Managing the transition from OBO to OWL: The COBrA-CT Bio-Ontology Tools

Stuart Aitken, Yin Chen, Bonnie Webber, Wenfei Fan and Jonathan Bard

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

Abstract

This paper presents the COBrA-CT ontology tools, which include an ontology server database and version manager client tool for collaborative ontology development, and an editor for bio-ontologies that are represented in the Web Ontology Language (OWL) format. The ontology server uses OGSA-DAI Grid technology to provide access to the ontology server database. These tools implement the agreed standard for representing Open Biomedical Ontologies (OBO) in OWL and interoperate with other tools developed for this standard. Such tools are essential for the uptake of OWL in the biomedical ontology community.

1. Introduction

Biomedical ontologies, which include the Gene Ontology and the anatomies of model organisms such as mouse and drosophila, are growing in size and their encoding languages are becoming more sophisticated. As a consequence, tools for verification, version control, meta-data attribution, provenance and archival are needed, in addition to the ontology editing tools that users are familiar with. This wider curation aspect has been recognised as a priority for e-Science, and is an important concern for many communities and in standards initiatives (Lord, 2003, 2004).

Ontologies are of central importance in data curation, as only by defining the meaning of the terms used to describe data points or fields can the (possibly implicit) content be clarified and its interpretation agreed upon among the research community, and thereby used consistently. For example, there are efforts to standardise the names used for tissue samples assayed by microarray (Parkinson, 2005), as well as the metadata that describes the experimental results (e.g. MGED/MIAME).

More generally, use of a consistent, shared ontology is of critical importance to the sharing of knowledge, and has long-term benefits. 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 that must be accounted for in a systems level view of biology. Ontologies have also been identified as key resources in numerous e-Science projects, including AstroGrid, MyGrid and the Advanced Knowledge Technologies IRC.

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. And, as we now discuss, biomedical ontologies, which are often defined rather informally, must be translated into the more formal languages of the Semantic Web (e.g. OWL) in order to take advantage of the range of tools and services being developed for the Semantic Web and computational Grid. This requires the meaning of terms and relationships to be clarified and the development of supporting tools. We present an ontology editor for biomedical ontologies that have been translated into OWL, and an ontology management system that supports the distributed, cooperative development of ontologies, which, in combination, can be used to transition Open Biological Ontologies to the Web Ontology Language and address the version management tasks that arise in the process. These tools interoperate with tools developed by others that perform the OBO format to OWL format translation according to an agreed standard. We now discuss the languages and tools used to develop bio-ontologies, then consider the Grid as a platform for such tools.

1.1 Bio-ontology Languages and Tools

Biomedical 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, and conceptual abstractions for aggregating results (Bard and Rhee, 2004).

Bio-ontology tools typically provide a graphical interface to a (largely) fixed set of ontology language constructs that specify the

term name, textual definition, synonyms etc, plus search functions to help the user find terms within the ontology. Tools are typically used outwith a methodological framework, i.e. users are not following a modelling process or paradigm of any kind. A gap between the theoretical understanding of the formal issues of ontology development and the modelling approaches apparent in many of the Open Biomedical Ontologies has previously been noted (Smith, 2003, 2005). Initiatives such as the Common Anatomy Reference Ontology[1] have the potential to make modelling practice more uniform across different user groups, and may be the most practical solution to clarifying the conceptual basis of bio-ontologies. However, to-date, many of the modelling decisions captured in biomedical ontologies 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 ontologies, including the interpretation of the *part-of* relation (but note that these comments do not apply to the Gene Ontology which has specified principles for curation).

Use of the OWL language brings with it the need to clarify the meaning of relations other than the *is-a* (or *subClassOf*) relationship: The OWL ontology developer is immediately exposed to the Description Logic oriented view of a concept definition. This logic-based view of concept definitions has implications for ontology editor design as a user who is a 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 logical definitions. The logic-based view of concept definitions is most at odds with current anatomy ontologies where (in many cases) the more pragmatic view of the ontology as a labelled graph holds sway over the logic-oriented view that all concepts require *subClassOf* relationship.

Recent developments in the translation of OBO to OWL are bringing this issue to the fore: In OWL ontologies, the *part-of* relation cannot be used to link between two concepts in the ontology graph. Rather, it is used to specify the set of parts of the parent entity, e.g. the parts of some Heart. The child entity, e.g. the Aortic Valve will be a subclass of this set. Presenting such a definition to a user in an intuitive way is a significant challenge for new tools that are based on OWL.

