Introduction to ontologies and tools; some examples

Josep Blat, Jesús Ibáñez, Toni Navarrete
Universitat Pompeu Fabra

Definition and objectives

Definition: explicit formal specifications of the terms in the domain and relations among them
Goal: encoding knowledge to make it understandable to software agents searching for information (role of RDF for the Web). Common vocabulary
Another tool: DARPA Agent Markup Language (DAML) which extends RDF with more expressive constructs to facilitate agent interaction on the Web
Reasons for using ontologies (1)

- To share common understanding of the structure of information among people or software agents: re-use of data, mix of data, ... (pirineus?)
- To enable reuse of domain knowledge: re-use of knowledge, mix knowledge (time?)

Reasons for using ontologies (2)

- To make domain assumptions explicit: easier to validate, to change, ...
- To separate domain knowledge from the operational knowledge: re-use of knowledge in other domains
- To analyze domain knowledge
Ontologies in practice

- **Ontology** is a formal explicit description of:
  - Concepts in a domain: **classes**, or **concepts**
  - **Subclasses** represent concepts more specific than their superclasses
  - Properties of each concept describing features and attributes of the concept: **slots**, **roles** or **properties**
  - Restrictions on slots: **facets** or **role restrictions**
- A **knowledge base**: an ontology and a set of individual **instances** of classes

Ontologies in practice: a simple example

- **Classes** **Wine** (**subclasses** Red, White, Rosé) **Winery**
- Two **Slots** of **Wine**: Maker **Body**
- **Instance** of **Wine**: Château Lafitte Rothschild Pauillac
- **Slot** Maker Château Lafitte Rothschild
- **Slot** Body **full**
  - We say that the wine Château Lafitte Rothschild Pauillac is made by Château Lafitte Rothschild and has got a full body; remark that the maker is a winery (that is why the class winery was introduced)
Methodological steps in ontology development

- Step 1. Determine the domain and scope of the ontology
- Step 2. Consider reusing existing ontologies
- Step 3. Enumerate important terms in the ontology
- Step 4. Define the classes and the class hierarchy
- Step 5. Define the properties of classes—slots
- Step 6. Define the facets of the slots
- Step 7. Create instances

Step 1. Determine the domain and scope (1)

- Domain of the ontology. Example:
  - Representation of food and wines
- Application intended. Example:
  - Recommending good combinations of wines and foods
- Competency questions ontology should provide answers. Useful for testing, too. Examples:
  - Which wine characteristics should I consider when choosing a wine?
  - Is Bordeaux a red or white wine?
  - Does Cabernet Sauvignon go well with seafood?
  - What is the best choice of wine for grilled meat?
Step 1. Determine the domain and scope (2)

- Who will use and maintain the ontology? Different users. Example:
  - Source of terms, ... could come from journals of food and wine
  - Users could be professionals (chefs), restaurant customers
  - This might mean different languages, which should be appropriately mapped

Step 2. Consider reusing existing ontologies

- Re-use of languages, communication with other applications
- Ontolingua library
  http://www.ksl.stanford.edu/software/ontolingua/
- DAML library
  http://www.daml.org/ontologies/
- Other commercial ones
- Usually there are import-export tools
- Multilinguality?
Step 3. Enumerate important terms

- We suppose we do not re-use ontology
- Start by making a comprehensive list of terms without worrying about categorization in class, hierarchy, property, facet, overlapping ...
- Example:
  - Wine, grape, winery, location; wine’s color, body, flavor and sugar content
  - fish and red meat
  - subtypes of wine such as white, red, rosé
  - ...

Step 4. Define the classes and the class hierarchy

- Approaches:
  - Top down
  - Bottom up
  - Combined
- Usually: establish classes, check for hierarchy
- Example, a taxonomy of French wines:
  - Wine
  - Red wine, White wine, and Rosé wine
  - ...
  - Pauillac, Margaux (subclasses of Red Burgundy)
Step 5. Define properties of classes—slots (1)

- Properties define the internal structure of classes
- Slots will likely be words which are not classes, we must assign each to a class (the most general one; remark that subclasses of a class inherit the slots); properties can be
  - *Intrinsic* such as the flavor of a wine
  - *Extrinsic* such as a wine’s name, and area it comes from
  - *Parts* (physical or abstract) in a structured object
  - *Relationships* to other individuals
Example: slots (and facets) of the wine class

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Cardinality</th>
<th>Other Facets</th>
</tr>
</thead>
<tbody>
<tr>
<td>body</td>
<td>Symbol</td>
<td>single</td>
<td>allowed-values={FULL,MEDIUM,LIGHT}</td>
</tr>
<tr>
<td>color</td>
<td>Symbol</td>
<td>single</td>
<td>allowed-values={RED,ROSE,WHITE}</td>
</tr>
<tr>
<td>flavor</td>
<td>Symbol</td>
<td>single</td>
<td>allowed-values={DELICATE,MODERATE,STRONG}</td>
</tr>
<tr>
<td>grape</td>
<td>Instance</td>
<td>multiple</td>
<td>classes={Wine grape}</td>
</tr>
<tr>
<td>maker</td>
<td>Instance</td>
<td>single</td>
<td>classes={Winery}</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
<td>single</td>
<td></td>
</tr>
<tr>
<td>sugar</td>
<td>Symbol</td>
<td>single</td>
<td>allowed-values={DRY,SWEET,OFF-DRY}</td>
</tr>
</tbody>
</table>

Step 6. Define the facets of the slots

- **Common slots:**
  - **Cardinality** (e.g. body of wine has cardinality 1; produces of winery multiple values)
  - **Type:**
    - String
    - Number (e.g. price)
    - Boolean
    - Enumerated (lists)
    - Instance-type slots allow definition of relationships between individuals: allowed values are called range of the slot
  - Domain of a slot are the classes where the slot belongs to
Step 7. Create instances.

