
Ontology Views for Collaborative Ontology Creation: The BioSphere Portal

Stuart Aitken¹ and Jonathan Bard²

**¹ Informatics, ² Biomedical Science
University of Edinburgh**

The BioSphere Ontology Portal

BioSphere: A BBSRC tools and resources project developing tools for the end user

- Collaborative ontology development tools

1. Ontologies

- Bio-ontology languages: OBO and tools
- W3C standards: Web Ontology Language (OWL)
- OBO => OWL
- OWL syntax (RDF/XML) and OWL 2

2. Archiving XML data

- Keys for XML

3. BioSphere Portal

- GridSphere and OGSA-DAI

Bio-ontologies

- **Taxonomies of organisms (NCBI)**

 - Cellular Organisms**

 - Eukaryota ...**

 - Rodentia ...**

 - Mus musculus (house mouse)**

- **Anatomy ontologies**

 - **Mouse, fly, worm and plant anatomies used for indexing spatial and temporal gene expression data**
 - **Tissues and regions are named and given IDs**
 - » **IDs used in many databases**
 - » **Mapped to other ID schemes**
 - » **interoperability**

Bio-ontologies

- **Gene Ontology (GO)**
 - Used for describing the molecular of gene products
 - Extensively adopted
 - Has enabled many new data mining and statistical analyses
 - The GO ontology representation is also used by the Open Biomedical Ontologies (OBO) community
 - » *obofoundry.org*

- **Ontologies complement other standards initiatives**
 - Standards for data formats and protocols

Bio-ontologies

- **Annotation**

- PAX6 is linked to GO term **camera-type eye development (GO:00430010)** in the mouse MGI database
- PAX6 is detected in the mouse **eye** (or substructures)
 - » data also found in the mouse MGI database
- Mouse anatomy is hierarchically organised, by part-of and by stage
 - » **lens** is a substructure of **eye**
 - » **eye** and lens have EMAPA identifiers
 - eye** has ID EMAPA:16192
 - » Identifiers serve important roles in the databases
- A biologist can then look for genes in other organisms annotated to **camera-type eye development**

Bio-ontologies: Syntax

- **GO flat file format**
 - ASCII rendering of the hierarchy using % (isa) and < (part-of)
 - indentation indicates parent-child relationships
 - **OBO flat file format**
 - uses a ‘frame-like’ representation, organising defining information about the class as a series of assertions, or ‘stanza’
 - Overcomes some limitations of GO
 - » Allows new relationships to be created
 - Recent versions introduce intersection, union and restrictions
 - **XML versions of GO and OBO are defined but not as extensively used**
 - **A Web Ontology Language encoding of OBO has been defined**
-

Bio-ontologies: Semantics

- **isa**
 - Always used in the sub-class-of sense [GO/OBO]
- **part-of**
 - Initial ambiguities have been clarified
 - Still overloaded (functional vs component etc)
 - Links classes: Heart *part-of* Cardiovascular System
 - » Some Hearts ? All Hearts? Some/All Cardiovascular Systems?
 - Comments apply both to GO and OBO
 - » needs clarification for translation to OWL
- **Other relations treated in the same way as part-of**
 - class-level relations in GO/OBO

Bio-ontologies: Annotations

Annotations to terms (classes)

- **Definition**
 - Textual definition (or description) of a term
 - Sometimes omitted leaving the name only
- **Cross references ‘DbXRef’ e.g [TAIR:ki]**

```
[Term]
id: CARO:0000063
name: portion of cell substance
def: "Portion of organism substance located within
a cell." [CARO:mah]
is_a: CARO:0000004 ! portion of organism substance
```


Bio-ontologies: Annotations

Annotations form nested structures:

- **Definition:** <text, **DbXRef** list>
- **DbXRef:** <name, accession, [description]>
e.g. ISBN:0471245208
- **Synonym:** <{exact, broad, narrow, related}, text, DbXRef list,
[synonym-type]>
e.g. CARO:0000048 gonochoristic organism
has synonym: “monoecious organism” RELATED []
- **Synonym-Type:** <name, description, scope>
e.g. synonymtypedef: UK_SPELLING "British spelling"
EXACT