[1] http://www.bioontology.org/wiki/index.php/CARO:Main_Page

1.2 Platforms and Infrastructure: The Grid

Ontologies are recognised as having a key role in data integration on the computational Grid. Metadata standards can themselves be considered to be a kind of ontology, and may characterise resources in terms of domain ontology concepts for the purposes of integration, service discovery, provenance, and long-term preservation. Conversely, the Grid provides an ideal platform for new ontology tools and databases as it aims for seamless resource sharing and global collaborations. The Grid has attracted enormous attention and gained popularity by supporting distributed resource sharing and aggregation across multiple administrative virtual organisations. It offers upgraded performance in terms of reliability, scalability and availability.

In the COBrA-CT project, we have developed Grid services to provide data storage and access so that users can share their ontologies in a secure, and dependable way. By enabling COBrA-CT to operate through the Grid, the software capabilities have been enhanced by taking advantage of Grid infrastructure. The following sections describe our solution to the curation and archiving problems that arise when individuals and communities develop ontologies, then introduce the ontology editor and its functions.

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, and archiving, curation also includes the review of the content of the ontology, and assessment of quality (Missier, 2005). As the use of ontologies widens, the problems of tracking versions, and the changes between versions, identifying flaws and of reconciling differences in conceptual modelling arise. Addressing the first of these issues is our main goal in the design of the curation tools.

Supporting the ontology development and curation effort in a distributed setting, providing access to current and past versions of ontologies and allowing collaboration among users requires an ontology management server. As we are making 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 use of an

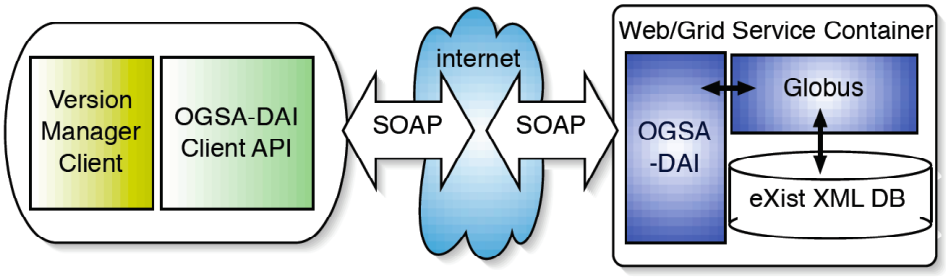

Fig. 1. The Client/Server Grid Architecture

XML database will also support querying across ontologies, for example, for concepts and synonyms. A simple CVS or Wiki solution would not provide such functionality. The further potential for XML-based methods is discussed under future work.

We next present the Ontology Management Server and the Version Manager, then describe the Protégé plug-in for editing OWL bio-ontologies: the OBO Explorer. These tools can be downloaded from the project website: http://www.aiai.ed.ac.uk/project/cobra-ct/

2.1 The COBrA-CT Ontology Management Server

The Ontology Management Server is built on OGSA-DAI. The OGSA-DAI project[2] aims to ease access to, and ease the integration of distributed data resources via the Grid. It provides various interfaces supporting data transformation and delivery, and is compatible with many popular (relational or XML) databases, such as Oracle, DB2, SQL Server, MySQL, Xindice, and eXist, and file systems, such as CSV, BinX, EMBL, OMIM. This middleware is compliant with two popular web services specifications, WS-I and WSRF, and is distributed with both the Globus Toolkit and the OMII-UK middleware distribution. The COBrA-CT installation currently employs the recently-released WS-RF distribution of OGSA-DAI (OGSA-DAI WSRF 2.2), which has been designed to work with the Globus Toolkit 4 implementation of WS-RF.

We use eXist[3], an Open Source native XML database, to store ontology data. Native XML databases provide powerful tools for XML processing, and so are 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 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[4] could only handle documents less than 5MB, and so did not satisfy our requirements.

As indicated in Fig. 1, the client triggers OGSA-DAI methods (*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 client submits its working plan in a so-called OGSA-DAI Perform Document, which is a XML document consisting of a sequence of requests. 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 messages are also encrypted in a SOAP message and sent back to the client.

