OCM Ontology and Ontology Services

August 14, 2012
NOAA, Boulder CO

Peter Fox (RPI* and WHOI**) pfox@cs.rpi.edu and OTHERS!!
*Tetherless World Constellation, ** AOP&E
Modern informatics

- Use cases
- Stakeholders
- Distributed authority
- Access control
- Ontologies
- Maintaining Identity
Use Case

• … is a collection of possible sequences of interactions between the system under discussion and its actors, relating to a particular goal.

• The collection of Use Cases should define all system behavior relevant to the actors to assure them that their goals will be carried out properly.

– is a prose description of a system's behavior when interacting with the outside world.

– is a technique for capturing functional requirements of business systems and, potentially, of an ICT system to support the business system.

– can also capture non-functional requirements
IOCM Use Case

• “Provide a list of NOAA datasets in a particular geographic area including quality, extent, responsible agency and datatype with the intent of justifying NOAA contributions to the IOCM.”
What might an implementation look like?
Systems v. Frameworks

• Rough definitions
  – Systems have very well-defined entry and exit points. A user tends to know when they are using one. Options for extensions are limited and usually require engineering
  – Frameworks have many entry and use points. A user often does not know when they are using one. Extension points are part of the design
  – Platforms are built on frameworks
Use Cases Expose System Requirements

- Exposes goals, outcomes, actors/roles, resources, preconditions, process flow, artifacts
- And … semantics, terms, concepts and their relations
Semantic Web Methodology & Technology Development Process

- Establish and improve a well-defined methodology vision for semantic technology based on application development
- Leverage controlled vocabularies, etc.

Open world: evolve, iterate redesign, redeploy

Use case
Rapid prototype
Leverage technology infrastructure
Adopt technology approach
Scientist/expert reviews and iteration

Small team, mixed skills
Analysis
Develop model ontology
Lite tools
Information modeling

• Conceptual models, sometimes called domain models, are typically used to explore domain concepts and often created
  – as part of initial requirements envisioning efforts as they are used to explore the high-level static business or science or medicine structures and concepts
  – as the precursor to logical models or as alternatives to them

• Followed by logical and physical models

• Introduced in ANSI processes in 1978!
Data Type and others...
Another use case

- To identify priority wind energy areas for potential development and accelerate the leasing process by evaluating existing ocean uses by humans and natural resources.
- Coastal and Marine Spatial Planning
Logical models

- For a logical data model to be normalized, it must include the **full population of attributes** to be implemented and those attributes must be defined in terms of their domains or **logical data types** (e.g., character, number, date, picture, etc.).
- A logical data model requires a complete scheme of **identifiers** or candidate keys for unique identification of each occurrence in every entity.
• A *logical* model is provable in the mathematics of data science. E.g. for relational databases, logical models generally conform to relational theory.

• Thus a logical model contains only **fully normalized entities**. Some of these may represent logical domains rather than, for e.g. potential physical tables.
Object oriented design

• Object-oriented modeling is a formal way of representing something in the real world (draws from traditional set theory and classification theory). Some basics to keep in mind in object-oriented modeling are that:
  – Instances are *things*.
  – *Properties* are attributes.
  – *Relationships* are pairs of attributes.
  – *Classes* are types of things.
  – *Subclasses* are subtypes of things.
Physical models

• A *physical* model is a single logical model instantiated in a specific information system (e.g., relational database, RDF/XML document, etc.) in a specific installation.

• The physical model specifies implementation details which may be features of a particular product or version, as well as configuration choices for that instance.
For example for relational DBs

<table>
<thead>
<tr>
<th>Feature</th>
<th>Conceptual</th>
<th>Logical</th>
<th>Physical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entity Names</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Entity Relationships</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Attributes</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Primary Keys</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Foreign Keys</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Table Names</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Column Names</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Column Data Types</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Not an isolated set of models

• Handle errors, iteration, evolution, …
  – To the logical model?
  – To the conceptual model?

• Relating to and/ or integrating with other information models?
  – General rule – integrate at the highest level you can (i.e. more abstract)
Working with knowledge

Expressivity

Implementability

Maintainability/ Extensibility
Working with knowledge

- Query
- Rule execution
- Inference
Or it may be this ...
Expressivity/ Implementation

Declarative

Linked open data
URI/http/RDF *

Procedural

Ontology encoded
Semantic Web Standards*

- Ontology - OWL 1.0 (Web Ontology Language, 2004)
- Query - SPARQL 1.0 (SPARQL Protocol and RDF Query Language, 2008)
- OWL 2.0 (2009)
- Taxonomy - SKOS (Simple Knowledge Organization System, 2009)
- Rules - RIF (Rule Interchange Framework, 2010)
- SPARQL 1.1 (in review)
- NB. No service standards!
An ontology specifies a rich description of the

- Terminology, concepts, nomenclature
- Properties explicitly defining concepts
- Relations among concepts (hierarchical and lattice)
- Rules distinguishing concepts, refining definitions and relations (constraints, restrictions, regular expressions)

relevant to a particular domain or area of interest.

