Ontology Views for Collaborative Ontology Creation: The BioSphere Portal

Stuart Aitken[1] and Jonathan Bard[2]

[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 []
- **Synomym-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**

  i. TermA isa TermB → TermA subClassOf TermB

  ii. 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

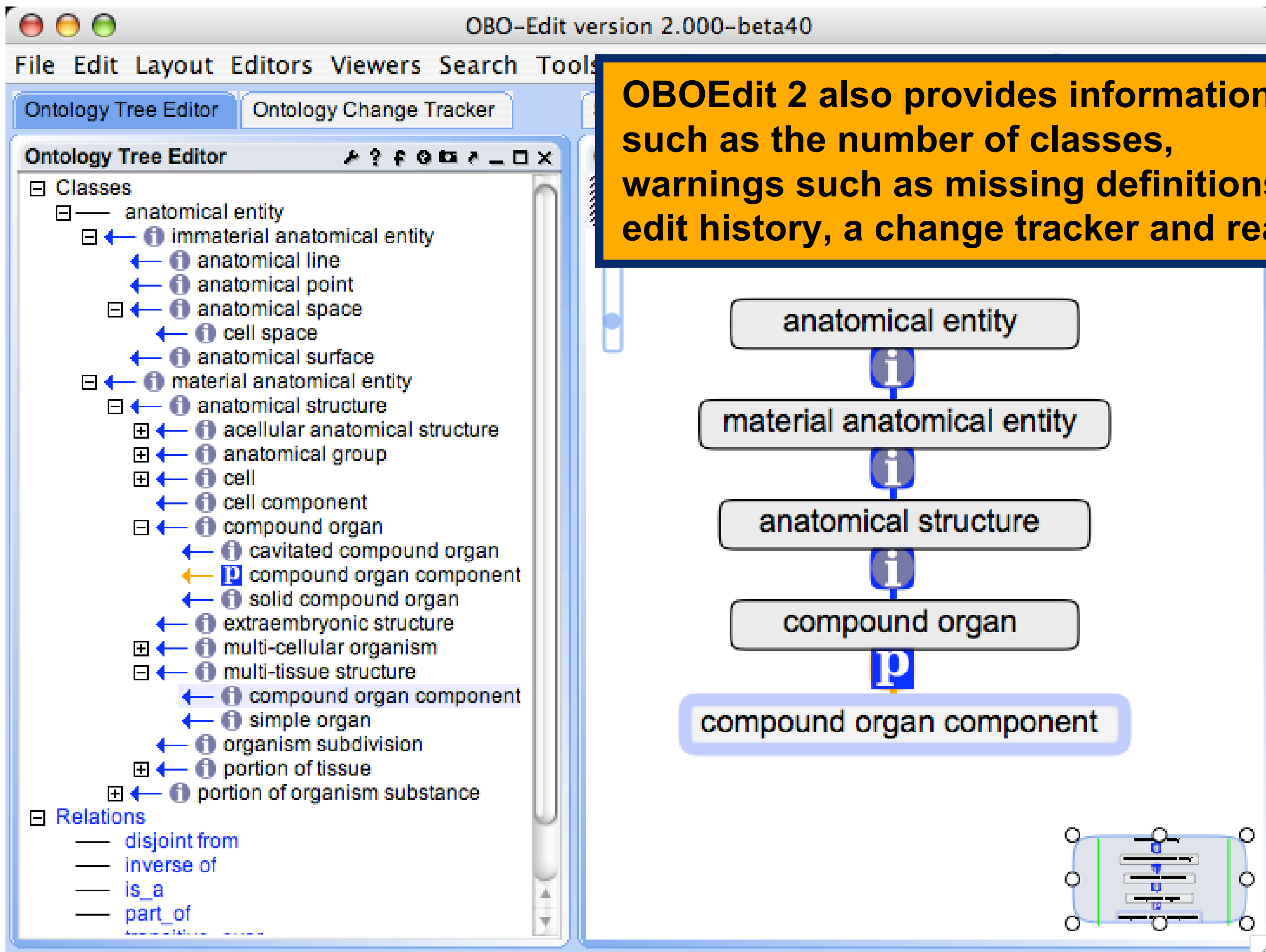