Ontology files are stored in hierarchical collections based on user unique identifiers, ontology identifiers and ontology version numbers in the eXist database. This means the physical location of an 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

[2] http://www.ogsadai.org.uk

[3] http://exist.sourceforge.net

[4] http://xml.apache.org/xindice/

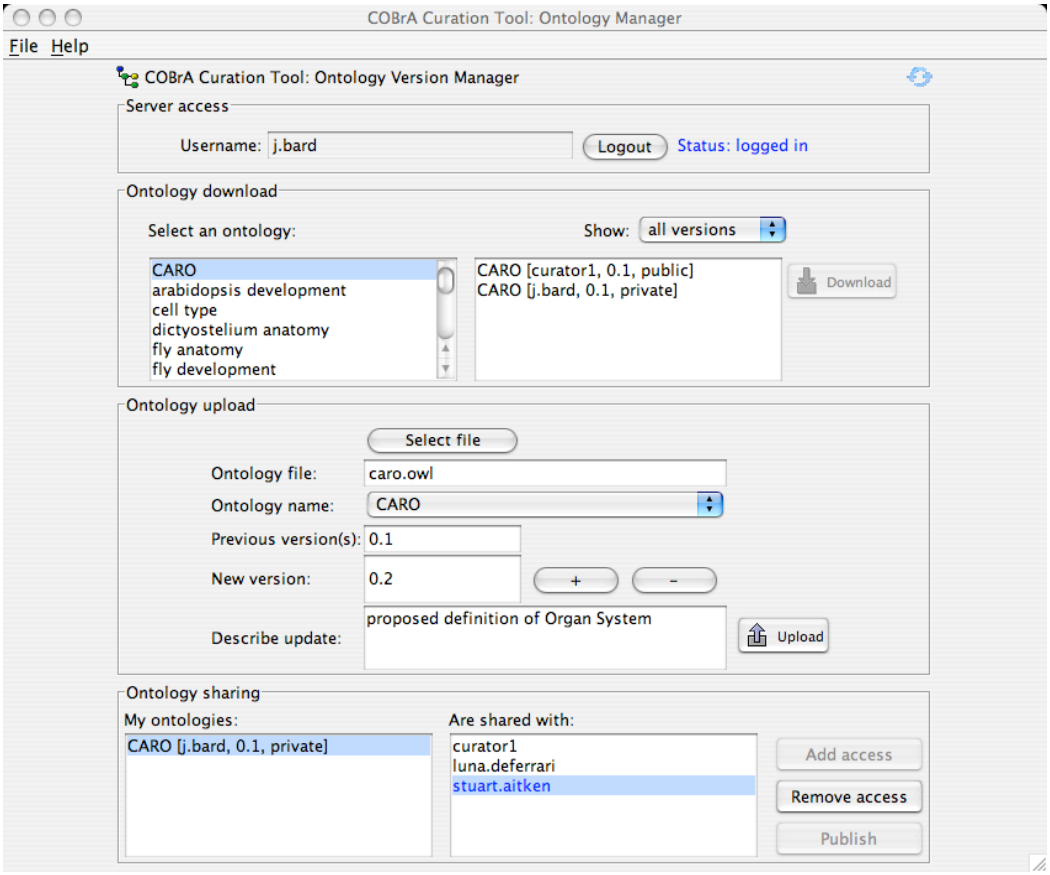

Fig. 2. The Ontology Version Manager Client

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).

A simple Java tool has also been developed to help the database administrator manage user accounts.

Security: Several options for maintaining security were explored. The simplest is for users to log in using their account name and password, and for these to be verified against the database records. This approach is currently used in version 1.0 of the Version Manager client. We have also explored a public/private key system which eases the user's account management problems by replacing passwords with key files, and allows authentication checks for the client. The Certification Authority (CA) method was also examined, however, we concluded that this is overly complicated and inefficient for our needs, and that it is unreasonable to ask all our users go to the relevant certifying organisations to obtain their CAs. In a small to middle scale Grid system, like COBrA-CT, it seems more applicable to self-issue CAs, and we shall explore this in future work.

