Ontologies

Lecture 3


Agenda

• Foundations
  – Definitions, features, classifications, applications, modelling principles

• Ontology engineering
  – Methodologies and methods and tools to build, manage and use ontologies

• How to build an ontology (Ontology 101)

• Selected tools
  – Protégé, Collaborative Protégé, myOntology, NEON
  – Thursday: WSMT
What is an ontology?

• **An ontology defines the basic terms and relations comprising the vocabulary of a topic area, as well as the rules for combining terms and relations to define extensions to the vocabulary**
  
  *Enabling Technology for Knowledge Sharing. AI Magazine.* Winter 1991. 36-56

• **An ontology is an explicit specification of a conceptualization**
  
What is an ontology (ii)?

- An ontology is a hierarchically structured set of terms for describing a domain that can be used as a skeletal foundation for a knowledge base
  

- An ontology provides the means for describing explicitly the conceptualization behind the knowledge represented in a knowledge base
  

What is an ontology (iii)?

“An ontology is a formal, explicit specification of a shared conceptualization”

- Machine-readable
- Concepts, properties relations, functions, constraints, axioms, are explicitly defined
- Consensual Knowledge
- Abstract model and simplified view of some phenomenon in the world that we want to represent

Features of an ontology

- Modelled knowledge about a specific domain
- Defines
  - A common vocabulary
  - The meaning of terms
  - How terms are interrelated
- Consists of
  - Conceptualization and implementation
- Contains
  - Ontological primitives
Classifications of ontologies (i)

- Issue of the conceptualization
  - Upper-level/Top-level
  - Domain
  - Application

- Degree of formality
  - Highly informal: in natural language
  - Semi-informal: in a restricted and structured form of natural language
  - Semi-formal: in an artificial and formally defined language
  - Rigorously formal: in a language with formal semantics, theorems and proofs of such properties as soundness and completeness

Classifications of ontologies (ii)

### Languages for building ontologies

- Ontologies can be built using various languages with various degrees of formality
  - Natural language
  - UML
  - ER
  - OWL/RDFS
  - WSML
  - FOL
  - ...

- The formalism and the language limit the kind of knowledge that can be represented
- A domain model is not necessarily a formal ontology only because it is written in OWL

### Applications of ontologies

- **Knowledge representation**
  - Ontology models domain knowledge

- **Semantic annotation**
  - Ontology is used as a vocabulary, classification or indexing schema for a collection of items
  - Semantic annotation of Web services

- **Semantic search**
  - Ontology is used as a query vocabulary or for query rewriting purposes

- **Configuration**
  - Ontology defines correct configuration templates
Principles for the design of ontologies

- **Clarity:**
  - To communicate the intended meaning of defined terms

- **Coherence:**
  - To sanction inferences that are consistent with definitions

- **Extendibility:**
  - To anticipate the use of the shared vocabulary

- **Minimal Encoding Bias:**
  - To be independent of the symbolic level

- **Minimal Ontological Commitments:**
  - To make as few claims as possible about the world

Clarity

An ontology should communicate effectively the intended meaning of defined terms.

Definitions should be objective.

Definitions can be stated on formal axioms, and a complete definition (defined by necessary and sufficient conditions) is preferred over a partial definition (defined by only necessary or sufficient conditions)…
Minimal Encoding Bias

The conceptualization should be specified at the knowledge level without depending on a particular symbol-level encoding.

```
(define-class Travel (?travel)
  "A journey from place to place"
 :axiom-def
 (and (Superclass-Of Travel Flight)
    (Subclass-Of Travel Thing)
    (Template-Facet-Value Cardinality
      arrivalDate Travel 1)
    (Template-Facet-Value Cardinality
      departureDate Travel 1)
    (Template-Facet-Value Maximum-Cardinality
      singleFare Travel 1))
 :iff-def
 (and (arrivalDate ?travel Date)
    (departureDate ?travel Date))
 :def
 (and (singleFare ?travel Number)
    (companyName ?travel String)))
```

No minimal encoding bias

Mininal Encoding Bias

(singleFare ?travel Number)

should be substituted by:

(singleFare ?travel CurrencyQuantity)
Extensibility

One should be able to define new terms for special uses based on the existing vocabulary, in a way that does not require the revision of the existing definitions.

