



## Knowledge acquisition, consistency checking and concurrency control for Gene Ontology (GO)

Iwei Yeh<sup>1</sup>, Peter D. Karp<sup>2</sup>, Natalya F. Noy<sup>3</sup> and Russ B. Altman<sup>1,\*</sup>

<sup>1</sup>Department of Genetics, Stanford University, Stanford, CA 94305-5120, USA,

<sup>2</sup>Bioinformatics Research Group, SRI International, 333 Ravenswood Avenue, Menlo Park, CA 94025, USA and <sup>3</sup>Stanford Medical Informatics, Stanford University,

Stanford, CA 94305-5479, USA

Received on December 27, 2001; revised on May 13, 2002; July 15, 2002; accepted on July 24, 2002

### ABSTRACT

**Motivation:** A critical element of the computational infrastructure required for functional genomics is a shared language for communicating biological data and knowledge. The Gene Ontology (GO; <http://www.geneontology.org>) provides a taxonomy of concepts and their attributes for annotating gene products. As GO increases in size, its ongoing construction and maintenance becomes more challenging. In this paper, we assess the applicability of a Knowledge Base Management System (KBMS), Protégé-2000, to the maintenance and development of GO.

**Results:** We transferred GO to Protégé-2000 in order to evaluate its suitability for GO. The graphical user interface supported browsing and editing of GO. Tools for consistency checking identified minor inconsistencies in GO and opportunities to reduce redundancy in its representation. The Protégé Axiom Language proved useful for checking ontological consistency. The PROMPT tool allowed us to track changes to GO. Using Protégé-2000, we tested our ability to make changes and extensions to GO to refine the semantics of attributes and classify more concepts.

**Availability:** Gene Ontology in Protégé-2000 and the associated code are located at <http://smi.stanford.edu/projects/helix/gokbms/>. Protégé-2000 is available from <http://protege.stanford.edu>.

**Contact:** russ.altman@stanford.edu

### INTRODUCTION

The Gene Ontology (GO) describes the roles of gene products and allows genomes to be annotated with a consistent terminology (The Gene Ontology Consortium, 2000). GO is organized into three sub-ontologies: *molecular function*, *biological process* and *cellular location*, each with its own hierarchical organization of concepts. Each concept has a numeric unique identifier or GOid, and is associated with

a 'term' (e.g. GO: 0015643, anti-toxin). GO has gained wide acceptance and several genome databases use GO for curation (<http://www.geneontology.org/>).

In addition to assigning GO annotations, the GO community provides tools for viewing, editing, and querying GO and its associated annotations. AmiGO! (<http://www.godatabase.org/cgi-bin/go.cgi>) is a browser that provides string based querying of GO terms and gene products. DAG-Edit (<http://sourceforge.net/projects/geneontology>) provides browsing, querying, and editing for ontologies in a DAG data structure.

Ontologies differ from controlled terminologies in that they represent the relevant concepts and relationships in a domain, whereas terminologies simply restrict the words used to describe the domain (Gruber, 1993). Well-structured ontologies allow querying of complex relationships within large data sets and facilitate automatic inference of new knowledge (Russell and Norvig, 1995; Bada and Altman, 2000).

An important method for representing ontologies is one based on frames (Fikes and Kehler, 1985; Minsky, 1987; Karp, 1992). Frame-based representations are the basis for several systems (Felbaum, 1998; Cimino, 2000; Karp *et al.*, 2002). In frame-based representations, concepts represent sets of objects with common properties. A frame representing a concept contains *slots* describing its attributes. *Facets* describe properties of slots, such as their allowed values and cardinality. *Relationships* are slots that have frames as their allowed value. A special relationship is the *is-a* relationship that organizes ontologies into a taxonomic hierarchy and allows for inheritance. Inheritance propagates slot values from concepts to their children, decreasing redundancy of stored information—thus reducing the chance of inconsistent or inefficient updates.

While we restrict our discussion to frame-based representations, description logics can also be used to capture biological ontologies (Stevens *et al.*, 2000). Description logics provide an expressive representation by defining concepts and the relationships between them

\*To whom correspondence should be addressed.

axiomatically, and can be used for automatic classification of concepts.

Semantic specification is an integral part of ontology development. Formal restrictions on the possible values of slots add semantic expressiveness to an ontology and enforce consistency. Non-taxonomic relationships between concepts (such as the *part-of* relationship) can be defined and used to make inferences. GO is compatible with frame-based representations, since each GO term can be mapped to a frame and the additional information can be mapped to slots. Once GO is in a frame-based representation, techniques developed for reducing redundancy, consistency checking and inference may be relevant and useful.