2.2 The COBrA-CT Ontology Version Manager

The motivation for the design of the Version Manager came from observing the development of the cell type ontology (Bard et al, 2005) where ontology versions were created by a small, geographically-dispersed and informally organised group who might meet at a conference to create and review content or exchange views and ontology files by email. We concluded that supporting this process would be best achieved by lightweight, easy-to-use client tools. In contrast, supporting a fully-fledged standards initiative (e.g. where there is a committee structure and members have roles

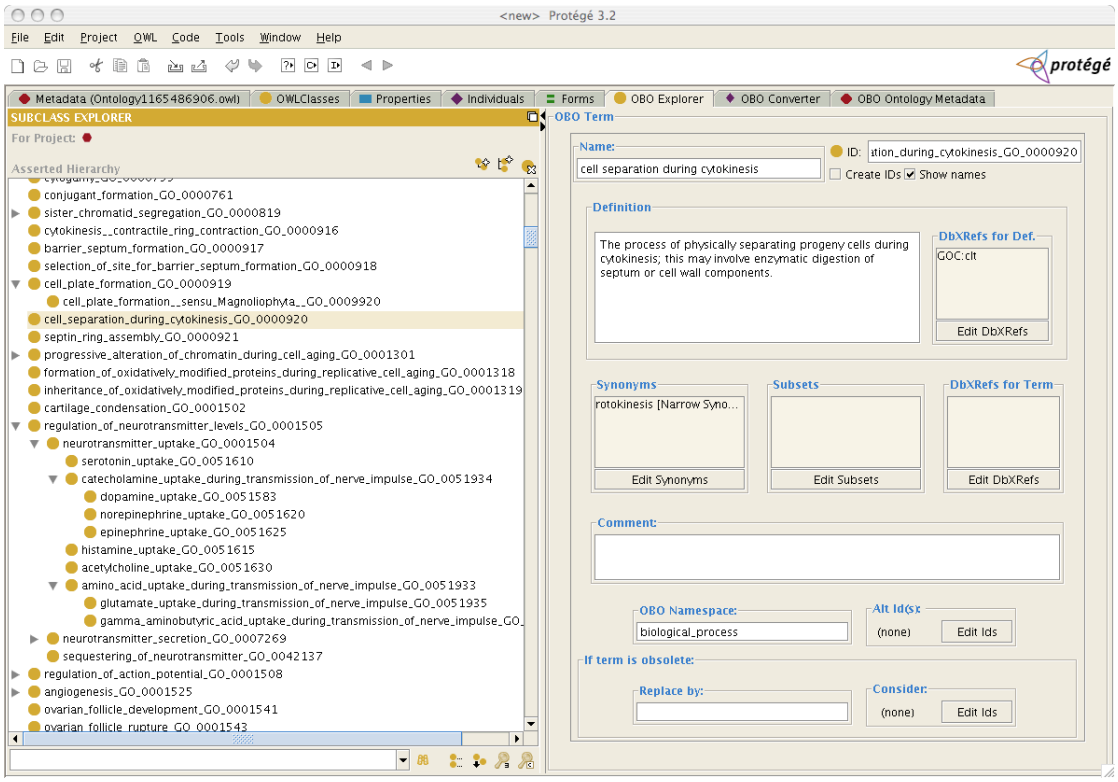

Fig 3. OBO Explorer Protégé tab

and rights etc[5]) might require designated persons such as the chair, and an explicit representation of the process.

The Ontology Version Manager is a client tool that 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 management system 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.

The client component of the ontology management system aims to provide an intuitive interface to the ontology repository. As shown in Figure 2, the tool shows the ontologies the user has access to, and their versions, allows download and upload, and manages version numbers. In this instance, the user (j.bard) has created a private version of the CARO ontology which is shared with stuart.aitken (CARO version 0.1 private) as indicated by the *Ontology sharing* panel. For simplicity, users are not organised into explicitly-named groups. Instead, users give access to others on an individual basis. This user is in the process of uploading version 0.2 of their private version of CARO as indicated by the *Ontology upload* panel. In addition to being shared with specific users, an ontology can be *Published*, in which case it will be accessible to guest users of COBrA-CT as well as to registered users.

Having described the archival and sharing of ontologies, we now describe the editing tools that allow the user to create and modify OBO ontologies in OWL (thereby creating new versions).

2.3 The OBO Explorer

A representation for OBO ontologies in OWL has been agreed[6] and tools for automatically converting ontologies from OBO to OWL, and for reading OBO ontologies into the Protégé 3 ontology editor[7] have recently been developed. The OWL representation of OBO, which we helped establish, is referred to as OBO-in-OWL.

OBO-in-OWL succeeds in capturing all of the content of OBO ontologies, both the logical structure and the informal annotations, e.g. synonyms and database cross-references

[5] http://www.w3.org/Consortium/Process-20010719/

[6] http://www.bioontology.org/wiki/index.php/OboInOwl:Main_Page

[7] http://www.bioontology.org/tools/oboinowl/obo_converter.html

(DbXRefs). Naturally, we want users to be able to edit all aspects of a term's definition.

