The Semantic Web — RDF, RDF Schema, and OWL (Part 2)

Mitchell W. Smith
Array BioPharma, Inc.
msmith@arraybiopharma.com
Agenda

- **Part One:**
  - RDF
  - RDF/XML Syntax
  - RDF Schema
  - SPARQL

- **Part Two:**
  - OWL
  - Ontologies
  - Reasoners
  - RDF, RDF Schema, and OWL *versus* UML
  - Tools
RDF Introduction

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."

Semantic Web Layering

From: Berners-Lee XML 2000
RDF

- RDF is an assertional language intended to be used to express propositions.
  - RDF triples are propositions – a labeled connection between two resources.
  - An RDF triple contains three components:
    - The subject, which is an RDF URI reference or a blank node.
    - The predicate, which is an RDF URI reference.
    - The object, which is an RDF URI reference, a literal or a blank node.
  - RDF is monotonic and is defined to have an open-world assumption.
**RDF – Graph**

- An RDF graph is a set of RDF triples.
RDF Schema

- Extension to RDF.
  - As we have seen, RDF has no notion of resource attributes or relationships between attributes and resources.
  - RDF Schema is defined as a series of RDF statements.
- Describes classes and properties somewhat similar to typical programming languages.
  - Based upon an open-world assumption.
  - Properties are described in terms of the classes of resource to which they apply.
RDF Schema

- Provides information as additional descriptions of resources.
  - Just like RDF, not prescriptive.
- RDF Vocabulary Description Language is the official name for RDF Schema.
SPARQL – RDF Query Language

- Facilities include:
  - To extract information in the form of URIs, blank nodes, plain and typed literals.
  - To extract RDF subgraphs.
  - To construct new RDF graphs based on information in the queried graphs.

- Working Draft (was Candidate Recommendation)
Web Ontology Language (OWL)

- A language for defining and instantiating web ontologies.
  - Vocabulary extension of RDF and RDF Schema.
- Allows for the creation of reasoning programs.
  - Logical consequences based upon OWL entailment.
- OWL Semantics allow for unambiguous ontologies.
- Still an open-world assumption.
  - Classes and properties can be extended.
**OWL Species**

- **OWL Lite**
  - Provides a classification hierarchy and cardinality constraints. Designed as a minimal useful subset that is easy to implement.

- **OWL DL (Description Logic)**
  - Maximum expressiveness w/o losing computational completeness and decidability.

- **OWL Full**
  - Maximum expressiveness, but may not be computable or decidable.
Full RDF and RDF Schema are not supported since the semantics are not well constrained.

- For example, in RDF, can add assertions to RDF definitions.
  - RDF:
    
    \[
    \text{rdf:type} \quad \text{rdf:type} \quad \text{rdf:Property} \\
    \text{rdf:nil} \quad \text{rdf:type} \quad \text{rdf:List}
    \]
  
  - Mitch RDF:
    
    \[
    \text{rdf:Property} \quad \text{rdf:type} \quad \text{rdf:nil} \\
    \text{rdf:nil} \quad \text{rdf:type} \quad \text{rdf:Property} \\
    \text{rdf:type} \quad \text{rdf:nil} \quad \text{rdf:Property}
    \]

- Can make new assertions about RDF Schema definitions as well.
  - No differences between assertions about the RDF vocabulary and ontological assertions.

- In general, RDF and RDF Schema are not well constrained.
  - No distinction between classes and instances.
  - Properties can have properties.
OWL Lite and DL Vocabulary Elements Supported

- Only the following RDF and RDF Schema vocabulary elements are supported:
  - `rdf:List`
  - `rdf:nil`
  - `rdf:type`
  - `rdfs:comment`
  - `rdfs:Datatype`
  - `rdfs:domain`
  - `rdfs:label`
  - `rdfs:Literal`
  - `rdfs:range`
  - `rdfs:subClassOf`
  - `rdfs:subPropertyOf`
```xml
<!DOCTYPE rdf:RDF [<!ENTITY beer "http://www.sc.org/2006/beer#" >
<!ENTITY food "http://www.sc.org/food#" > ]>
<rdf:RDF
 xmlns ="&beer;"
 xmlns:beer ="&beer;"
 xml:base ="&beer;"
 xmlns:food ="&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:Ontology rdf:about="">
 <rdfs:comment>An example OWL ontology</rdfs:comment>
 <owl:imports rdf:resource="http://www.sc.org/food"/>
 <rdfs:label>Beer Ontology</rdfs:label>
 ...
 </owl:Ontology>
</rdf:RDF>
```
OWL Version Information

