The Semantic Web and Java

September 16, 2010

David Read

Semantic What?

© Blue Slate Solutions 2010
The Semantic Web

• Philosophy – “Web”
  – URI->Resource static mapping
  – Creates navigable "space"
  – Shared space = new genre of communication
  – Self-describing documents
  – URIs as identifiers not recipes
• Philosophy – “Semantic”
  – Machine processable
    • not natural language, human inference
  – For data: what you can do with it
  – For the future: conversion
  – Declarative (Statements)
  – Relationships

Tim Berners-Lee, 2000
Vision for Sharing Data

• Machine readability
• Reduced data friction
• Robust data failure management
• Unique, resolvable names
• Dynamic and flexible data
  – use inferencing to figure things out and create additional data

Semantic Web Bus¹

1 Semantic Web Bus diagram showing the integration of data, ontologies, logic, rules, and a heuristic engine for search functionality.
Semantic Web Evolution (Web 3.0 - 2020)²

Path from Data to Reasoning²
Semantic Wave Markets

- Research and Development
- Consumer
- Information and Communications Technology
- Industry Verticals
- Enterprise Horizontal

Dublin Core Metadata Initiative

Levels of Interoperability

1. **Description Set Profile Interoperability**
   - Shared formal vocabularies and constraints in records

2. **Description Set syntactic interoperability**
   - Shared formal vocabularies in exchangeable records

3. **Formal semantic interoperability**
   - Shared vocabularies based on formal semantics

4. **Shared term definitions**
   - Shared vocabularies defined in natural language
Formal Disciplines Form Basis for Semantic Web

- **Graph Theory**
  - 1736, Leonhard Euler, “The Seven Bridges of Konigsberg”
  - Nodes and relationships

- **Description Logic**
  - 1980s
  - Derived from first-order logic
  - Produce decideable (versus undecidable) knowledge representations
    - Doesn’t say how long it will take to “decide”
Semantics

- The study of meaning
- Cratylus of Plato (427-347 BC)
  Words → things

Ontology

- A formal organization of a body of knowledge
  - A domain vocabulary (with domain rules)
Referents from an Upper Ontology (gist)

Detail of Referents from gist

http://ontologies.semanticarts.com/gist/gist.owl
Detail of Predicates from gist

RDF Triple (Assertion)

- **Referent**: the object or idea to which a word or phrase refers (thing)
- **Subject**: A URI reference (to a referent)
- **Predicate**: The property of the triple (how the object is related to the subject) – also a URI
- **Object**: A URI reference (again, to a referent) or a constant

Subject → Predicate → Object

David → fatherOf → Sarah
RDF Triples Use URIs

Why?

Identify:
- Subject
- Predicate
- Object

Note: Triples are “directed”

RDF as an XML Schema

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
xmlns:si="http://www.w3schools.com/rdf/">
<br:Description
   rdf:about="http://www.w3schools.com">
   <si:title>W3Schools.com</si:title>
   <si:author>Jan Egil Refsnes</si:author>
</rdf:Description>
</rdf:RDF>
```
Wine Ontology (excerpted)

```xml
<owl:Class rdf:ID="Wine">
  <owl:Restriction>
    <owl:allValuesFrom rdf:resource="#Winery"/>
  </owl:Restriction>
  <owl:onProperty rdf:resource="#hasMaker"/>
  <owl:Restriction>
    <owl:minCardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:minCardinality>
  </owl:Restriction>
  <owl:onProperty rdf:resource="#madeFromGrape"/>
  <owl:Restriction>
    <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger">1</owl:cardinality>
  </owl:Restriction>
  <owl:onProperty rdf:resource="#hasFlavor"/>
  <owl:Restriction>
  </owl:Restriction>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>
</owl:Class>
```

Wine Ontology (graphically)
Food Ontology (excerpt)