However, Protégé 3 is unable to display the annotations associated with OBO terms using the default interface configuration, and therefore the user cannot edit this information. Hence we implemented the OBO Explorer. This tool is distributed as a Protégé tab, compatible with other Protégé components, and contributing to the large user community that supports Protégé development[8].

The OBO Explorer interface is implemented as a tab that presents the class hierarchy on the left hand panel, and term annotations on the right. Where possible, the user interface components are present on the main panel, and immediately update the underlying OWL model. The synonym, subset and DbXRef information is displayed in list form in a concise manner to enable users to see all this information in one place. These data are edited by calling up dialogs that allow new items to be added and existing items to be deleted from the lists. Figure 3 shows the OBO Explorer tab.

When an OWL ontology is being created from scratch, it will lack the agreed OBO-in-OWL classes and relationships needed to represent OBO terms. In this case, the tool creates the appropriate definitions for these elements. These features hide the underlying details of the OWL representation from the user – a contrasting feature with the built-in editor.

In on-going work, we are developing a browser component that, for a selected term, shows the entities that the term is necessarily part of, and the parts that the term necessarily has. These two types of assertions are conjunctions in the definition of the selected term (and are shown in the built-in class editor in Protégé). This browser also shows references to the selected class from other classes. That is, the tool searches the definitions of other classes to find entities that are, by definition, necessarily part of the selected class, or have the selected class as a part. The existing interfaces to not provide this functionality, which we have already found useful - uncovering errors in the conversion of the Foundational Model of Anatomy ontology to OWL, and making explicit the differing approaches formalising the part-whole relation.

2.4 Evaluation

A simple task-based evaluation of the OBO Explorer is underway. Users are asked to install the tool and perform a number of searching and editing operations. The trial addresses the installation and configuration tasks as these involve navigating the numerous dialogs that Protégé users must complete when opening an ontology and adding tabs to the interface. The results will indicate whether the OBO Explorer should be packaged such that these steps are avoided.

It is also important to investigate whether the translation from OBO to OWL causes confusion to users, for example, between OBO term names/IDs and OWL URIRefs. On translation to OWL, the unique term ID (for example, GO:0000920) becomes the local name in the URI, and will be used as the label for the concept in Protégé's display (in the default configuration). However, the user will expect to see the term name ‘cell separation during cytokinesis’. The OBO Explorer has a feature to cause the name to be displayed with the term ID as a postfix (shown in Figure 3), and we are interested to investigate any problems in the use of URIs and the usability of the features provided.

A potentially more significant change that the user will observe is the displacement of all terms that have no *is-a* definition (in the original OBO) to the top level of the OWL ontology. For ontologies that are *is-a* complete such as the Gene Ontology, which completed the process of assigning *is-a* parents to all terms in January 2007, there will be no change in the taxonomic structure. But for other ontologies (and for the OBO anatomies in particular), the user will see that the taxonomy is deficient. The tool allows this problem to be solved, but, in certain cases, significant effort will be required to complete the transition of an ontology from OBO to the more formal OWL representation.

Initial results from the trial suggest that the Protégé configuration task is time consuming for users (and barrier for some). The OBO Explorer tab follows the Protégé interface style where changes to text fields are confirmed by typing return, however, this was noted as being inconvenient. The procedure for generating new term IDs was not sufficiently clear. Overall, users completed the tasks successfully.

3. Related Work

In KAON, a comprehensive infrastructure for ontology management[9], ontology edits are stored in an ‘evolution log’ that also records metadata about the author’s identity and a

[8] http://protege.stanford.edu

[9] http://kaon.semanticweb.org/documentation

description of the change. This level of tool integration (not easily achived in Protégé's plug-in architecture) allows changes to be reversed. The semVersion approach to ontology versioning (Völkel, 2005) is based on the RDF representation that OWL is layered on. The RDF layer can be analysed for structural changes in the graph - a task that is complicated by the existence of 'blank nodes' (unnamed nodes which may have no semantic type). Semantic diffs (i.e. comparisons between two versions of the same ontology) are computed accounting for the semantics of the ontology language. This approach has been implemented as a Protégé tab (Groza, 2006). Structural diffs between versions of ontologies can also made in Protégé using the Prompt tools (Noy, 2004). The latter approaches stress the importance of visualising changes between versions. In addition, Prompt supports the process of accepting and rejecting individual changes to class definitions. Protégé also has plug-ins for project management and database connectivity. OBOEdit[10], an ontology editor supported by the Gene Ontology consortium, now has an OWL import/export facility that is based on the same code as the OBO Converter Protégé tab[7] and so has comparable functionality. The adoption of a client server model, and the improved treatment of annotations in Protégé 4 also has parallels with our design.