For this work, we define a knowledge base (**KB**) as an ontology with the addition of instances. KBMSs facilitate knowledge modeling, knowledge acquisition, consistency checking, and concurrency control of ontologies and KBs (Paley et al., 1997; Altman et al., 1999; Karp et al., 1999). *Knowledge modeling* consists of creating concepts and organizing them into a taxonomy. The Unified Modeling Language (UML) was developed for object-oriented modeling which is in many ways similar to frame-based systems (Booch et al., 1998). *Knowledge acquisition* involves extracting knowledge about the domain from various sources. *Consistency checking* ensures that classes and instances in the KB have attributes which conform to the knowledge model. *Concurrency control* maintains appropriate behavior of the KB during and after simultaneous operations.

Clarity of the knowledge model and the consistency of the KB are especially important when applying automated computational methods to the KB. Hand-assigned GO annotations provide a foundation for automated annotation transfer, such as transfer based on protein domain similarity (Schug et al., 2002) and abstract similarity (Raychaudhuri et al., 2002).

We translated GO into Protégé-2000, an open-source KBMS environment (Noy et al., 2000). We accessed the utility of ontology queries, constraint specification/verification, and visualization in the context of GO.

## METHODS

GO is available in text format (<http://www.geneontology.org/>).

### Protégé-2000

Protégé-2000 was developed in the Musen Laboratory at Stanford Medical Informatics. Protégé-2000 is written in Java and contains a Java API for independent development of plug-ins.

### Translation of GO into Protégé-2000

Despite the compatibility of GO with frame-based systems, modeling decisions were required to represent

GO in Protégé-2000. Each GO concept is a class, reproducing the hierarchical nature of GO. The root concept, `Gene_Ontology_Entity`, has children `biological_process`, `molecular_function`, and `cellular_component`. Many concepts in GO do not have an *is-a* relationship, but only a *part-of* relationship. This decision has some cost because in most ontologies, different classes have different slots defined for them, determined by the position of the class in the *is-a* hierarchy. Currently, GO defines the same set of slots for all of its concepts. Over time, however, the GO developers may add additional slots at appropriate places in the GO *is-a* hierarchy, and a fully elaborated *is-a* hierarchy may be valuable.

Protégé-2000, as is the case with many KBMSs, requires every concept to have an *is-a* relationship. `Gene_Ontology_Entity` has additional children:

```
Molecular_Function_Unclassified,
Biological_Process_Unclassified, and
Cellular_Component_Unclassified
```

(Figure 1). All terms without an *is-a* relationship in GO received an *is-a* relationship to one of these classes.

The GO attributes are: name, synonyms, database references, and *part-of* (see <http://www.geneontology.org/GO.doc.html> for documentation).<sup>†</sup> We modeled most slots as strings, but modeled *part-of* as a relationship between two classes. We included slots `Definition`, `Definition_Reference`, `SwissProt_Keyword`, and `InterPRO_ID`. The biological domain knowledge resides in the `Definition` and the *is-a* and *part-of* relationships, the other attributes represent metadata, or data about data.

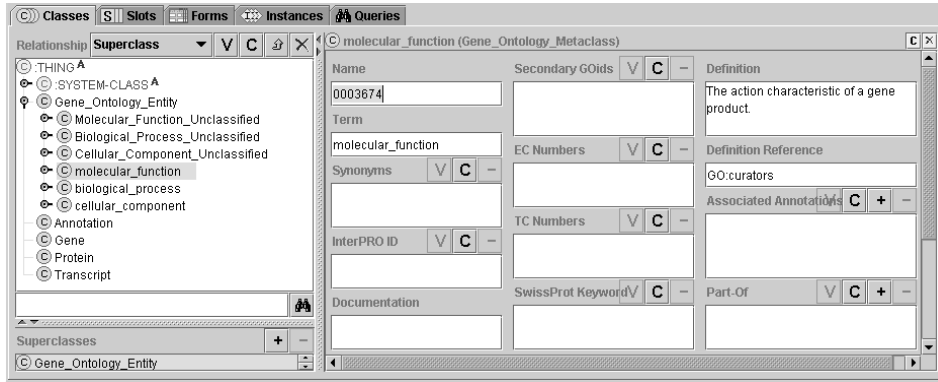
The classes `Annotation`, `Gene`, `Protein`, and `Transcript` are used to represent annotations. Each `Annotation` captures the assignment of a GOid to a particular gene product and the associated metadata. Figure 2 illustrates how the `Annotation` and `Gene` classes are structured.

We wrote a Java program using the Protégé-2000 API, to create a Protégé project (an ontology or KB) for GO. We translated GO from February 20, 2002. We added `Genes` and `Annotations` from the SGD annotations of *Saccharomyces cerevisiae* and the FlyBase annotations for *Drosophila melanogaster* from February 19, 2002.

