1

COBrA and COBrA-CT: Ontology Engineering Tools

Stuart Aitken and Yin Chen

School of Informatics, The University of Edinburgh, Edinburgh EH8 9LE, United Kingdom

Summary. COBrA is a Java-based ontology editor for bio-ontologies and anatomies that differs from other editors by supporting the linking of concepts between two ontologies, and providing sophisticated analysis and verification functions. In addition to the Gene Ontology and Open Biology Ontologies formats, COBrA can import and export ontologies in the Semantic Web formats RDF, RDFS and OWL.

COBrA is being re-engineered as a Protégé plug-in, and complemented by an ontology server and a tool for the management of ontology versions and collaborative ontology development. We describe both the original COBrA tool and the current developments in this chapter.

Bio-ontologies play a crucial role in the indexing of experimental data - providing both unique IDs for aspects of anatomy, phenotype, process, cellular structure and molecular function [1, 2], and conceptual abstractions for aggregating results [3]. As discussed elsewhere in this volume, constructing ontologies of anatomy poses particular challenges including the choice of an appropriate level of granularity, how to represent spatial relationships (if at all) and how to represent the development of the organism over time. Many of the modelling decisions have been guided by the immediate use of the ontologies for indexing gene expression data, and the net result is a diversity of approaches and of interpretations for the basic elements in the anatomies, including the interpretation of the *part-of* relation. In many current anatomies the more pragmatic view of the ontology as a graph (where a *part-of* assertion is sufficient to define a concept) holds sway over the logic-oriented view that all concepts require an *is-a* relationship. This has implications for ontology editor design as the biologist will expect to see a graph that mixes *is-a* and *part-of*, rather than a pure *is-a* hierarchy that corresponds to the definitions that have been specified. These features of current anatomy ontologies had to be accounted for in the COBrA ontology editor, and its successor.

Over recent years, anatomies and other biological ontologies have grown in size, and their encoding languages have become more sophisticated, with the result that tools for creating, editing, verifying and maintaining them (e.g. version control, meta-data attribution, provenance, etc) have become essential. This wider curation activity has been recognised as a priority for e-Science [4], and is an important concern for many communities and in standards initiatives. For example, there are efforts to standardise the names used for tissue samples assayed by microarray [5], as well as the metadata that describes the experimental results (MGED/MIAME). Ontologies are of central importance in curation, as only by defining the meaning of the terms used to describe a particular field can the underlying concepts be clarified and agreed upon among the research community, and used consistently for annotating data. A consistent, shared ontology is of critical importance to the sharing of knowledge, and has long-term value in supporting a systems-level approach to biology. For example, the Gene Ontology is in widespread use for data mining and data visualisation, and has great potential for further integration of data across the different levels of biological granularity. However, ontologies are not static: they must change to reflect changes in science, to adapt to new uses, to broaden their community or to remedy flaws. Ontologies have also been identified as key resources in numerous e-Science projects, including AstroGrid, MyGrid and the Advanced Knowledge Technologies IRC.

