

*A Methodology to Migrate the Gene Ontology to a Description Logic Environment Using
DAML+OIL*

C.J. Wroe, R. Stevens, C.A. Goble, M. Ashburner

Pacific Symposium on Biocomputing 8:624-635(2003)

A METHODOLOGY TO MIGRATE THE GENE ONTOLOGY TO A DESCRIPTION LOGIC ENVIRONMENT USING DAML+OIL

C.J. WROE, R. STEVENS, C. A. GOBLE

*Department of Computer Science, University of Manchester,
Oxford Rd, Manchester, M13 9PL, UK
{cwroe robert.stevens}@cs.man.ac.uk*

M. ASHBURNER

*EMBL – European Bioinformatics Institute,
Wellcome Trust Genome Campus,
Hinxton, Cambridge CB10 1SD, UK*

The Gene Ontology Next Generation Project (GONG) is developing a staged methodology to evolve the current representation of the Gene Ontology into DAML+OIL in order to take advantage of the richer formal expressiveness and the reasoning capabilities of the underlying description logic. Each stage provides a step level increase in formal explicit semantic content with a view to supporting validation, extension and multiple classification of the Gene Ontology. The paper introduces DAML+OIL and demonstrates the activity within each stage of the methodology and the functionality gained.

1 Introduction

The Gene Ontology Consortium set out to provide ‘a structured precisely defined common controlled vocabulary for describing the roles of genes and gene products in any organism’.¹ The resulting, publicly available, Gene Ontology (GO) has become the *defacto* standard used to provide ~250,000 annotations for entries in at least 14 major bioinformatics databases. GO has been successful in supporting the needs of molecular biologists due to its comprehensive coverage in a relatively simple but consistent structure acceptable to the biological communities. However, its growing success and size now leads to several challenges for ongoing manual curation.

The Gene Ontology Next Generation project (GONG) aims to demonstrate that, in principle, migrating to a finer grained formal conceptualization will allow computation techniques such as description logics to aid in the curation and delivery of the ontology. This migration must be practical. Providing a fine-grained conceptualization in a formal language is a significant knowledge acquisition process and it is unrealistic to approach it as a one-off effort. We aim to prove the exercise can be undertaken in a staged manner, both in terms of number and granularity of formal concept definitions, with useful benefits received at each increment. The paper is organized as follows: This section continues with an introduction to the existing structure and use of GO, and the challenges it faces. Section 2 provides an overview of the ontology language DAML+OIL. Section 3 provides an overview of the methodology we propose and then a detailed look at

each stage examining its aim, procedure and results. We conclude with a discussion of the current status of the project and plans for the future.

1.1 *Current structure and use of the Gene Ontology*

The Gene Ontology (GO) is split into three orthogonal sub-ontologies containing a total of about 11,000 concepts. The ‘cellular component’ ontology is used to annotate the location at which a gene product acts. ‘Molecular function’ terms are used to annotate the specific capabilities of a gene product, while ‘biological process’ terms capture the higher order processes in which the gene product is involved. GO is more than a controlled vocabulary. The aim is to associate a textual definition to each term to promote an explicit shared understanding and currently 60% of terms have such a definition. Each term is also placed in a directed acyclic graph (DAG) allowing multiple parents both along ‘is-a’ relationships and ‘part-of’ relationships. The hierarchical arrangement of terms is primarily used by humans rather than software to accomplish three main tasks:

1. **Query/ browse bioinformatics databases.** GO can act as an index into databases. GO Browsers, e.g. AmiGO (<http://www.godatabase.org>) allow users to link directly from the hierarchical view of the ontology to database entries annotated with those terms.
2. **Interpret results.** GO annotations provide biologists with more meaningful yet concise alternatives to the cryptic abbreviations used to label experimental results and so help in interpretation of the large data sets, e.g. microarray data.¹
3. **Aggregate information.** A GO Slim is a non-overlapping subset of high-level GO terms. Aggregating all entries annotated with hierarchical descendants of each GO Slim term can produce useful summary statistics. Several GO Slims have been created to aggregate different sets of annotations for different purposes.^a The ‘GO summary’ feature of the AmiGO browser demonstrates how this information is used to provide a high level view of GO annotation statistics.