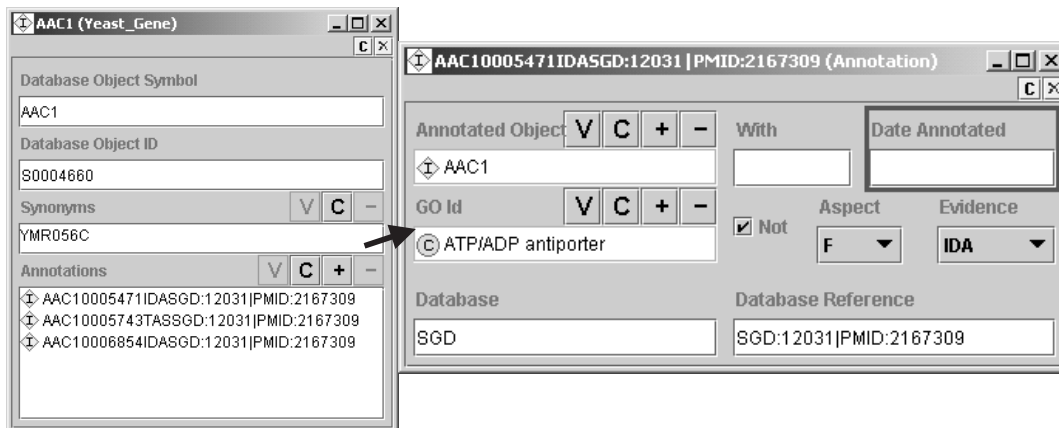
### Application of KBMS tools to GO

The Protégé Axiom Language (PAL) allows querying of concepts based on their features (<http://protege>).

<sup>†</sup>The Protégé-2000 metaclass architecture (Noy and Musen, 2000) allows definition of class-level attributes with metaclasses—templates according to which new classes are defined. The class-level attributes define properties of a class itself. Unlike the slots of normal classes, the values of these attributes are not inherited by its subclasses or its instances. The template for all the GO concepts (`Gene_Ontology_Metaclass`) describes these attributes and the constraints on their values (Figure 1).



**Fig. 1.** Class hierarchy and Gene\_Ontology\_Metaclass in Protégé-2000. A screen shot from Protégé-2000 shows the class hierarchy used for GO and an example instance of the Gene\_Ontology\_Metaclass. An instance of the Gene\_Ontology\_Metaclass for the concept molecular\_function is shown with its slots on the right. The form on the right is an automatically generated knowledge acquisition form, the fields can be edited directly, and when the attribute is a relation, clicking on the '+' allows developers to choose from a list of allowed values.



**Fig. 2.** On the left is an instance of Gene. Each Gene instance can be associated with several instances of Annotation. One example of a corresponding Annotation for this particular Gene is shown on the right. The Gene class has attributes Database\_Object\_ID, Database\_Object\_Symbol, Synonyms, and Annotations. The Annotation class contains single-valued string slots for: Database, Database\_Reference, Date\_Annotated and With. Not is a Boolean value while Aspect and Evidence are of type Symbol. GO\_id is of type Gene\_Ontology\_Metaclass and Annotated\_Objects is of type instance of Gene, Protein, or Transcript. The form highlights areas where constraints are violated. Date\_Annotated is highlighted because this is a required attribute with a missing value.

stanford.edu/plugins.html). To test the utility of PAL, we wrote queries to identify unnecessary *is-a* relationships (Figure 3A).

The OntoViz tab (<http://protege.stanford.edu/plugins.html>) automatically generates a graphical representation of concepts in an ontology. We used the OntoViz tab to diagram the knowledge model for GO.

PROMPT is a suite of tools for managing multiple ontologies (Noy and Musen, 2000). PROMPT automatically generates a list of differences and similarities between two input ontologies.

To examine the performance of PROMPT with GO, we translated the release of GO from June 8, 2001, and examined its differences with the February 20, 2002 version.

### Extending GO

To test the ability of Protégé-2000 to facilitate extension of GO, we experimentally modified GO with specific goals in mind:

*Introduction of new relationships* We created new relationships to capture more biological information. The

**Name**: Strict\_Inheritance

**Description**: This PAL constraint disallows redundant subclassing. If A is a direct parent of C, and B is a direct parent of C, then A cannot also be a direct parent of C, since the relationship between A and C is inferable from the relationship between A and B and between B and C.