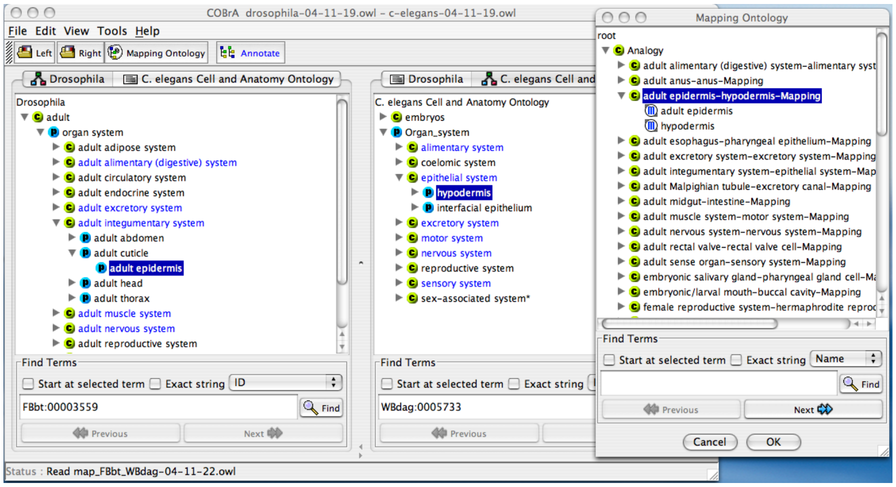
In parallel with expanding the range of domains being captured in bio-ontologies, and the number of terms in key resources such as the Gene Ontology (GO), researchers have been examining the formal and conceptual bases underlying ontology languages and modelling principles [6]. Initially constructed on an intuitive basis, many bio-ontologies are being scrutinised with regard to their underlying principles, and their support of inference - this being critical for automated verification. Ontologies of the same or similar conceptual domains are also being examined with respect to how they map to one another. The languages of the Semantic Web have a role to play as they provide standards, tools and techniques. For example, the Web Ontology Language (OWL `www.w3.org/TR/owl-ref`) has an XML syntax and a semantics designed for the sharing and reuse of ontologies over the Web. Utilising reasoners for OWL-Lite, OWL-DL and fragments of OWL-Full, OWL provides the mechanisms to address outstanding issues in bio-ontologies. For the ontology editor described here, OWL provides solutions to the problems of concept mapping and ontology verification.

Having chosen to work with OWL as the primary representation language, and to translate to and from the other bio-ontology languages, we are able to use XML databases for storage. XML querying tools can also be used for accessing and updating OWL ontologies providing we view them as XML documents.

The following sections introduce the COBrA ontology editor and its functions, then describe our solution to the curation and archiving problems that arise when individuals and communities develop ontologies.

1.1 COBrA

COBrA is an editor that allows GO and OBO ontologies to be created and explored. COBrA is also a mapping tool for ontologies that allows users to explore two ontologies simultaneously and to make links between them.


COBrA drosophila-04-11-19.owl - c-elegans-04-11-19.owl
File Edit View Tools Help
Left
Right
Mapping Ontology
Annotate
Drosophila
C. elegans Cell and Anatomy Ontology
Drosophila
adult
organ system
adult adipose system
adult alimentary (digestive) system
adult circulatory system
adult endocrine system
adult excretory system
adult integumentary system
adult abdomen
adult cuticle
adult epidermis
adult head
adult thorax
adult muscle system
adult nervous system
adult reproductive system
Find Terms
Start at selected term
Exact string
ID
Find
Previous
Next
C. elegans Cell and Anatomy Ontology
embryos
Organ_system
alimentary system
coelomic system
epithelial system
hypodermis
interfacial epithelium
excretory system
motor system
nervous system
reproductive system
sensory system
sex-associated system*
Find Terms
Start at selected term
Exact string
Previous
Mapping Ontology
root
Analogy
adult alimentary (digestive) system-alimentary syst
adult anus-anus-Mapping
adult epidermis-hypodermis-Mapping
adult epidermis
hypodermis
adult esophagus-pharyngeal epithelium-Mapping
adult excretory system-excretory system-Mapping
adult integumentary system-epithelial system-Map
adult Malpighian tubule-excretory canal-Mapping
adult midgut-intestine-Mapping
adult muscle system-motor system-Mapping
adult nervous system-nervous system-Mapping
adult rectal valve-rectal valve cell-Mapping
adult sense organ-sensory system-Mapping
embryonic salivary gland-pharyngeal gland cell-M
embryonic/larval mouth-buccal cavity-Mapping
female reproductive system-hermaphrodite reproc
Find Terms
Start at selected term
Exact string
Name
Find
Previous
Next
Cancel
OK

FBbt:00003559

WBdag:0005733

Status : Read map_FBbt_WBdag-04-11-22.owl

Fig. 1.1. Two anatomies displayed in COBrA, with a Mapping Ontology dialog inset right