- OBO-In-OWL
  - Simple uniform procedure not involving any extended analysis of term meaning
  - Gives ‘surprising’ results for ontologies that are not *isa-complete*
- “OBO-In-OWL 2”
  - Achieved simply using the Manchester/Wonderweb Java API
  - Retains the advantage of the OWL 2 schema
- Can now 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>
```

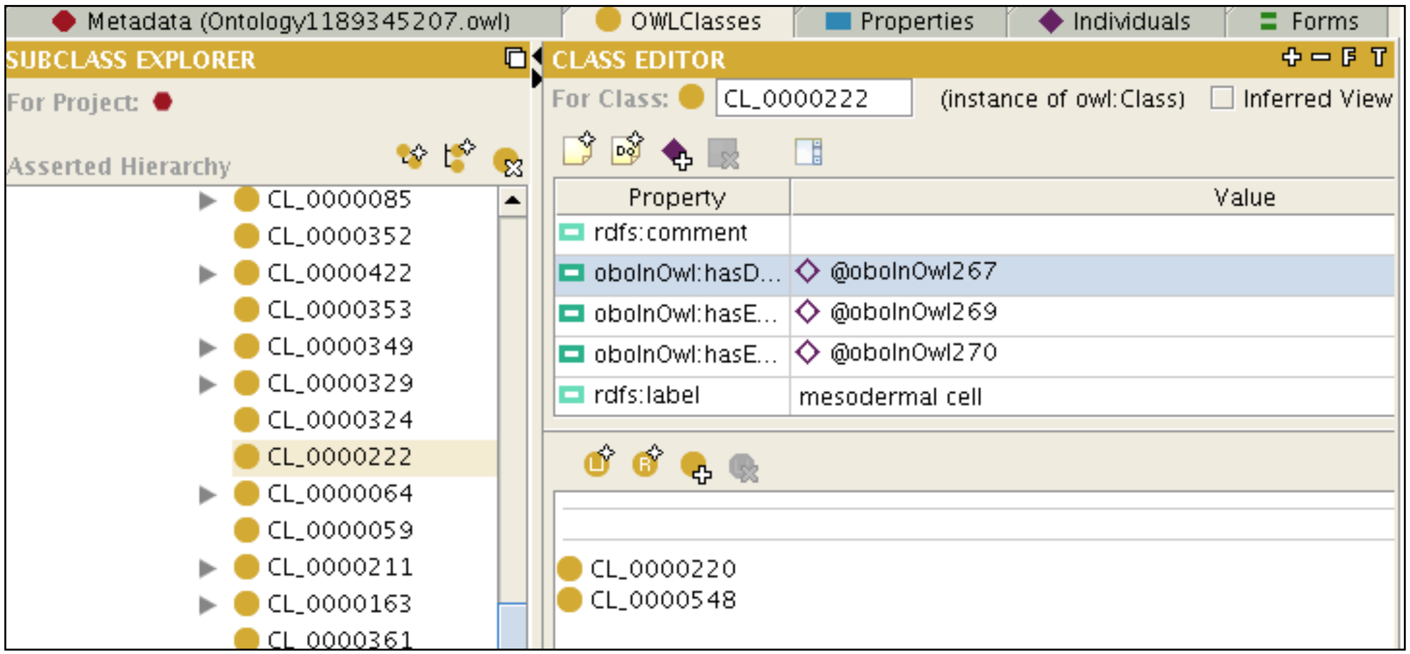
Tools: OBOEdit 2



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

Tools: Protégé 3

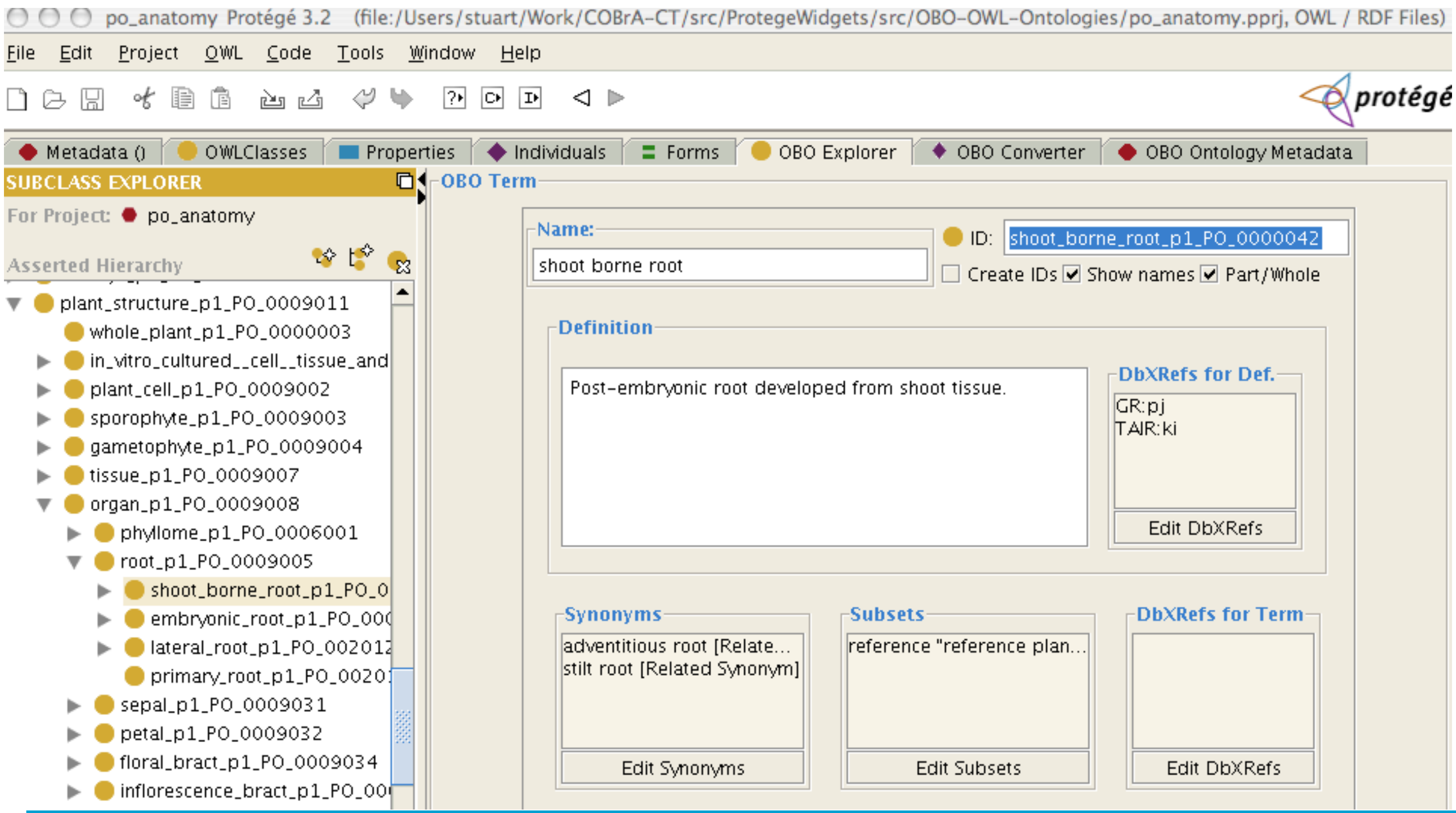
Protégé - viewing OBO OWL files



Tools: Protégé 3 + OBO-In-OWL Tabs

Tools: OBO Converter (Moreira and Musen, bioontology.org)

OBO Explorer - both Protégé plugins