**Statement**:  

```
(forall ?C
  (forall ?A
    (=> (direct-subclass-of ?C ?A)
      (forall ?B
        (=> (direct-subclass-of ?B ?A)
          (not (subclass-of ?C ?B)))))))
```

?C	?A
C 0001524 <sup>M</sup>	C 0019825
C 0001639 <sup>M</sup>	C 0008066
C 0001640 <sup>M</sup>	C 0008066
C 0003850 <sup>M</sup>	C 0016302
C 0004001 <sup>M</sup>	C 0016301
C 0004001 <sup>M</sup>	C 0016773
C 0004005 <sup>M</sup>	C 0005386
C 0004005 <sup>M</sup>	C 0016820
C 0004005 <sup>M</sup>	C 0016897 <sup>M</sup>
C 0004012 <sup>M</sup>	C 0016887 <sup>M</sup>
C 0004017 <sup>M</sup>	C 0016301
C 0004054 <sup>M</sup>	C 0016301
C 0004072 <sup>M</sup>	C 0016301
C 0004127 <sup>M</sup>	C 0016301
C 0004127 <sup>M</sup>	C 0016773
C 0004136 <sup>M</sup>	C 0016301
C 0004136 <sup>M</sup>	C 0016773
C 0004137 <sup>M</sup>	C 0016301
C 0004137 <sup>M</sup>	C 0016773
C 0004138 <sup>M</sup>	C 0016301

**Query Responses**

**Violations**

- oxygen binding
  - cytochrome P450
  - oxygen sensor
  - oxygen transporter<sup>M</sup>
    - globin<sup>M</sup>
    - hemerythrin<sup>M</sup>
    - hemocyanin<sup>M</sup>
  - globin<sup>M</sup>
  - hemerythrin<sup>M</sup>
  - hemocyanin<sup>M</sup>

**Fig. 3.** (A) PAL constraint. This constraint specifies that if A is a direct parent of B and B is a direct parent of C, then A should not be a direct parent of C, since this relationship is inferable from the first two relationships. (B) Violations of strict inheritance. This is a partial list of results from the PAL constraint. For example we find that the relationship GO:0001524 (globin) *is-a* GO:0019825 (oxygen binding) is explicitly specified when it is inferable from the relationships GO:0001524 (globin) *is-a* GO:0005344 (oxygen transporter) and GO:0005344 (oxygen transporter) *is-a* GO:0019825 (oxygen binding).

study of part–whole relationships has identified different types of part–whole relationships with different properties (Winston *et al.*, 1987; Artale *et al.*, 1996). We examined the *part-of* relation in GO and dissected out different types of part–whole relations.

**Classification of concepts** To classify concepts, we created additional concepts and created *is-a* relationships from unclassified concepts to these additional concepts. To clearly show which *is-a* relationships we created, all of our classifications exist under parent concepts contained in ‘Additions.To’.<sup>‡</sup>

**Extraction of attributes implicitly contained in terms** We created an attribute for Gene\_Ontology\_MetaClass called Sensu. For each GO term containing ‘sensu’, we added the corresponding organism or taxon to the domain of the slot and explicitly encoded the information. For

<sup>‡</sup> If the parent concept is already present in GO, we create a dummy concept in ‘Additions.To’ and note the correspondence in the documentation field.

other terms we deduced the taxon. For instance, all terms containing ‘imaginal discs’ without ‘sensu’ were assigned a value of Endopterygota, as imaginal discs are only seen in this class of insects. Simple scripts were used to add Sensu values.

We used the OntoViz tab during our modification to view our extensions to the GO knowledge model.

## RESULTS

### Properties of GO

The version of GO from February 20, 2002 contained 10,603 concepts. 1535 concepts have a *part-of* relation, and 1108 of these do not have an *is-a* relationship. Of the concepts without an *is-a* relationship, 17 were *part-of* a molecular\_function, 636 were *part-of* a biological\_process, and 455 were *part-of* a cellular\_component. Of the concepts with an *is-a* relationship, 4935 were molecular\_function, 3644 were biological\_process and 159 were

**Table 1.** Statistics compiled for Gene Ontology from February 20, 2002

	Component	Function	Process
<b>Classified</b>	<b>159</b>	<b>4935</b>	<b>3644</b>
with definition	30	606	993
with <i>part-of</i>	4	18	155
with multiple inheritance	3	967	1264
<b>Unclassified</b>	<b>455</b>	<b>17</b>	<b>636</b>
<b>Total</b>	<b>614</b>	<b>4952</b>	<b>4280</b>

cellular\_component (Table 1).

In the text version, we found that out of 2278 concepts exhibiting multiple inheritance, all except two concepts specified parents redundantly. These two concepts therefore have ambiguous positions in the hierarchy, causing potential semantic confusion.

The constraints generated by Protégé-2000 pinpointed inconsistencies in GO. SwissProt keywords and InterPRO ids were modeled as attributes of *Gene\_Ontology\_Entity*, and we were alerted when SwissProt keywords or InterPRO ids mapped to non-existent GOids.