• Currency dimension
• Definition of currencies
• Relationship between currencies

(define-individual Euro (Unit-of-Measure)
"An Euro is the currency on the European Union"
:= (* 0,96 USDollar)
:axiom-def
(- (Quantity.dimension Euro) CurrencyDimension))

Coherence

An ontology should be coherent: that is, it should sanction inferences that are consistent with the definitions. [...] If a sentence that can be inferred from the axioms contradicts a definition or example given informally, then the ontology is incoherent.
**Minimal Ontological Commitments**

Since ontological commitment is based on the consistent use of the vocabulary, ontological commitment can be minimized by specifying the weakest theory and defining only those terms that are essential to the communication of knowledge consistent with the theory.

```prolog
(define-class Travel (?travel)
   "A journey from place to place"
 :axiom-def
 { .... }
 :iff-def
 (and (arrivalDate ?travel Date)
      (departureDate ?travel Date))
 :def
 (and (singleFare ?travel Number)
      (companyName ?travel String)))
```

- **Absolute/relative date?**
- **Format?**

---

**Ontological Commitments**

- Agreements to use the vocabulary in a coherent and consistent manner (Gruber)
  - An agent commits (conforms) to an ontology if it “acts” consistently with the definitions

- Connection between the ontology vocabulary and the meaning of the terms of such vocabulary
Ontology Engineering

What is ontology engineering?

• “the set of activities that concern the ontology development process, the ontology life cycle, and the methodologies, tools and languages for building ontologies”

Gomez-Perez et al, 2004
Ontology life cycle

Project management: controlling, planning, quality assurance etc.

Domain analysis
motivating scenarios, competency questions, existing solutions

Conceptualization
conceptualization of the model, integration and extension of existing solutions

Implementation
implementation of the formal model in a representation language

Usage/Maintenance
adaptation of the ontology according to new requirements

Ontology engineering activities

Management
- Scheduling
- Control
- Quality assurance

Development oriented
- Pre-development
  - Environment study
  - Feasibility study
- Development
  - Specification
  - Conceptualization
  - Formalization
- Implementation
  - Post-development
  - Maintenance
  - Use

Support
- Knowledge acquisition
- Evaluation
- Integration
- Documentation
- Merging
- Configuration management
- Alignment
Methodologies for building ontologies

- Validated guidelines on how the ontology building process should be structured
- Well-organized process instead of “ontology development driven by inspiration and intuition only”
- Methodologies do not support all of the aforementioned activities
- They implicitly assume a particular development paradigm for the ontology engineering process
- Some of them also provide supporting methods and tools

Collaborative ontology engineering

- Two or more people interact and exchange knowledge in order to build a common, shared ontology in pursuit of a shared, collective, bounded goal.
  - Interaction may be indirect but required.
  - Argumentation as a common interaction means.
  - Simple contributions not enough.
  - Bounded goal: beginning and end.
  - Collaborators may have individual goals.
### Uschold and King Methodology

- Identify purpose
- Build ontology
  - Capture
  - Coding
  - Integrating
- Evaluation
- Documentation

### Grüninger and Fox Methodology

- Identify motivating scenarios
- Elaborate informal competency questions
- Specify terminology in FOL
  - Identify objects
  - Identify predicates
- Formal competency questions
- Specify axioms in FOL
- Specify completeness theorems
**METHONTOLOGY methodology**

- Develop + Management Activities + Support in parallel

- Develop:
  - Specification
  - Conceptualization
  - Formalization
  - Implementation
  - Maintenance

- Focus on the conceptualization activity
  - Advantage: Integration of existing ontologies considered from early on
  - Conceptualization is evaluated early on, which prevents propagation of errors.

---

**On-To-Knowledge Methodology**

- Feasibility study
- Kick-off: describe domain and goals, design guidelines (e.g. naming conventions), available sources
- Refinement
  - Knowledge elicitation with domain experts
  - Formalization
- Evaluation
- Maintenance
DILIGENT

1. **Build**: a small team builds an ontology which is not required to be complete.

2. **Local adaption**: users adapt it to their own needs in their local environments.

3. **Analysis**: the board selects which changes go to the next version of the ontology.

4. **Revision**: carried out regularly to avoid that the ontology diverges too far from the local adaptations.

5. **Local** updates may be performed if the users wish to align their local ontologies with the new version of the ontology.

Ontology Maturing

1. **Emergence of ideas**. In the first phase, new concept ideas are collected in an ad-hoc fashion. This is done using simple tags.