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

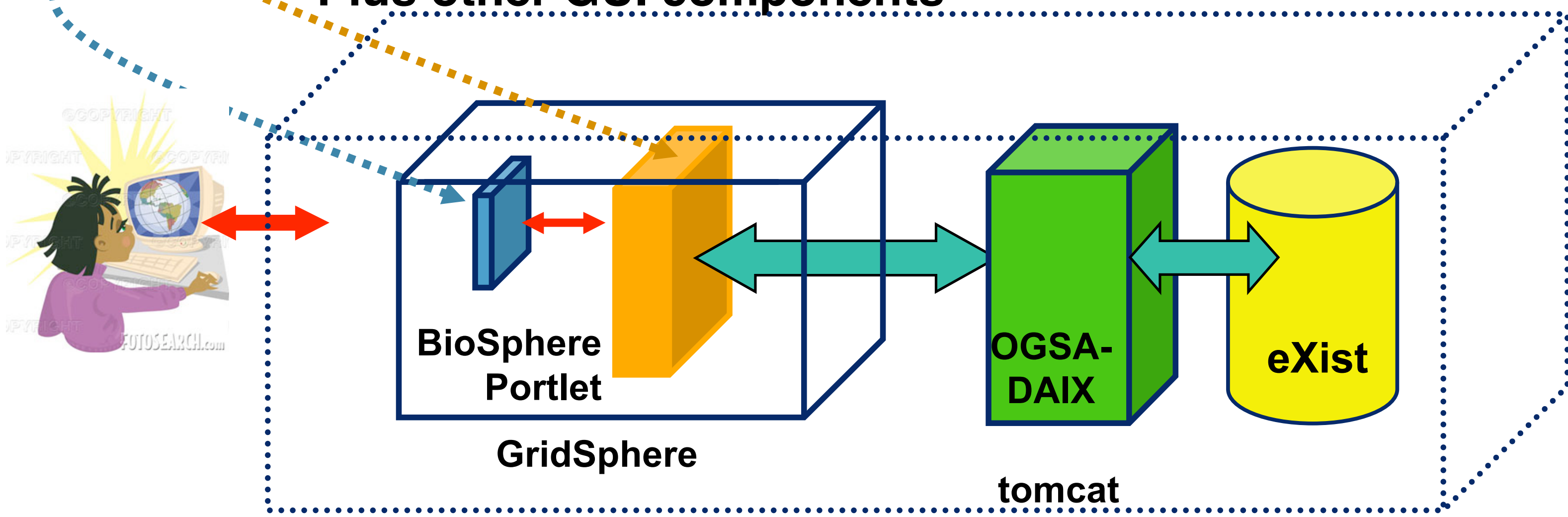
The BioSphere portal integrates several existing technologies:

- 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:**

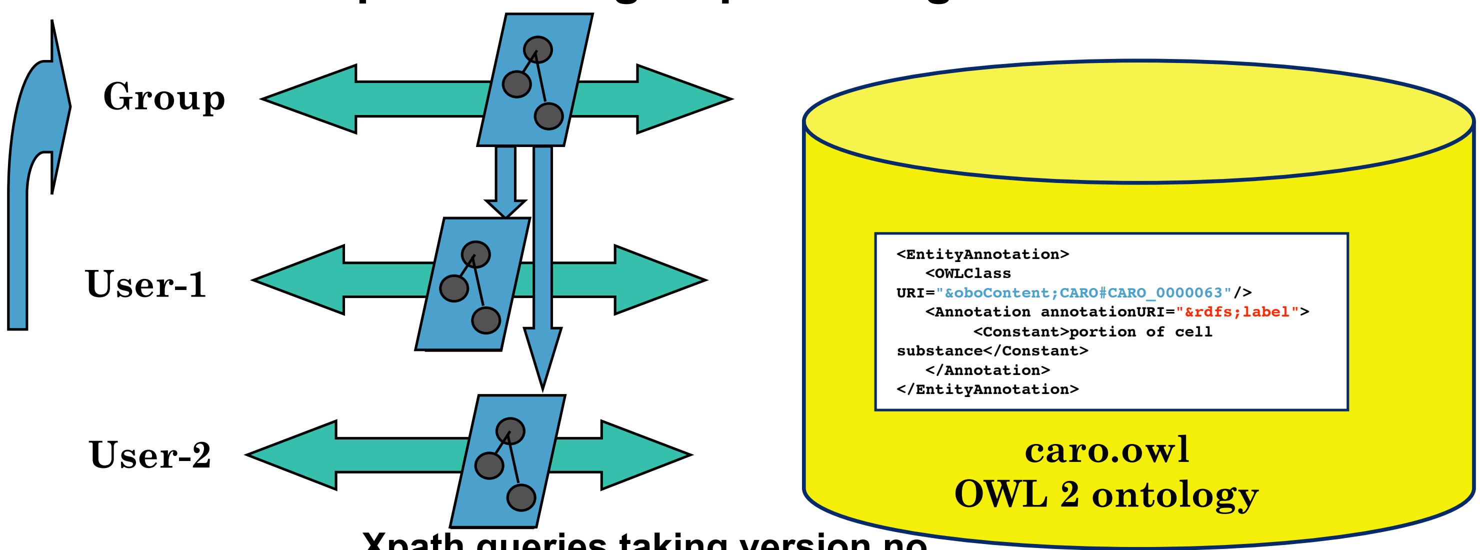
  <Assert begin-group=Version-0 end-group=FUTURE> add new:

  <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