```xml
  <owl:Class rdf:ID="ConsumableThing"/>
  <owl:Class rdf:ID="NonConsumableThing"/>
  <owl:complementOf rdf:resource="#ConsumableThing"/>
  <owl:Class rdf:ID="EdibleThing">
    <rdfs:subClassOf rdf:resource="#ConsumableThing"/>
  </owl:Class>
  <owl:Class rdf:ID="PotableLiquid">
    <rdfs:subClassOf rdf:resource="#ConsumableThing"/>
    <owl:disjointWith rdf:resource="#EdibleThing"/>
  </owl:Class>
</rdf:RDF>
```

Food Ontology (graphically)
Two Instances Using the Wine Ontology

-Winery rdf:ID="SchlossVolrad"/
-SweetRiesling rdf:ID="SchlossVolradTrochenbierenausleseRiesling"
-locatedIn rdf:resource="#GermanyRegion"
-hasMaker rdf:resource="#SchlossVolrad"
-hasSugar rdf:resource="#Sweet"
-hasFlavor rdf:resource="#Moderate"
-hasBody rdf:resource="#Full"
</SweetRiesling>

-Winery rdf:ID="SeanThackrey"
-PetiteSyrah rdf:ID="SeanThackreySiriusPetiteSyrah"
-locatedIn rdf:resource="#NapaRegion"
-hasMaker rdf:resource="#SeanThackrey"
-hasSugar rdf:resource="#Dry"
-hasFlavor rdf:resource="#Strong"
-hasBody rdf:resource="#Full"
</PetiteSyrah>

An Instance of a Wine (Triples as RDF and Graphic)

-Zinfandel rdf:ID="ElyseZinfandel"
-locatedIn rdf:resource="#NapaRegion"
-hasMaker rdf:resource="#Elyse"
-hasSugar rdf:resource="#Dry"
-hasFlavor rdf:resource="#Moderate"
-hasBody rdf:resource="#Full"
</Zinfandel>
Two Types of Statements to Define a Knowledgebase

- **TBox (a.k.a. T-Box)**
  - Terminological component
  - Controlled vocabulary
  - Define structure and rules
  - Wine \( \rightarrow \) isA \( \rightarrow \) PotableLiquid

- **ABox (a.k.a. A-Box)**
  - Assertion component
  - Fact associated with the TBox
  - Instances, individuals (*a.k.a. The Data!*)
  - ElyseZinfandel \( \rightarrow \) hasFlavor \( \rightarrow \) Dry
Inference (Defined via OWL and SWRL)

• Create an assertion based on a rule
  – e.g. derive new data
• Body (antecedent, precondition)
• Head (consequent, action)
• Common algorithms: tableaux, Rete

if
Person1 spouseOf Person2
then
Person2 spouseOf Person1

Brainstorm Rules and Inferences
5 Dimensions to Utilize a Technology

- Standards
  - XML, XSD, RDF, RDFS, OWL, SPARQL
- Infrastructure
  - Integration libraries, Triple stores, Inferencing engines (reasoners), Semantic services
- Content
  - Ontologies, Data \(\rightarrow\) Triples
- Tools
  - Ontology editors, Rule editors, IDEs
- Discipline
  - Methodologies, Standards, Best practices
Semantic Standards (W3C Recommendations, 2004)

- RDF
  - Resource Description Framework
  - All information represented as triples: Subject → Predicate → Object
  - Assertions (also inferences)
- RDFS
  - Resource Description Framework Schema
  - Necessary to support ontologies
  - Principally adds classes, properties, types
- OWL
  - Web Ontology Language
    - (figures that a standard about semantics would have a confused acronym)
  - Relationships between classes, properties, equality, cardinality
  - Required for computer processing of content (inferencing, reasoning)
  - Very limited – single assertion inferencing

Additional (more volatile) “Standards”

- SPARQL
  - SPARQL Protocol and RDF Query Language
  - W3C Recommendation, 2008
  - Query language for RDF triples
    Select ?name
    where {<http://blueslate.net/#BlueSlate>
    ?person <http://blueslate.net/#hasName> ?name }
- SWRL
  - Semantic Web Rule Language
  - W3C Member Submission, 2004
  - Supports rule authoring, overcoming single-instance inferencing limitations in OWL
  - Combines OWL and RuleML
    hasParent(?x1,?x2) ^ hasBrother(?x2,?x3) = hasUncle(?x1,?x3)
Software Technologies

- **Jena** – a framework for working with semantic web concepts
- **Pellet** – a semantic reasoner using the tableau algorithm
- Sesame – RDF triple store
- Virtuoso – RDB supporting SQL and SPARQL
- R2RQ – language to define RDB-to-RDF mappings
- Oracle – RDF triple store (A-Box)

RDF Serialization Syntaxes

- **RDF/XML** – “general purpose language for representing information on the Web”
