Brief Introduction to Ontology

Just use XML, right?

XML is Syntax
- DTDs talk about element nesting
- XML Schema schemas give you data types
- need anything else? => write comments!

Domain Semantics is complex:
- implicit assumptions, hidden semantics
- sources seem unrelated to the non-expert
Need Structure and Semantics beyond XML trees:
- employ richer OO models
- make domain semantics and "glue knowledge" explicit
- use ontologies to fix terminology and conceptualization

XML DTDs and XML Schemas are sufficient for exchanging data between parties who have agreed to meaning of terms beforehand
Ontologies are critical for exchanging data between applications from diverse parties in order to ensure terms used are semantically equivalent

Sharing Semantics

Metadata
- Data describing the content and meaning of resources and services.
- But everyone must speak the same language...

Terminologies
- Shared and common vocabularies
- For search engines, agents, curators, authors and users
- But everyone must mean the same thing...

Ontologies
- Shared and common understanding of a domain
- Essential for search, exchange and discovery
  - Ontologies aim at sharing meaning

The Meaning Triangle

Humans require words (or at least symbols) to communicate efficiently. The mapping of words to things is indirect. We do it by creating concepts that refer to things.

- The relation between symbols and things has been described in the form of the meaning triangle:

  “Jaguar”
  Concept
  Symbol
  Thing

Ontologies: A definition

An "ontology" describes the common words, concepts and relationships between concepts used to describe and represent an area of knowledge.

An ontology can range from:
- Taxonomy (knowledge with minimal hierarchy or a parent/child structure) to a...
- Thesaurus (words and synonyms) to a...
- Conceptual Model (with more complex knowledge) to a...
- Logical Theory (with very rich, complex, consistent and meaningful knowledge)

A well-formed ontology is one that is expressed in a well-defined syntax that has a well-defined machine interpretation consistent with the above ontology definition
Ontology Spectrum

**Weak Semantics**
- Terminological
- "is a" vs. "instance of"
- Taxonomy
- "is a" vs. "instance of" (e.g., "tank")
- Relation Model
- "is a" vs. "instance of" (e.g., "part of")
- **Logical Theory**
- "is disjoint subclass of with transitivity"

**Strong Semantics**
- Can be arbitrary
- Term Semantics relations used
- Synonymy, homonymy, superclass/subclass, inverse
- Concepts, Properties, Relations, Axioms
- "x is a homonym of y, e.g. "tank"

Ontology Modeling

**An explicit description of a domain**

**Concepts** (class, set, type, predicate)
- event, gene, gammaBur1st, atium, molecule, cat

**Properties of concepts and relationships between them** (slot)
- Taxonomy: generalisation ordering among concepts (isa, partOf, subProcess)
- Relationships, Role or Attribute: functionOf, hasActivity, location, eats, size

XML based ontology languages

**OWL**
- DAML = DARPA Agent Markup Language
- OIL = Ontology Inference Layer
- OWL = Ontology Web Language

Ontology Modeling

**An explicit description of a domain**

**Constraints or axioms** on properties and concepts:
- value: integer
- domain: cat
- cardinality: at most 1
- range: $0 \leq X \leq 100$
- oligonucleotides $< 20$ base pairs
- cows are larger than dogs
- cats cannot eat only vegetation
- cats and dogs are disjoint

Values of concrete domains
- integer, strings
- 20, tryptophan-synthetase

Individuals or Instances
- sulphur, tryA Gene, felix

Ontology versus Knowledge Base
- An ontology = concepts+properties+axioms+values
- A knowledge base = ontology+instances

Difference between Relational Schema and Ontology

Relational Schema primary purpose is to organise data within a database
- As we have seen previously relationships between entities are implicit and interpretation is necessary by program/human who accesses
- If the program/human lacks knowledge of semantics of the data or database then becomes unusable

Ontology
- Relationships defined formally and interpretable by both human and program
Ontology Design

Ontology terminology that we will use

Class: Description of a concept in the domain of discourse
- E.g. Room in the intelligent space domain

Slot/Property: Describes attribute of a concept or relationship of a concept
- E.g. longitude and latitude of a room provides its coordinates
- E.g. isSpatiallySubsumedBy — points to instance of class “Building”

Facet/Restriction: Describes a constraint on a slot
- Slot cardinality e.g. one or many values allowed in slot
- Slot value type, e.g. String, Number, Boolean, Enumerated, Instance

Introduction to Ontologies © Declan O’Sullivan 14

Fundamental Rules in Ontology Design

1) There is no one correct way to model a domain — there are always viable alternatives.
   - The best solution almost always depend on the application that you have in mind and the extensions that you anticipate.