### Enforcing strict inheritance using PAL

Our strict inheritance constraint identified violations (Figure 3), for example:

- (1) **globin** (GO:0001524) *is-a* **oxygen transporter** (GO:0005344);
- (2) **oxygen transporter** (GO:0005344) *is-a* **oxygen binding** (GO:0019825);
- (3) **globin** (GO:0001524) *is-a* **oxygen binding** (GO:0019825).

Relationship 3 can be inferred from relationships 1 and 2. This is inconsistent with standard ontological design because, when an *is-a* relationship is inferable, its specification introduces a redundancy that increases the likelihood of incorrect or inconsistent updating.

### Visualizing the knowledge model

We created diagrams using the OntoViz tab (Figure 4A). When extending GO, OntoViz clearly summarized the changes, contributing to the knowledge modeling process (Figure 4B).

### Tracking changes to GO

We used PROMPT to compare June 2001 and February 2002 releases of GO. Between June and February 30 concepts became obsolete and 2888 concepts were added. In addition, PROMPT pointed out concepts that changed but that were very similar in their structure (Figure 5). For example, ‘peripheral plasma membrane protein’

(GO:000157) in the June version has exactly the same subclasses and superclass as ‘extrinsic plasma membrane protein’ (GO:0019897) in the February version. Therefore, PROMPT identifies them as related to each other.

### Representing implicit information

We added Sensu values for 185 concepts with ‘sensu’ in their terms and 262 concepts without ‘sensu’ in their terms. Specifying the Sensu slot as type ‘symbol’ restricts the possible values to a set of organisms and taxa, facilitating querying over this attribute.

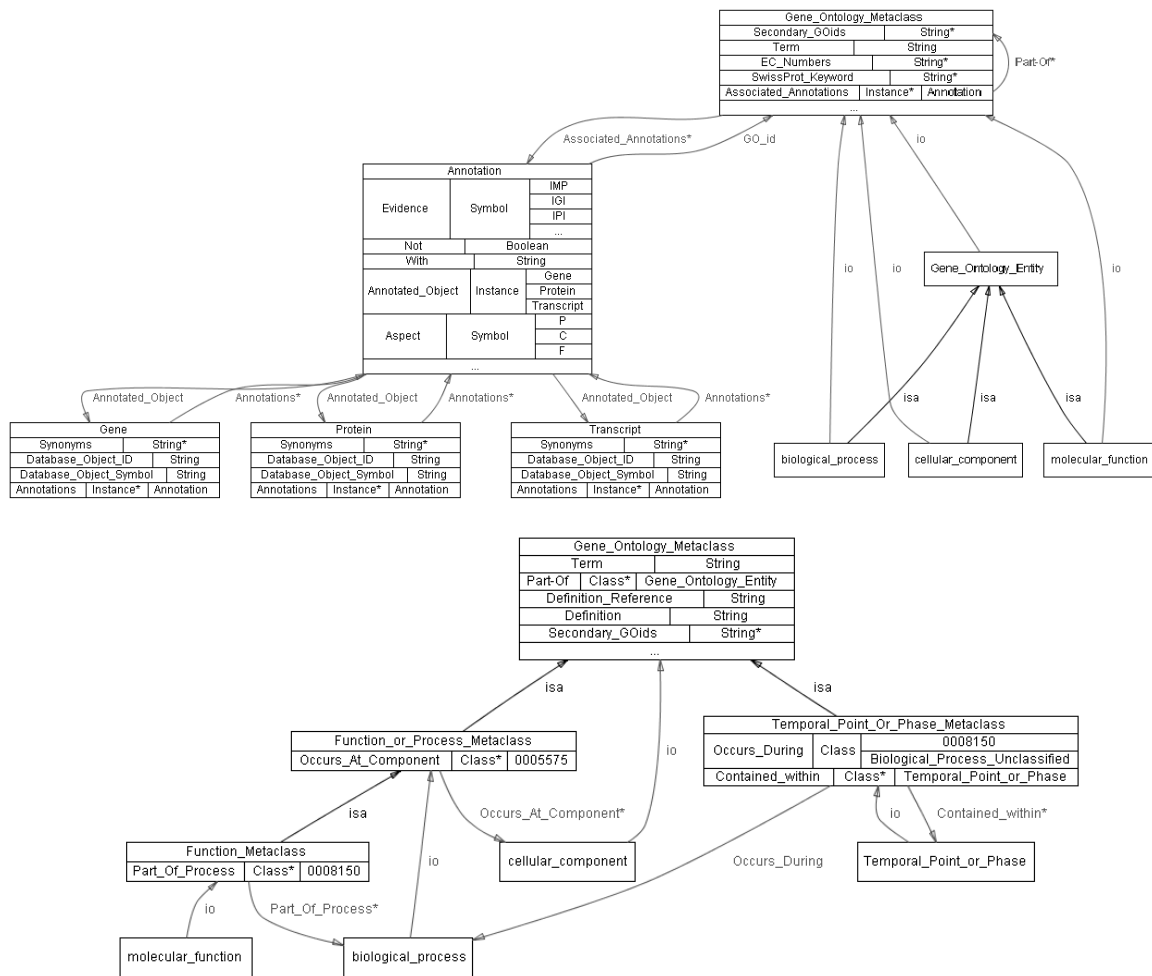
We found one concept whose term included ‘sensu *Drosophila*’. Ad hoc methods to pull out all terms containing ‘sensu *Drosophila*’ might miss this term.