Example

Rules of thumb for ontology development

- Attach closely to the application intended (no ‘correct’ way)
- Develop the ontology iteratively
- *Concepts* likely to be **nouns**, and *relationships* **verbs** in sentences describing the domain
Further advanced questions

- Defining classes and a class hierarchy properly
- When to introduce a new class (or not)
- A new class or a property value?
- An instance or a class?
- ...

Ontology languages for the semantic web (I)

- RDF (Resource Description Framework)
  - It describes resources through triples <resource, property, value>
  - W3C recommendation
  - Good resources:
    - http://www.w3c.org/RDF/
    - Tutorial in xfront.com by Roger Costello: http://www.xfront.com/rdf-schema/ (currently unavailable)
    - RDF Primer: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/
Ontology languages for the semantic web (II)

- RDFS (RDF Schema)
  - Allows the definition of types of resources and properties
  - Taxonomies can be created through subClasses relationships
  - W3C recommendation
  - Good resources:
    - [http://www.w3c.org/RDF/](http://www.w3c.org/RDF/)

Ontology languages for the semantic web (III)

- OIL (Ontology Interchange Language or Layer) and DAML+OIL
  - Supports more powerful semantic primitives
  - OIL is superseded by DAML+OIL which is the base of OWL, the W3C standard
Ontology languages for the semantic web (IV)

- **OWL Web Ontology Language**
  - Much more powerful than RDFS
  - Supports subclasses, equivalence and disjointness among classes, definition of classes as intersection, union and complement of others, among other axioms.
  - There are also different types of properties
  - 3 profiles
    - Full, too complex for most reasoners
    - DL, based on Description Logics
    - Lite, quite reduced although still much richer than RDFS
  - Good resources:
    - WebOnt working group from W3C: [http://www.w3.org/2001/sw/WebOnt/](http://www.w3.org/2001/sw/WebOnt/)

Other Ontology languages

- Other mark-up languages:
  - SHOE, XOL,...

- Non-mark-up languages:
  - KIF – Knowledge Interchange Format- (and Ontolingua) based on frames and first order logic with Lisp-like syntax
  - LOOM based on DL
  - FLogic based on frames and first order logic but without Lisp-like syntax
  - OKBC
  - OCML
  - ...

Ontology development tools

- Protégé is a Java-based editor that works with RDF(S), DAML+OIL, OWL, and others. Many plugins available. Relatively easy to program new functionalities
  - Available at [http://protege.stanford.edu/](http://protege.stanford.edu/)
  - PROMPT is an ontology merging tool for Protégé
- OilEd is a Java-based editor for DAML+OIL
  - Available at [http://oiled.man.ac.uk/](http://oiled.man.ac.uk/)
- Ontolingua: the Ontolingua server has an on-line editor and other tools (including for merging)
- ...

Parser

- Jena is an API from HP that handles ontologies expressed in RDF(S), DAML+OIL and OWL (since version 2)
- It supports some reasoning mechanisms based on DL
- It is probably the most used parser for RDF and OWL. Protégé uses it for the OWL plugin
Reasoning with ontologies

- Propositional Logic
- First Order Logic
- **Description Logic**
  - Description Logics courses and tutorials
    - [http://dl.kr.org/courses.html](http://dl.kr.org/courses.html)
  - Book: Description Logic HandBook
    Edited by Franz Baader, Diego Calvanese, Deborah McGuinness, Daniele Nardi, Peter Patel-Schneider.
    PDF version at
    [http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-01.pdf](http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-01.pdf) (chapter 1)
    [http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-02.pdf](http://www.inf.unibz.it/~franconi/dl/course/dlhb/dlhb-02.pdf) (chapter 2)

Reasoners for DL

- FaCT
  - [http://www.cs.man.ac.uk/~horrocks/FaCT/](http://www.cs.man.ac.uk/~horrocks/FaCT/)
- Racer
  - [http://www.sts.tu-harburg.de/~7Er.f.moeller/racer/index.html](http://www.sts.tu-harburg.de/~7Er.f.moeller/racer/index.html)
- Protégé and OilEd can be connected to both
Motivation

- ICT for the *Pirineu Català*
- 10 years ago: Catalonia's most depressed area
- Current goal: ICT for
  - promoting economic development
  - providing services for locals
  - giving access to all information about the area
TOPICS

- information and portals: why and improvements
- information as service? New frontiers in service provision
- information retrieval invisible to the user
- techniques, tools, architecture for solution

A portal for the Pyrenees Web

- because of human factors
  - information rich and personalised to motivate not ICT-educated actors involvement
  - items in the portal to cater for a wide variety of users
  - clear personalization
A portal for the the Pyrenees Web

- because of the scope of the project
- access to wide variety of items of information
- enable communication amongst the actors
- becoming part of the community
- portal not only for information, but for intercommunication, for building a community.