Web Ontology Language (OWL)

In parallel with the development of these biomedical ontologies, standards for a web-compatible ontology language emerged from the W3C:

- **OWL is a general-purpose ontology language**
 - **Based on Description Logic**
 - » **With well-defined semantics**
 - » **Tractable reasoning algorithms**
 - **Web-compatible**
 - » **Concepts and relations referred to through URIs**
 - » **RDF/XML syntax**
 - **Compatible with other W3C standards**
 - » **RDF**
 - » **RDFS**

Web Ontology Language (OWL)

- **Semantics**

- owl:subClassOf is the isa/subclass relation and holds between two classes (or class descriptions)**

- All other relationships in the ontology are used to construct class descriptions**

- differs considerably from the use of relations in bio-ontologies**

- **Syntax**

- RDF/XML has multiple syntaxes**

- No guarantee that an OWL ontology will be written in the same way twice**

- » This does not matter in the construction of the RDF graph, the model is always the same**

Web Ontology Language (OWL)

- RDF Syntax

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <exterms:creation-date>August 16, 1999</exterms:creation-date>  
</rdf:Description>
```

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>  
</rdf:Description>
```

```
<rdf:Description rdf:about="http://www.example.org/index.html">  
  <exterms:creation-date>August 16, 1999</exterms:creation-date>  
  <dc:creator rdf:resource="http://www.example.org/staffid/85740"/>  
</rdf:Description>
```

Web Ontology Language 1.1 and 2

- OWL 1.1 extended the expressivity of OWL
 - New cardinality constraints
 - Role composition
- OWL 2 has a new XML syntax, not based on RDF
 - XML Schema has been defined

```
<owl:Class rdf:about="http://purl.org/obo/owl/CARO#CARO_0000063">  
  <rdfs:label xml:lang="en">portion of cell substance</rdfs:label>  
  <rdfs:subClassOf  
    rdf:resource="http://purl.org/obo/owl/CARO#CARO_0000004" />  
</owl:Class>
```

RDF/XML

```
<EntityAnnotation>  
  <OWLClass URI="&oboContent;CARO#CARO_0000063" />  
  <Annotation annotationURI="&rdfs;label">  
    <Constant>portion of cell substance</Constant>  
  </Annotation>  
</EntityAnnotation>
```

OWL 2 XML

OBO into OWL: First steps to using OWL in Biomedical Ontologies

- **Aims for converting OBO to OWL (NCBO/GO effort with Moreira, Mungall and Shah):**
 - to translate OBO to OWL, and back,
 - staying within OWL-DL, and
 - round-trip files without error.
- **IDs - remain the primary index → local name in URL**
`CARO:0000063 → namespace#CARO_0000063`
- **Semantics**
 - TermA isa TermB → TermA subClassOf TermB**
 - TermA relation TermB →**
TermA subClassOf some.relation TermB

Therefore, terms with no isa parent will become direct subclasses of Thing

OBO into OWL: OWL 2

- OBO-In-OWL

- Simple uniform procedure not involving any extended analysis of term meaning
- Gives ‘surprising’ results for ontologies that are not *isa-complete*

- “OBO-In-OWL 2”

- Achieved simply using the Manchester/Wonderweb Java API
- Retains the advantage of the OWL 2 schema

- Can now consider working with ontologies at the XML document level

- XML methods
- First, look at ontology tools

```
<EntityAnnotation>
  <OWLClass URI="&oboContent;CARO#CARO_0000063" />
  <Annotation annotationURI="&rdfs;label">
    <Constant>portion of cell substance</Constant>
  </Annotation>
</EntityAnnotation>

<EntityAnnotation>
  <OWLClass URI="&oboContent;CARO#CARO_0000065" />
  <Annotation annotationURI="&rdfs;label">
    <Constant>basal lamina</Constant>
  </Annotation>
</EntityAnnotation>

<EntityAnnotation>
  <OWLClass URI="&oboContent;CARO#CARO_0000067" />
  <Annotation annotationURI="&rdfs;label">
    <Constant>simple cuboidal epithelium</Constant>
  </Annotation>
</EntityAnnotation>
```