- `owl:versionInfo` – a string that specifies the version.
- `owl:priorVersion` – the prior version.
  - Instance of `owl:OntologyProperty` (this has `owl:Ontology` as its domain and range).
- `owl:backwardCompatibleWith` – the prior version with which this version is compatible.
  - Instance of `owl:OntologyProperty` (this has `owl:Ontology` as its domain and range).
- `owl:incompatibleWith` – the prior version with which this version is incompatible.
  - Instance of `owl:OntologyProperty` (this has `owl:Ontology` as its domain and range).
- `owl:DeprecatedClass` – a class that being preserved for backwards compatibility only.
- `owl:DeprecatedProperty` – a property that being preserved for backwards compatibility only.
**owl:imports**

- A reference to another ontology whose meaning is part of the meaning of this ontology
  - Instance of owl:OntologyProperty (this has owl:Ontology as its domain and range).
  - Is transitive.
  - Just a description; does not infer an action.

    `<owl:imports
     rdf:resource="http://www.example.org/bikes"/>`

    Note that this does not create a namespace shorthand.
owl:Class

- A group of resources.
- owl:Class rdfs:subClassOf rdfs:Class
  - rdfs:Class is not allowed in OWL Lite or DL.
- Classes are described via:
  - Class identifier.
  - Enumeration of classes.
  - Properties and property restrictions.
  - Intersection, Union, and Complement of classes.
owl:Thing and owl:Nothing

- owl:Thing is the set of all individuals.
  - owl:Class rdfs:subClassOf owl:Thing
- owl:Nothing is the empty set.
  - owl:Nothing rdfs:subClassOf owl:Class
    - Nothing is a subclass of every class.
The class is described by all of its instances.

```xml
<owl:Class rdf:ID="Continents">
  <owl:oneOf rdf:parseType="Collection">
    <owl:Thing rdf:about="#Eurasia"/>
    <owl:Thing rdf:about="#Africa"/>
    <owl:Thing rdf:about="#NorthAmerica"/>
    <owl:Thing rdf:about="#SouthAmerica"/>
    <owl:Thing rdf:about="#Australia"/>
    <owl:Thing rdf:about="#Antarctica"/>
  </owl:oneOf>
</owl:Class>
```
rdfs:subclassOf

- All the instances of one class are instances of another.
- This property is transitive.

```xml
<owl:Class rdf:ID="ScreenHouse">
  <rdfs:subClassOf rdf:resource="#Tent" />
</owl:Class>
```

This also entails that ScreenHouse is a rdfs:Class since the rdfs:domain of rdfs:subclassOf is a rdfs:Class.
**owl:equivlentClass**

- Defines that one class has the same members as another class.
- Does not mean that they are the same class.

```xml
<owl:Class rdf:about="#Transporter">
  <owl:equivalentClass rdf:resource="#AzgardBeam"/>
</owl:Class>
```
The two classes have no members in common.

<owl:Class rdf:about="#Wine">
    <owl:disjointWith rdf:resource="#Beer"/>
</owl:Class>
### `owl:intersectionOf`

- Restricts instances to all of the classes described.

```xml
<owl:Class>
<owl:intersectionOf rdf:parseType="Collection">
  <owl:Class>
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#ImperialPorter" />
      <owl:Thing rdf:about="#OatmealStout" />
    </owl:oneOf>
  </owl:Class>
  <owl:Class>
    <owl:oneOf rdf:parseType="Collection">
      <owl:Thing rdf:about="#ImperialIPA" />
      <owl:Thing rdf:about="#Bock" />
    </owl:oneOf>
  </owl:Class>
</owl:intersectionOf>
</owl:Class>
```
owl:unionOf

- Restricts instances to at least one of the classes described.

```xml
<owl:Class>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="#Ale"/>
    <owl:Class rdf:about="#Lager"/>
    <owl:Class rdf:about="#Bock"/>
    ...
  </owl:unionOf>
</owl:Class>
```
**owl:complementOf**