### Extensions to GO

To capture the biological relationships between molecular function and biological process, we created the relation *Part\_Of\_Process* as an attribute of *molecular\_function* to specify the biological processes in which a molecular function participates. We also created the relation *Occurs\_At\_Component* as an attribute of both *molecular\_function* and *biological\_process* to specify the cellular component where these functions and processes occur. In its current state, the three hierarchies of GO do not contain relationships that relate concepts in one hierarchy to those in another. However, more biological knowledge can be encoded by introducing relationships between the different hierarchies.

We manually classified 79 GO concepts by creating *is-a* relations to an existing or newly created concepts. We created the concept *Cellular\_Space* as a child of *cellular\_component* as a parent class for unclassified concepts such as cytoplasm, mitochondrial matrix, and Golgi lumen. Instances of *Cellular\_Space* are distinct volumes that are contained within membranes and can contain various other cellular components. Unique attributes can be defined for this concept to describe the functions, processes and components contained within a particular space.

To classify the GO concepts that were polypeptides and nucleotides, we created several classes that are analogous to those used in the EcoCyc ontology (Karp, 2000). We created classes *Macromolecule* and *Complex* as subclasses of *cellular\_component*. *Complex* has subclasses *Protein\_Complex* and *Protein\_Nucleic\_Acid\_Complex*. Under *Macromolecule* we created further subcategories *Polypeptide* and *Nucleic\_Acid*. We classified 31 concepts under *Polypeptide* and 10 concepts under *Nucleic\_Acid*. We created a distinct *part-of* relation where a *Macromolecule* has the attribute *Part\_of\_Complex* and the value must be of type *Complex*. This



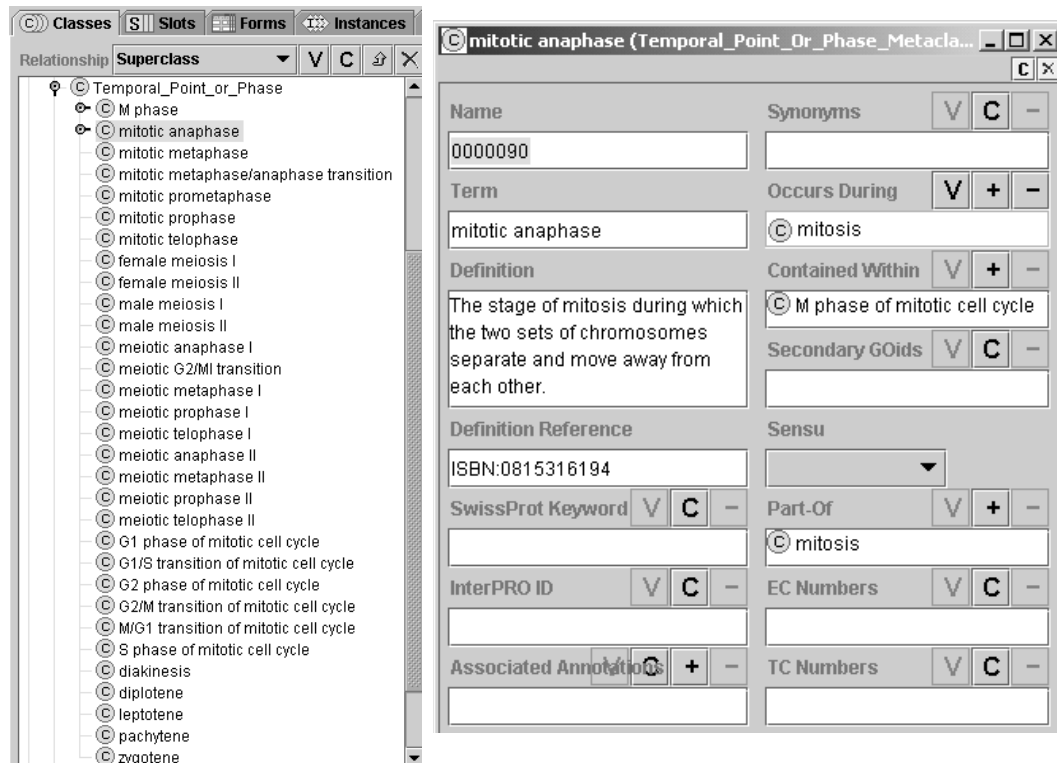
**Fig. 4.** (A) This output from the OntoViz tab in Protégé-2000 represents the frame-based knowledge model we developed to capture the GO knowledge base. The squares represent classes and display the slots and their value types. The arcs represent relations between classes or between classes and instances. The io link shows when one concept is an *instance-of* another concept. (B) We extended GO and we can display how the new concepts created interact with the existing concepts using the OntoViz tab. The full knowledge model can be viewed at the supplemental web page.

