

OWL Functionality

(Adapted from slides by Jennifer Golbeck, University of Maryland)

Foundations

- XML
- RDF
- RDFS
- DAML+OIL
- OWL

OWL

- W3C Recommendation
- `xmlns:owl="http://www.w3.org/2002/07/owl#"`
- Successor to DAML+OIL
- Three Species
 - OWL Lite
 - OWL DL
 - OWL Full

OWL Lite Features

Equality

- `equivalentClass`
- `equivalentProperty`
- `sameAs`
- `differentFrom`
- `allDifferent`

Example

- ZipCode equivalentClass PostalCode
- If zip code and postal code are supposed to be different - e.g. zip is for american addresses and postal is for foreign ones - then we can say they are different

- ZipCode differentFrom PostalCode

```
<owl:Class rdf:ID="ZipCode">
```

```
  <owl:differentFrom
```

```
    rdf:resource="http://example.com/ont.owl#PostalCode/>
```

```
</owl:Class>
```

Property Characteristics

- inverseOf
 - hasParent is the inverseOf hasChild
- TransitiveProperty
 - E.g. - ancestorOf - if Bob is an ancestorOf Joe and Joe is an ancestorOf Fred, then Bob is an ancestorOf Fred
- SymmetricProperty
 - E.g. if Tom is marriedTo Michelle, then Michelle is marriedTo Tom
- FunctionalProperty (unique value)
 - Wine hasMaker - hasMaker is functional (there can be only one)
- InverseFunctionalProperty
 - The inverse of a functional property - makesWine is the inverse of hasMaker and is an inverseFunctionalProperty

Restrictions

- Property Type Restrictions
 - allValuesFrom
 - The hasMother property has allValuesFrom the class Woman
 - someValuesFrom
 - The hasChild property has someValuesFrom the class Woman
- Restricted Cardinality (can be 0 or 1 in Lite)
 - minCardinality
 - maxCardinality
 - Cardinality

Local Restrictions on Property Ranges

- Instead of setting a range for a property, each class can have its own range
- E.g. The range of eats for vegetarians is different than for non-vegetarians
- Done with subclasses and a restriction

```
<owl:Class rdf:ID="Vegetarian">  
  <rdfs:subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#eats"/>  
      <owl:allValuesFrom rdf:resource="#VegetarianFood"/>  
    </owl:Restriction>  
  </rdfs:subClassOf>  
  ...
```

Versioning

- versionInfo
- priorVersion
- backwardCompatibleWith
- incompatibleWith
- DeprecatedClass
- DeprecatedProperty

OWL DL and Full

Class Axioms

- one of
 - An enumeration of instances
 - E.g. fromContinent must have a value that is one of the following: Antarctica, N. America, S. America, Africa, Europe, Asia, Australia
- hasValue
 - A property must have a specific instance of a value
 - E.g. a U.S. Citizen's citizenOf property is restricted to have the value USA

Combinations

- unionOf (uses ParseType)
 - E.g. European Union Citizenship is the unionOf the citizenship of the member states
- intersectionOf (uses ParseType)
 - E.g. Fire engines are found in the intersection of RedThings and Trucks
- complementOf (used like subClassOf)
 - E.g. the complementOf livingThings are all things that are non-living
- disjointWith (used like subClassOf)
 - E.g. Man and Woman are disjoint classes

```
<owl:Class rdf:ID="Man"/>  
<owl:Class rdf:ID="Woman">  
  <owl:disjointWith rdf:resource="#Man"/>  
</owl:Class>
```

Intersection Example

```
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class>
      <owl:equivalentClass
rdf:resource="http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml#Food-
Beverage-and-Tobacco-Products"/>
    </owl:Class>

    <owl:Class>
      <owl:equivalentClass
rdf:resource="http://www.ksl.stanford.edu/projects/DAML/UNSPSC.daml#Meat"/>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```

ComplementOf Example

```
<owl:Class>
  <owl:complementOf>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Class>
          <owl:equivalentClass rdf:resource="#Food"/>
        </owl:Class>

        <owl:Class>
          <owl:equivalentClass rdf:resource="#Meat"/>
        </owl:Class>
      </owl:intersectionOf>
    </owl:Class>
  </owl:complementOf>
</owl:Class>
```

Cardinality

- Cardinality restrictions without limits