Module I: Overview of Semantic Technologies and the Semantic Web

- Introduction and Orientation to the Course
- What is Semantic Technology? What is it Good For?
- The Appreciating Value of "Meaning"
- Demo I
- Mapping the Semantic Terrain

............. Lunchtime .............

- Comparing Semantic with Conventional Technologies
- Demo II
- Stocking Your Semantic Toolbox
- Knowledge Management and the Semantic Web
The Semantic Wave is NOT one thing ... there are differing major streams within it!

- The Semantic Web
  - Information sharing on a global scale
  - Intranets vs. Internet

- Semantic Technology
  - Enhanced knowledge access and search
  - Semantic Interoperability
  - Information syndication
  - ... and so forth

---

Semantic Web: Make web content machine-readable!

“The Semantic Web is a vision: the idea of having data on the Web defined and linked in a way that it can be used by machines not just for display purposes, but for automation, integration and reuse of data across various applications.” [W3C 2001]

“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.” [Tim Berners-Lee et al 2001]
What could the Web do?

Can this sort of interaction become part of the Web itself?

How could the Web do it?

Built-in by the Webmaster

Agree upon an “interlingua”
What about XML? Doesn’t it support semantics?

- HTML gave us formatting tags
- XML gave us custom tags
  - You get to pick your tags/attributes
  - Tags can have “meaning” specific to your application
- Many dialects have blossomed
- XML and XML Schema became W3C standards
- Standard dialects are being developed by many industry groups – XBRL.org, FpML.org, TaxML.org, ...
- Every large organization has their own XML Schemas
Figure 2.

Gartner: All Tied Up with XML: 2001

From 2001 through 2004 enterprises spent $3 billion on modeling activities with no return on investment from $2 billion of it.

Source: Gartner Research

A new Web of terminology

What's the Interlingua for the Interlingua?

Use the same technology for mapping web pages to terminology to map terminology to one another
AAA Slogan -

Anyone can say Anything about Any topic

Non-unique naming

public String getContextPath() {
    try {
        Method getContextPathMethod =
            delegate.getClass().getMethod("getContextPath", null);  //$NON-NLS-1$
        return (String) getContextPathMethod.invoke(delegate, null);
    } catch (Exception e) {
        // ignore
    }
    return null;
}

“Java”

“Java”?

“Coffee”?

Programming language?

Hot Beverage?
What is Semantic Technology?

“Semantic technology (software) allows the meaning of and associations between information to be known and processed at execution time.

For a semantic technology to be truly at work within a system, there must be a knowledge model of some part of the world (an active ontology) that is used by one or more applications at execution time.”

-- TopQuadrant

Semantic Technology and the Internet

<table>
<thead>
<tr>
<th>Static</th>
<th>Dynamic</th>
<th>Transactional</th>
<th>Semantic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encoding</td>
<td></td>
<td></td>
<td>+ RDF, OWL ?</td>
</tr>
<tr>
<td>HTML, CGI, Perl, ...</td>
<td>+ RDBMS JSP, ASP, Java, ...</td>
<td>+ XML J2EE, .NET, ...</td>
<td></td>
</tr>
<tr>
<td>Creation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hand crafted by people for people</td>
<td>Generated applying specific templates, used by people</td>
<td>Generated by applications based on fixed schemas, used by applications and people</td>
<td>Generated by applications based on models, used by applications, devices and people</td>
</tr>
<tr>
<td>Paradigm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertisement, information, large newspaper</td>
<td>A newspaper becomes a catalog</td>
<td>A catalog becomes a transaction platform</td>
<td>Platforms connect Set of mind = “Interoperable”</td>
</tr>
<tr>
<td>Static</td>
<td>Dynamic</td>
<td>Transactional</td>
<td>Semantic</td>
</tr>
<tr>
<td>HTML, CGI, Perl, ...</td>
<td>+ RDBMS JSP, ASP, Java, ...</td>
<td>+ XML J2EE, .NET, ...</td>
<td>+ RDF, OWL ?</td>
</tr>
<tr>
<td>Creation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hand crafted by people for people</td>
<td>Generated applying specific templates, used by people</td>
<td>Generated by applications based on fixed schemas, used by applications and people</td>
<td>Generated by applications based on models, used by applications, devices and people</td>
</tr>
<tr>
<td>Paradigm</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advertisement, information, large newspaper</td>
<td>A newspaper becomes a catalog</td>
<td>A catalog becomes a transaction platform</td>
<td>Platforms connect Set of mind = “Interoperable”</td>
</tr>
<tr>
<td>Killer Apps</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marketing</td>
<td>Sales</td>
<td>Service</td>
<td>Integration</td>
</tr>
<tr>
<td>• Browser</td>
<td>• Search, Content Mgmt, Web Application Servers</td>
<td>• Portals, Process Integration, Web Services</td>
<td>• Advisors, Personal Agents, IP Apps, Cognitive Engines</td>
</tr>
</tbody>
</table>