Tools: OBOEdit 2

OBOEdit 2.000-beta40

File Edit Layout Editors Viewers Search Tools

Ontology Tree Editor Ontology Change Tracker

Ontology Tree Editor

- Classes
 - anatomical entity
 - immaterial anatomical entity
 - anatomical line
 - anatomical point
 - anatomical space
 - cell space
 - anatomical surface
 - material anatomical entity
 - anatomical structure
 - acellular anatomical structure
 - anatomical group
 - cell
 - cell component
 - compound organ
 - cavitated compound organ
 - compound organ component
 - solid compound organ
 - extraembryonic structure
 - multi-cellular organism
 - multi-tissue structure
 - compound organ component
 - simple organ
 - organism subdivision
 - portion of tissue
 - portion of organism substance

- Relations
- disjoint from
- inverse of
- is_a
- part_of

OBOEdit 2 also provides information such as the number of classes, warnings such as missing definitions, edit history, a change tracker and reasoner

anatomical entity
↓
material anatomical entity
↓
anatomical structure
↓
compound organ
↓
compound organ component

Tools: Protégé 3

Protégé - viewing OBO OWL files

The screenshot displays the Protégé interface with two main panes: SUBCLASS EXPLORER and CLASS EDITOR.

SUBCLASS EXPLORER: Shows an asserted hierarchy for the project. The classes listed are:

- CL_0000085
- CL_0000352
- CL_0000422
- CL_0000353
- CL_0000349
- CL_0000329
- CL_0000324
- CL_0000222 (highlighted)
- CL_0000064
- CL_0000059
- CL_0000211
- CL_0000163
- CL_0000361

CLASS EDITOR: Shows the details for class CL_0000222. The table below represents the data shown in the interface:

| Property | Value |
|------------------------|-----------------|
| rdfs:comment | |
| oboInOwl:hasDisjoint | @oboInOwl267 |
| oboInOwl:hasEquivalent | @oboInOwl269 |
| oboInOwl:hasEquivalent | @oboInOwl270 |
| rdfs:label | mesodermal cell |

Below the table, the instance list shows:

- CL_0000220
- CL_0000548

Tools: Protégé 3 + OBO-In-OWL Tabs

Tools: OBO Converter (Moreira and Musen, bioontology.org) OBO Explorer - both Protégé plugins

The screenshot shows the Protégé 3.2 interface with the following components:

- Window Title:** po_anatomy Protégé 3.2 (file:/Users/stuart/Work/COBRA-CT/src/ProtegeWidgets/src/OBO-OWL-Ontologies/po_anatomy.pprj, OWL / RDF Files)
- Menu Bar:** File Edit Project OWL Code Tools Window Help
- Toolbar:** Standard Protégé icons for file operations and navigation.
- Tab Bar:** Metadata (0), OWLClasses, Properties, Individuals, Forms, OBO Explorer, OBO Converter, OBO Ontology Metadata.
- Left Panel (SUBCLASS EXPLORER):**
 - For Project: po_anatomy
 - Asserted Hierarchy
 - plant_structure_p1_PO_0009011
 - whole_plant_p1_PO_0000003
 - in_vitro_cultured_cell_tissue_and
 - plant_cell_p1_PO_0009002
 - sporophyte_p1_PO_0009003
 - gametophyte_p1_PO_0009004
 - tissue_p1_PO_0009007
 - organ_p1_PO_0009008
 - phylome_p1_PO_0006001
 - root_p1_PO_0009005
 - shoot_borne_root_p1_PO_0000042 (selected)
 - embryonic_root_p1_PO_0009006
 - lateral_root_p1_PO_002012
 - primary_root_p1_PO_002013
 - sepal_p1_PO_0009031
 - petal_p1_PO_0009032
 - floral_bract_p1_PO_0009034
 - inflorescence_bract_p1_PO_0009035