The range of applications of GO is constantly growing,^b which places increasingly exacting requirements on GO’s internal structure as detailed below:

1. **Multiple classification and consistency.** There are multiple ways to organize terms in a classification. The exact choice depends on the task at hand. Multiple classification within GO is currently maintained by hand but experience from the medical domain has shown that numerous parent-child links are omitted in such hand crafted, phrase based controlled vocabularies.² While of less importance to manual interpretation, machine interpretation will falter in the face of such inconsistencies.
2. **Extension.** There is a growing desire to extend the content of processes such as embryonic development. The effort to manually pre-enumerate and maintain the

^a A list of GO Slims can be found at ftp://www.geneontology.org/pub/go/GO_slims

^b A list of published uses of GO can be found at <http://www.geneontology.org/doc/GO.biblio.html>

cross product of developmental processes against all anatomical structures in every organism would be immense.

3. **Machine interpretation.** Biologists are able to interpret information both within term names and the lexical definitions. However this implicit information is inaccessible to computer applications. The hierarchical structure of GO has been used for automated processing.³ However, the definition of a concept is only implicitly and incompletely encoded by its hierarchical position, e.g. for 'protein kinase C' (GO:0004697), its parentage implies it is an 'enzyme function' that can 'transfer a phosphorous containing group to an alcohol group', is a 'serine/threonine kinase', and is a 'phorbol ester receptor'. However, we cannot synthesize a complete formal definition from this information.

Many formal ontology representation languages have been developed in the AI community to capture formal concept descriptions including frame-based systems, conceptual graphs and description logics (DLs). DLs offer a new paradigm in modeling vocabulary. Rather than annotate manually classified concepts with additional properties, explicit concept definitions actually form the basis for calculating a classification or checking the logical consistency of an existing classification.

2 DAML+OIL

DAML+OIL arose from EU and US DARPA research programs and is currently undergoing standardization through the W3C WebOnt activity,^a to become the Ontology Web Language (OWL). Irrespective of its reasoning capabilities it is becoming a standard language for ontology interchange. As an interchange language it has been designed to encode a wide range of ontologies from taxonomies, frame based ontologies, to ontologies that include logic based concept definitions. This flexibility allows the staged evolution of an ontology within a single representation, greatly simplifying the process.

Within a DAML+OIL ontology each concept is represented as a class. At its simplest, DAML+OIL allows each class to be placed in a taxonomy with the use of the subclass relationship e.g.

```
class isocitrate dehydrogenase (NAD+) (GO:0004449) b
  subclassOf 'oxidoreductase, acting on the CH-OH group of donors, NAD or NADP as
  acceptor' (GO:0016616)
```

Classes can be further described (or restricted in DAML+OIL terms) by their attributes, specified as property/value pairs, e.g.

```
class isocitrate dehydrogenase (NAD+) (GO:0004449)
  restriction onProperty has_substrate hasClass isocitrate
```

^a <http://www.w3c.org/2001/sw/WebOnt/>

^b Space precludes the reproduction of DAML+OIL in its standard XML format. An abridged non-standard format is used the purposes of illustration.

Both universal and existential quantification can be used to represent such definitions, as ‘carbohydrate metabolism is the metabolism of some carbohydrate and only carbohydrate’.

```
class carbohydrate metabolism (GO:0005975) defined
  subclassOf metabolism
  restriction onProperty acts_on hasClass carbohydrate
  restriction onProperty acts_on toClass carbohydrate
```

Each restriction can also be associated with numerical cardinality constraints:

```
class tricarboxylic acid defined
  subclassOf organic acid
  restriction onProperty has_part 3 (carboxyl group or carboxylate group)
```

The above class ‘tricarboxylic acid’ can be specified as *defined* because the description completely captures its definition and as such its place in the classification can be inferred by merit of its definition using description logic reasoners such as FaCT.⁴ Note also the use of anonymous embedded expressions and logical operators ‘(carboxyl group or carboxylate group)’, which provides greatly increased expressive power with respect to standard frame-based languages.

Horrocks⁴ gives a more detailed description of the capabilities of DAML+OIL and Stevens⁵ describes the use of DAML+OIL in capturing molecular biology domain knowledge with a high degree of fidelity.

