Introduction to the Semantic Web

Example Applications:

**ITTALKS**

- **ITTALKS** is a database driven website of IT related talks at UMBC and other institutions. The database contains information on:
  - Seminar events
  - People (speakers, hosts, users, ...)
  - Places (rooms, institutions, ...)
- Web pages with DAML markup are generated.
- The DAML markup supports agent-based services relating to these talks.
- Users get talk announcements based on the interests, locations and schedules.

[http://ittalks.org/](http://ittalks.org/)
**ITTALKS Architecture**

People

Web server + Java servlets

Web Services

Agents

Web Services

Web server + Java servlets

MapBlast, CiteSeer, Google...

Apache Tomcat

HTTP

RDBMS

DAMLS reasoner

DAMLS files

DAML reasoning engine

**Travel Agent Game in Agentcities**

**Motivation**
- Market dynamics
- Auction theory (TAC)
- Agent collaboration (FIPA & Agentcities)

**Features**
- Open Market Framework
- Auction Services
- DAML message content
- OWL Ontologies
- Global Agent Community

**Technologies**
- FIPA (JADE, April Agent Platform)
- Semantic Web (RDF, OWL)
- Web Services (SOAP, WSDL, DAML-S)
- Internet (Java Web Start)

**Ontologies**
- FIPA platform infrastructure services, including directory facilitators enhanced to use DAML-S for service discovery

**http://taga.umbc.edu/**

- Our research group’s web site generate both HTML and OWL.
- **HOW?** This is relatively easy since the content is in a database.
- **PHP** is sufficient for the job.
- **HTML** pages have links to corresponding OWL.
- **WHY?** This exposes the information to programs and agents – no more web scraping.

**Mobile & Pervasive Computing Uses**
How does OWL Help?

OWL provided a uniformed language which met many needs in developing a complex pervasive computing system.

OWL as an Ontology Language

- Key Benefits
  - Helps to separate the task of knowledge engineering and system engineering
  - Helps to define “semantic” specifications for applications that exploit KR and reasoning
  - Opens the door to the Semantic Web for mobile and pervasive computing applications
    - Gaining access to a vast amount of information on the Web
    - Applications will be less restricted by their sensing capability

OWL as a Service Description Lang.

- Key benefits
  - Enables semantic service discovery and matching
    - Expressing more detailed and more precise service description
  - Provides a means for ubiquitous service composition
  - Allows intelligent applications to have fine-grain control over system execution
    - E.g. is it economic to print using a close by printer?
    - E.g. is it polite to display my email using the room’s project?

OWL as a Language for Interoperability

- Key benefits
  - Encourages independently developed systems to interoperate
    - A standard language backed up the W3C
    - Industrial organizations tend to follow W3C standards
    - Amateurs tend to develop programs based on W3C standards
  - Enables knowledge sharing and reasoning
    - APIs for processing RDF/XML -- the normative exchange syntax of OWL -- are widely available and suitable for building commercial strength applications
    - OWL has well defined language semantics for building OWL reasoners. A few number of OWL reasoners are now available.
  - Provides standard constructs for ontology mapping
    - Multiple ontologies will likely to exist in a shared PerCom space
    - Ontology mapping can help apps. that adopt different ontologies to interoperate
OWL is XSLT/XML Friendly

- Key benefits
  - Information expressed in OWL can be transformed into other languages for external processing
    - OWL => Prolog rules or Jess rules
    - OWL => XHTML
    - OWL => PHP, JavaScript
  - Maximizes the reusability of the knowledge that is encoded in OWL
    - Not all useful tools and applications can process OWL
    - Not all XML developers are willing to switch to OWL
    - Not all users think OWL (esp. in RDF/XML) is easily readable

OWL as a Meta Language

- Key benefits
  - Helps to define new languages to control the high level behavior of a complex system (e.g. policy languages)
    - It’s inflexible to adjust the dynamic behavior of a complex system by writing low level code
    - Using meta languages, users can change system behavior without needing to change the low-level system implementation
  - Meta languages (e.g. policy) defined using OWL can be used to work with other knowledge that is expressed in OWL
    - Security -- define policy to control actions that are expressed in OWL
    - Privacy protection -- define policy to protect user private information that are expressed in OWL

OWL for Defining Context Model

- Key benefits
  - Helps to overcome semantic ambiguities in representing contexts using programming languages
    - Java representations of contextual knowledge has limited expressiveness
    - OWL representations have well defined semantics
  - Encourages the reuse of previously defined context model
    - Generic context models (e.g., time, space, actions, policy) can shared and reused by different context-aware systems
    - Tools (e.g., reasoners, APIs) associated with these generic context models often can also be used by different system implementations

Ontology-Driven PerCom Systems

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>myCampus (CMU)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Task Computing (Fujitsu)</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EasyMeeting (UMBC)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Context Broker Architecture (UMBC)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CMU MyCampus Project

- **Objective**: Enhance campus life through context-aware services accessible over the WLAN
- **Ontologies**
  - Personal/contextual: location, calendar, organizational etc.
  - Privacy preferences: who has access to what, “obfuscation” rules
  - Web services: automated service identification and access (OWL-S)

http://www.cs.cmu.edu/~sadeh/mycampus.htm#Video

Fujitsu Task Computing

- **Objective**: Make computing available throughout the physical environment while it is effectively invisible to the users

http://www.taskcomputing.org/

STEER-SIS for Web Services
The Context Broker Architecture

The EasyMeeting System

An EasyMeeting Scenario

An EasyMeeting Scenario