- Restricts instances to not be a member of a class.

```xml
<owl:Class rdf:ID="kabinett">
  <owl:Class>
    <owl:complementOf>
      <owl:Restriction>
        <owl:onProperty rdf:resource="#hasSugar" />
        <owl:hasValue rdf:resource="#Dry" />
      </owl:Restriction>
    </owl:complementOf>
  </owl:Class>
</owl:Class>
```
OWL Properties

- **owl:ObjectProperty** – the relationship from one class member to another.
- **owl:DataProperty** – the relationship from a class member to a value.

Note: these are disjoint in OWL Lite and DL.
rdfs:subPropertyOf

- Resources related by one property are also related by another.
- This property is transitive.

```
<owl:ObjectProperty rdf:ID="hasMother">
  <rdfs:subPropertyOf rdf:resource="#hasParent"/>
</owl:ObjectProperty>
```
**rdfs:domain**

- Defines that any class that has a given property is an instance of the class.

```
<owl:ObjectProperty rdf:ID="hasBankAccount">
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Class rdf:about="#Person"/>
        <owl:Class rdf:about="#Corporation"/>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>
</owl:ObjectProperty>
```

**Note:** Without the unionOf, this would be an intersection instead.
rdfs:range

- Defines that values of a property are instances of a class or data range (literal, enumeration, etc.).

```
<owl:ObjectProperty rdf:ID="isOakAged">
  <rdfs:domain rdf:resource="#Beer" />
  <rdfs:range rdf:resource="&xsd:boolean" />
</owl:ObjectProperty>
```

Note: Multiple ranges are interpreted as an intersection.
**owl:equivalentProperty**

- Defines that one property is equivalent to another property.
- Does not mean that they are the same property.

```xml
<owl:ObjectProperty rdf:ID="numberOfEuros">
  <rdfs:subPropertyOf rdf:resource="#numberOfDollars"/>
</owl:ObjectProperty>
```
owl:inverseOf

- Supports bi-directional properties.

```xml
<owl:ObjectProperty rdf:ID="hasChild">
  <owl:inverseOf rdf:resource="#hasParent"/>
</owl:ObjectProperty>
```
owl:FunctionalProperty

- Defines that the range of a property can have only one value given a domain.

```xml
<owl:ObjectProperty rdf:ID="captain">
  <rdfs:domain rdf:resource="#Plane" />  
  <rdfs:range rdf:resource="#Person" /> 
</owl:ObjectProperty>

<owl:FunctionalProperty rdf:about="#captain" />
```
owl:InverseFunctionalProperty

- Defines that the domain of a property can have only one value given a range.

```xml
<owl:InverseFunctionalProperty rdf:ID="sired">
  <rdfs:domain rdf:resource="#Person"/>
  <rdfs:range rdf:resource="#Vampire"/>
</owl:InverseFunctionalProperty>
```
owl:TransitiveProperty

- If a property, P, is specified as transitive then for any x, y, and z:
  
P(x,y) and P(y,z) implies P(x,z)

<owl:TransitiveProperty rdf:ID="partOf"/>
  <rdfs:domain rdf:resource="#Part"/>
  <rdfs:range rdf:resource="#Part"/>
</owl:TransitiveProperty>
If a property, P, is tagged as symmetric then for any x and y:

\[ P(x,y) \iff P(y,x) \]

```xml
<owl:SymmetricProperty rdf:ID="sister">
  <rdfs:domain rdf:resource="#Women"/>
  <rdfs:range rdf:resource="#Women"/>
</owl:SymmetricProperty>
```
Individual Properties

- Expresses constraints about individuals, not classes or properties.
  - owl:sameAs
  - owl:differentFrom
  - owl:allDifferent
owl:sameAs

- Links two distinct individuals as having the same identity.

```
<rdf:Description rdf:about="#Mitchell_Smith">
  <owl:sameAs rdf:resource="#MitchSmith"/>
</rdf:Description>
```

- Used to define mappings between ontologies.
- In OWL Full, owl:sameAs can identify two classes as being the same.
**owl:differentFrom**