Xpath queries taking version no. and user ID return views of the OWL 2 XML document

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:

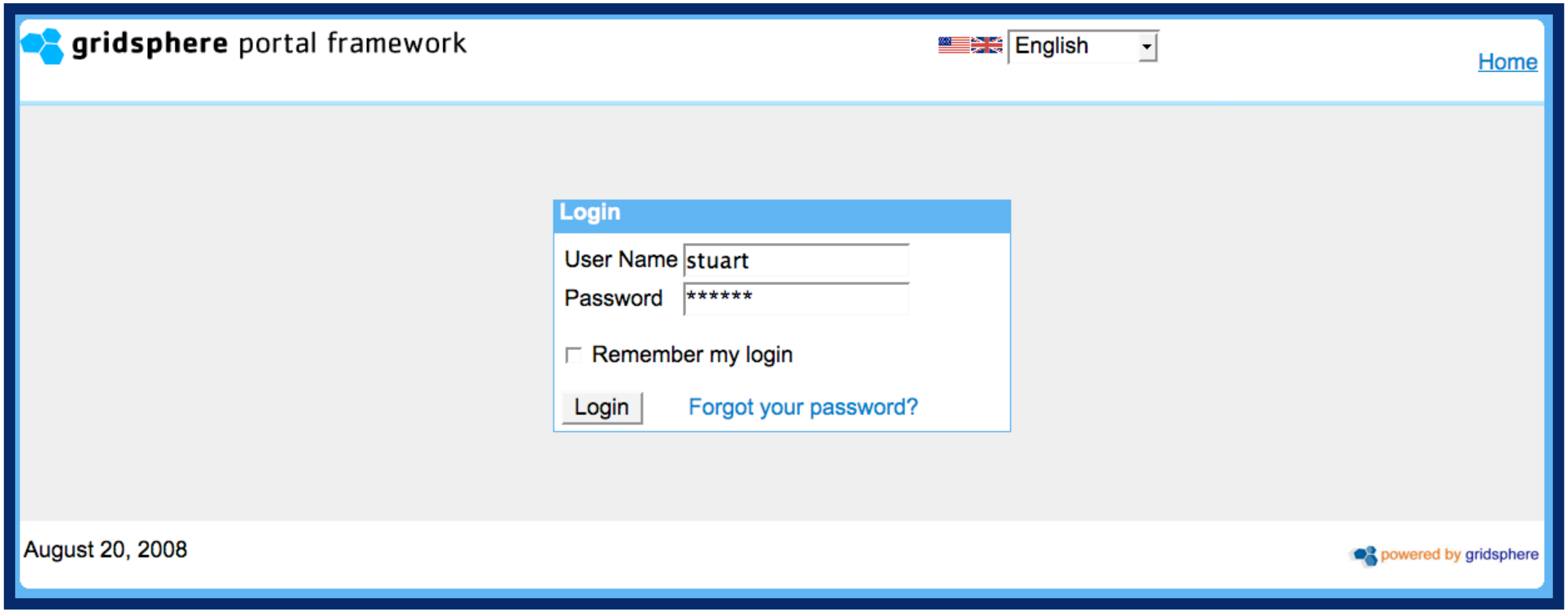
- How a user has constructed their ontology over time
- Where a user agrees with, and differs from, the group
- Where a user agrees with and differs from, other users

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