© Copyright 2007-2008 TopQuadrant Inc.
Contrasting the Semantic Web and Semantic Technology Solutions

- **They differ in their**
  - Goals – *what* they want to accomplish
  - Value propositions – *why* bother with them
  - Readiness for commercial application – *when* to get on the train

- **They have some overlap in their use of**
  - Tools (e.g., Ontology editors)
  - Vendors and Products
  - Solution architectures

- **They share in common**
  - Semantic languages and standards (developed by the W3C--World Wide Web Consortium)
  - For More Information see:
    * http://www.w3.org/
    * http://www.w3.org/2001/sw

W3C standards for semantic models

- **W3C Semantic stack is built on XML**
- **XML-based Ontology languages are being developed to support semantic interoperability.**

“Semantic Web is stimulating a whole new class of applications at individual, enterprise and web scales”
– Eric Miller, W3C, Semantic Technologies for eGOV’2003
  * www.w3.org/
  * www.w3.org/2001/sw
Application integration today

Yahoo! Finance

MySimon

How do they do it?

- Yahoo! and MySimon are collecting simple, well-understood data (personal financial records, retail prices).
- Programmers built a single program for all data sources
- De-facto standards (like Quicken) are already in place

Simplest kind of Application Integration – everyone agrees on a simple representation. Just use it!
It’s supposed to be a web, after all!

Mash-ups are not the responsibility of some service on the web …

They are the responsibility of the web infrastructure!!

How does it work?

RDF – the Ultimate Mash-up Language!!
A view of TopBraid Composer being used to connect a real estate ontology and other RDF resources with geospatial ontologies.

Capability Case: Model-enabled Application Integrator
Solution Story: Geospatial Mashup in TopBraid Composer

Capability Case: Semantic Multi-Faceted Search
Solution Story: BeachHouse – search and bring the beach home

© Copyright 2007-2008 TopQuadrant Inc.
Semantic Model-driven Applications

Semantic Technology supports new types of Dynamic Business Applications

“Designed for users ... Built for change”
Capability Case: Semantic Portal

http://del.icio.us/CapabilityCases/SemanticPortal

Integrating multiple sources of information – talk submissions, attendee registrations, user profiles.
Systems developed in different work practice settings have different semantic structures for their data. Time-critical access to data is made difficult by these different semantics. Semantic Data Integration allows data to be shared and understood across these settings.

Aviation Security – Passenger Threat Analysis

Customer Story: Major retailer deploys customer site in 12 weeks

A web portal for consumers to maintain information about their homes and belongings. Many different product – all have different types of information.
TopBraid Live – has an open Architecture

Graph stores

- Example products: Sentences, Semagix Freedom™, JARG, Cogito

- Role in Semantic Web:
  - Pre-date Semantic Web Standards
  - Strictly speaking, not semantic web;
  - typically not W3C conformant

- How to evaluate one:
  - Scalability (number of nodes, throughput)
  - Query language
### Comparing Semantic and Relational

<table>
<thead>
<tr>
<th></th>
<th>RDB</th>
<th>Semantic Model (Ontology)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Ability to Answer Questions</strong></td>
<td>Database must be designed to answer the questions. Specific, typically complex, queries must be developed.</td>
<td>Ontology must be designed to answer the questions. Queries can be generic and very simple.</td>
</tr>
<tr>
<td><strong>Ability to Accommodate Change</strong></td>
<td>Inflexible: Database structure must be modified so it can continue to answer the questions. Queries must be re-written. Data must be ported.</td>
<td>Flexible: Ontology can be easily extended so it can continue to answer new questions. No data porting required.</td>
</tr>
<tr>
<td><strong>Processing Speed</strong></td>
<td>Can be very fast with proper tuning - mature technology: Known optimization approaches. Certain queries, such as multi-table joins and self-joins are known to cause problems.</td>
<td>Not as fast, but improving, tuning does not affect flexibility: Adding more processing power and distributed computation helps. Performs better than RDBMS for certain query types.</td>
</tr>
</tbody>
</table>