- Links two distinct individuals as having different identities.

```xml
<rdf:Description rdf:about="#JimmieDuffyJr">
  <owl:differentFrom rdf:resource="#JimmieDuffySr"/>
</rdf:Description>
```
owl:allDifferent

- Links numerous distinct individuals as having different identities.

```xml
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <Person rdf:about="#JimmieDuffy"/>
    <Person rdf:about="#JimmieDuffyI"/>
    <Person rdf:about="#JimmieDuffyII"/>
  </owl:distinctMembers>
</owl:AllDifferent>
```
Local Property Restrictions

- An anonymous class description that describes what is required to satisfy the restriction.
  - Value Constraint – constrains the range of a property.
  - Cardinality Constraint – constrains the number of values a property can have.

Note: unlike RDF Schema, these apply only in the scope of the class definition.
**owl:**allValuesFrom

- The properties range is restricted to a class of individuals.

```xml
<owl:Class rdf:ID="Mead">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker"/>
      <owl:allValuesFrom rdf:resource="#Meadery"/>
    </owl:Restriction>
  </rdfs:subClassOf>
  ...
</owl:Class>
```
\texttt{owl:someValuesFrom}

- At least one of the class of individuals is in the property range.

\begin{verbatim}
<owl:Class rdf:ID="Mead">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasMaker"/>
      <owl:someValuesFrom rdf:resource="#Meadery"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
\end{verbatim}
**owl:hasValue**

- Restricts the properties range to a single literal value (instances of rdfs:literal) or a single resource (or ones that are the same via the owl:sameAs relation).

```xml
<owl:Class rdf:ID="Blur">
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#bestColor" />
      <owl:hasValue rdf:resource="#PowderBlue" />
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```
owl:maxCardinality

- Restricts the maximum number of (semantically distinct) values.

```xml
<owl:Class rdf:ID="Child">
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasParent" />
      <owl:maxCardinality
        rdf:datatype="&xsd;nonNegativeInteger">2
      </owl:maxCardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```
owl:minCardinality

- Restricts the minimum number of (semantically distinct) values.

```xml
<owl:Class rdf:ID="Child">
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasParent" />
      <owl:minCardinality
        rdf:datatype="&xsd;nonNegativeInteger">2
      </owl:minCardinality>
    </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```
**owl:cardinality**

- Restricts the number of (semantically distinct) values exactly.

```xml
<owl:Class rdf:ID="Child">
  ...
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hasParent" />
      <owl:cardinality rdf:datatype="&xsd;nonNegativeInteger">2</owl:cardinality>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```
OWL Datatypes

- OWL supports:
  - rdf:XMLLiteral
  - rdfs:Literal
  - XML Schema simple datatypes (most)
    - Only xsd:string and xsd:integer are required for reasoning.
  - Application defined datatypes
    - These are all unsupported.
  - Enumerated datatype – rdf:List
**OWL Datatypes – Enumerated Example**

- High, Medium, Low as an enumerated datatype:

```xml
<owl:DatatypeProperty rdf:ID="descretizedScore">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf>
        <rdf:List>
          <rdf:first rdf:datatype="&xsd;string">High</rdf:first>
          <rdf:rest>
            <rdf:List>
              <rdf:first rdf:datatype="&xsd;string">Medium</rdf:first>
              <rdf:rest>
                <rdf:List>
                  <rdf:first rdf:datatype="&xsd;string">Low</rdf:first>
                  <rdf:rest rdf:resource="&rdf;nil"/>
                </rdf:List>
              </rdf:rest>
            </rdf:List>
          </rdf:rest>
        </rdf:List>
      </owl:oneOf>
    </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>
```

...
OWL DL Restrictions Again

- OWL DL requires a pairwise separation between classes, datatypes, datatype properties, object properties, annotation properties, ontology properties (i.e., the import and versioning stuff), individuals, data values and the built-in vocabulary. This means that, for example, a class cannot be at the same time an individual.

- OWL DL requires that no cardinality constraints (local nor global) can be placed on transitive properties or their inverses or any of their superproperties.

- All axioms must be well-formed, with no missing or extra components, and must form a tree-like structure.