2. **Consolidation in communities**. The concept symbols generated in the first phase are re-used and adapted by the user community. The phase aims at extracting concepts from the available tags leading to a common terminology.

3. **Formalization**. This phase adds taxonomic and ad-hoc relations to the common terminology yielding lightweight but formal ontologies.

4. **Axiomatization**. The final step in the methodology addresses knowledge workers, i.e. ontology engineers, to add axiom leading to a heavy-weight ontology.
**Holsapple and Joshi's approach**

1. Preparation phase defines design criteria, determines boundary conditions and determines standards that can then be used for evaluation.

2. Anchoring produces a first ontology that helps for orientation of the participants. This ontology is refined in several steps.

3. Iterative improvement is an adaptation of the Delphi method, a technique for collecting views of several stakeholders. This process is repeated until consensus is reached.

4. Application is the actual usage of the ontology.

---

**UPON**

1. Requirements workflow
   a) Determining the domain of interest and scope.
   b) Defining the purpose involves a description of the application of the ontology or a motivating scenario.
   c) Define storyboard and terminology.
   d) Creating an application lexicon can be carried out in a semi-automatic fashion that can be supported by automatic tools that extract knowledge from documents.
   e) Identifying the competency questions.
   f) Use case identification.

2. The analysis workflow depicts the conceptual analysis of the domain in order to derive ontology requirements and ontological commitments.
   a) Considering reuse of existing resources.
   b) Modeling the application scenario using UML diagrams.
   c) Building the glossary.

3. The design workflow refines results of previous steps and specifies relationships.

4. The implementation workflow leads to a serialized ontology in the preferred ontology representation formalism. The authors recommend OWL.

5. The test workflow validates the ontology by testing against the motivation scenario.
   a) identified the collaborators (team building), the knowledge workers
   b) discuss requirements, produce specification documents, and
   c) reach consensus on the scope and aim of the ontology.

2. Conceptualization first takes place in personal spaces addressing the following tasks:
   a) import of existing ontologies from ontology libraries,
   b) consult generic top ontologies for better understanding,
   c) improvising of ontologies, i.e. from-scratch-development, based on domain experts' views,
   d) management, mapping and merging of various ontology versions,
   e) comparison of available ontology versions,
   f) enriching ontology concepts with more information and specification details.

3. Exploitation. In this phase, the developed ontologies are pushed from the personal spaces to the shared space in order to reach a common understanding via structured conversation and criticism.
   a) inspection of agreed or shared ontologies by collaborators,
   b) comparisons of versions of one ontology in order to spot differences, and
   c) publication of comments and feedback.

How to build an ontology

Step 1: Determine the domain and scope of the ontology

- What is the domain that the ontology will cover?
- For what are we going to use the ontology?
- For what types of questions the information in the ontology should provide answers?
- Who will use and maintain the ontology?

Competency Questions

- A set of queries which place demands on the underlying ontology.
- Ontology must be able to represent the questions using its terminology and the answers based on the axioms.
- Ideally, in a staged manner, where consequent questions require the input from the preceding ones.
- A rationale for each competency question should be given.
Step 2: Consider reusing existing ontologies

- Reuse ensures interoperability and reduces costs
- Ontology libraries and tools for customization are required for this step
- Sub-steps
  - Discover potential reuse candidates
  - Evaluate their usability
  - Customize ontologies to be reused
  - Integrate and merge to the target ontology
- Dr. Watson (Open University, Knowledge Media Institute)
- SWOOGLE

Step 3: Enumerate important terms in the ontology

- 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?
Step 4: Define classes and class hierarchy

- A top-down development process starts with the definition of the most general concepts in the domain and subsequent specialization of the concepts.
- A bottom-up development process starts with the definition of the most specific classes, the leaves of the hierarchy, with subsequent grouping of these classes into more general concepts.
- Middle-out approach: define the more salient concepts first and then generalize and specialize them appropriately.

