - motorcycle subset-of vehicle
  - has-favorite.Brewery subrelation-of drinks.Beer
Description Logics: Aboxes

- Assertional knowledge
- Concept assertions
  - John is a Man
- Role assertions
  - has-child(John, Bill)

Description Logics: Basic Inferencing

- Subsumption
  - Is C1 subclass-of C2
  - Compute taxonomy
- Consistency
  - Can C have any individuals

Namespaces and OWL

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

<owl:Class rdf:ID="Wine"/>
<owl:equivalentClass rdf:resource="#Food/FoodableLiquid"/>
<rdf:label xml:lang="en">Wine</rdf:label>
<rdf:label xml:lang="fr">Vino</rdf:label>
```

Why owl:Class vs. rdfs:Class

- Rdfs:Class is "class of all classes"
- In DL class cannot be treated as individuals (undecidable)
- Thus owl:Class, which is expressed as rdfs:subClassOf 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 Definition

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

<owl:Class rdf:ID="Wine"/>
<owl:equivalentClass rdf:resource="#Food/FoodableLiquid"/>
<rdf:label xml:lang="en">Wine</rdf:label>
<rdf:label xml:lang="fr">Vino</rdf:label>
```

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

- Transitive Property
  - \( P(x,y) \) and \( P(y,z) \) \( \rightarrow P(x,z) \)
- Symmetric Property
  - \( P(x,y) \) if \( P(y,x) \)
- Functional Property
  - \( P(x,y) \) and \( P(x,z) \) \( \rightarrow y = z \)
- inverseOf
  - \( P(x,y) \) if \( P(y,x) \)
- Inverse Functional Property
  - \( P(x,y) \) and \( P(z,x) \) \( \rightarrow y = z \)
- 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

```xml
<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

- Class/Property
- Ontology
- Annotation
- Restrictions
- Properties
- Cardinality
- Inverse Information

OWL DL and Full Summary

- Class Actions:
  - subClassOf
  - subPropertyOf
  - equivalentClass
  - owl:equivalence
- Boolean Combinations of Class Expressions:
  - and
  - or
  - xor
  - inclusiveOr
- Arbitrary 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

Language Comparison

<table>
<thead>
<tr>
<th></th>
<th>XML</th>
<th>XSD</th>
<th>RDF(S)</th>
<th>OWL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounded lists</td>
<td>X</td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Cardinality constraints (Kleene operations)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Class expressions (intersection, complement)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data types</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enumerations</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Equivalence (properties, classes, instances)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Formal semantics (model-theoretic &amp; axiomatic)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inheritance</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inference (transitivity, inverse)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Qualified constraints (“all children are of type person”)</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Realization</td>
<td>X</td>
<td>X</td>
<td></td>
<td></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 (Kowari, Jena)
  - SPARQL
  - W3C working draft
    - [http://www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/)

Protégé and RACER – 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
- Protégé OWL plug-in
  - [http://protege.stanford.edu/plugins/owl/](http://protege.stanford.edu/plugins/owl/)
- Other semantic web related plug-ins
- Racer
  - Description Logic based reasoning engine
  - Server-based
  - Integrates with Protégé-OWL

RDQL-by-example

- RDF source
- Queries
  - [http://www.cs.cornell.edu/courses/cs431/2006sp/examples/RDQL/vc.q1](http://www.cs.cornell.edu/courses/cs431/2006sp/examples/RDQL/vc.q1)