relation is a Component/Integral-Object relationship, where each complex is a structure with separable components that have specific functionality. Upon examination of the *part-of* relationships contained in GO, we found instances of this relationship, for example, ‘heterotrimeric G-protein GTPase, alpha-subunit’ (GO:0000262) is *Part-of-Complex* ‘heterotrimeric G-protein GTPase’ (GO:0003927).

We created the concept *Temporal\_Point\_or\_Phase* outside of the original GO hierarchy. For this concept, we identified two distinct *part-of* relations. For instance, ‘M phase’ (GO:0000279) is *part-of* ‘cell cycle’ (GO:0007049), which is a relation between a *Temporal\_Point\_or\_Phase* and a

*biological\_process*. This we capture by the relation *Occurs\_During* of *Temporal\_Point\_or\_Phase*, whose value is restricted to *biological\_process*. We created the slot *Contained\_Within* to denote a *Temporal\_Point\_or\_Phase* completely contained within another *Temporal\_Point\_or\_Phase* and created the relation ‘mitotic anaphase’ (GO:0000090) is *Contained\_Within* ‘M phase of mitotic cell cycle’ (GO:0000087; Figure 5). This slot is a Portion/Mass relationship where the whole is homogeneous (an interval of time) and the portions are separable (intervals within an interval).

The knowledge base resulting from our extension of GO is available on the supplemental materials web page.



**Fig. 5.** The subclasses of *Temporal\_Point\_or\_Phase* are displayed on the left. A specific class, 'mitotic anaphase' (GO:0000090) is displayed. The relation *Occurs\_During* captures information from the original *part-of* relation. Here we denote that 'mitotic anaphase' (GO:0000090) occurs during the biological process 'mitosis' (GO:0007067). The *Contained\_Within* relation relates 'mitotic anaphase' (GO:0000090) to 'M phase of mitotic cell cycle' (GO:0000087) another class of *Temporal\_Point\_or\_Phase*, stating that the interval 'mitotic anaphase' is completely within the interval 'M phase of mitotic cell cycle'.

## DISCUSSION

The Gene Ontology Consortium has created a controlled taxonomy of terms for annotating gene products. In addition, genes across several organisms have been carefully annotated using GO. By defining a standard for annotating gene products, GO integrates genomic data and protein annotations from many sources. With high quality annotations we can explore a variety of hypotheses about evolutionary relationships and other aspects of comparative genomics.

Much the success of GO stems from its open and collaborative development process, which has led to widespread community acceptance. Biologists from specialized domains plan to contribute concepts to GO (Berriman *et al.*, 2001). Improvements and extensions to GO are released in monthly updates. Concurrency control has been addressed for databases by sophisticated locking mechanisms during editing. Strategies developed for KB concurrency control merge separate edits and resolve conflicts in a semi-automated fashion (Karp *et al.*, 1999; Noy and Musen, 2000). Merging tools help users automatically

update attribute values as classes or instances change, while PROMPT facilitates the comparison of different versions of the same KB. As the size of a collaboration increases, the ability to coordinate and integrate individual work becomes extremely important. Protégé-2000 has been used to develop and manage ontologies with over 80,000 concepts (Noy *et al.*, 2002). Our results suggest that a KBMS can support organizational scalability.

KBMSs are developed to help users maintain and develop ontologies and knowledge bases. Once GO was loaded into Protégé-2000, we were able to add additional concepts, slots, and slot values to the ontology with minimal effort. We also added classes to classify concepts without *is-a* relationships. This can be difficult for annotations that are ascribed to genes and gene products that do not neatly fit into the current hierarchy. For instance in our extended ontology we leave some concepts 'unclassified', such as 'maintenance of cell polarity' (GO:0030011). However, the classes that are classified allow us to develop more specialized relations to capture biological knowledge.