A portal for different users / uses

- **locals**: virtual community with intercommunication and services
- **visitors**: information coming with services
- **experts**: thematic access, indexes, search
- alive (news, agenda)
- dynamic and participative (locals, visitors, experts ...)
Services?

- Information is a service (hotels in the area, medical info)
- Added services are added value (booking hotels, doctor's visit improves usefulness)
- New Web frontiers in service provision: Amazon (and how libraries could take advantage)

Searching / finding?

- A link repository is of great help
- Better: information served to the user (invisible information retrieval)
- Assume typical behaviour and help the information flow to the user (space, time are important clues, there might be others)
- Always complemented with useful services (delivered!!)
The goals from a technical point of view (summary)

- Access to heterogeneous information sources
- Heterogeneous input formats
- From heterogeneous locations
- To extract specific data
- Which have to be integrated in a single information node
- To be presented according to the user
- Taking into account the context of the user

The technical point of view

- **Heterogeneous information sources**
  - Databases
  - Static pages
  - Dynamic pages (cgi outputs)

- **Heterogeneous input formats**
  - HTML
  - Relational DBs
  - GIS DBs

- **Heterogeneous locations**
  - Local computers
  - LAN connected computers
  - Internet connected computers

- **To extract specific data**
  - Second picture in a page
  - List of answers from a search
  - Third column from a table in a HTML page
  - Two columns from three DB tables
The technical point of view

- To be uniquely integrated
  - the page shown to the user

- Taking into account the typology of the user
  - Expert
  - Local
  - Visitor

- And the user context
  - Space
  - Time
  - ...

Solutions

- For cataloguing information
  - XML for labeling information
  - Ontologies for capturing the semantics

- For information without pre-defined format and access mechanisms
  - Java application to check HTML pages, deciding what to access and where it is
  - Again XML labeling

- Separation of content and presentation, and multimodality
  - XML+XSL (different output formats: HTML, e-mails, SMS, WAP,...)
Solutions

"Real time" response
- Some things directly accessed (in the DB, in the page)
- Some things replicated with some periodicity

Mixing information sources
- Specialized Java libraries
- XML

Personalization by user typology
- Different information, different design (XSL)
- Some AI

Contextualization of information
- Information depending on space and time
- Some AI

Response time (adding all up)
- time of accessing the information * no. accesses + mixing time + contextualization time + connection delays + telecoms delays
The structure for content cataloguing (1)

- **Basic tables**
  - Those containing information (hotels, monuments,...)

- **Dictionary tables**
  - First metainformation level
  - Describing contents of the basic tables: names, columns (and types), relationships

The structure for content cataloguing (2)

- **Tables of semantic type relationships**
  - Defining the semantic relationships between different information elements

- **Semantic relationship tables**
  - Second metainformation level
  - Describe relationships between tables and columns
  - Example: advertising in terms of information requested
Pirineus: structure for the presentation

- Typologies and user profiles
  - Each user has a default profile, corresponding to his/her typology (local, tourist, expert) which defines which information is to be presented and how.
  - The user (in principle, only the local one) can modify these options.

- Sections
  - Each page is dynamically generated made up of sections (in turn made up of others). The user decides which sections are to be presented, and how.

Some tools used

- Apache
  - Web Server
  - JServ
  - Cocoon
    - XSP
    - XQL
- DB2XML
- XML (Xerces) and XSLT (Xalan) Parsers
- The spider
Global diagram

Architecture
A second example: virtual worlds on the web

Motivating question

Find virtual worlds, zones and objects in them, help navigation in these worlds
E.g.:
Classical question: find on the web based on a comment “I’ve seen a page talking about Bernie Roehl’s new book on VR”
versus
New question: what to do if we are told “I’ve seen a virtual word with a red VW parked in front of a library”

Application: furniture for a house
Characteristics of the problem (1)

- Access from natural language to visual information? differences in representation? shared representation necessary
- Structure of objects (e.g.: several objects together make up a new one)? shared common classification pattern

Characteristics of the problem (2)

- Visual nature of the information (color tones, sizes, textures,...)? imprecise information
- Concepts built on top of others (good weather beach?)? fuzzy inference
Information building (1)

- Define ontologies (facts, rules)
  - Spatial relationships
  - Temporal relationships
  - Furniture composition
  - Fuzzy relationships
  - Agents’ communication
- Labeling worlds (db, xml)
- Get information from worlds

Information building (2)

- Induce knowledge
  - fuzzy ilp
  - mechanisms for inducing temporal relationships
- Refining rules (genetic algorithms)
- And we have the info !! (for some time)
The user

- Multimodal user interface
  - Graphical interface
  - Natural language
- Understanding requests
- Translation into the shared format
- Build the user profile
- ... (Agents and more)