Bottom-up overview

Relational databases

Models in biology

Thesauri

Outline

Bottom-up overview

Relational databases

Data analysis

Automatic Extraction of Ontologies

Example: manual extraction

Models in biology

General idea

Case study

Thesauri

Bottom-up

- From some seemingly suitable legacy representation to an OWL ontology
  - Database reverse engineering
  - Conceptual model (ER, UML)
  - Frame-based system
  - OBO format
  - Thesauri
  - Formalizing biological models
  - Excel sheets
  - Text mining, machine learning, clustering
  - etc...

A few languages
Levels of ontological precision

(from Gangemi, 2004)

Examples: OBO and Protégé-frames

• Frame (as in Protégé) into OWL-DL (see Zhang & Bodenreider, 2004), and its problems doing that to the FMA
  • Not a formal transformation
  • Slot values generally correspond to necessary conditions—so they took a first guess to define an anatomical entity as the sum of its parts
  • Global axioms dropped (with an eye on the reasoner)
  • After the conversion of the 39,337 classes and 187 slots from FMA in Protégé (ignoring laterality distinctions), FMAinOWL contains 39,337 classes, 187 properties and 85 individuals
  • Additional optimizations: optimizing domains and subClassOf axioms
  • But still caused Racer to fail to reason over the whole file; restricting properties further obtained results

General considerations

• Let us for a moment ignore the issues of data duplication, violations of integrity constraints, hacks, outdated imports from other databases to fill a boutique database, outdated conceptual data models (if there was one), and what have you
• Some data in the DB—mathematically instances—actually assumed to be concepts/universals/classes
• each tuple is assumed to denote an instance and, by virtue of key definitions, to be unique in that table, but such a tuple has values in each cell of the participating columns; however, OWL ABox expects objects (impedance mismatch)
• instances-but-actually-concepts-that-should-become-OWL-classes and real-instances-that-should-become-OWL-instances
General considerations

- Reuse/reverse engineer the physical DB schema
- Reuse conceptual data model (in ER, EER, UML, ORM, ...)
- But,
  - Assumes there was a fully normalised conceptual data model,
  - Denormalization steps to flatten the database structure, which, if simply reverse engineered, ends up in the ontology as a class with umpteen attributes
  - Minimal (if at all) automated reasoning with it
- Redo the normalization steps to try to get some structure back into the conceptual view of the data?
- Add a section of another ontology to brighten up the ‘ontology’ into an ontology?
- Establish some mechanism to keep a ‘link’ between the terms on the ontology and the source in the database?

Manual Extraction

- Most database are not neat as assumed in the ‘Automatic Extraction of Ontologies’ (e.g., denormalised)
- Then what?
  - Reverse engineer the database to a conceptual data model
  - Choose an ontology language for your purpose
- Example: the HGT-DB about horizontal gene transfer (the same holds for the database behind ADOLENA)
Manual mapping to $DL$-$Lite_A$

- Basic statistics:
  - 38 classes
  - 34 object properties of which 17 functional
  - 55 data properties of which 47 functional
  - 102 subclass axioms

- Subsequently used for Ontology-Based Data Access (more about that in the next block)

Overview

- Pure and applied life sciences use many diagrams
- Some diagram hand drawn, but more and more with software
- Come with their own ‘icon vocabulary’ and many diagrams
- Exploit such informal but structured representation of information to develop automatically (a preliminary version of) a domain ontology
- Formalize the ‘icon vocabulary’ in a suitable logic language, choose a foundational ontology (taxonomy, relations), categorise the formalised icons accordingly, load each diagram into the ontology, verify with the domain expert

Example of a PathwayAssist diagram

Motivation

- Experiment in 2005 (Keet, 2005), but progress made in ecology (Madin et al, 2008; MTSR’09 proceedings)
- Extensive use of modelling in ecology, but not much shared (depending on sub-discipline)
- Models used with independent software tools (DB and other applications)
- ‘Legacy code’ (procedural), moving toward more OO, and ontologies
- Requirement for (re re-)analysis to upgrade legacy SW), develop new SW to meet increasing, complexities and rising demands.
- use the opportunity to create a more durable, yet computationally usable, shared, agreed upon representation of the knowledge about reality

Key aspects in the ecological model: Flow, Stock, Converter, Action Connector

Informal ‘Translation’

- A Stock correspond to a noun (particular or universal)
- Flow to verb
- Converter to attribute related to Flow or Stock
- Action Connector relates the former
- Object is candidate for an Endurant
- Event_or_activity for a method or Perdurant
- Converter maps to Attribute_or_property
- Action Connector candidate for relationship between any two of Flow, Stock and Converter
‘Translation’ w.r.t. DOLCE categories

- Basic mapping to DOLCE categories:
  - $\forall x ((\text{Stock}(x) \leftrightarrow \text{Entity}(x)) \rightarrow \text{ED}(x))$
  - $\forall x ((\text{Flow}(x) \leftrightarrow \text{Entity}(x)) \rightarrow \text{PD}(x))$
  - $\forall x ((\text{Converter}(x) \leftrightarrow \text{Entity}(x)) \rightarrow (Q(x) \lor \text{ST}(x)))$
  - $\forall x (\text{ActionConnector}(x, y) \rightarrow \text{Relationship}(x, y))$

