Modularisation & Normalisation of Ontologie

Alan Rector & Julian Seidenberg
School of Computer Science, University of Manchester
Northwest institute of BioHealth Informatics
rector@cs.man.ac.uk
www.clinical-esciences.org
www.co-ode.org

with thanks to the protégé team
Why modularise an ontology?

- **Manageability**
  - Independent development of modules
- **Re-use**
  - Most applications require only part of another ontology
    - How many applications need 65K anatomical entities
- **Abstraction**
  - Use only as much detail as is needed
- **Evolution, Maintenance & tailoring**
  - Controlled management of changes
  - Replace one variant with another
    - Hue-saturation-brightness vs Red-Blue-Green for colour
Loosely coupled distributed ‘just in time’ ontology development

local cycles: work by users

global cycle & work at centre
Referencing, mapping & importing

• Mapping
  – Pointers via annotations only

• Referencing
  – Common identifiers only
    • OWL: URI as a unique reference

• Referencing with partial import
  – Upwards taxonomy?

• Importing
  – Inclusion of all information
Referencing

- Common ID / Controlled vocabulary
  - Need the reference but not the structure
    - If everyone uses the same reference, we can know if we agree or disagree
      - May not need to merge
      - Or may make later merging easier
Importing / Inclusion
True modularisation:

• The rest of this talk
  – Important practicalities
  – Ontology structure and Normalisation
Important practicalities

“Nonsemantic Identifiers”

• Lexical names make lousy identifiers
  – Spelling errors
  – Arguments over ‘terms’
  – Changes in usage
  – Variation between sites

• Meaningless identifiers much safer
  – Almost impossible to edit a set of coherent modularised ontologies using names to link
    • Made particularly hard by the use of URIswww.
important practicalities

• Namespaces vs Base URIs/Modules
  – Namespaces are often used to indicate modules
    • … but do not depend on them

• Local copies vs web sources
  – Need local redirection and ‘repositories’
    • Still working on this in Protégé OWL
      – Let us know if it works well.

• Notion of a “package”
  – Still ill defined
Other work

- Swoop circle diagrams
- Oscar Corcho and KnowledgeWeb deliverables
Swoop - “Fly the mother ship”
Ontology Structure

Assertion:

*The arrival of logic-based ontologies/OWL gives new opportunities to make ontologies more manageable and modular*

- Let the ontology authors
  - create discrete modules
  - describe the links between modules
- Let the logic reasoner
  - Organise the result
Two dimensions for modularisation

- Segmentation
  - Drugs, anatomy, diseases,
  - Micro and macro anatomy
  - ...
Some scenarios: Segmentation

- Re-use of parts of a high level ontology
  - Self-standing vs modifiers
    - Continuants vs Occurrents
      - Parts and whole - mereology
        » Biology
          * Core Anatomy
          * Gross anatomy
            * Organs
            * Vessels
            * Limbs
            * Organs + vessels
            * Limbs + vessels
            ...
          * Cellular
Two dimensions for modularisation

• Abstraction - specialisation
  – Vertebrate anatomy, mammalian anatomy, human anatomy
    • What is common?
  – hand
    • left, right hand
  – Vertebra
    • cervical (neck…)
      – 2nd, 3rd, 4th..
    • thoracic (chest)
      – 8th, 9th, 10th…
  – Specialisation of guideline for my hospital
Segmentation & locking
Primitive & defined classes

- **Primitive classes**
  - Cheese, Mozarella, Pizza,
    - Cannot be split
      - Must be in one module

- **Defined classes**
  - Cheesy Pizza, Vegetarian Pizza
    - Span modules
      - Glue modules together
      - “Come apart” when classes are split
        - Often collapse logically
        - Can’t tell them apart without the ontology referenced
  
  - **But need to know which module**
    - Therefore want them to have just one primitive parent

- **Additional descriptions**
  - References between classes
    - Cheese comes from Milk
      - Milk is new extra information
Reasons normalise an ontology?

• Modularisation, Evolution and Maintenance
  – The general modularisation problem for expressive ontologies is intractable
    • Need additional constraints
    • Ease of defining mutually exclusive segments
  – Ease maintenance behaviour
    • 1 change in one place
      – No “side effects” or “update anomalies”

• Defeat combinatorial explosions
  – Reduce an exponential structure to two linear structures
    • “Just in time” ontologies

• A first step towards comprehensive meta models and ontology schemas
The scaling problem: The combinatorial explosion

- It keeps happening!
  - “Simple” brute force solutions do not scale up!