Currently the three hierarchies of GO are completely separated, and we created relationships between the three hierarchies. Addition of these relations captures additional biological knowledge (and improves the expressive power of GO). For example, we can say that a particular molecular function is always a part of a particular biological process. These relations also provide a more compact representation of information. When annotating a gene product with a molecular function, associated processes are automatically associated with the gene product through the link between function and process.

The extensions added to GO could be useful in hypothesis generation over the KB. For example, when searching for possible interactions between molecular functions, one might want to limit the possibilities to functions in the same or adjacent cellular components.

Over the last 20 years the artificial intelligence community has developed standard operating procedures for creating ontologies that have been implemented in many different KBMSs. Our results suggest that the tools produced during this time are immediately and directly relevant to the ongoing development of the GO.

## ACKNOWLEDGEMENTS

The work was funded by the Burroughs Wellcome Fund, NIH grant GM07365, GM61374, and a grant from Glaxo-SmithKline. Dr Noy is supported by Contract 21XS067A from the National Cancer Institute.

## REFERENCES

- Altman, R.B., Bada, M.A. et al. (1999) RiboWeb: An ontology-based system for collaborative molecular biology. *IEEE Intelligent Systems and Their Applications*, **14**, 68–76.
- Artale, A., Franconi, E. et al. (1996) Part-whole relations in object centered systems: an overview. *Data and Knowledge Engineering*, **20**, 347–383.
- Bada, M.A. and Altman, R.B. (2000) Computational modeling of structural experimental data. *Meth. Enzymol.*, **317**, 470–491.
- Berriman, M., Aslett, M. et al. (2001) Parasites are GO. *Trends Parasitol.*, **17**, 463–464.
- Booch, G., Rumbaugh, J. et al. (1998) *The Unified Modeling Language User Guide*. Addison-Wesley, Menlo Park, CA.
- Cimino, J.J. (2000) From data to knowledge through concept-oriented terminologies: experience with the Medical Entities Dictionary. *Journal of the American Medical Informatics Association*, **7**, 288–297.
- Felbaum, C. (Ed.) (1998) *WordNet An Electronic Lexical Database*. MIT Press, Cambridge, MA.
- Fikes, R. and Kehler, T. (1985) The role of frame-based representations in reasoning. *Commun. ACM*, **28**, 904–920.
- Gruber, T. (1993) *Toward principles for the design of ontologies used for knowledge sharing*. Knowledge Systems Laboratory, Stanford University.
- Karp, P.D. (1992) *The design space of frame knowledge representation systems*. SRI International Artificial Intelligence Center.
- Karp, P.D. (2000) An ontology for biological function based on molecular interactions. *Bioinformatics*, **16**, 269–285.
- Karp, P.D., Chaudhri, V.K. et al. (1999) A collaborative environment for authoring large knowledge bases. *Journal of Intelligent Information Systems*, **3**, 155–194.
- Karp, P.D., Riley, M. et al. (2002) The EcoCyc database. *Nucleic Acid. Res.*, **30**, 56.
- Minsky, M. (1987) A framework for representing knowledge. *Readings in Knowledge Representation*. Levesque, H.J. (ed.), Morgan Kaufmann, Los Altos, CA, pp. 246–262.
- Noy, N.F. and Musen, M.A. (2000) PROMPT: algorithm and tool for automated ontology merging and alignment. *Seventeenth National Conference on Artificial Intelligence, Austin, TX*.
- Noy, N.F., Musen, M.A. et al. (2002) Pushing the envelope: challenges in a frame-based representation of human anatomy. Stanford Medical Informatics, Stanford, CA.
- Noy, N.F., Sintek, M. et al. (2001) Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems*, **16**, 60–71.
- Paley, S.M., Lowrance, J.D. et al. (1997) A generic knowledge-base browser and editor. *AAAI-97/IAAI-97*. American Association for Artificial Intelligence, Providence, RI.
- Raychaudhuri, S., Chang, J.T. et al. (2002) Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Res.*, **203**, 214.
- Russell, S.J. and Norvig, P. (1995) *Artificial Intelligence A Modern Approach*. Prentice Hall, Upper Saddle River, NY.
- Schug, J., Diskin, S. et al. (2002) Predicting gene ontology functions from ProDom and CDD protein domains. *Genome Res.*, **12**, 648–655.
- Stevens, R., Goble, C. et al. (2000) Ontology-based knowledge representation for bioinformatics. *Briefings In Bioinformatics*, **1**, 398–414.
- The Gene Ontology Consortium (2000) Gene ontology: tool for the unification of biology. *Nature Genet.*, **25**, 25–29.
- Winston, M., Chaffin, R. et al. (1987) A taxonomy of part-whole relations. *Cognitive Science*, **11**, 417–444.