COBrA is a product of the XSPAN project (`www.xspan.org`) which uses concept mapping to express judgements of homologies and analogies between tissues across different anatomy ontologies. The resulting knowledge base will contribute to a community resource for exploring gene expression data. In XSPAN, mappings can be used to express correspondences between tissues in terms of their evolution (*Evolutionary Homology*), development (*Common Lineage Homology*) or function (*Analogy*). Creating a mapping is necessarily a human decision, made complex by the nature of the task and the size of the anatomies. Within XSPAN, COBrA supports acquisition and exploration of these human-specified mappings.

COBrA provides both a tree-based view and a node-based view of an ontology, where the latter displays the selected term's parents, children and definitional information. The tree includes all relationships used in the ontology and is not limited to only the *is-a* or only *part-of* relationships (however, the user can hide relationships if they choose to). The ontology can be edited by direct manipulation of the tree or by calling a term editor. Initial evaluation of the tool over a range of tasks, and user-types, confirms the design choices [7]. Figure 1.1 shows a mapping between *adult epidermis* (Drosophila) and *hypodermis* (C Elegans).

Concepts and relations in Semantic Web languages such as OWL require both a name and a namespace (combined into a URIRef), and COBrA provides visualisations and interfaces to these new (and potentially unfamiliar) elements of the ontology. COBrA maps OBO relationships into their OWL-Full equivalent, that is, a relationship such as *part-of* is represented as a relationship between classes (these can be formally interpreted using a translation to first-order logic as in [8])[1]. COBrA

[1] In parallel with the development of tools for OWL-DL, a consensus on the interpretation of *part-of* in OWL-DL is emerging and so we expect to work in the OWL-DL sublanguage in future.

also provides a graphical interface to a number of analysis functions which we now describe.

Concept mapping, ontology merging, and verification are problems that COBrA solves through the use of OWL. A mapping is a pointer created to link a concept in one ontology to a concept in another: A mapping is a new term that relates two existing URIRefs. It can be created and saved without modifying the original ontologies. Meta-data such as authorship is associated with the mapping term, and mapping terms can be organised hierarchically, as illustrated in the right hand side of Figure 1.1. Terms with an associated mapping are shown in blue, and the user can click on such terms to automatically locate the matching term: clicking on *adult epidermis* in the Drosophila ontology causes the mapped term *hypodermis* to be found and displayed in the C. elegans ontology. The use of colour for mapped terms helps the user to locate anatomical entities that have been given a mapping. The user might critique existing mappings or seek to complete the mapping between ontologies.

Turning to ontology comparison and merger, these can be computed by finding the intersection and union, respectively, of the RDF graphs derived from the OWL representations of two ontologies. These graph-based operations improve on the equivalent operations that might be performed on textual representations of the ontologies (e.g. in CVS), but do not involve verification of the results.

For ontology verification, the semantics of the GO *is-a* and *part-of* relations must be defined, hence we use OWL *subClassOf* and define the interpretation of *partOf* [8]. These steps allow verification. An inference mechanism implements rule-based reasoning over the RDF graph, for example, to propagate properties across *partOf* links. COBrA can also perform a more complex ontology analysis that checks for cycles in the graph and in the ontology. Both graph manipulation and inference methods are provided by the Jena Semantic Web toolkit which provides Java methods to read, write and create RDF graphs (`www.hpl.hp.com/semweb`).

In addition, COBrA supports the import and export of bio-ontologies in RDF, RDFS and OWL. However, COBrA is not a generic OWL editor. The GO RDF format is that specified by the Gene Ontology Consortium, the RDFS format is a modification of that where *is-a* is replaced by *rdfs:subClassOf*. The OWL format is defined by a top-level ontology [8] which specifies a number of classes and relations that are required to state GO-style ontologies in OWL.