  ```xml
  <rdf:Description rdf:about="http://monead.com/semantic/education#claimant">
    <rdfs:subPropertyOf rdf:resource="http://monead.com/semantic/education#policyHolder"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#ObjectProperty"/>
  </rdf:Description>
  ```
- **N3** (Notation 3) – readable alternative to RDF/XML
  ```n3
  dsr:AutoPolicy
  a owl:Class;
  rdfs:subClassOf dsr:Policy;
  owl:equivalentClass
  { a owl:Restriction
    owl:minCardinality 1;
    owl:onProperty dsr:manufacturer
  };
  ```
- **N-Triples** – subset of N3
  ```n3
  ```
- **Turtle (Terse RDF Triple Language)** – subset of N3
  ```turtle
  dsr:AutoPolicy
  a owl:Class;
  rdfs:subClassOf dsr:Policy;
  owl:equivalentClass
  { a owl:Restriction
    owl:minCardinality 1
    owl:onProperty dsr:manufacturer
  }.
  ```
For you OO folks, anything seem strange?
Individual

dsr:CivicPolicy
   a    dsr:Policy ;
   dsr:deductable "500" ;
   dsr:manufacturer "Honda" ;
   dsr:premium "342" .

dsr:McCoyLH
   a    dsr:Person ;
   dsr:legalName "Leonard H. McCoy" ;
   dsr:name "Bones" , "Leonard McCoy" ;
   dsr:policyHeld dsr:CivicPolicy , dsr:ColonialPolicy .

Restriction Class and Special Property

dsr:PolicyHolder
   a    owl:Class ;
   owl:equivalentClass
      [ a    owl:Restriction ;
        owl:onProperty dsr:policyHeld ;
        owl:someValuesFrom dsr:Policy
      ] .

dsr:policyHolder
   a    owl:ObjectProperty ;
   owl:inverseOf dsr:policyHeld .
Inferences

<http://monead.com/semantic/education#PolicyHolder>
<http://www.w3.org/2000/01/rdf-schema#subClassOf>
<http://monead.com/semantic/education#PolicyHolder> ,
<http://www.w3.org/2002/07/owl#Thing> ;
<http://www.w3.org/2002/07/owl#equivalentClass>
[ a <http://www.w3.org/2002/07/owl#Restriction> ;
<http://www.w3.org/2002/07/owl#onProperty>
<http://monead.com/semantic/education#policyHeld> ;
<http://www.w3.org/2002/07/owl#someValuesFrom>

<http://monead.com/semantic/education#CivicPolicy> a <http://www.w3.org/2002/07/owl#Thing> ;
<http://monead.com/semantic/education#policyHolder>
<http://monead.com/semantic/education#McCoyLH> ;
= <http://monead.com/semantic/education#CivicPolicy> .
Basic SPARQL Query


<table>
<thead>
<tr>
<th>s</th>
<th>p</th>
<th>o</th>
</tr>
</thead>
</table>

Number of Results: 160

Another SPARQL Query

prefix dsr: <http://monead.com/semantic/education#>

select ?s ?o where { ?s dsr:policyHolder ?o }

<table>
<thead>
<tr>
<th>s</th>
<th>o</th>
</tr>
</thead>
</table>
## Many More Features Within RDF, RDFS, OWL

- **Dynamic Classes (Categorization)**
  - `someValuesFrom`
  - `Intersection`
  - `Union`
  - `Complement`
  - `Boolean`
  - `Enumerations`
- **Special Properties**
  - `Inverse`
  - `Transitive`
  - `Symmetric`
  - `Functional`
  - `Inverse Functional`
- **And much more…**

## What About Java?

- In the semantic realm we’ve seen syntax to create classes, properties, individuals and rules. How do we use this information in our applications?
- For Java, we need to connect between the triple-based and OO worlds.
- Just like with O/R, there are frameworks to handle triple mapping and give us APIs to leverage:
  - We will look at **Jena**
- We also need to be able to support the inferencing aspect of the semantic web:
  - We will use **Pellet**
**Key Jena (and a Pellet) Classes**

- com.hp.hpl.jena.ontology.Individual
- com.hp.hpl.jena.ontology.OntModel
- com.hp.hpl.jena.query.Query
- com.hp.hpl.jena.query.QueryExecution
- com.hp.hpl.jena.query.QueryFactory
- com.hp.hpl.jena.query.QuerySolution
- com.hp.hpl.jena.query.ResultSet
- com.hp.hpl.jena.rdf.model.StmtIterator
- com.hp.hpl.jena.rdf.model.Statement
- com.hp.hpl.jena.rdf.model.RDFNode
- com.hp.hpl.jena.rdf.model.ModelFactory
- com.hp.hpl.jena.rdf.model.Model
- com.hp.hpl.jena.query.QueryExecution
- com.hp.hpl.jena.query.Query
- com.hp.hpl.jena.ontology.OntModel
- com.hp.hpl.jena.ontology.Individual
- com.hp.hpl.jena.reasoner.Reasoner