3 Methodology

The methodology is designed to embrace evolution not revolution. We have therefore partitioned development into well-defined stages. At each stage we increase both the quantity and complexity of the explicit semantic content by incremental extension of class descriptions. Figure 1 illustrates the five steps involved and the resources involved at each stage. Step 0 is a foundation stage in which GO is translated into a DAML+OIL ontology. Step 1 uses DL reasoning to group related components based on part-of relationships specified in the current GO. Step 2 programmatically creates partial class descriptions from existing structured information in bioinformatics databases, enabling the grouping of existing terms under abstractions which could form a novel GO Slim. Step 3 manually completes the partial descriptions of step 2 to enable the reasoner to check the consistency of the existing hierarchy and detect missing is-a relationships. Step 4 allows annotation applications to dynamically extend GO as required. At each stage it should be possible to re-express a subset of the information within the DAML+OIL ontology in the original GO XML format enabling existing applications to take advantage of reorganized hierarchies and additional concept information. The feedback of results as a static snap-shot is similar to the creation of thesauri from description logic ontologies described by Bechhofer *et al.*⁶

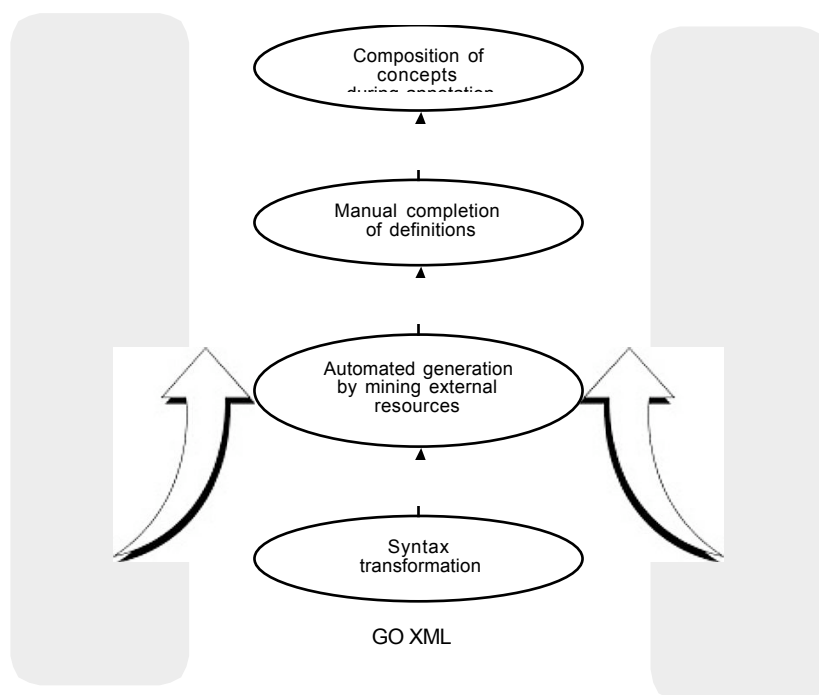


Figure 1. Overview of the staged migration described in this paper.

3.1 Materials

The XML version of GO released January 2002 (<http://www.godatabase.org/dev/database/archive/2002-01-01/>) was used throughout the work described in the paper and all references are to that version. OilEd version 3.4 (<http://oiled.man.ac.uk>) was used to edit DAML+OIL ontologies, and provided the DAML+OIL data structures manipulated by scripts. OilEd also provided the GO XML to DAML+OIL conversion capability. The COHSE ontology server (<http://cohse.semanticweb.org>) provided an API to link to the FaCT reasoner, and provided a server to demonstrate client side composition of ontology concepts. DAGEdit version 1.302 (<http://www.geneontology.org/#tools>) was used to browse the current Gene Ontology in its native format.

The KEGG enzyme database (downloaded 17/05/02 from <http://www.genome.ad.jp/kegg>) was used to extract enzyme substrate, product and cofactor information. BioPython (<http://www.biopython.org>) was used to parse KEGG enzyme flat file format and load into a MySQL database (<http://www.mysql.com>). UMLS knowledge sources 2002AA (<http://www.nlm.nih.gov/research/umls>) loaded into a MySQL database were used as a source of the MeSH chemical taxonomy. Lexical

tools bundled with UMLS version 2002AA were used to lexically normalise chemical terms in the KEGG enzyme database.