Archiving XML data

XML documents and XML keys

- XML documents, serving as databases, often contain unique keys (in the database sense)
- Top-level keys and secondary keys identify elements in hierarchically-structured databases [Buneman 01]
 - If *name* and *course* in `<name>joe</name><course>maths-1</course>...` constitute a key, then all elements with this key must agree everywhere, define:
 - path expressions
 - *value equality* \Rightarrow *node equality*
 - *Relative keys* - motivated by scientific databases

Archiving XML data

Versioning XML documents using timestamps

- **Changes to scientific databases are accretive [Buneman 01]**
 - Documents have a key structure
 - Additions/deletions are infrequent
 - diff-based approaches may be inefficient
- **'keys' can be used to merge versions of documents**
 - Keys that are the same in two versions indicate that no change has taken place - supports merging
 - In fact, the element need only be stored once, along with a timestamp representing a time interval
 - The document structure can be exploited - timestamps stored only at child nodes that are changed

BioSphere: An ontology portal

Now consider development efforts and the tools needed to support end users

- **Sharing ontologies**
 - For loosely-organised groups of ontology developers
- **Version management**
- **Visualising user's view points**
- **Supporting discussion and collaboration**

Why a portal?

- **The architecture is ideal for managing data resources centrally**
- **Details of formats, plug-ins etc can be hidden**
 - But off-line working needs to be supported

Portal and Grid Technologies

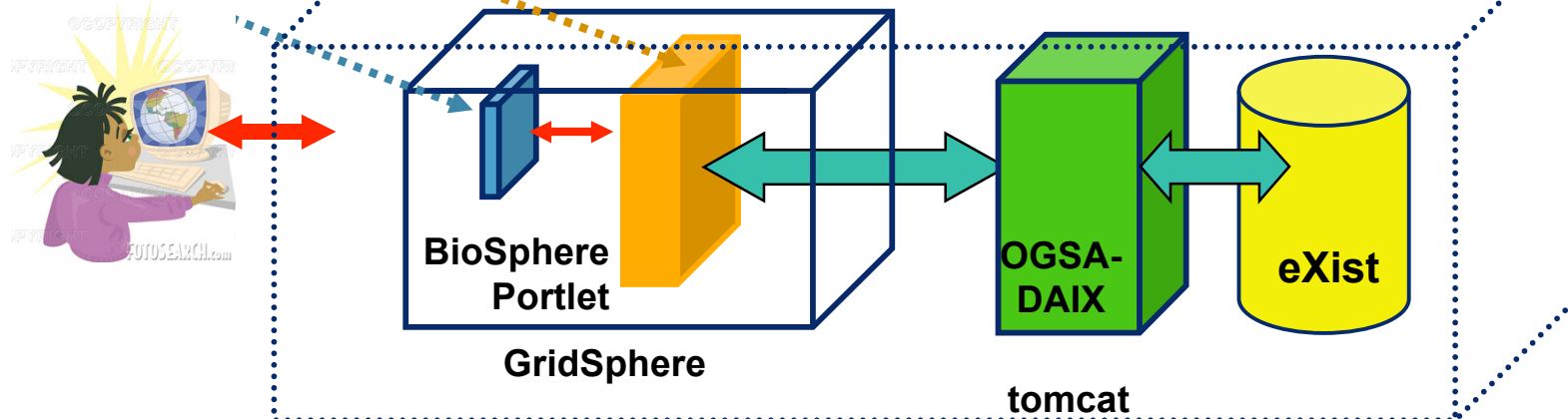
The BioSphere portal integrates several existing technologies:

- **GridSphere**
 - A environment for accessing portlets
 - » Java enterprise-style designs
 - » GridSphere and portlets are deployed in tomcat
 - Provides user account management
 - Integration with Grid security where needed
- **OGSA-DAI / DAIX**
 - Middleware providing read and read/write/update access to data resources
 - DAIX version is designed to operate with XML databases
- **eXist XML database**

Portal and Grid Technologies

In addition to GridSphere:

- Spring Framework
 - Portal event handling
- Dojo JavaScript toolkit
 - Provides a drag-and-drop tree library
 - Plus other GUI components



Version Control for OWL 2 Ontologies

Relying on OWL ontologies conforming to the OWL2 Schema:

- New metadata elements are introduced into the OWL 2 XML document to represent the duration during which an assertion is believed by a named user

- Durations are indicated by a begin and end version number, or
- by a constant denoting a future version

`<begin-user-ID=Version-I end-user-ID=Version-J>`

`<begin-user-ID=Version-K end-user-ID=FUTURE>`

New attributes added to `<Assert/>` elements in OWL 2 logic definitions, or

To existing `<Constant>` elements in OWL 2 annotations

- The objective is to capture changes for the group, for each user and each version, e.g. user 1 rejects an assertion:

`<Assert begin-group=Version-0 end-group=FUTURE>` add new:

`<Assert begin-user-1=Version-0 end-user-1=Version-2>`

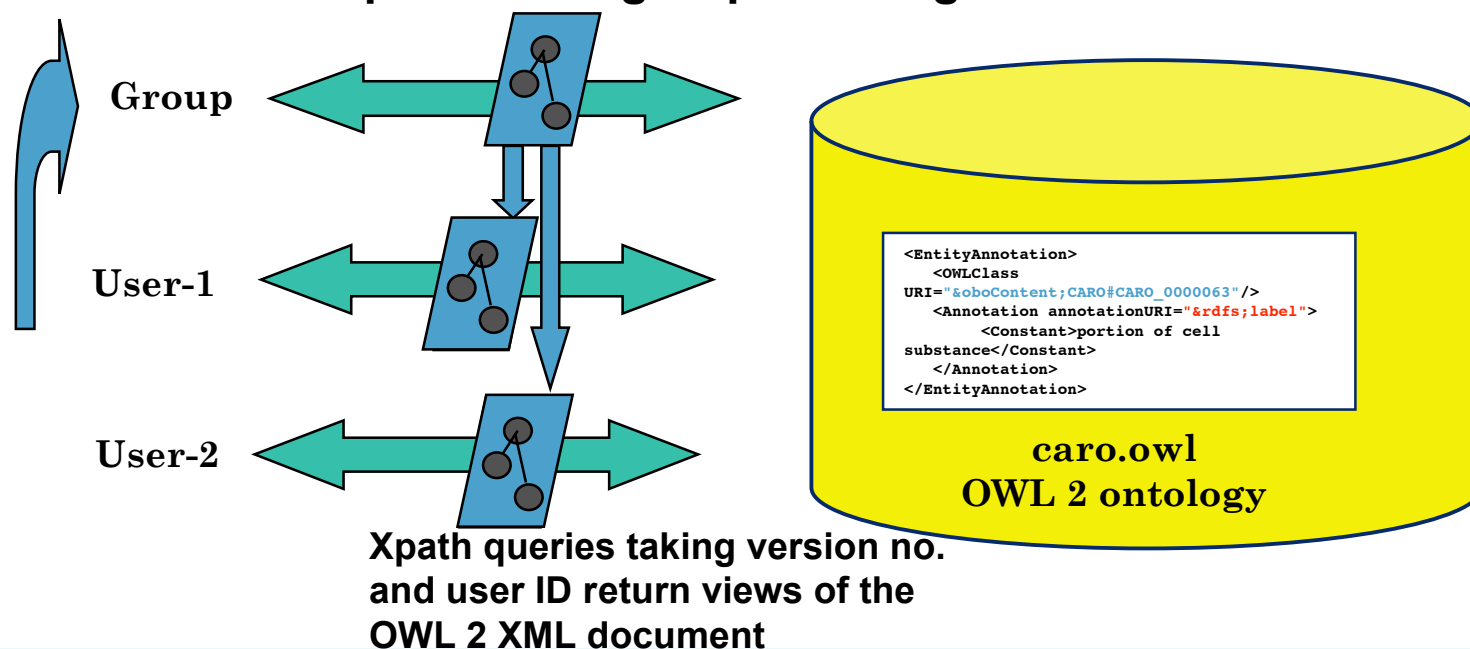
Version Control for OWL 2 Ontologies

Collaborative editing model

Users inherit the group view

Users make edits / notes / act on assigned tasks

User edits are copied to the group when agreement is established



Version Control for OWL 2 Ontologies

Objectives:

- To efficiently retrieve (sub)sections of the ontology document within the BioSphere portal
 - Design of OWL 2 schema modifications
 - Design of Xpath/Xupdate queries + Java to get/set version information
 - Export to pure OWL 2 syntax, to OWL RDF/XML and OBO
- To package as a stand-alone OGSA-DAI workflow
 - assuming the user and version database files are suitably initiated:
 - Provide generic version management functions
getOWL2Ontology(user, version, document-name, collection,...)
updateOWL2Ontology(user, version,...)