```
org.mindswap.pellet.jena.PelletReasonerFactory
```

**Loading a Serialized Model**

```
Reasoner reasoner = 
PelletReasonerFactory.getInstance().create();
Model infModel = 
ModelFactory.createInfModel(reasoner, 
ModelFactory.createDefaultModel());
OntModel model = 
ModelFactory.createOntologyModel(OntModelSpec
  .OWL_DL_MEM, infModel);
InputStream = new 
ByteArrayInputStream(assertions.getText()
  .getBytes("UTF-8"));
model.read(inputStream, null, “TURTLE”);
```
Obtain the Classes from a Model

```java
OntClass ontClass;
ExtendedIterator<OntClass> classesIterator;

classesIterator = ontModel.listClasses();
while (classesIterator.hasNext()) {
    ontClass = classesIterator.next();
    if (ontClass.isAnon()) {
        oneClassNode = new DefaultMutableTreeNode("Anonymous Class (" + ontClass.getId().getLabelString() + ")");
    } else {
        oneClassNode = new DefaultMutableTreeNode(ontClass.getLocalName() + " (" + ontClass.getURI() + ")");
    }
    // Do something with the oneClassNode instance
}
```

Obtain the Individuals in a Class

```java
Individual individual;
ExtendedIterator<Individual> individualsIterator;