Step 4: Define classes and class hierarchy (ii)

- From the list created in Step 3, select the terms that describe objects having independent existence rather than terms that describe these objects.
  - These terms will be classes in the ontology.
- Organize the classes into a hierarchical taxonomy by asking if by being an instance of one class, the object will necessarily (i.e., by definition) be an instance of some other class.
  - If a class A is a superclass of class B, then every instance of B is also an instance of A.
- Classes as unary predicates—questions that have one argument. For example, “Is this object a wine?”
  - Later: binary predicates (or slots)—questions that have two arguments. For example, “Is the flavor of this object strong?” “What is the flavor of this object?”
How to find classes

- Interview: talk to subject matter experts.
- Documentation: read what experts have written about the subject matter, read the requirements documentation, read proposals and invitations to tender.
- Observation and reflection.

- Typical candidates for classes: NOUNS.
  - But: actors of use cases do not necessarily correspond to classes.

- Other terms as well
  - Gerund: „My eyes glazing over…” ~ withdrawal
  - Verbs: an association which starts to take on attributes and associations of its own turns into an entity: „Officer arrests suspect“.
  - Verbs: events: „Illness episode“.
  - Passive form: re-formulate in active form.

Class hierarchy

- A subclass of a class represents a concept that is a “kind of” the concept that the superclass represents.
- Classes represent concepts in the domain and not the words that denote these concepts. Synonyms for the same concept do not represent different classes.
- All the siblings in the hierarchy (except for the ones at the root) must be at the same level of generality.
- If a class has only one direct subclass there may be a modeling problem or the ontology is not complete.
- If there are more than a dozen subclasses for a given class then additional intermediate categories may be necessary.
- Subclasses of a class usually (1) have additional properties that the superclass does not have, or (2) restrictions different from those of the superclass, or (3) participate in different relationships than the superclasses.
Step 5: Define attributes and relationships

- Step 4 selected classes from the list of terms we created in Step 3.
  - Most of the remaining terms are likely to be properties of these classes.
  - For each property in the list, we must determine which class it describes.
- Types of properties
  - Attributes (Data properties, OWL)
  - Relationships (Object properties, OWL)
- Properties are inherited and should be attached to the most general class in the hierarchy.
Characterizing classes

- Two types of principal characteristics:
  - Measurable properties: attributes.
  - Inter-class connections: relationships.
  - Color of an image as attribute vs. class.

How to find attributes

- Interview: talk to subject matter experts.
- Documentation: read what experts have written about the subject matter, read the requirements documentation, read proposals and invitations to tender.
- Observation and reflection.
- Nouns in „-ness“
  - Velocity-ness, job-ness, arrested-ness...
- „How much, how many“ test.
  - If you evaluate this, then it is probably an attribute.
  - If you enumerate these, it is probably an entity.
Relationships

- Are defined on sets of instances.

- Properties: reflexivity, cardinality, many-to-many, all values from, some values of, transitivity, symmetry etc.

- Arity.

Examples

- Crime
- Suspect
- Person
- Vehicle
- Officer

0..1 0..* investgates
How to find relationships

• Interview: talk to subject matter experts.
• Documentation: read what experts have written about the subject matter, read the requirements documentation, read proposals and invitations to tender.
• Observation and reflection.

• Verbs, verbal phrases and things that could have been verbs.
  – “The butler murdered the duchess”

Step 6: Define the restrictions of the properties

• Refine the semantics of the properties
  – Cardinality
  – Domain and range
    • When defining a domain or a range for a slot, find the most general classes or class that can be respectively the domain or the range for the slots.
    • Do not define a domain and range that is overly general

• Example
  – Class Dog
  – Attribute Age
  – Relationship hasOwner
  – Dog hasAge Integer
  – Dog hasOwner Person
  – Domain: Dog in both cases
  – Range: XML data type: integer; Class Person
Step 7: Create instances

- Define an individual instance of a class requires
  - choose a class
  - create an individual instance of that class
  - filling in the values of the properties

Selected Tools
Selected Tools

- Protege
- Collaborative Protege
- myOntology
- OntoGame

Protege and Collaborative Protege
### Overview*

- Protege is a free, open-source platform to construct domain models and knowledge-based applications with ontologies.
- Go to [http://protege.stanford.edu/doc/owl/getting-started.html](http://protege.stanford.edu/doc/owl/getting-started.html) to download Protege (version 4.x)

* Based on the Protege OWL Tutorial on the Protege website

### Features

- Protege supports the following ways of OE:
  - Frame-based
  - OWL
- Protege Frames editor: enables users to build and populate ontologies that are frame-based, in accordance with ÖKBC (Open Knowledge Base Connectivity Protocol).
  - Classes
  - Slots for properties and relationships
  - Instances for class
- Protege OWL editor:
  - Classes
  - Properties
  - Instances
  - Axioms

