0.1 Upper ontologies and ontology matching

0.1.1 Upper ontologies

Basics

What are upper ontologies?

- **Upper ontologies** (sometimes also called *top-level* or *foundational* ontologies) describe very general concepts that are independent of particular problem or domain.

- They provide categories of kinds of things and relations that can provide a basic structure for "lower-level" ontologies such as domain ontology.

Why should we use upper ontologies?

- **Pros:**
  - "top-down approach" and modelling guidance for ontology development
  - basic categories and relations that we don’t need to reinvent again
  - interoperability among ontologies

- **Cons:**
  - a lot of effort needed to understand
  - too abstract

Basic ontological commitments

- Universals vs. Particulars – Universals can have instances, while Particulars don’t

- Descriptive vs. Realist – represent world using natural language and common sense vs. represent it as is

- Multiplicative vs. Reductionist – different objects can be co-located at the same time vs. only one object may be located at the same region at one time

- Endurantism vs. Perdurantism – an object is wholly present at all times vs. an object has temporal parts

- Actualism vs. Possibilism – everything that exists in the ontology is real vs. objects are allowed independent of their actual existence

- Concrete & Abstract entities – entities that exist in space and time & entities that exist neither in space nor time
Overview of upper ontologies

Existing upper ontologies

- DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering)
- BFO (Basic Formal Ontology)
- GFO (General Formal Ontology)
- YOMATO (Yet Another More Advanced Top-level Ontology)
- UFO (Unified Foundational Ontology)
- PROTON (PROTo ONtology)
- SUMO (Suggested Upper Merged Ontology)
- Cyc
- WordNet

Comparison of ontological commitments

<table>
<thead>
<tr>
<th>Term and meaning</th>
<th>DOLCE</th>
<th>BFO</th>
<th>GFO</th>
<th>SUMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ontological Commitments</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Descriptive vs. Realist</td>
<td>Descriptive</td>
<td>Realist</td>
<td>Descriptive and Realist</td>
<td>Descriptive</td>
</tr>
<tr>
<td>Universal vs. Particular</td>
<td>Universal</td>
<td>Universals and Particulars</td>
<td>Universals and Particulars</td>
<td>Universals and Particulars</td>
</tr>
<tr>
<td>Multiplicative vs. Reductionist</td>
<td>Multiplicative</td>
<td>Reductionist</td>
<td>Unclear</td>
<td>Multiplicative</td>
</tr>
<tr>
<td>Endurant vs. Perdurant</td>
<td>Endurant</td>
<td>Endurant and Perdurant</td>
<td>Endurant and Perdurant</td>
<td>Endurant and Perdurant</td>
</tr>
<tr>
<td>Actuality vs. Possibility</td>
<td>Actuality</td>
<td>Unclar</td>
<td>Unclar</td>
<td>Unclar</td>
</tr>
<tr>
<td>Eternalism vs. Pluralism</td>
<td>Eternalist</td>
<td>Eternalist</td>
<td>Eternalist</td>
<td>Eternalist</td>
</tr>
<tr>
<td>Concrete &amp; abstract entities</td>
<td>Concrete and Abstract</td>
<td>Concrete</td>
<td>Concrete and Abstract</td>
<td>Concrete and Abstract</td>
</tr>
<tr>
<td>Temporal aspects</td>
<td>Provided</td>
<td>Not provided</td>
<td>Provided</td>
<td>Provided</td>
</tr>
<tr>
<td>Granddaddy vs. Concrete (different levels of detail contained in an ontology)</td>
<td>High level</td>
<td>Sensitive</td>
<td>Unclear</td>
<td>Unclear</td>
</tr>
<tr>
<td>Properties and relations (attributes)</td>
<td>Included</td>
<td>Some support</td>
<td>Included</td>
<td>Included</td>
</tr>
<tr>
<td>World for space and time (Concords of time and space regions and boundaries)</td>
<td>Not included</td>
<td>Included</td>
<td>Not included</td>
<td>Not included</td>
</tr>
<tr>
<td>One-layered vs. Three-layered architecture (a basic level only, an abstract top level, abstract core level and basic level)</td>
<td>One-layered</td>
<td>One-layered</td>
<td>Three-layered</td>
<td>One-layered</td>
</tr>
<tr>
<td>Situations and worlds (an aggregate of facts that may be comprehended as a whole and without certain conditions of time; Situations is a part of the world and a world is a part of the reality)</td>
<td>Not included</td>
<td>Not included</td>
<td>Included</td>
<td>Not included</td>
</tr>
</tbody>
</table>