// Assume setup from previous slide

individualsIterator = ontModel.listIndividuals(ontClass);
while (individualsIterator.hasNext()) {
    individual = individualsIterator.next();
    if (individual.isAnon()) {
        // can use individual.getId().getLabelString() for anon label
    } else {
        // can use individual.getLocalName() and individual.getURI()
    }
```
Get Predicates and Objects for Subjects

Statement statement; Property property; RDFNode rdfNode; StmtIterator stmtIterator;

// Assume setup from previous slide

stmtIterator = individual.listProperties();
while (stmtIterator.hasNext()) {
    statement = stmtIterator.next();
    property = statement.getPredicate();
    rdfNode = statement.getObject();
    if (property.isAnon()) {
        // can use property.getId().getLabelString() to get anon label
    } else {
        // can use property.getLocalName() and property.getURI() for predicate (property)
    }
    // Now get the object (could be literal or object)
    if (rdfNode.isLiteral()) {
        // Is a literal: get String representation with statement.getString()
    } else {
        // Is an object, can use statement.getResource().getLocalName()
        // and statement.getResource().getURI()
    }
}

Demo

• RDB Conversion to RDF
• Differences in Inferencing Levels
• Exploring an RDF Graph
• Querying the Cloud: DBPedia

Sample code: http://monead.com/semantic/
Topics

Visions

Technology

leadTo

produce

require

Terminology

Challenges

Status

transformTo

Summary

Semantics: Change in Multiple Dimensions

<table>
<thead>
<tr>
<th>Data Interpretation Topic</th>
<th>Current Technology</th>
<th>Description Logics Ontology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Structural Bias</td>
<td>Format input to conform to model</td>
<td>Accept input and attempt to make sense of it</td>
</tr>
<tr>
<td>Completeness</td>
<td>Model is assumed complete “Negation as failure”</td>
<td>Model is assumed partial “Open-world reasoning”</td>
</tr>
<tr>
<td>Class</td>
<td>Template for all instances</td>
<td>Set for relating instances</td>
</tr>
<tr>
<td>Disjointness</td>
<td>All rows (tuples) assumed to refer to different referents</td>
<td>Unless declared, any two instances may be referring to the same referent</td>
</tr>
<tr>
<td>Fact Model</td>
<td>Used as defined</td>
<td>Altered through inference</td>
</tr>
<tr>
<td>Data Volatility</td>
<td>Facts come and go</td>
<td>Facts can only be added (well, let me explain…)</td>
</tr>
</tbody>
</table>
### Shift in Design and Programming Paradigms

- **Class** indicates membership
  - not structure or behavior
- May create instances without creating a class
- Instances may belong to multiple classes
  - without any inheritance relationship between those classes
- **Property (predicate)** is a first-class concept
  - sub-properties and super-properties
- Properties are not associated with a particular class
- **Provably** versus **satisfiably** true or false
- **Necessary** and/or **sufficient** for inferencing
Sounds Good…

… Why the Long Journey to Adoption?

Types of Innovation

• **Sustaining / Incremental Innovation**: generally small innovations in products and processes aimed at existing customers

• **Disruptive / Discontinuous Innovation**: significant innovations generally aimed at unknown or non-existent customers
Normal Distribution and the Adoption S-Curve

Geoffrey Moore

Sustaining Innovation: Java

Java Adoption History

- Work starts (Oak)
- Demo on PDA
- Target Web (Java)
- Public Release
Disruptive Innovation: Relational Databases

Edgar Codd begins relational algebra -> relational data research

What is the Semantic Web’s Timeline?

• 1994: Tim Berners-Lee posits the value of semantics on the web at the (first) International WWW Conference
• 1995: OCLC/NCSA Metadata Workshop, Dublin, OH
• 1999: Dan Brickley submits “Nodes and Arcs 1989-1999” as a proposal to the W3C in November, which leverages RDF
• 2001: Tim Berners-Lee publishes an article entitled, “The Semantic Web” in the May issue of “Scientific American”
• 2004: RDF, RDFS and OWL become W3C Recommendations
• 2005: FACT++ released with support for OWL DL
• 2006: Metatomix release eclipse-based toolkit for RDF/OWL editing
• 2006: Oracle adds a native triple-store to their Oracle 10g product (release 2)
• 2007: TopQuadrant releases development tool for RDF/OWL, integrates with multiple engines

Sources: product web sites and W3C.org as of May 9, 2010
Topics

Visions

require

leadTo

produce

Terminology

hasA

Technology

Challenges

transformTo

Status

transformTo

Summary

Standards

• RDF/RDFS: Stable (implemented in multiple products)
• OWL: version 2 a W3C recommendation
  – second system effect
• SPARQL: Lacks support for querying RDFS and OWL-based extensions to RDF
• SWRL: Not yet a W3C recommendation
• Focus needed on rule expression standards
Technology

- Development Tools
  - Lack usability beyond hardcore semantic-savvy developers
  - Buggy, inconsistent interpretation of standards
- Runtime environments
  - Limited feature sets
  - Performance issues (time/space tradeoffs)
- Products and libraries include: Oracle, Metatomix, FACT++, Protégé, Siderean, TopBraid, SWiFT Sublème, Calais, Jena

Linked Data Cloud
Querying the Cloud – DBpedia Objects

```sparql
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbt: <http://dbpedia.org/thing/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbt: <http://dbpedia.org/thing/>