4. Conclusions and Future Work

The COBrA-CT version manager tools allow any ontology that can be saved in an XML syntax, including all RDF and OWL ontologies, to be stored centrally, shared among developers via internet connection to the database, and managed throughout its development lifecycle. The editor tool provides specific support for bio-ontologies in the OWL format. The translation of bio-ontologies to OWL requires such tools, both for making the modifications required in the formal OWL-DL language, and for organising the development effort among multiple users.

In future work, we shall re-examine efficiency issues in storing the OWL ontologies. Viewing the ontologies as XML data allows a range of XML techniques to be applied. It has been noted that changes to scientific data archives are accretive - most changes are additive - although deletion and modification also occur (Buneman, 2002). Scientific data is typically structured hierarchically, allowing a hierarchical key structure to be exploited in archiving changes to the data. The central notions of hierarchical organisation, objects and timestamps described in (Buneman, 2001) also apply to ontologies and ontology management and can be expected to improve efficiency. The specification of an XML schema for OWL 1.1 widens the potential for applying these methods to ontologies.

Explicitly modelling the ontology development and publication lifecycle, and deriving measures of ontology quality (in analogy to Misser (2005)) are the next methodological steps that our tools should support.

Acknowledgements This work is supported by BBSRC grant BB/D006473/1

[10] http://oboedit.org/

References

Bard, J.B.L. and Rhee, S.Y. (2004) Ontologies in biology: design, applications and future challenges. *Nature Review Genetics* 5(3) :213-222.

Bard, J.B.L., Rhee, S.Y. and Ashburner, M. (2005) An ontology for cell types. *Genome Biology* 6:R21 doi:10.1186/gb-2005-6-2-r21

Buneman, P., Davidson, S., Fan, W., Hara, C. and Tan, W. (2001) Keys for XML. *Proc. WWW 10* :201-210.

Buneman, P., Khanna, S., Tajima, K. and Tan, W.J.S. (2002) Archiving scientific data. *Proc. ACM SIGMOD*

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

Groza, T., Völkel, M. and Handschuh, S. (2006) Semantic Versioning Manager: Integrating SemVersion in Protégé. Proc 9th International Protege Conference Stanford, California.

Lord, P. and MacDonald, A. (2003) Data curation for e-Science in the UK: an audit to establish requirements for future curation and provision. *JISC Report*.

Lord, P. and MacDonald, A., Lyon, L. and Giaretta, D. (2004) From data deluge to data curation. *Proc 3rd e-Science All Hands Meeting* :371-375.

Missier, P., Embury, S., Greenwood, M., Preece, A. and Jin, B. (2005) An Ontology-Based Approach to Handling Information Quality in e-Science, *Proc 4th e-Science All Hands Meeting*.

Noy, N. F., Kunnatur, S., Klein, M. and Musen, M.A. (2004) Tracking Changes During Ontology Evolution *Proc Third International Conference on the Semantic Web* (ISWC-2004), Hisroshima, Japan :259-273.

Open Biomedical Ontologies
http://obo.sourceforge.net

Parkinson, H., Aitken, S., Baldock, R.A, Bard, J.B.L., Burger, A., Hayamizu, T.F., Rector, A., Ringwald, M., Rogers, J., Rosse, C., Stoeckert Jr, C.J. and Davidson, D. (2004) The SOFG anatomy entry list (SAEL): an annotation tool for functional genomics data. *Comparative and Functional Genomics* 5,(6-7) :521-527.

Smith, B. Williams, J. and S. Schulze-Kremer, S. (2003) The Ontology of the Gene Ontology *Proc. AMIA 2003*

Smith, B., Ceusters, W., Klagges, B., Köhler, J., Kumar, A. Lomax, J., Mungall, C., Neuhaus, F., Rector, A.L. and Rosse, C. (2005) Relations in biomedical ontologies. *Genome Biology* 6:R46.

Völkel, M., Winkler, W., Sure, Y., Kruk, S.R. and Synak, M. (2005) SemVersion: A Versioning System for RDF and Ontologies *Proc. 2nd European Semantic Web Conference* ESWC'05.