Protégé (`protege.stanford.edu`), a generic ontology editor, and OBOEdit (`www.geneontology.org`) provide comparable editing functions to COBrA. However, neither address mapping between ontologies. Protégé would require adaptation to read GO and OBO formats, but is more fully compatible with OWL (such a plug-in tool is described below).

COBrA demonstrates the practical application of Semantic Web techniques in the Bioinformatics context by combining familiar ontology-editing functions, and compatibility with existing file formats, with additional features such as mapping, merging and verification that make use of RDF and OWL.

1.2 Ontology Curation and the COBrA Curation Tools

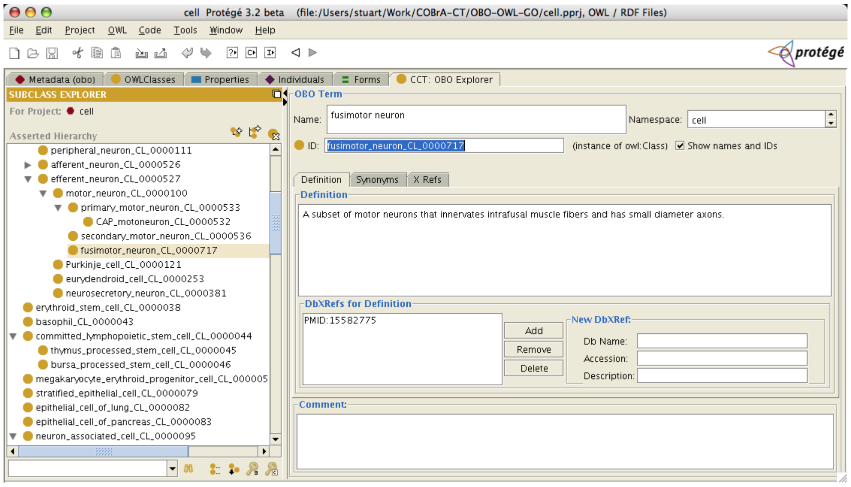
In common with experimental data, ontologies are created, published, and revised. Tracking and managing such changes requires new curation tools. In addition to version management, curation also includes the review of the content of the ontology, and assessment of quality. A related issue is the maintenance of ontological annotations assigned to data under a given ontology, as the ontology may change after a term has been used as an annotation and therefore one may wish for the annotation to be updated as well.

As the use of ontologies widens, the problems of tracking versions, and the changes between versions, and of reconciling differences in conceptual modelling arise. Addressing these are our main goals in the design of curation tools. Problems such as inconsistency that might arise in individual ontologies can be addressed by the graph checking that tools such as the COBrA editor can perform, or by more formal reasoning should the ontology be expressed in the description logic sub-language of OWL. We propose a server-based model for curation that allows remote users to create and submit annotated changes to ontologies and also to participate in the review process by applying some simple critiquing techniques that help identify errors. The COBrA-CT tools will support the curator by providing the appropriate management support and visualisations.

Organising the curation effort in a distributed setting, providing access to current and past versions of ontologies and providing search and related services requires an ontology management server. While it is possible (and certainly common) to simply archive different versions of an ontology, there is much to be gained from an explicit record of the changes made and their rationale. Ontology version *mappings* will be considered in the curation process, as they provide the explanation for the proposed changes. The focus on bio-ontologies is important as the problems we address are very complex in the general case. However, strategies have evolved in bioinformatics to address them, for example, concepts have IDs that are unique, and rather than being deleted, IDs persist as annotations to other concepts, or are categorised as obsolete terms.

As we are continuing to make use of the Web Ontology Language with its XML syntax as the means of data exchange, we shall be able to take advantage of both ontology-based and XML-based techniques for capturing changes. The difference between these can be illustrated as follows: from the document structure perspective, an edit to an XML-encoded ontology that asserts that class C, known to be a subclass of A, is also a subclass of B would be viewed as modifying node C in the XML document (irrespective of anything else we know about A and B). If classes A and B are known to be disjoint, then from the ontology perspective we would note a contradiction in the semantics as there can be no common subclass of disjoint classes. We propose to layer semantic checks on an XML-based ontology archiving mechanism. This approach is flexible, as XML is very widely adopted, and can exploit (but is not committed to) the logical language an ontology is expressed in.