2) Ontology development is necessarily an iterative process.

3) Concepts in the ontology should be close to objects (physical or logical) and relationships in your domain of interest.
   - These are most likely to be nouns (objects) or verbs (relationships) in sentences that describe your domain

Introduction to Ontologies © Declan O’Sullivan 15

Step 1: Determine Scope and Domain of Ontology

What is the domain that the ontology will cover?
- E.g. representing an intelligent University space

For what are we going to use the ontology?
- E.g. applications that support roaming students, lecturers and visitors are going to use the ontology

For what types of questions the information in the ontology should provide answers?
- E.g. to help roaming student print to the nearest printer etc.
- E.g. to help the university intelligent space applications to implement energy conservation policies etc.

Who will use and maintain the ontology?
- E.g. if it is maintained by someone in TCD for TCD users but needs also to be used by visiting users from UCD etc., then mappings need to be included

Introduction to Ontologies © Declan O’Sullivan 16

Step 1: Competency Questions

One way to determine scope is to list out the questions the knowledge base based on the ontology should be able to answer for example
- Which rooms have data projectors?
- Is DR1 G.14 a conference room?
- Is there a conference room near restrooms?
- What is the best choice of room for a student tutorial?

Judging from this list of questions, the ontology will include information on
- Various room types and characteristics
- Various event types and characteristics
- Classifications of rooms with respect to appropriateness for events
- Classifications of rooms with respect to location
- Classifications of rooms with respect to furniture/equipment

Introduction to Ontologies © Declan O’Sullivan 17

Step 2: Consider reusing other ontologies

Examples
- UNSPSC ontology which provides terminology for products and services (www.unspsc.org)
- DAML ontologies
Step 3: Enumerate important terms in the ontology

List of all terms we would like either to make statements about or to explain to a user:
- What are the terms we would like to talk about?
- What properties do those terms have?
- What would we like to say about those terms?

For example, important university intelligent space related terms will include:
- Building, campus, restroom, room, hallway, stairs, chairs, projectors;
- different types of event, such as lecture, tutorial, seminar, meeting;
- and so on.

Get a comprehensive list of terms without worrying about:
- overlap between concepts they represent,
- relations among the terms,
- or any properties that the concepts may have,
- or whether the concepts are classes or slots.

Step 4: Define the classes and class hierarchy

From the list created in Step 3, we select the terms that describe objects having independent existence rather than terms that describe these objects. These terms will be classes in the ontology and will become anchors in the class hierarchy.

Multiple Inheritance allowed

Step 5: Define the properties of classes - slots

In general, there are several types of object properties that can become slots in an ontology:
- "intrinsic" properties such as the capacity of the room;
- "extrinsic" properties such as a room's name;
- parts, if the object is structured; these can be both physical and abstract "parts" (e.g., the courses of a meal);
- relationships to other individuals; these are the relationships between individual members of the class and other items (e.g., spatiallySubsumedBy, representing a relationship between a room and a CompoundPlace (subclasses are Building and Campus))

Step 6: Define the Facets of each Slot

Facet: Describes a restriction on a slot
- Slot cardinality e.g. one or many values allowed in slot
- Slot value type, e.g. String, Integer, Boolean, Enumerated, Instance

In Protege

Blue rectangle means property defined on this class
Round brackets around rectangle means property inherited
Step 7: (Optionally) Create Instances

Chief Honcho is an instance of the class Editor.
This instance has the following slot values defined:

<table>
<thead>
<tr>
<th>Slot</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>Title</td>
<td></td>
</tr>
<tr>
<td>Contact</td>
<td></td>
</tr>
</tbody>
</table>

Seven steps to…

1. Determine scope and domain of ontology
2. Consider reusing existing ontologies
3. Enumerate important terms in the ontology
4. Define the classes and class hierarchy
5. Define the properties of classes – slots
   - Properties, relationships
6. Define the facets of slots
   - Value type (String, Number, Boolean, Enumerated, Instance)
   - Value cardinality
7. Define the instances