ML to Microbial Loop domain ontology

- Aim: to test translations with a real STELLA model
- ML’s initial mapping to ontological categories contain 38 STELLA elements: 11 Stock/ED, 21 Flow/PD, 2 Converters/ST, 4 Action Connectors/Relationships
- The MicrobialLoop ontology has 59 classes and 10 properties
- Increase due to including DOLCE categories and implicit knowledge of ML that is explicit in MicrobialLoop
Section of more refined mapping to DOCLE categories

<table>
<thead>
<tr>
<th>Phyto C</th>
<th>NAPO</th>
<th>Phyto C = phytoplankton organic carbon. Phytoplankton is an APO, but 'phyto C' is part of the APO: only the organic carbon of the phytoplankton, not the organism as an active agent as such</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phyto N</td>
<td>NAPO</td>
<td>Phyto N = phytoplankton nitrogen</td>
</tr>
<tr>
<td>DOC</td>
<td>NAPO</td>
<td>DOC = dissolved organic carbon. DOC is an ED with no unity, thus an amount of matter (M), but here, like with the organisms, there is focus on only a part of the NAPO</td>
</tr>
<tr>
<td>Nitrate</td>
<td>NAPO</td>
<td>Dissolved nitrate. Molecules are non-aggregative physical objects.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flow</th>
<th>PRO</th>
<th>To phytoplankton N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Respiration</td>
<td>PRO</td>
<td>From phytoplankton N</td>
</tr>
<tr>
<td>Prot gr bac</td>
<td>PRO</td>
<td>Prototrozoa that are grazing on the Bacterial C</td>
</tr>
</tbody>
</table>

**Grazing pressure**

- **ST** acts on a PRO affecting the process of grazing: 'grazing pressure' is there (might reach zero), hence a ST.
- **T** acts on the mesozooplankton grazing on the protozoa, and acts on the mesozooplankton grazing on the phytoplankton: relation hasGrazingPressure

more mappings at http://www.meteck.org/supplDILS.html

---

**Discussion**

- **Formalising ecological natural, functional and integrative concepts**
- aids comparison of scientific theories
- makes the implicit explicit, and more expressive than other modelling practices, therefore useful:
  - points to ambiguous sections,
  - part of/extra tool for doing science,
  - importance ontology maintenance, comparisons
- **Modular, backbone or all-encompassing ontology/ies**
- With the mappings, a quicker bottom-up development of ecological ontologies
To summarize

- Taxonomies insufficiently expressive compared to existing ecological modelling techniques
- Perspective of flow in ecological models cannot be represented adequately in a taxonomy
- More comprehensive semantics of formal ontologies
- Formalised mapping between STELLA and ontology elements facilitates bottom-up ontology development and has excellent potential for semi-automated ontology development
- STELLA as intermediate representation, widely used by ecologists and is translatable to a representation usable for ontologists

Overview

- Thesauri galore in medicine, education, agriculture, ...
- Core notions of BT broader term, NT narrower term, and RT related term (and auxiliary ones UF/USE)
- E.g. the Educational Resources Information Center thesaurus:
  - reading ability
  - BT ability
  - RT reading
  - RT perception
- E.g. AGROVOC of the FAO:
  - milk
  - NT cow milk
  - NT milk fat
- How to go from this to an ontology?

Problems

- Lexicalisation of a conceptualisation
- Low ontological precision
- BT/NT is not the same as is_a, RT can be any type of relation: overloaded with (ambiguous) subject domain semantics
- Those relationships are used inconsistently
- Lacks basic categories alike those in DOLCE and BFO (ED, PD, SDC, etc.)

A rules-as-you-go approach

- A possible re-engineering procedure:
  - Define the ontology structure (top-level hierarchy/backbone)
  - Fill in values from one or more legacy Knowledge Organisation System to the extent possible (such as: which object properties?)
  - Edit manually using an ontology editor:
    - make existing information more precise
    - add new information
    - automation of discovered patterns (rules-as-you-go)

see (Soergel et al, 2004)
A rules-as-you-go approach

- A possible re-engineering procedure:
  - Define the ontology structure (top-level hierarchy/backbone)
  - Fill in values from one or more legacy Knowledge Organisation System to the extent possible (such as: which object properties?)
  - Edit manually using an ontology editor:
    - make existing information more precise
    - add new information
    - automation of discovered patterns (rules-as-you-go); e.g.
      - observation: cow NT cow milk should become cow
        <hasComponent> cow milk
      - pattern: animal <hasComponent> milk (or, more generally animal <hasComponent> body part)
        — derive automatically: goat NT goat milk should become goat <hasComponent> goat milk
        other pattern examples, e.g., plant <growsIn> soil type and geographical entity <spatiallyIncludedIn> geographical entity

see (Soergel et al, 2004)