Jython version 2.1 (Java Python, <http://www.jython.org>) was used as the scripting environment, with which to integrate large-scale programmatic manipulation of DAML+OIL ontologies, database queries and lexical tools.

Step 0. Transforming GO XML into DAML+OIL

GO is not currently published in DAML+OIL, so the first stage of any migration must be a syntactical transformation from an available format, e.g. GO XML into DAML+OIL. The transformation involves the simple mapping of XML elements to equivalent constructs in DAML+OIL as shown in Table 1.

Table 1. Mapping between GO XML and DAML+OIL

GO XML	DAML+OIL
<code><go:term></code>	<code><daml:Class></code>
<code><go:isa></code>	<code><daml:subClassOf><daml:Class></code>
<code><go:part-of></code>	<code><daml:subClassOf><daml:Restriction></code> <code> <daml:onProperty><daml:ObjectProperty</code> <code> rdf:resource="go:part-of"/></code> <code><daml:hasClass><daml:Class></code>

Not all GO terms have a subsumption relationship (orphan terms), but instead are related to another term by a part-of relationship. Formal ontologies require the majority of concepts to be at least a kind of one other concept, as the is-a (subsumption) network forms a key substrate on which reasoning occurs. At this stage our solution is simply to add three additional abstractions during the transformation: 'part_of_cellular component', 'part_of molecular function' and 'part_of biological process'. Orphan terms become a kind of one of these respective abstractions.

This purely syntactic step paves the way for future work, but there is no additional functionality gained at this stage.

Step 1. Reasoning over existing semantic information

In the previous stage we placed all orphan terms under at least one parent, e.g. 'part_of_cellular component'. When viewed as a hierarchy these orphan terms form a long unorganized list in which it is difficult to associate related terms. Native Gene Ontology browsers, such as AmiGO, circumvent this problem by presenting both the 'is-a' and 'part-of' relationships as parent-child links within the same tree structure. Terms subsumed by nothing (orphan terms) can still be visually related to the structures or processes that contain them (Fig. 2a). To replicate this organization

within a pure is-a hierarchy requires the addition of abstractions that group together terms that are ‘part-of’ of a common structure. This would be a laborious task to undertake by hand, but it can be straightforward to achieve using a DL reasoner.

To demonstrate this step, we manually identified 20 biologically significant cellular structures that have numerous components specified in the cellular component ontology. We then added 20 corresponding DAML+OIL classes to the ontology in order to group those components, e.g.

```
'component of mitochondrion' defined
  subclassOf cellular component
  restriction onProperty part-of hasClass mitochondrion
```

Only the definitions were manually created. The grouping of terms was achieved by submitting the ontology with newly defined classes to the FaCT reasoner, which automatically inferred the required is-a links. This creates a similar organization to that displayed in native GO browsers, but is based purely on is-a links. Figure 2 shows the evolution of the hierarchy focused on the GO term ‘TCA cycle enzyme complex’ (GO:0030062). These novel abstractions are for organization only and should not be used for annotation. Therefore metadata should be applied to these abstractions, to prevent their use in annotation tools.

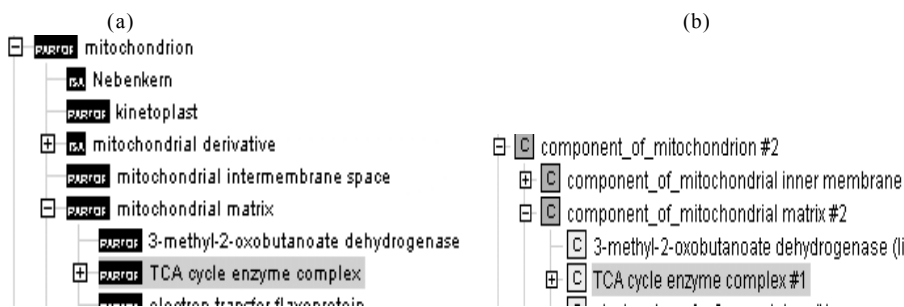


Figure 2. Two screenshots showing an extract of the hierarchical position of ‘TCA cycle enzyme complex’ (GO:0030062) as shown in (a) DAG edit and (b) Oiled after addition of ‘component of’ abstractions and inference of new subsumption relationships using the FaCT reasoner.

