Ontology Design

CS 431 - March 15, 2006
Carl Lagoze - Cornell University
What is an ontology?

• Many definitions from different domains
  - Philosophy - A Systematic Account of Existence
  - A.I. - An explicit specification of a conceptualization (the objects, concepts, and other entities that are assumed to exist in some area of interest and the relationships that hold among them) Gruber 1993

• Our context – web semantic interoperability
  - Formal explicit description in a domain of discourse (classes, concepts)
  - Attributes of concepts (slots, properties, relationships)
  - Slot restrictions (facets)

• A Knowledge Base is an ontology combined with instance data
Overview of Ontology and Knowledge Base Development

• Define the classes
• Arrange the classes into a taxonomic hierarchy
  – establish class/sub-class relationships
• Define slots and their restrictions
• Define instances
Why Ontologies (1)?

• Sharing a formalized definition of information structure among people or software
  - e.g., ShopBots extracting and aggregating information from different sites
  - formalization of notation and decidability is important
Why Ontologies (2)?

• Enable reuse of domain knowledge
  - modularize development process
  - e.g., share common concepts of time (events, situations) in domain specific ontologies
Why Ontologies(3)?

- Separate operational from domain knowledge
  - avoid hard-coding domain knowledge into programs
  - parameterize code to allow use in different domains
  - allow easy modification of domain knowledge without code changes
Some guiding rules of ontology design

• In most cases there are many ways to model a domain
• Ontology development, like program development, is by nature iterative
• The ontology should closely correspond to the objects (nouns) and relationships (verbs) in the sentences describing your domain of interest
Ontology Development (1)

• Define the scope
  - What domain does it describe?
  - What applications will be built upon it?
  - What are the questions for which it should provide answers?
    * competency questions that serve as tests of ontology.
  - Who are its users and maintainers?
  - Limiting the scope is vital to a usable ontology.
    * Don’t include extraneous information!
Ontology Development (2)

• Search available online ontologies and determine utility of them.
  - [http://www.daml.org/ontologies/](http://www.daml.org/ontologies/)
  - [http://protege.stanford.edu/plugins/owl/owl-library/](http://protege.stanford.edu/plugins/owl/owl-library/)

• Increases possibility of interoperability with other applications
Ontology Development (3)

- Enumerate important terms in ontology
  - Concepts and properties
  - Ignore relationships for now, just brainstorm
- Establish a naming convention
  - capitalization
  - use of delimiters
  - singular or plural
  - prefixes
Ontology Development (4)

• Define concepts and concept hierarchy
  - Top-down
  - Bottom-up
  - Remember transitivity of class hierarchy
  - Depth and breadth issues
    • Avoid single sub-class
    • Excessive # of siblings (> 12) indicates possible need for new sub-classing
Ontology Development (5)

- Define slots or properties of classes
  - data properties
    - names
    - flavors
    - colors
  - object properties
    - whole/part relationships
    - other semantic relationships among individuals
  - Reflect class/sub-class hierarchy
    - Slots should distinguish sub-classes
    - Attach slot at most general point in hierarchy
    - Remember that all sub-classes inherit slot
Ontology Development (6)

• Define facets of slots
  - Data type of data slots
  - Domain and range of object slots
    • Again obey class generality rule
  - Slot cardinality
Ontology Development (7)

- Test with instances
Issues (1) - Multiple Inheritance

- Most systems allow it
- Frequently necessary to model a domain
- Make sure slot inheritance works
Issues (2) - Classes vs. Slots

• E.g., wine with slot color, or sub-classes for red, white, rose

• If classes with different slot values become restrictions for other slots in other classes, create a new class for distinction
  - example - consider car color vs. wine color
Issues (3) - Instance or Class?

• **Answer is domain specific and application specific**
  - Magnet Pinot Noir vs. Magnet Pinot Noir 2003

• **Remember that instances are essentially the leaves in the knowledge base hierarchy**
  - no notion of sub-instance

• **Instances should be answers to competency questions**
More Issues

• Disjoint classes
  - Can’t have any instances in common
  - Pay attention to open world issues

• Inverse slots
  - Usually unnecessary to represent
    • system can infer information
    • “reverse queries” are possible
  - Sometime useful for understanding
    • system provide way of automatically completing
Ontology Tool

- Protégé
  - [http://protege.stanford.edu/](http://protege.stanford.edu/)
- Open Source, Java Based
- Export to a variety of formats