* Based on the Protege OWL Tutorial on the Protege website
Properties

• Object properties
  – Domain: Class
  – Range: Class
  – E.g.: Dog hasOwner Person

• Datatype properties
  – Domain: Class
  – Range: Datatype
  – E.g. Dog hasBirthyear integer

• Property hierarchies

Individuals tab
Reasoning

- For an ontology that falls into the scope of OWL-DL, we can use a DL Reasoner to infer information that isn’t explicitly represented in the ontology. Standard ‘reasoning services’ are:
  - Subsumption checking
  - Equivalence checking
  - Consistency checking
  - Instantiation checking
Importing ontologies

- Re-use of ontologies
- Local or on the Web

Collaborative Protege

- Extension of Protege
- Supports:
  - Annotating ontology components and changes in the ontology
  - Discussion threads (live chat)
  - Proposal and voting
  - Searching and filtering
  - Defining users, groups, policies
- Works with Protege OWL and Frames
Collaborative Protege

Collaborative panel and tabs

myOntology

→ See Freebase
myOntology

- Community-driven ontology building
- Wiki principles are fundamental
- Combination of human and computational intelligence
- Easy-to-use
- Low entrance barriers
- Lightweight ontology
Create new ontology

Create new concept within ontology
Create new concept

Create new property
Create new property

Assign properties
Game play

- Consensus as underlying principle.
- 2 players in a team.
- Anonymously, randomly paired.
- They have to solve a task.
- They only get points when they provide consensual replies.
- Best strategy: say the truth.
- 2 types
  - Selection agreement games (most common → knowledge structures).
  - Output agreement games (cf. Von Ahn).

OntoPronto: Step 1
OntoTube

Final modeling guidelines
General issues

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

- Ontology development is necessarily an iterative process.

- Classes 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.

Domain analysis

- Ontology engineer needs to find a balance between learning about the domain without becoming a full domain expert.

- Talk to people in the organization who have to talk to experts, but are not experts themselves.

- Avoid diving into detailed, complicated theories unless their usefulness is proven.

- Construct a few typical scenarios which you understand at a global level.
### Purpose and scope of the ontology

- Be clear about why the ontology is being built and what its intended usages are
  - Interoperability between systems
  - Systems engineering
  - Semantic search, semantic annotation
  - Communication between people and organizations

- Example: semantic search
  - Semi-formal ontology
  - Usage of natural language labels and naming conventions
  - Well-balanced at schema and instance level
  - Rich conceptualization
  - Syntactical and semantic correctness

- The ontology should not contain all the possible information about the domain.

---

### Degree of formality

- The more formal an ontology is, the more does it exclude unwanted interpretations.
- Also, it includes the amount of inferences that can be drawn.
- However, it is more costly in terms of labor and engineering delay to create a more formal conceptualization.
- Also, achieving consensus is more difficult and time-consuming for a higher degree of formality.
- A higher degree of formality also imposes high entry barriers on the ones who are to model the respective domains and thus excludes potential contributors.
Further reading

- Ontology 101, Stanford University (Google)
- Presentation on ontologies
- Object-Oriented Analysis and Design (John Deacon, Addison-Wesley).
  - Chapters 5-8.
  - [http://www.johndeacon.net/OOAandD/index.asp#TutorLecturerResources](http://www.johndeacon.net/OOAandD/index.asp#TutorLecturerResources)

Assignment overview

- Assignment 2 due Dec 7 2009
- Assignment 3 due Dec 7 2009
- Assignment 4 due Jan 11 2010
Assignment 3 (7.12.2009)

- Read about Ontology 101, Stanford University (Google)
- Look at presentation on ontologies
- Take a look at Freebase
- SMW experiment
  - Finalize experiment with SMW
  - Fill in questionnaire

Assignment 4 (11.1.2010)

- Imagine an online portal for sharing cooking recipes. Examples of such portals are http://www.rezepte-ideen.de/, http://www.cuisine.at/ and many others. Develop an ontology for this domain following steps 1-7 of the Ontology 101 presented in the lecture.
- Prepare a presentation of your ontology following the steps of the ontology 101.
- Provide your ontology including documentation and download online. See example here: http://members.sti2.at/~katharinas/resume/