Step 2. Programmatically adding partial descriptions from other sources

Step 1 allowed the classification of GO terms using existing ‘part-of’ information. The creation of further novel abstractions grouping current GO terms in alternative ways requires the addition of the relevant explicit concept information on which the reasoner can operate. For example, the descendants of ‘enzyme’ (GO:0003824) are manually organized from a biochemical point of view derived from the Enzyme Classification (EC).⁷ Biologists from other disciplines may prefer to group enzyme functions by the type of chemical substances they function on rather than detailed chemical substructures involved in the reactions. Given there will inevitably be

effort required to manually create the information required to support this alternative classification, it is advisable to first investigate the reuse of existing structured information from other sources.

There are numerous bioinformatics resources available that contain structured information characterizing various aspects of enzymes. To support the reclassification described above, we need to capture the substrates and products of that reaction and any cofactors involved. To do this we used the enzyme database published as part of the Kyoto Encyclopedia of Genes and Genomes (KEGG).⁸ Each substrate, product and cofactor entry in the relevant KEGG enzyme record (cross referenced by EC identifier) was converted into an existential restriction on the relevant DAML+OIL class as shown in section 2.

Of the 2960 enzyme functions in GO, 2513 were annotated with an EC identifier and so could be linked to external databases. Of these 1596 had a corresponding entry in the KEGG enzyme database. The reasoner is unable to classify enzyme functions based on chemical class specified in these restrictions unless we provide a classification of said chemicals. Chemical thesauri do exist, the most widely known is that embedded within the Medical Subject Headings MESH *et seq.*⁹ We therefore represented the relevant subset of MeSH as a DAML+OIL ontology and linked the chemicals specified in enzyme description with this MESH chemical ontology. No direct cross-reference exists between the KEGG enzyme database and MeSH. Linking based on an exact term name match, yields links for only 4% (106/2443). Use of lexical tools and synonym information available with the Unified Medical Language System,¹⁰ led to the resolution of three sources of mismatch, resulting in an increase in matches to 35% (856/2443):

- | | |
|--------------------------|--|
| 1. Syntactic differences | e.g. Divalent cation --> Cations, Divalent |
| 2. Abbreviations | e.g. dUMP --> 2'-deoxyuridylic acid |
| 3. Synonyms | e.g. 20-Hydroxyecdysone --> Ecdysterone |

Of the remaining 1587 unmatched chemicals, most covered specializations of concepts within MESH, e.g. 'manganese²⁺ ion' as opposed to the term 'manganese' present in MESH. This points to the need for ontology integration tools that interleave related concepts rather than provide just an exact mapping between equivalent terms. Ontology integration tools do exist such as Chimera and PROMPT,^{11,12} and the next phase of the project will evaluate their utility for this task.

The reasoner can now group enzyme functions by the class of chemicals they involve, (as shown in fig. 3) providing those chemicals are linked to the MeSH ontology.

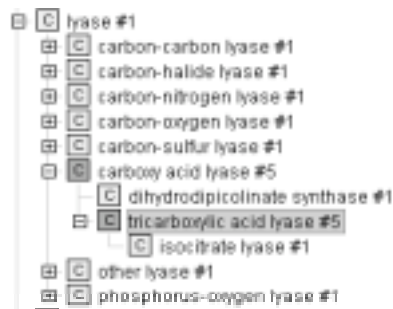


Figure 3. Automated grouping of 'isocitrate lyase' under novel abstractions 'tricarboxylic lyase' and 'carboxy acid lyase' using the FaCT reasoner.

Step 3. Manually adding semantic information to support validation of existing classification.

The previous step added partial semantic information in a shallow and broad manner that can be used to index specific leaf node terms along additional axes of classification. In most cases the partial definition mined from existing resources must then be completed and checked by hand. Only then can they be used to verify the existing classification. The resulting definitions can be simple, such as GO metabolism concepts, or complex such as GO enzyme function concepts.