- ...
OWL Lite Restrictions Again

- OWL Lite forbids the use of owl:oneOf, owl:unionOf, owl:complementOf, owl:hasValue, owl:disjointWith, and owl:DataRange.
- OWL Lite requires that the subject of owl:equivalentClass and rdfs:subClassOf triples be class names and the object of owl:equivalentClass and rdfs:subClassOf triples be class names or restrictions.
- OWL Lite requires that the object of owl:allValuesFrom and owl:someValuesFrom triples be class names or datatype names.
- Cardinality can only be from zero to one.
Semantic Web Layering

From: Berners-Lee XML 2000
Tool Demos

- Want to learn more about entailment?
  - OWL Reasoning Examples and Hands-On Session
    - [http://owl.man.ac.uk/2005/07/sssw/](http://owl.man.ac.uk/2005/07/sssw/)

- Pellet – Java OWL DL reasoner
Ontologies

- Ontology is the science about the nature and relations of being and existence.
- (Philosophy) Concerned with what kinds of things really exist.
- (Applied) Concerned with the details of a formal description of some topic or domain.

*Not an Object Model!*
Developing Ontologies

- Check other related ontologies first!
  - Don’t invent one when another one is available.
  - Think about extending existing ones, if needed.
    - (remember the open-world assumption)
  - Not a UML Object Model.
    - More and less concepts – more on this later.
  - Look at others in similar domains and pattern after them.

- Check Swoogle for existing ontologies.
  - [http://swoogle.umbc.edu/](http://swoogle.umbc.edu/)
Ontology – Dublin Core Metadata Initiative

The Dublin Core Metadata Initiative provides simple standards to facilitate the finding, sharing and management of information.

- **Examples:**
  - **Element Name:** Title
    - **Label:** Title
    - **Definition:** A name given to the resource.
    - **Comment:** Typically, Title will be a name by which the resource is formally known.

  - **Element Name:** Creator
    - **Label:** Creator
    - **Definition:** An entity primarily responsible for making the content of the resource.
    - **Comment:** Examples of Creator include a person, an organization, or a service. Typically, the name of a Creator should be used to indicate the entity.
Ontology - FOAF Vocabulary Specification

- (Friend of a Friend) Ontology
  - Creating a Web of machine-readable pages describing people, the links between them and the things they create and do.
- Defined in RDF, RDF Schema, and OWL.
- A small bit of the FOAF vocabulary:
  
  Agent
  Person
  name
  nick
  title
  homepage
  mbox
  mbox_sha1sum
  img
  depiction (depicts)
  surname
  family_name
  givenname
  firstName
Ontology – Biological Pathways

Exchange Ontology (BioPAX)

- A data exchange format that enables sharing of pathway information, such as signal transduction, metabolic and gene regulatory pathways.

  ➢ [http://biopax.org](http://biopax.org)
Reasoners

- The logic part of the "stack".
  - It is in the most flux.
- Allow the specification of if-then rules/logical assertions.

\[
\{ \ ?x : \text{daughter} \ ?y \ \} \Rightarrow \{ \ ?y \ a : \text{Female} \ \}.
\]

\[
\{ \ ?x : \text{daughter}\?y. \ ?y.\text{age} \text{math:lessThan} 12. \ \} \Rightarrow \\
\{ \ ?y \ a : \text{Girl}. \ \}
\]
W3C Rule Interchange Format Working Group

- Chartered to produce a core rule language plus extensions which together allow rules to be translated between rule languages and thus transferred between rule systems.
  - Recently released "Use Cases and Requirements" public draft.
Common Logic Draft Standard

  - Specifies a family of logic languages designed for use in the representation and interchange of knowledge among disparate computer systems.
  - RDF, RDF Schema, and OWL can be represented in CL.
Reasoners

- Two main types of rule engines:
  - Forward chaining starts with the available data and uses inference rules to make new assertions.
    - Cwm (pronounced coom) is a general-purpose data processor for the semantic web.
      - [http://www.w3.org/2000/10/swap/doc/cwm.html](http://www.w3.org/2000/10/swap/doc/cwm.html)
      - Written in Python!
  - Backward chaining starts with a possible assertion and works backwards to see if it can be derived.
    - Euler is an inference engine supporting logic based proofs.
    - SWI-Prolog – implementation of ISO Standard
      - Has RDF and OWL parsers.
SKOS

- Alternative to OWL.
  - Built on RDF and RDF Schema.
  - Easier to use and understand.
  - Expresses the basic structure and content of concept schemes such as thesauri, classification schemes, subject heading lists, taxonomies, 'folksonomies', other types of controlled vocabulary, and also concept schemes embedded in glossaries and terminologies.