Comparison of ontological commitments within selected upper ontologies taken from [http://www.thezfiles.co.za/ROMULUS/ontologicalCommitments.html](http://www.thezfiles.co.za/ROMULUS/ontologicalCommitments.html)
DOLCE overview

- **Descriptive Ontology for Linguistic and Cognitive Engineering**
- developed by researchers from the Laboratory of Applied Ontology, headed by N. Guarino
- first module of the WonderWeb Foundational Ontologies Library
- ontology of particulars, multiplicative, possibilism
- strong cognitive/linguistic bias – descriptive attitude with categories mirroring cognition, common sense, and the lexical structure of natural language

DOLCE’s taxonomy of basic categories

DOLCE basic relations

- parthood (immediate and temporary)
- constitution
- participation
- representation
- specific/generic constant dependence
- inherence (between a quality and its host)
- quale (immediate and temporary)
DOLCE’s primitive relations between basic categories

DOLCE’s relations about qualities

BFO overview

- Basic Formal Ontology
- developed in Saarland University mainly by B. Smith, P. Grenon
0.1 Upper ontologies and ontology matching

- designed for use in supporting information retrieval, analysis and integration in scientific and other domains
- realistic and reductionist view of the word, actualism
- limited granularity
- contains both SNAP (endurants) and SPAN (perdurants) sub-ontologies

**BFO’s taxonomy of basic categories**

**BFO’s realizable entity example**
0.1.2 Ontology matching

Why do we need ontology matching?

- One of the main challenges of semantic technologies is to solve problem of managing semantic heterogeneity among various information sources
- This heterogeneity introduces variations in meaning as well as ambiguity in entity interpretations
- Ontology matching is semantic technology that focus to solve this issue by automating integration of distributed information sources

Motivating example

Simple example of matching, taken from [1], between two different information sources: 1) ontology $O_1$ on the left side, 2) ontology $O_2$ on the right side. Classes are shown in rectangles with round corners (e.g. Book being subclass of Product), while relations are not bordered (e.g. price is an integer data type, while creator is an object property). Albert Camus: *L’Étranger* is a shared instance between the ontologies. Correspondence between entities of the ontologies are shown by thick blue arrows annotated with the relation that express it (e.g. Person in $O_1$ is subclass of Human from $O_2$, which is marked by ⊑ symbol).

Basics

Definitions of core concepts (1)

- Given 2 ontologies, a correspondence is a 4-tuple $(id, e_1, e_2, r)$, where $id$ is identifier for correspondence, $e_1$ and $e_2$ are entities, e.g. classes and properties of the first and second ontology respectively, $r$ is a relation holding between $e_1$ and $e_2$, e.g. equivalence, generalization/specialization, disjointness. An example of correspondence expressing that Book in $O_1$ is more general that Monograph in $O_2$ would be tuple $(id_{3,4}, \text{Book}, \text{Monograph}, \sqsubseteq)$.
- Correspondence can have some associated metadata such as author name or confidence in correspondence typically expressed by a number within the range [0,1].
0.1 Upper ontologies and ontology matching

- **Alignment** is set of correspondences between entities belonging to the examined ontologies. Alignments are of cardinality 1:1, n:1 or n:m.

Definitions of core concepts (2)

- **Matching** is an operation that determines an alignment $A'$ for a pair of ontologies $O1$ and $O2$.

- Within context of matching it is usually referred to terms:
  - matching operation – focusing on the input and the result
  - matching task – focusing on the goal and the insertion of the task in a wider context
  - matching process – focusing on the internals of the task.

- Matching operation can be defined by use of: (i) an input alignment $A$, which is to be extended, (ii) the matching parameters, such as weights, or thresholds, and (iii) external resources, such as common knowledge or thesauri.

Overview of matching systems

Comparison of matching systems
Analytical comparison of matching systems taken from [1]. The first half of the table shows the system name, the input format, cardinality of output alignments, weather GUI is provided, the ways in which system can process alignments, respectively. The second half of the table shows the available matching methods classified by kind of data that the algorithm work with.

### Applications

#### Applications of matching systems

The ontology matching systems are used within two scenarios:

- **design-time matching**
  - used in traditional applications, characterized by heterogeneous models (e.g. ontologies or database schemas), such as ontology evolution, ontology integration, data integration, data warehouses.
  - analysis and matching is done manually or semi-automatically

- **run-time matching**
  - used in emerging applications, characterized by dynamics, such as peer-to-peer information sharing, web service composition, search and query answering.
  - matching is done usually automatically with use of more explicit conceptual models compared to traditional applications.

### References