It has been noted that changes to scientific data archives are accretive [9] - most changes are additive - although deletion and modification also occur. Scientific data


cell Protégé 3.2 beta (file:/Users/stuart/Work/COBrA-CT/OBO-OWL-GO/cell.pprj, OWL / RDF Files)
File Edit Project OWL Code Tools Window Help
protégé
Metadata (obo) OWLClasses Properties Individuals Forms CCT: OBO Explorer
SUBCLASS EXPLORER
Asserted Hierarchy
OBO Term
Definition Synonyms X Refs
Add
Remove
Delete
New DbXRef:
Comment:

For Project: cell

peripheral_neuron_CL_0000111
afferent_neuron_CL_0000526
efferent_neuron_CL_0000527
motor_neuron_CL_0000100
primary_motor_neuron_CL_0000533
CAP_motoneuron_CL_0000532
secondary_motor_neuron_CL_0000536
fusimotor_neuron_CL_0000717
Purkinje_cell_CL_0000121
eurydendroid_cell_CL_0000253
neurosecretory_neuron_CL_0000381
erythroid_stem_cell_CL_0000038
basophil_CL_0000043
committed_lymphopoietic_stem_cell_CL_0000044
thymus_processed_stem_cell_CL_0000045
bursa_processed_stem_cell_CL_0000046
megakaryocyte_erythroid_progenitor_cell_CL_000005
stratified_epithelial_cell_CL_0000079
epithelial_cell_of_lung_CL_0000082
epithelial_cell_of_pancreas_CL_0000083
neuron_associated_cell_CL_0000095

Name: fusimotor neuron
Namespace: cell
ID: fusimotor_neuron_CL_0000717 (instance of owl:Class) ☑ Show names and IDs

Definition
A subset of motor neurons that innervates intrafusal muscle fibers and has small diameter axons.

DbXRefs for Definition
PMID: 15582775

Db Name:
Accession:
Description:

Fig. 1.2. The OBO Explorer interface

is typically structured hierarchically, allowing a hierarchical key structure to be exploited in archiving changes to the data. Managing versions of a data resource can be performed on the basis of diffs (i.e. by recording the editing steps that cause the change). However, there are advantages for an approach where all objects have an associated timestamp. The central notions of hierarchical organisation, objects and timestamps [10] also apply to ontologies and ontology management, and this is the approach we plan to adopt. Given the problems noted by [11] with the simple diff approach, our approach will also be structure-based. We shall identify types of ontological changes that occur in practice, taking the procedures used in practice, e.g. by the Gene Ontology, as a starting point. As we do not assume that ontologies will make use of formalisms such as Description Logic, our approach is not reliant on the widespread uptake of this particular logic. However, we will exploit any formalism that is associated with an ontology, which may be DL or first-order logic [12]

We now present the Protégé plug-in for editing OWL bio-ontologies, named the OBO Explorer. The Ontology Version Manager is then introduced.

1.2.1 The COBrA-CT OBO Explorer

Methods for automatically converting ontologies in the Open Biological Ontologies formats into OWL have been proposed and can be utilised to create files that can be read into the Protégé ontology editor. Protégé has a large user community, and an active developer community that has created a wide range of plug-in utilities. However, Protégé is unable to display the annotations associated with OBO terms such as the database cross-references. As we aim to capture all of the content of OBO formated ontologies in OWL, both the logical structure of the ontology and the annotations, this is a significant barrier to the uptake of OWL. Therefore, there is a need for a COBrA-like plug-in that will allow full visualisation and editing for OBO OWL ontologies: the OBO Explorer.