SELECT ?subject WHERE {
}
```

Results:
- dbpedia.org/resource/Subject "Aristotle"
- dbpedia.org/resource/Subject "Socrates"
- dbpedia.org/resource/Subject "Plato"
- dbpedia.org/resource/Subject "Immanuel Kant"
- dbpedia.org/resource/Subject "Max Weber"
- dbpedia.org/resource/Subject "Emile Durkheim"
- dbpedia.org/resource/Subject "Georg Simmel"
- dbpedia.org/resource/Subject "Herbert Blumer"
- dbpedia.org/resource/Subject "Evelyn B. Coontz"
- dbpedia.org/resource/Subject "Jonathan Haidt"

Number of Results: 92

Querying the Cloud – Social Theories

```sparql
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbr: <http://dbpedia.org/resource/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbt: <http://dbpedia.org/thing/>
PREFIX dbo: <http://dbpedia.org/ontology/>
PREFIX dbp: <http://dbpedia.org/property/>
PREFIX dbt: <http://dbpedia.org/thing/>

SELECT ?subject WHERE {
}
```

Results:
- dbpedia.org/resource/Subject "Aristotle"
- dbpedia.org/resource/Subject "Socrates"
- dbpedia.org/resource/Subject "Plato"
- dbpedia.org/resource/Subject "Immanuel Kant"
- dbpedia.org/resource/Subject "Max Weber"
- dbpedia.org/resource/Subject "Emile Durkheim"
- dbpedia.org/resource/Subject "Georg Simmel"
- dbpedia.org/resource/Subject "Herbert Blumer"
- dbpedia.org/resource/Subject "Evelyn B. Coontz"
- dbpedia.org/resource/Subject "Jonathan Haidt"

Number of Results: 92
Some Thought Leaders

- Sir Tim Berners-Lee at TED
  - 2009
    - http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html
  - 2010
    - http://www.ted.com/talks/tim_berners_lee_the_year_open_data_went_worldwide.html

- Dave McComb, Semantic Arts Inc.
- Dr. James Hendler, Professor, RPI
- Dean Allemang, Chief Technology Consultant, TopQuadrant
POCs, Partial SDLC, Limited Deployments

- Lower-level (structural) standards are sound
- Tools exist to test concepts and develop ontologies
- Structured semantic ontology definitions supported
- Specific use cases can leverage existing semantic technology
- FOAF, RDFa and microformats can be leveraged on websites now
  - “Bridging the Human and Data Webs”
- Integration protocols are absent
  - currently use known standards (web services, SOA)
  - theories abound (location-agnostic services)
- Rule authoring standards in flux
- Query language in its infancy
- Upper ontologies required to achieve shared meaning

A Few Relevant Resources

- Books
  - Semantic Web Programming (2009)
    Heblert, John et al
    Excellent coverage of semantic modeling (RDF, RDFS, OWL) as well as inferencing. Serves as a great reference guide as well.
    Allemang, Dean and Handler, Jim
    Uses Python to drill into the low-level implementation details around semantically-based application operation.
  - Programming the Semantic Web (2009)
    Segaran, Toby et al
    Uses Python to drill into the low-level implementation details around semantically-based application operation.

- Web
  - W3C Semantic Web Activity Homepage: [http://www.w3.org/2001/sw/](http://www.w3.org/2001/sw/)
  - Dave’s Semantic Homepage: [http://monead.com/semantic/](http://monead.com/semantic/)
  - Cheat Sheet: [http://ebiquity.umbc.edu/ file_directory_/resources/97.pdf](http://ebiquity.umbc.edu/ file_directory_/resources/97.pdf)
Thank You

Q&A

David.Read@blueslate.net

Semantic Workbench
Goal: To Create
An Open Source Semantic Technology Exploration Tool
Please participate:
semanticwb.sourceforge.net

Sources

1. http://www.w3.org/2000/Talks/1206-xml2k-tbl/slide1-0.html
17. http://linkeddata.org/
18. Hebeler et al. (2009), Semantic Web Programming, Wiley Publishing Inc, Indiana, 9
19. http://www.w3.org/TR/xhtml-rdfa-primer/

All website content retrieved May 9, 2010 unless otherwise noted