Uses of Ontology Version Metadata

The version and user metadata can provide important information, including:

- **How a user has constructed their ontology over time**
- **Where a user agrees with, and differs from, the group**
- **Where a user agrees with and differs from, other users**

But to be useful, this information must be easily understood

- **A good visualisation method that allows ontologies to be compared and contrasted is needed**
 - **A tree is the traditional visualisation of the ontology graph [bearing in mind this may not be ideal]**
 - **A simple approaches are to colour nodes according to user/version & to present post-it notes for discussion**
- **A good technique would have other uses too**

BioSphere Portal (v0.5)

gridsphere portal framework English [Home](#)

Login

User Name

Password

Remember my login

[Forgot your password?](#)

August 20, 2008 powered by gridsphere

Gridsphere provides the user with login and layout customisation

BioSphere Portal (v0.5)

BioSphere 0.5

Ontology Editor Form

Class URI and Name Assertion ●

Class URI: Name:

SubClass - SuperClass Assertion ●

SubClass URI: SuperClass URI:

User and Version IDs for Action

User ID: Version ID:

Action: Assert Retract

○ CARO

- ObsoleteClass
- Subset
- SynonymType
- anatomical cluster
- anatomical entity
 - immaterial anatomical entity (CARO_0000007)
 - material anatomical entity (CARO_0000006)
 - anatomical structure (CARO_0000003)

The portlet currently has two components:
a form for entering information; and
a tree visualisation of the ontology;

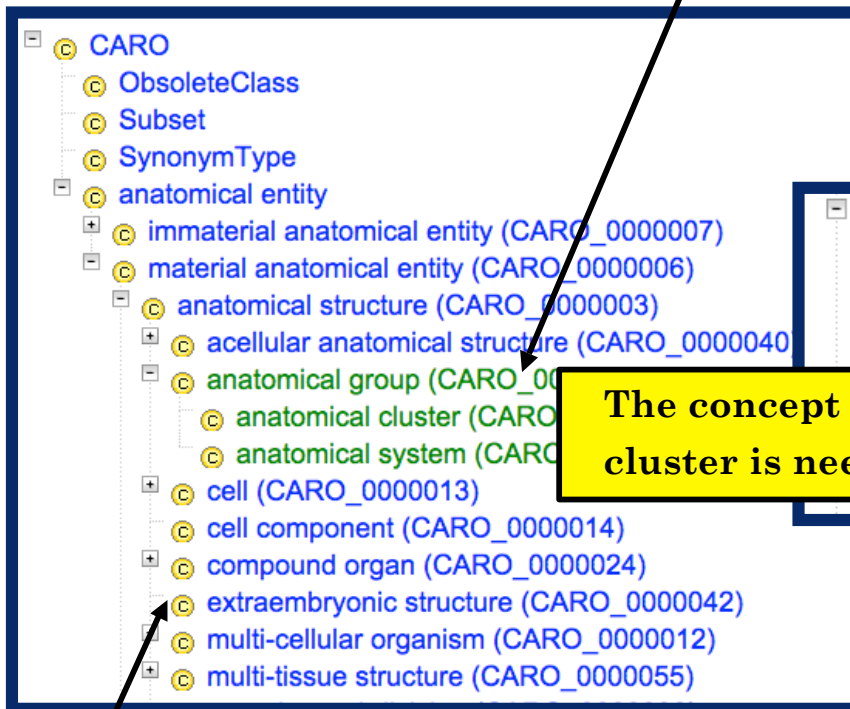
BioSphere Portal (v0.5)