The OBO Explorer is tightly integrated to the Protégé architecture to ensure interoperability with other Protégé tools. The interface is implemented as a ‘tab’ that presents the term annotations on the right hand panel, with the class hierarchy on the left. The user interface components are all present on the main panel, and automatically update the underlying OWL model, thus eliminating pop-up editors and ‘confirm change’ actions. Where the OWL ontology lacks the OWL and RDF relationships needed to represent OBO annotations, the tool creates the appropriate definitions. These features hide the underlying details of the OWL representation from the user - another contrasting feature with the built-in editor. Figure 1.2 shows the OBO Explorer tab.

1.2.2 The COBrA-CT Ontology Version Manager

The COBrA-CT Ontology Version Manager allows users to access ontologies that have been published to the community and stored on the ontology server, and to store, manage and share their own ontologies. The version manager implements a simple model for assigning rights to users to allow them to download, upload, and publish ontologies. Guest users can access all public ontologies, while registered users have rights to upload and share their own ontologies. We also plan to consider explicitly representing the ‘process’ of curation in explicit process models, e.g. from authoring, through review, to publication and revision. The ontology curator will require a visualisation of the differences between two versions of an ontology and we can provide this through COBrA’s dual-view capability. The Version Manager is implemented using Grid middleware, developed under the UK e-Science initiative, as we now describe.

Over recent years, the Grid has attracted enormous attention and gained popularity by supporting distributed resources sharing and aggregation across multiple administrative virtual organisations. Compared to the web, the Grid offers upgraded performance in terms of reliability and availability. In COBrA-CT, we developed Grid services to provide data storage and access that allow users to share their ontology information in a more scalable, secure, and dependable way. By enabling COBrA-CT to operate through the Grid, the software capabilities have been enhanced greatly.

The implementation was built on top of Grid middleware, OGSA-DAI. The OGSA-DAI project (`www.ogsadai.org.uk/`), proposed by the University of Edinburgh, is designed to ease access to, and integration of distributed data resources via the Grid. It provides various interfaces supporting data operations, transforming and delivering with many popular (relational or XML) databases, such as Oracle, DB2, SQL Server, MySQL, Xindice, eXist etc., and file systems, such as CSV, BinX, EMBL, OMIM etc. This middleware is based on the GGF-defined OGSI specification and layered on top of the Globus Toolkit implementation. The COBrA-CT currently employs the recently-released WS-RF distribution of OGSA-DAI (OGSA-

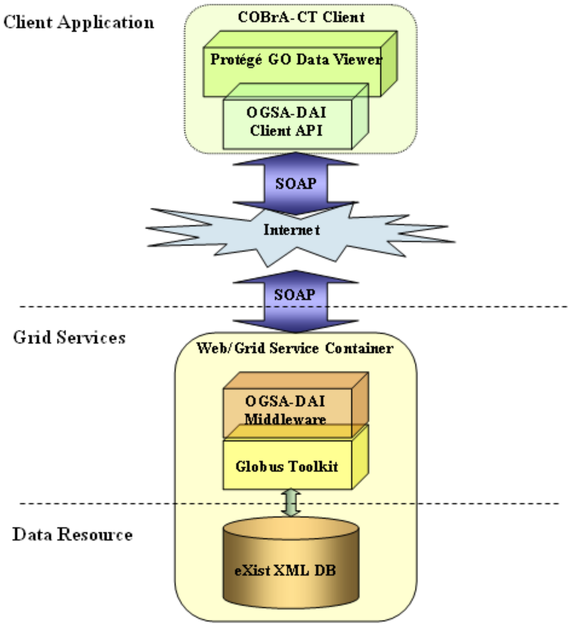


Fig. 1.3. The COBrA-CT system architecture

DAI WSRF 2.2), which has been designed to work with the Globus Toolkit 4 implementation of WS-RF.

The client, shown in Figure 1.3, can be implemented as part of the Protégé plug-in and uses the OGSA-DAI client libraries. Via these interfaces, the client triggers OGSA-DAI activities for uploading and downloading both ontologies and metadata. Both are passed as XML documents. XPath and XUpdate have been applied to query and modify XML database objects. XUpdate supports node-level updating in a DOM tree, which gives much more flexibility and efficiency.