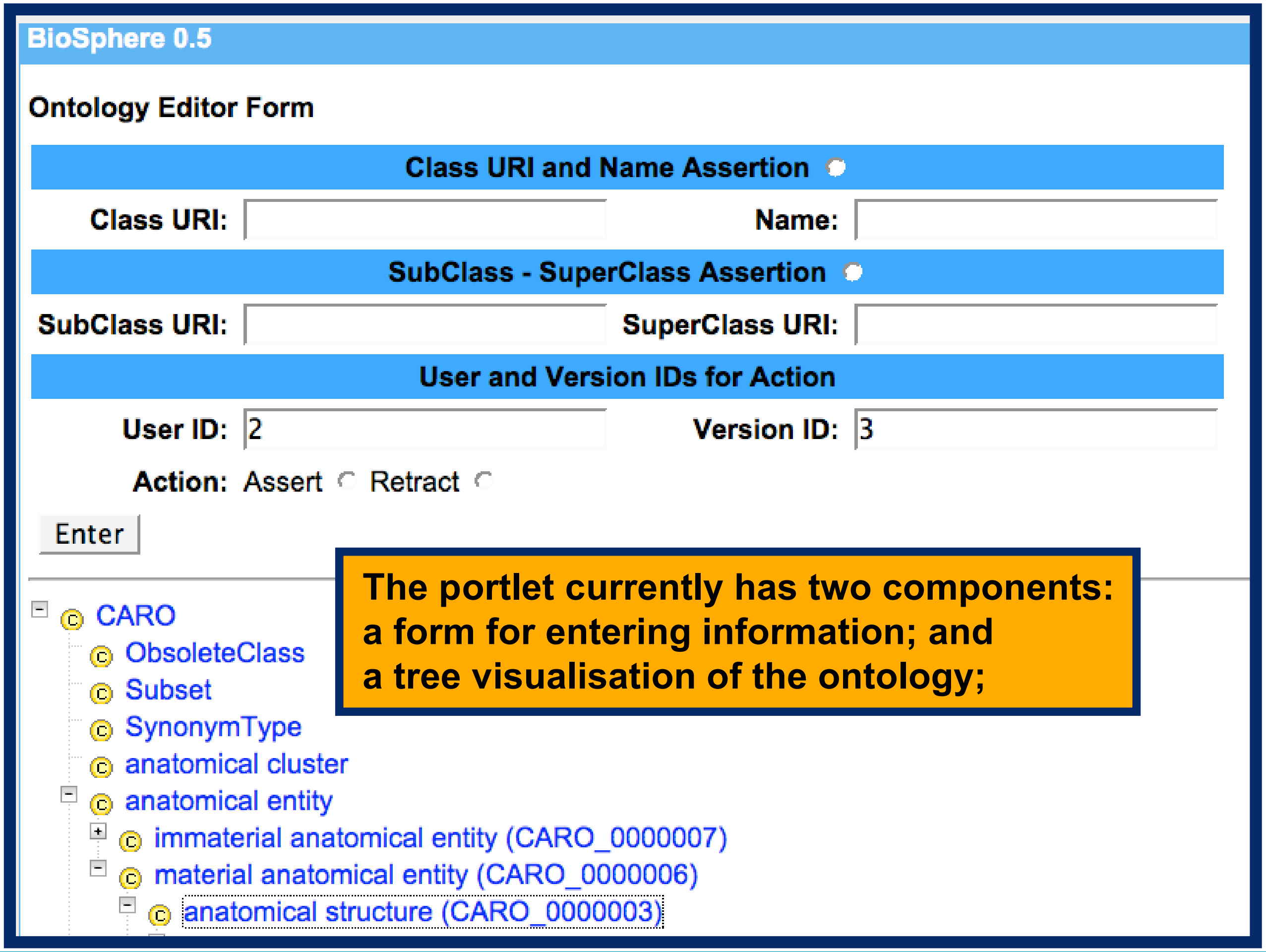
  [bearing in mind this may not be ideal]
  - A simple approaches are to colour nodes according to user/version & to present post-it notes for discussion
- A good technique would have other uses too

BioSphere Portal (v0.5)



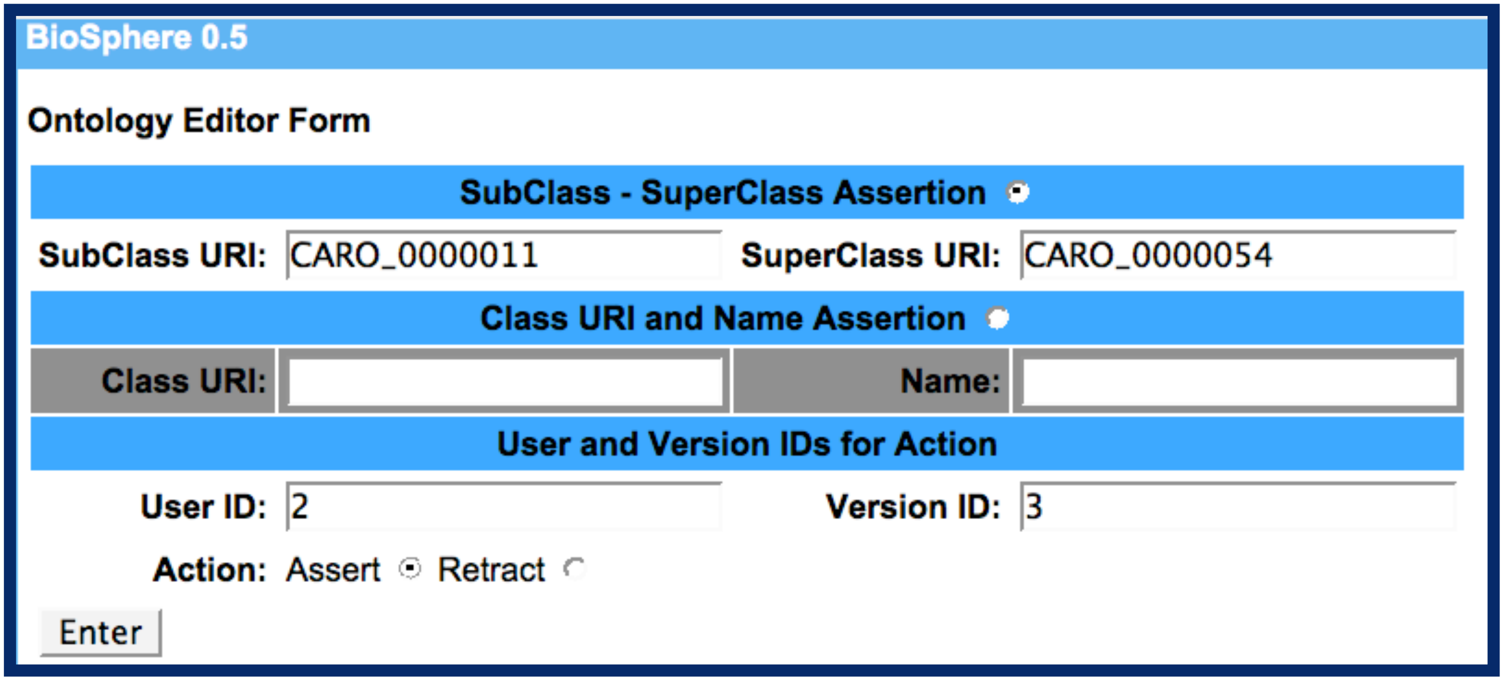
Gridsphere provides the user with login and layout customisation