The majority of 'metabolism' (GO:008152) descendants convey only two aspects of information: the subtype of metabolism ('catabolism' (GO:009056), 'biosynthesis' (GO:009058)), and the chemical being metabolized. This results in many metabolism concepts having multiple is-a parents. As mentioned in the introduction, manual maintenance of multi-axial hierarchies is known to be prone to error.² To detect and resolve possible omissions we explicitly represented the chemical involved and metabolism subtype as DAML+OIL restrictions, for 250 descendants of 'carbohydrate metabolism' (GO:0005975). Chemicals referenced by these restrictions were defined in a separate ontology derived from MeSH (see stage 2). The resulting class definitions were submitted to the FaCT reasoner and any additional inferred is-a relationships recorded. The reasoner inferred 22 new is-a relationships, e.g. 'fructosamine catabolism' (GO:0030392) is now inferred to be a kind of 'carbohydrate catabolism' (GO:0016052). The set of inferred 'is-a' links have been reviewed by the editorial team resulting in 17 additional is-a links in the published Gene Ontology.

A complete definition of a metabolism term is usually simple. However, complete descriptions of, for example, enzyme functions are far more complex. Reaction process terms such as 'oxidising' and 'reducing' need to be specified for each enzyme function term, and these in turn require properties detailing which chemical the reaction acts on and at what site. Reaction sites need specifying in terms of chemical substructure such as 'carboxyl group', and each chemical must

include a description of which and how many substructures it contains. It is at this stage that the full expressive power of DAML+OIL is brought to bear. For example, exact cardinality constraints in combination with logic operators are used to precisely define 'tricarboxylic acid' as a chemical, which has exactly three 'carboxy' or 'carboxylate anion' groups. The dedicated DAML+OIL ontology editor OilEd¹³ was used to support the authoring, but it was not designed to support the complexity of descriptions required for these GO concepts. For example, it took one day to author the nine enzyme function definitions involved in the tricarboxylic acid cycle.^a This issue was encountered during the construction of the large medical ontologies in the GALEN-IN-USE project,¹⁴ and alternative environments were needed to manage this complexity. A simplified intermediate representation was developed to capture or 'dissect' concept definitions in a single domain (surgical procedures). These were then automatically expanded into the more complex underlying description logic representation for reasoning. The next phase of the project will develop tools to support an intermediate representation for dissecting GO terms.

Even with a simplified representation, adding detailed semantic information by hand is time consuming. Term names often follow a stereotyped pattern. Simple scripts were used to help parse metabolism terms in the work described above, but more sophisticated term mining tools would greatly increase productivity.

Step 4. On-demand composition of new concepts within client annotation software

Realistically, only central maintainers of the ontology can undertake steps 0 to 3. However, the dynamic nature of description logic ontologies enables on demand concept composition to occur in client annotation applications. As discussed in the introduction, cross products are a key future challenge for the Gene Ontology. Many concepts such as metabolism and development are highly compositional in nature, encompassing a process acting on a chemical or anatomical structure. Manually enumerating these cross products is untenable. Automatic enumeration would lead to a huge and largely redundant ontology. Therefore these compositions should only be created if required at the point of use. Experience with medical terminologies, such as SNOMED, shows that manual compositions are not correctly classified and users often find multiple methods for expressing the same concept.¹⁵ Use in an automated environment also opens up the possibility of nonsense compositions. The fine-grained semantic information and constraints added in previous steps together with a DL reasoner can be used to provide a logical basis for such compositions. Such an activity requires an *ontology server* with which client applications can communicate in order to compose new logically consistent concepts and integrate them in the existing classification. A DAML+OIL ontology server has been constructed during the COHSE project¹⁶ and the recently

^a Difficulty authoring complex expressions is compounded by the absence of a human readable syntax for DAML+OIL.

funded GOAT project (Gene Ontology Annotation Tool, <http://goat.man.ac.uk>) will explore the utility of concept composition within client annotation tools.

4 Discussion

We have shown how DAML+OIL can be used to represent a range of ontologies from taxonomies such as subsets of MESH, to ontologies in which complete formal definitions are given to each term. We have also shown the utility of reasoning in helping to maintain existing hierarchies, and in grouping terms along novel axes of classification.

The aim of the project was to enable the migration to DAML+OIL rather than provide significant DAML+OIL content. However, the stereotypical pattern of many definitions in categories of concept such as metabolism, has enabled us to automatically generate ~1500 metabolism definitions with three weeks programming effort and a further 1600 partial enzyme definitions with a similar effort. We are investigating how successfully this approach can be extended to other categories.