The interaction between OGSA-DAI activities is illustrated in Figure 1.4. The client submits its working plan in a so-called *Perform Document*, which is a XML document consisting of a sequence of requests(*Activities*). The request is sent as encrypted SOAP message to the Grid services, which will invoke *Data Resource Accessors* (DRA) methods to connect with specific data resources. The return datasets or response message are also encrypted in a SOAP message and sent back to the client.

We use eXist (`http://exist.sourceforge.net`), an Open Source native XML database, to store ontology data. Compared to relational databases, the native XML database provides more powerful tools for XML processing, and so is suitable for keeping ontology and metadata information. For example, eXist supports XPath, XQuery, XUpdate, XInclude, XPointer and XSL/SXLT XML standards, and provides XML:DB API, and both DOM and SAX parsers. We also choose the eXist

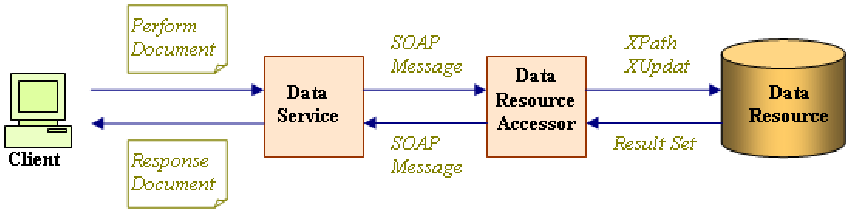


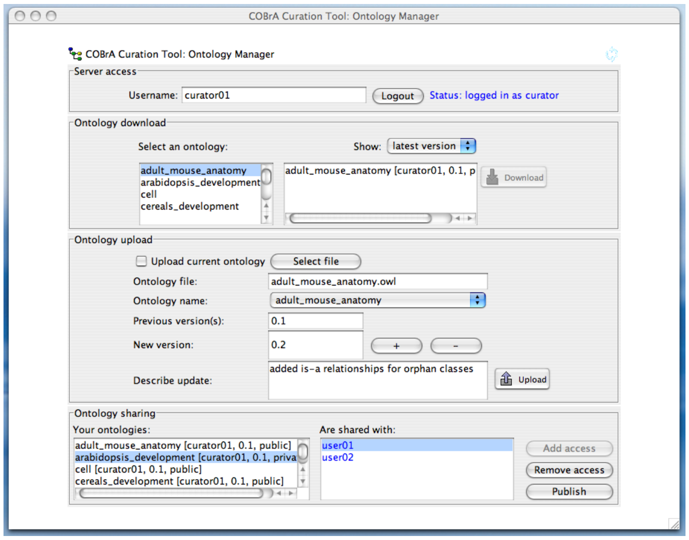
Fig. 1.4. The OGSA-DAI data flow

database because it is able to deal with large XML documents. In COBrA-CT, the ontology files sizes range from 78KB to 10,000KB. Other XML databases, e.g. Apache Xindice (`xml.apache.org/xindice/`) only handle documents less than 5MB, and so cannot satisfy our requirements.

In the eXist database, we store ontology files in hierarchical collections, based on user unique identifiers, ontology identifiers, and ontology version numbers. This means the physical location of a ontology OWL file is determined by these ids. To accelerate data searching, we have implemented a registry to record the ontology and metadata information, and the mapping to the physical location. Current metadata information includes but not limited to:

- Ontology ownership: owner's name, id and database user roll;
- Ontology descriptions: ontology name, a text description of the version;
- Ontology file location: including the XML resource name and subcollection.
- A trace of ontology version changes , including version numbers, upload dates, and a set of previous ontologies that an ontology has been derived from. In the typical case, an ontology will simply have one previous version, but we allow for ontology merging from diverse sources, and for the concurrent editing and subsequent merging of ontology versions.
- Ontology sharing information: COBrA-CT allows a registered user to share his/her ontologies with a group of users. This is supported by associating a set of sharing users with the ontology – these users are able to download the ontology for inspection (and subsequently they may upload a modified version under their own user name). In addition to being shared with specific users, an ontology can be declared to be public, in which case it will be accessible to guest users of COBrA-CT as well as to registered users.