- Conditions x sites x modifiers x activity x context →
  - Huge number of terms to author
  - Software CHAOS
Combination of things to be done & time to do each thing

- Terms and forms needed
  - Increases exponentially

- Effort per term or form
  - Must decrease to compensate

- To give the effectiveness we want
  - Or might accept
The exploding bicycle

- 1972 ICD-9 (E826) 8
- READ-2 (T30..) 81
- READ-3 87
- 1999 ICD-10 ...
1999 ICD10: 587 codes

• V31.22 Occupant of three-wheeled motor vehicle injured in collision with pedal cycle, person on outside of vehicle, nontraffic accident, while working for income

• W65.40 Drowning and submersion while in bath-tub, street and highway, while engaged in sports activity

• X35.44 Victim of volcanic eruption, street and highway, while resting, sleeping, eating or engaging in other vital activities
Evolution Maintenance & tailoring

• What might a change affect?
  – What needs to be locked?
• What has to be changed to make a variation?
  – What can be “swapped out” and replaced?
Logic-based Ontologies: Conceptual Lego

- hand
- extremity
- body
- chronic
- acute
- abnormal
- normal
- ischaemic
- deletion
- polymorphism
- gene
- protein
- cell
- expression
- Lung
- inflammation
- infection
- bacterial
Logic-based Ontologies: Conceptual Lego

“SNPolymorphism of CFTRGene causing Defect in MembraneTransport of Chloride Ion causing Increase in Viscosity of Mucus in CysticFibrosis…”

“Hand which is anatomically normal”
Normalisation for Segmentation: Logical Constructs build complex concepts from modularised primitives
Normalising (untangling) Ontologies

Structure

Part-whole

Function
Normalisation and Untangling
Let the reasoner do multiple classification

- **Tree**
  - Everything has just one parent
    - A ‘strict hierarchy’
- **Directed Acyclic Graph (DAG)**
  - Things can have multiple parents
    - A ‘Polyhierarch’
- **Normalisation**
  - Separate primitives into disjoint trees
  - Link the trees with restrictions
    - Fill in the values
Untangling and Enrichment
Using a classifier to make life easier

Substance
- Protein
  - ProteinHormone
  - Insulin
- Steroid
  - SteroidHormone
  - Cortisol
- Hormone
  - ProteinHormone
  - Insulin
  - SteroidHormone
  - Cortisol
- Catalyst
- Enzyme
- ATPase
Most normalisation results in three potential modules: *two primitive skeletons*.
And an interface

Definitions

Plus extra description
Unified ontology after classification
Consider the steps to make a change

- What do we have to do to organise hormones as metabolic hormones and sex hormones and stress hormones and add in testosterone?
A Few of the changes the hard way

<table>
<thead>
<tr>
<th>Substance</th>
<th>- Protein</th>
</tr>
</thead>
<tbody>
<tr>
<td>- ProteinHormone</td>
<td>- - Insulin</td>
</tr>
<tr>
<td>- Steroid</td>
<td>- - Cortisol</td>
</tr>
<tr>
<td>- SteroidHormone</td>
<td>- - Hormone</td>
</tr>
<tr>
<td>- - Sex Hormone</td>
<td>- - ProteinSexHrmn</td>
</tr>
<tr>
<td>- ProteinHormone</td>
<td>- - ProteinMetabolicHrmn</td>
</tr>
<tr>
<td>- - Insulin</td>
<td>- - Sex Hormone</td>
</tr>
<tr>
<td>- - ProteinSexHormone</td>
<td>- - SteroidHormone</td>
</tr>
<tr>
<td>- - Cortisol</td>
<td>- - Catalyst</td>
</tr>
<tr>
<td>- Catalyst</td>
<td>- - Enzyme</td>
</tr>
<tr>
<td>- Enzyme</td>
<td>- - ATPase</td>
</tr>
</tbody>
</table>
The easy way - and only create the ones I need

- Substance
  - Protein
  - Insulin
  - ATPase
  - Steroid
  - Cortisol
  - Testosterone

- PhsioloicRole
  - HormoneRole
  - SexHR
  - MetabolicHR
  - StressHR
  - CatalystRole

Substance
  - Protein
  - Insulin
  - Enzyme
  - ATPase
  - Steroid
  - SteroidHormone
  - Cortisol
  - ProteinHormone
  - Insulin
  - SteroidHormone
  - SexHR
  - Testosterone
  - StressHormone
  - Cortisol
  - MetabolicHrmn
  - Catalyst
  - Enzyme
  - ATPase
Changes to Structure

& Roles
Update the overall structure
stress roles are metabolic roles: One change
Another Example

From

Diameter
  Large_diameter
    Large_increasing_diameter
    Large_decreasing_diameter
  Small_diameter
    Small_increasing_diameter
    Small_decreasing_diameter

To add regular/irregular we get

Diameter
  Large_diameter
    Large_increasing_diameter
    Large_increasing_regular_diameter
    Large_increasing_irregular_diameter
  Large_decreasing_diameter
    Large_decreasing_regular_diameter
    Large_decreasing_irregular_diameter
  Small_diameter
    Small_increasing_diameter
    Small_increasing_regular_diameter
    Small_increasing_irregular_diameter
    Small_decreasing_diameter
    Small_decreasing_regular_diameter
    Small_decreasing_irregular_diameter
Instead of

Which would you rather maintain?

Level
- Large
- Small

Trend
- Increasing
- Decreasing

Regularity
- Regular
- Irregular
Summary

• Modularisation for
  – Manageability
  – Locking and collaborative development
  – Adaptation and Localisation

• Two dimensions:
  – Abstraction & Segmentation

• Normalisation
  – Clean modularisation
  – Control the combinatorial explosion
  – Support smooth evolution
  • Changes only in one place