BioSphere Portal (v0.5)



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

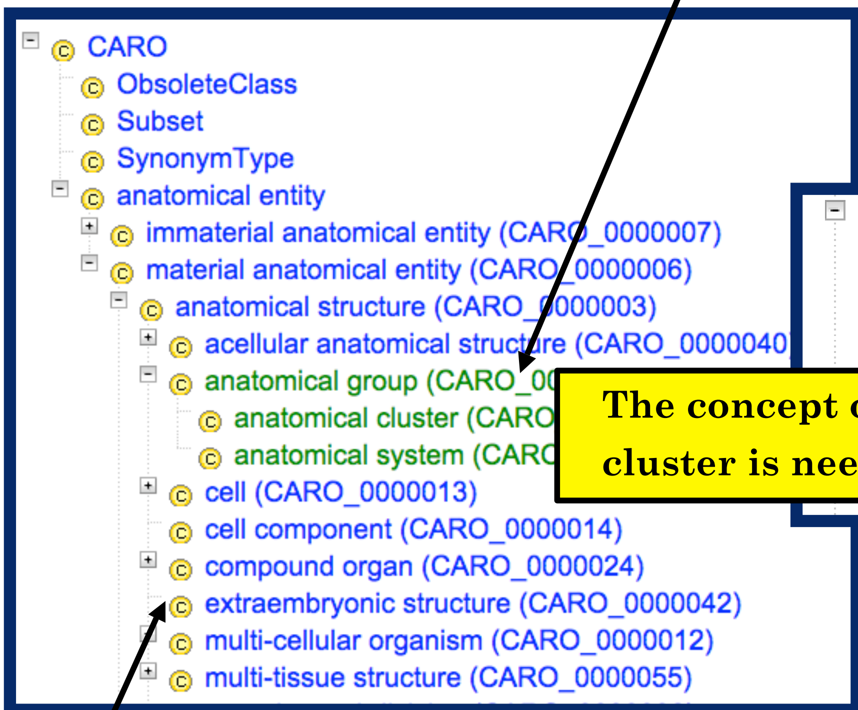
BioSphere Portal (v0.5)



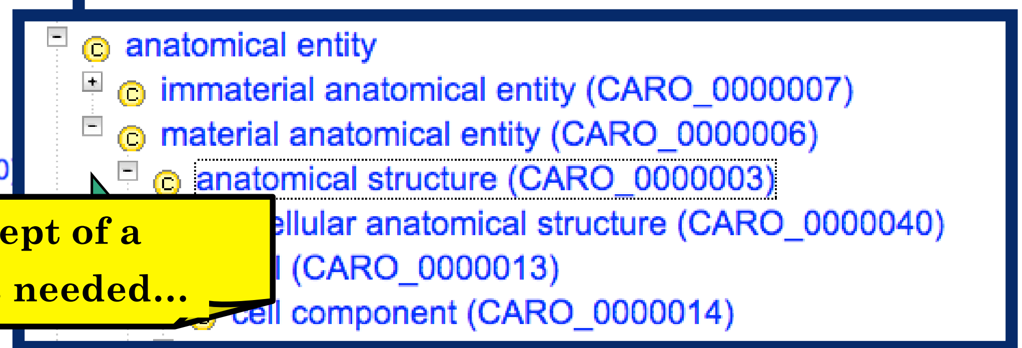
User interaction is currently via a form, but this will be replaced by drag-and-drop operations where possible

BioSphere Portal (v0.5)

User-1 View:
SubClassOf structure shown in green



Group View - excludes anatomical group



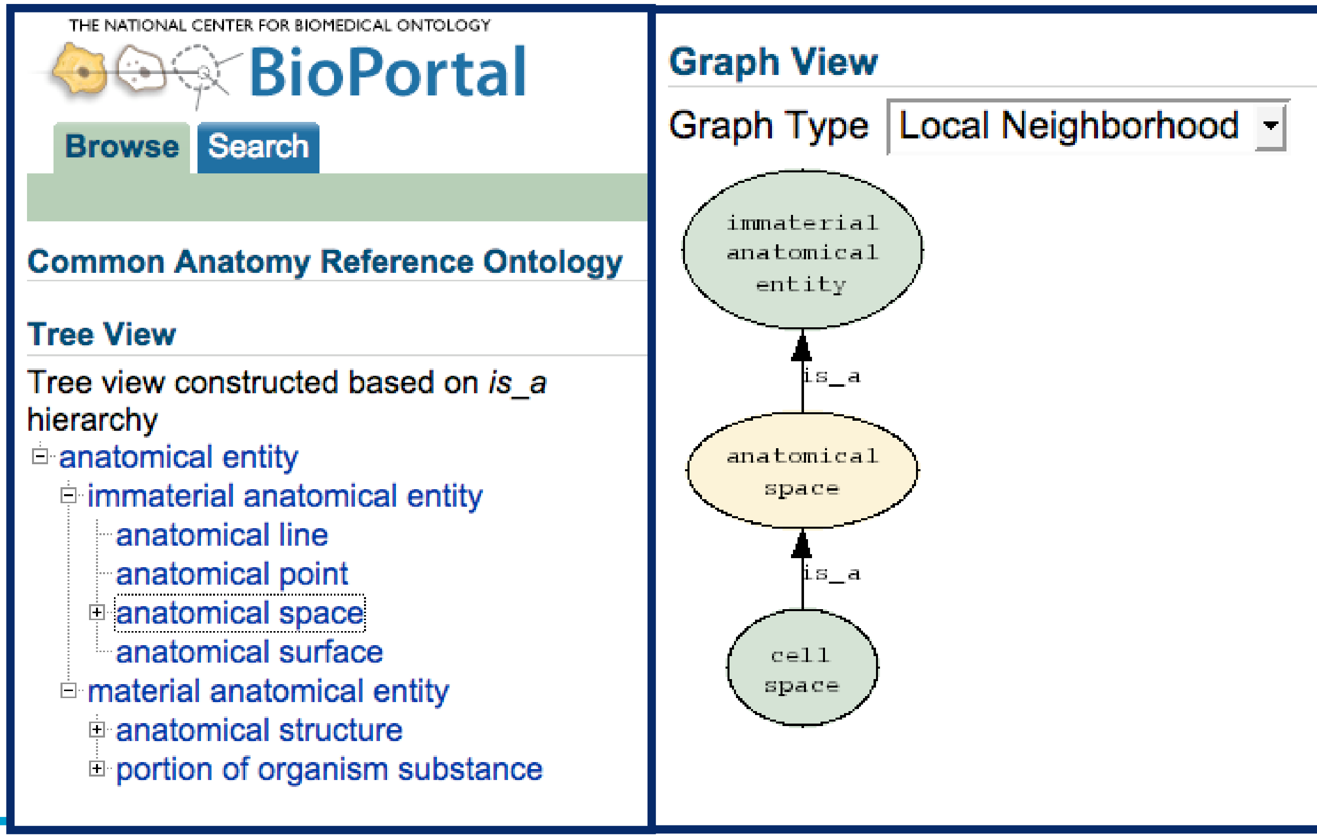
The concept of a cluster is needed…

Group SubClassOf structure shown in blue

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

Ontology Portals

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



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