### Key differences in the representation of relationships

<table>
<thead>
<tr>
<th></th>
<th>RDB</th>
<th>Semantic Model (Ontology)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cardinality of relationships</strong></td>
<td>Relationships are either 1:1, many:1 or many:many. Many:many relationships must be broken into many:1 relationships by creating join tables.</td>
<td>By default all relationships are many:many. Functional properties and cardinality restrictions are used to specify 1:1, many:1 as well as other cardinalities. It is possible to specify, for example, 1:4 or min 2, etc.</td>
</tr>
<tr>
<td><strong>Information bearing relationships</strong></td>
<td>Additional information about the relationship is represented by the extra columns in the join table.</td>
<td>Relationship is reified (made into a class). Additional information is represented as properties of the class.</td>
</tr>
<tr>
<td><strong>The nature of relationships</strong></td>
<td>Implicit Embedded in the name of the join table or in the name of the column. Typically these names are not designed for ease of understanding of the nature of the relationship.</td>
<td>Explicit Care is taken to name a relationship in a way that its nature and intentions are well understood.</td>
</tr>
</tbody>
</table>
Properties (ST) vs. Attributes and Relations (OO)

- OWL Properties represent relations between two individuals (Not classes)

- OWL Property types:
  - Object Properties link an individual to an individual
  - Datatype Properties link an individual to simple values
    - integers, floats, strings, booleans, and so forth
    - an XML Schema Datatype primitive value or an RDF literal

- Those with OO experience/expertise must overcome the typical pre-conception that properties belong to the class!

Properties are first-class constructs

In contrast to most OO paradigms, where properties are “owned” or “contained in” Classes

This allows relationships between Properties

hasParent

hasMother

hasFather

This is not a class diagram!

"wherever I use the property 'author', tell the world that they can read 'dc:creator'"
In OWL, Properties may have Sub Properties

- It is possible to form hierarchies of properties (these are not Class hierarchies)

  ![Property Hierarchy Diagram]

- The rdfs:subPropertyOf construct allows relationships to be abstracted up the sub-property tree.

(SST) In OWL, Classes are *inferred* or *computed*

- OWL classes are interpreted as sets that contain individuals
  - A class is *not* a kind of template as in OO technology
  - In OWL, classes are built up of descriptions that specify the conditions that must be satisfied by an individual to be a member of the class

- Subclasses are subsets of their parent classes.

- Superclass-subclass relationships can be computed automatically by a *reasoner*
Properties (ST) vs. Attributes and Relations (OO)

- OWL Properties represent relations between two individuals (*Not classes*)

- OWL Property types:
  - *Object Properties* link an individual to an individual
  - *Datatype Properties* link an individual to simple values
    - integers, floats, strings, bools, and so forth
    - an XML Schema Datatype primitive value or an RDF literal

- Those with OO experience/expertise must overcome the typical pre-conception that properties belong to the class!

Semantic Web – OO Gotchas!

In the Semantic Web, you infer the class of an object.

The class of an object can change:
  - over time
  - with what you know/believe
  - with whom you trust

Properties are first-class objects (independent of classes!)
  - Properties form hierarchies as well as classes

No behavior is described anywhere – only *inferencing*

Multiple set membership is commonplace
  - No OO inheritance
How Semantic Languages Work

Bring information together
Draw *inferences* for further processing

RDF

RDFS

OWL

What is RDF?

- RDF *(Resource Description Framework)* is an infrastructure for:
  - Encoding,
  - Exchange and
  - Distributing metadata

RDF Triple:

Subject

Predicate

Object

Safety Harbor

offers

Massage
RDF: A distributed network of data!

RDF Files: “bags of triples”

Safety Harbor offers Massage

Facial offeredBy Safety Harbor

RDFS is a schema language for RDF

RDFS allows us to create vocabularies

Resort

Spa

Activity

Treatment

Safety Harbor

Massage

Rdfs:subClassOf

rdfs:domain

rdfs:range

rdf:type

offers

 rdf:type

rdfs:domain

rdfs:range

offers
If the bags contain RDFS key symbols, then RDFS can infer certain conclusions.