- http://www.w3.org/TR/swbp-skos-core-guide
RDF, RDF Schema, and OWL versus UML

- RDF, RDF Schema, and OWL semantics are based upon first-order logic and description logics.
  - These support the use of automated inference.
- UML semantics are not complete and consistent.
  - Does not support the use of automated inference.
  - OCL remedies some of this, but is not built on top of a formal model theory and proof theory.
RDF, RDF Schema, and OWL versus UML

- OWL supports complement, disjoint w/o one superclass, specify individuals w/o classes, functional properties, overlapping of classes, and properties, etc.

- UML supports operations, responsibilities, interface classes, qualified associations, aggregation, etc.

However, development of the Ontology Definition Metamodel (ODM) is underway.

- Maps RDF, RDF Schema, OWL, and Common Logic to UML and visa-versa.
- Recommended for adoption.
Embedding RDF in XHTML

- W3C Working draft available.

```html
<html>  
<head>  
<title>Mitch Home Page</title>  
</head>  
<body>  
<p>Go biking and then read this.  
<span property="foaf:name">Mitch Smith</span>'s home page.  
<a rel="foaf:mbox" href="mailto:msmith@arraybiopharma.com">Email me.</a>  
</p>  
</body>  
</html>

foaf:name "Mitch Smith"^^rdf:XMLLiteral ;
foaf:mbox <mailto:msmith@arraybiopharma.com>.
Tools for Semantic Web 2.0

Best guides that I've seen so far:

- Developers Guide to Semantic Web Toolkits for different Programming Languages
  - http://www.wiwiss.fu-berlin.de/suhl/bizer/toolkits/

- Semantic Web Development Tools: Introduction
  - http://esw.w3.org/topic/SemanticWebTools
RDF Data Sources

- WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory. 
  http://www.semanticweb.org/library

- Publicly available collections of RDF and web services that return RDF (a little dated at this point).
  ➢ http://rdfdata.org/data.html

- Swoogle (get ontologies, then look up RDF data)
  ➢ http://swoogle.umbc.edu/
Resources

- RDF Primer
  - http://www.w3.org/TR/rdf-primer/

- RDF Vocabulary Description Language 1.0: RDF Schema
  - http://www.w3.org/TR/rdf-schema/

- RDF Semantics
  - http://www.w3.org/TR/rdf-mt/

- SPARQL Query Language for RDF
  - http://www.w3.org/TR/rdf-sparql-query/
Resources

- OWL Web Ontology Language Overview
  - http://www.w3.org/TR/owl-ref/

- OWL Web Ontology Language Guide
  - http://www.w3.org/TR/owl-guide/

- Ontology Definition Metamodel (ODM)
  - http://ontology.omg.org/

- Semantic Web Best Practices and Deployment Working Group
  - http://www.w3.org/2001/sw/BestPractices/
Resources

- Semantic Web Best Practices and Deployment Working Group
  ```
  http://www.w3.org/2001/sw/BestPractices/
  ```

- RDF/A Primer 1.0 (Embedding RDF in XHTML)
  ```
  http://www.w3.org/TR/xhtml-rdfa-primer/
  ```

- Dave Beckett's Resource Description Framework (RDF) Resource Guide
  ```
  http://planetrdf.com/guide/
  ```
Resources

- SemanticWebDOAPBulletinBoard
  - [http://esw.w3.org/topic/SemanticWebDOAPBulletinBoard](http://esw.w3.org/topic/SemanticWebDOAPBulletinBoard)
- FOAF Vocabulary Specification
  - [http://xmlns.com/foaf/0.1/](http://xmlns.com/foaf/0.1/)
- Dublin Core Metadata Initiative
  - [http://www.dublincore.org/](http://www.dublincore.org/)
- Open source tools for the Semantic Web
  - [http://semwebcentral.org/](http://semwebcentral.org/)
Questions