BioSphere 0.5
Ontology Editor Form
SubClass - SuperClass Assertion
SubClass URI: SuperClass URI:
Class URI and Name Assertion
Class URI: Name:
User and Version IDs for Action
User ID: Version ID:
Action: Assert Retract

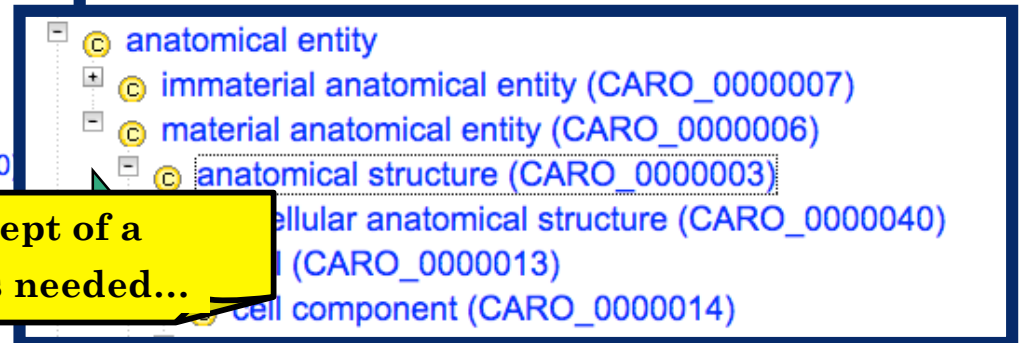
User interaction is currently via a form,
but this will be replaced by drag-and-drop operations
where possible

BioSphere Portal (v0.5)

User-1 View:
SubClassOf structure shown in green



Group View - excludes anatomical group



The concept of a cluster is needed...

ToDo:
[user-1] Complete definition for ...
[curator] Publish to group

Group SubClassOf structure shown in blue

Ontology Portals

- The NCBO has a *bio-portal* that allows OBO ontologies to be browsed

The screenshot displays the BioPortal interface for the Common Anatomy Reference Ontology. The left panel shows the 'Tree View' with a hierarchical list of terms: anatomical entity, immaterial anatomical entity, anatomical line, anatomical point, anatomical space (highlighted), anatomical surface, material anatomical entity, anatomical structure, and portion of organism substance. The right panel shows the 'Graph View' with a 'Local Neighborhood' graph type, illustrating the 'is_a' relationships between 'cell space', 'anatomical space', and 'immaterial anatomical entity'.

THE NATIONAL CENTER FOR BIOMEDICAL ONTOLOGY
BioPortal
Browse Search

Common Anatomy Reference Ontology

Tree View
Tree view constructed based on *is_a* hierarchy

- anatomical entity
 - immaterial anatomical entity
 - anatomical line
 - anatomical point
 - anatomical space**
 - anatomical surface
 - material anatomical entity
 - anatomical structure
 - portion of organism substance

Graph View
Graph Type Local Neighborhood

```
graph BT; cell_space((cell space)) -- is_a --> anatomical_space((anatomical space)); anatomical_space -- is_a --> immaterial_anatomical_entity((immaterial anatomical entity));
```


Ontology Versioning and Collaboration

- **KAON ontology management [Gabel 04]**
- **SemVersion [Volkel 06]**
- **Protégé**
 - Prompt - change tracking [Noy 04]
 - Collaborative Protégé [Tudorache 07]
- **Ontology views**
 - Sub-ontology extraction [Bhatt 04]
 - Segmentation [Seidenberg 06]
 - Views [Noy 04] [Volz 03]

Future Work

- **Completing the coverage of the OWL 2 schema**
 - Define the modified schema
 - Complete a set of test cases
- **Address performance issues in querying / rendering large amounts of data**
 - Construct the visualisation incrementally
- **Allow the upload/download of ontology versions**
 - Support format conversion
- **Visualisation**
 - Compare user/group/version views of an ontology
 - » Changes to logic
 - » Changes to annotations
 - Compare classified and asserted ontologies

**BBSRC grant
BB/F015976/1**