The technical feasibility of a methodology does not necessarily translate into a successful practical solution. We have shown in the later stages of the methodology the complexity of DAML+OIL expressions that may be required to realize the full functionality of such an approach. In the next stage of the GONG project, we plan to develop ontology engineering tools to support these later stages on a large scale with emphasis on ontology integration, automated and assisted content authoring from existing sources, ontology versioning and targeted deployment. Once these tools start to become available we can evaluate the impact of such a methodology in terms of resources and changes to the current maintenance process for GO.

Other Ontology editing environments exist. Protégé 2000, a frame based ontology environment, implements many analogous content creation tasks in the form of tabs.¹⁷ However, there is currently no DAML+OIL reasoning support from within Protégé, although such functionality is in development. Members of the GO consortium are also actively pursuing DAML+OIL as an ontology representation language and are developing a new editing environment to support it called GOET (Gene Ontology Editing Tool) (<http://sourceforge.net/projects/gmod/>). We are not clear to what extent GOET will use the reasoning capabilities of DAML+OIL, but we hope to inform the design process for the tool and the working practices which evolve around its use.

5 Acknowledgments

This work is supported by the GONG project grant (DARPA DAML subcontract PY-1149 from Stanford University) and the ^{my}Grid eScience pilot project grant (EPSRC GR/R67743). We would like to thank Midori Harris and Jane Lomax of the GO Consortium Editorial Team for their advice and guidance.

References

1. The Gene Ontology Consortium, Gene Ontology: tool for the unification of biology. *Nature Genetics* **25** (2000) pp. 25-29.
2. Rogers J.E., Price C., Rector A.L., Solomon W.D., Smejko N. Validating Clinical Terminology Structures: Integration and Cross-Validation of Read Thesaurus and GALEN. *AMIA Fall Symposium, Orlando, USA* (1998)
3. Chang, J., Raychaudhuri, S., Altman, R., Including Biological Literature Improves Homology Search. *Pacific Symposium on Biocomputing* **6** (2001) pp. 374-383.
4. Horrocks I., DAML+OIL: a reason-able web ontology language. *Proc. of EDBT 2002* (March 2002).
5. Stevens R., Horrocks I., Goble C., Bechhofer S., Building a Reason-able Bioinformatics Ontology Using OIL. *IJCAI'01 Workshop on Ontologies and Information Sharing, Seattle, USA*, (August 2001) pp. 81-90.
6. Bechhofer S., Goble C., Thesaurus construction through knowledge representation. *Data & Knowledge Engineering* **37** (April 2001) pp. 25-45.
7. Enzyme Nomenclature. *Academic Press, San Diego, California, ISBN 0-12-227165-3* (1992).
8. Kanehisa M., Goto, S., KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **28** (2000) pp. 27-30.
9. Medical Subject Headings, MESH. <http://www.nlm.nih.gov/mesh/>
10. McCray A.T., Srinivasan S., Browne A.C., Lexical methods for managing variation in biomedical terminologies. *Proc Annu Symp Comput Appl Med Care* (1994) pp. 235-239.
11. McGuinness, D.L., Fikes R, Rice J., Wilder S., The Chimaera Ontology Environment. *AAAI-2000, Austin, TX* (2000).
12. Noy N.F., Musen. M.A., PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. *AAAI-2000, Austin, TX* (2000).
13. Bechhofer S., Horrocks I., Goble C., Stevens R., OilEd: a Reason-able Ontology Editor for the Semantic Web. *Proceedings of KI2001, Joint German/Austrian conference on Artificial Intelligence, Vienna. Springer-Verlag LNAI 2174* (2001) pp. 396-408.
14. Rogers J.E., Solomon W.D., Rector A.L., Pole P.M., Zanstra P., van der Haring E., Rubrics to Dissections to GRAIL to Classifications. *Medical Informatics Europe, IOS Press* **43** (1997) pp: 241-245.
15. Rector A.L., Clinical terminology: Why is it so hard? *Methods of Information in Medicine, Schattauer, ISSN 0026-1270* **38** (1999) pp. 239-252.
16. Bechhofer S., Carr L., Goble C., Hall W., Conceptual Open Hypermedia = The Semantic Web? *Second International Workshop on the Semantic Web* (2001).
17. Musen M.A., et al., Component-Based Support for Building Knowledge-Acquisition Systems. *IIP 2000, Beijing* (2000).