The client component of the Version Manager aims to provide an intuitive interface to the ontology repository. As shown in Figure 1.5, the tool shows the ontologies the user has access to and their versions, allows download and upload, and manages version numbers. User log-in using a password, however, the Grid provides other more secure methods that we shall explore in future work.



COBrA Curation Tool: Ontology Manager

Server access

Username: curator01

Status: logged in as curator

Ontology download

Select an ontology:

adult_mouse_anatomy
arabidopsis_development
cell
cereals_development

Show: latest version

adult_mouse_anatomy [curator01, 0.1, p

Ontology upload

□ Upload current ontology

Ontology file: adult_mouse_anatomy.owl

Ontology name: adult_mouse_anatomy

Previous version(s): 0.1

New version: 0.2

Describe update: added is-a relationships for orphan classes

Ontology sharing

Your ontologies:

adult_mouse_anatomy [curator01, 0.1, public]
arabidopsis_development [curator01, 0.1, priva
cell [curator01, 0.1, public]
cereals_development [curator01, 0.1, public]

Are shared with:

user01
user02

Fig. 1.5. The Version Manager client tool

1.3 Future Work

In future work, we shall address efficiency issues in storing the OWL ontologies. Viewing the ontologies as XML data allows a range of XML techniques to be applied. We can distinguish updates to the ontology structure from updates to the annotations when analysing changes between versions. We also aim to visualise the differences between ontology versions by simultaneously displaying two versions and highlighting the additions and deletions graphically.

The Grid environment can provide a very high level of security covering data transmission and access to services. The Grid offers integrity (i.e. it can ensure that data has not been altered or destroyed since transmission), confidentiality, authentication, and, perhaps most importantly, availability. Currently, we have not made use of all of these features, for example, the use of certificates, and aim to explore alternative security models in future releases of the ontology tools.

Acknowledgements

This work is supported by BBSRC grants BBSRC 15/BEP 17046 and BB/D006473/1.

References

1. Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.

2. Open Biological Ontologies. *http://obo.sourceforge.net*.
3. J.B.L. Bard and S.Y. Rhee. Ontologies in biology: design, applications and future challenges. *Nature Review Genetics*, 5(3):213–222, 2004.
4. P. Lord and A. MacDonald. Data curation for e-science in the uk: an audit to establish requirements for future curation and provision. JISC Report.
5. H. Parkinson et al. The SOFG Anatomy Entry List (SAEL): an annotation tool for functional genomics data. *Comparative and Functional Genomics*, 5(6-7):521–527, 2004.
6. B. Smith, J. Williams, and S. Schulze-Kremer. The ontology of the Gene Ontology. Proc. AMIA 2003.
7. R. Korf. COBrA - a concept ontology browser for anatomy, 2003. Informatics Report EDI-INF-IM030022.
8. J.S. Aitken, B.L. Webber, and J.B.L. Bard. part-of relations in anatomy ontologies: A proposal for RDFS and OWL formalisations. In *Proc. PSB*, pages 166–177, 2004.
9. P. Buneman, S. Khanna, K. Tajima, and W.J.S. Tan. Archiving scientific data. In *Proc. ACM SIGMOD*, pages 1–12, 2002.
10. P. Buneman, S. Davidson, W. Fan, C. Hara, and W. Tan. Keys for xml. In *Proc. WWW 10*, pages 201–210, 2001.
11. N. Noy and M. Musen. The PROMPT suit: Interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, 59(6):983–1024, 2003.
12. S. Aitken. Formalising concepts of species, sex and developmental stage in anatomical ontologies. *Bioinformatics*, 21(11):2773–2779, 2005.