RDF - next step

URIs for the relations is important
- common naming (RDF)
- dereferencing! (RDFS, OWL)
Adding the Semantics

URI

- This is an RDF "Triple" (resources can also be URIs)
- Make the URI dereferencible for a machine-readable description
  - That is, put the ontology at that URI!
  - RDFS and OWL are basically XML dialects for this
Adding the Semantics

- Ex: Asserts that email address is a many to one relation (i.e. different people with same email address will be considered equivalent)
  - Can merge multiple FOAF files describing same user from different applications
Adding the Semantics

- RDFS and OWL provide other property restrictions as well
  - Which are properties needed for data modeling, domain vocabularies, etc.
- This is the heart of the Semantic Web
  - Why we talk links, not documents
Why?

• Semantics allows inferencing
  – RDFS allows domain, range mapping
    • :sister a rdfs:class;
      rdfs:range foaf:person;
      rdfs:domain ex:female.
  – OWL adds
    • Property inferencing
      – Symmetric, transitive, 1-1, 1-many, many-1
      – (similar to Entity-Relationshop Modeling)
    • "restricted" Class inferencing
      – The mother of a kitten is a cat (but not all mothers are cats)
      – Cardinality (a baseball team has 9 in the lineup…)
    • Some mapping relations
      – US-sport:soccer owl:sameAs UK-sport:football
Reasoners

• Procedural
  – A special purpose piece of code "doing the right thing"
    • RDFS, some subset of OWL

• Rule-based
  – Map constructs to rules
    • RDFS, many subsets of OWL (almost OWL Lite)
    • Non-OWL applications (cf. CWM& N3 rules)

• Special purpose
  – Subset of FOL optimized for decidable subset
    • OWL DL

• General
  – Full FOL reasoner
    • Beyond OWL
RDF Schema

• (Almost named RDF Vocab Def. Lang)
  – Some archival documents say RDF VDL
    • Luckily it stopped there
  – Graphs => Semantic networks
    • Class, subClassOf, property, subPropertyof, domain and range
    • BUT for the web (URIs!)
      – Also datatypes and literals
  – Also some usability stuff
    • label, comment
      – Defined to be strings with internationalization
RDFS example

http://home.fhtw-berlin.de/~engelh/div/hamster.rdfs
Intro to OWL (ca. 2003)

http://www.w3.org/2003/Talks/0522-webont-hendler/
OWL

- W3C Recommendation
- xmlns:owl=“http://www.w3.org/2002/07/owl#”
- Successor to DAML+OIL
- Three Species
  - OWL Lite
  - OWL DL
  - OWL Full
**Language Layers**

- **OWL Full**
  - Allow meta-classes etc

- **OWL DL**
  - Negation
  - Disjunction
  - Full Cardinality
  - Enumerated types

- **OWL Light**
  - (sub)classes, individuals
  - (sub)properties, domain, range
  - conjunction
  - (in)equality
  - cardinality 0/1
  - datatypes
  - inverse, transitive, symmetric
  - hasValue
  - someValuesFrom
  - allValuesFrom

- **RDF Schema**
  
  - Full
  - DL
  - Lite
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.

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

...
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

```xml
<owl:Class>
  <owl:intersectionOf rdf:parseType="Collection">
    <owl:Class>
    </owl:Class>

    <owl:Class>
    </owl:Class>
  </owl:intersectionOf>
</owl:Class>
```
ComplementOf Example

```xml
<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
Emerging Language

• A subset of OWL that is
  – Tractable
  – Easier syntactically
  – Easily mapped to rules
  – "understandable"
  – Powerful enough to be useful

• Sometimes called "RDFS+" or "OWL Mini"
  – Owl Fast, Owl Prime, Owl Ultralite, OWLET
## Vocabulary Comparison Matrix

<table>
<thead>
<tr>
<th>RDF Schema Features:</th>
<th>Current proposal</th>
<th>Oracle's ISWC</th>
<th>OWL Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>* Class (Thing, Nothing)</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:subClassOf</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdf:Property</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:subPropertyOf</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:domain</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:range</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:comment</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:label</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:seeAlso</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:isDefinedBy</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>&quot;Annotation&quot; Properties:</td>
<td>Current proposal</td>
<td>Oracle's ISWC</td>
<td>OWL Lite</td>
</tr>
<tr>
<td>* rdfs:label</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:comment</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:seeAlso</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* rdfs:isDefinedBy</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

**ADD FROM OWL**

<table>
<thead>
<tr>
<th>(In)Equality:</th>
<th>Current proposal</th>
<th>Oracle's ISWC</th>
<th>OWL Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>* equivalentClass</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* equivalentProperty</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* sameAs</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* differentFrom</td>
<td>?</td>
<td>yes</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Property Characteristics:**

<table>
<thead>
<tr>
<th>Current proposal</th>
<th>Oracle's ISWC</th>
<th>OWL Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>* inverseOf</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* TransitiveProperty</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* SymmetricProperty</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* FunctionalProperty</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* InverseFunctionalProperty</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* AnnotationProperty</td>
<td>no</td>
<td>yes</td>
</tr>
</tbody>
</table>

**MAYBE:**

<table>
<thead>
<tr>
<th>Current proposal</th>
<th>Oracle's ISWC</th>
<th>OWL Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>* AllDifferent, distinctMembers</td>
<td>?</td>
<td>yes</td>
</tr>
<tr>
<td>* ObjectProperty</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* DatatypeProperty</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>* disjointWith</td>
<td>?</td>
<td>yes</td>
</tr>
</tbody>
</table>

**Class/Restriction/MISC**

<table>
<thead>
<tr>
<th>Current proposal</th>
<th>Oracle's ISWC</th>
<th>OWL Lite</th>
</tr>
</thead>
<tbody>
<tr>
<td>* complementOf</td>
<td>no</td>
<td>yes*</td>
</tr>
<tr>
<td>* oneOf</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* Restriction</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* allValuesFrom, someValuesFrom</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* minCardinality, maxCardinality</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* cardinality</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* unionOf</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* intersectionOf</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* hasValue</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>* DataRange</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

**Note:** OWL DL/FULL has everything.
Foaf is pretty close to OWL Mini

Demo:

Browsing FOAF in Swoop

Foaf: http://xmlns.com/foaf/0.1/
Swoop: http://www.mindswap.org/2004/SWOOP/