*Based On AaaI '99 Ontologies Panel - Mcguinness, Welty, Ushold, Gruninger, Lehmann

Unit of exchange – the triple example (linked data)

Heath (2009)
• Note: XMLS *not* an ontology language
  
  – Changes format of DTDs (document schemas) to be XML
  
  – Adds an *extensible type hierarchy*
    
    • Integers, Strings, etc.
    
    • Can define sub-types, e.g., positive integers
  
• RDFS *is* recognisable as an ontology language
  
  – *Classes* and *properties*
  
  – *Sub/super-classes* (and properties)
  
  – *Range* and *domain* (of properties)
However

- **RDFS too weak** to describe resources in sufficient detail
  - No **localized range and domain** constraints
    - Can’t say that the range of hasChild is person when applied to persons and elephant when applied to elephants
  - No **existence/cardinality** constraints
    - Can’t say that all *instances* of person have a mother that is also a person, or that persons have exactly 2 parents
  - No **transitive, inverse or symmetrical** properties
    - Can’t say that isPartOf is a transitive property, that hasPart is the inverse of isPartOf or that touches is symmetrical

- **Difficult to provide reasoning support**
  - No "native" reasoners for non-standard semantics
  - May be possible to reason via First Order axiomatisation
Ontology Services

• SEAVOX
  – http://www.bodc.ac.uk/products/web_services/vocab/
• SEEGRID
• GeoNetwork
  – http://trac.osgeo.org/geonetwork/wiki/proposals/D CATandRDFServices
NERC Vocabulary Server

The NERC Vocabulary Server gives data managers the means to access lists of controlled terms to describe data, thus saving the time and costs associated with unraveling the meaning of a given data set.

- **Introduction** — a brief summary of the value of implementing the Vocabulary Server
- **Connectivity** — consumer access options for the Vocabulary Server
- **Collection, concept and scheme URIs** — how to browse the content of the Server
- **An example of the ReSTful and SOAP API methods** — the GetCollection method
- Access the NERC Vocabulary Server version 2.0 (NVS2.0) [documentation](#)

**Introduction**

The NERC Vocabulary Server provides access to lists of standardised terms that cover a broad spectrum of disciplines of relevance to the oceanographic and wider community.
Vocab Configuration

Introduction

In this activity, SISS team will create configuration and associated documentation required for SISSVoc 3.x deployment and work with BCM to deploy/understand the service and deploy into Production.

Workflow

- Step 1: Prepare and Pre-Condition the Vocabulary as an RDF/XML Document
- Step 2: Publish the Vocabulary as a Single Document
- Step 3: Create SPARQL Endpoint thru Loading the Vocabulary in OpenRDF from the Ontology URI
- Step 4: Prepare ELDA Configuration and Deploy to SISSVoc Service
- Step 5: Configure the Concept URI Server to Get Concept Descriptions From SISSvoc

Step 1: Prepare and Pre-Condition the Vocabulary as an RDF/XML Document

- The role of a vocabulary owner is to formalize and precondition the vocab. BCM as vocabulary owner has identified to mark the SISS4BoM as the recommended RDF/XML document.
Vocabulary web service — version 1.0

This document describes version 1.0 of the BGS Vocabulary Service.

Contents

- Usage
- Resources
- Supported media types
- Status codes
- Clients

Usage

The vocabulary service has been implemented as a RESTful web service using the Restlet framework.

Information is exposed through a series of resources, where each resource implements the standard HTTP interface and is identified by a URL. Each resource provides representations of information for a number of supported media types.

The vocabulary service supports content negotiation to provide the most appropriate representation for a particular client. For example to request a JSON representation of a resource, a client should use the following HTTP header in its request:

```plaintext
Accept: application/json
```

Not all clients will be able to add or modify HTTP headers, therefore the vocabulary service supports other ways to specify the required representation format. Each media type has been mapped to a file extension. Each resource supports a media parameter in the query string which may be used to tunnel the required media type. A JSON representation would be returned by the following:

```
/vocabularies?media=application/json
```

Additionally like in Ruby on Rails, the file extension may be appended to the URL:
```
/vocabularies.json
```

Resources

All resources are relative to [http://webservices.bgs.ac.uk/data/services/vocabulary/1.0](http://webservices.bgs.ac.uk/data/services/vocabulary/1.0)
Don’t forget - the metadata

• Metadata (maybe data too) is “materialized” into instances of the ontologies and accessed via services
• I.e. linked in a machine processable (e.g. queryable) way
• And… linked among vocabularies (more general than “mapping”)
• For discovery, inventory, access (and use…) but only what’s needed…
Status/ schedule

• First iteration of use cases are being analyzed and modeled
• Early fall – first models and ontologies (expect lots of re-use, minimal development), design of vocab service (re-use), prototype application to use it
• Then – evaluation with GeoPortal, etc.
Questions?

- Thanks…
- Peter Fox pfox@cs.rpi.edu @taswegian