Collaborative Ontology Development on the (Semantic) Web

Natalya F. Noy and **Tania Tudorache**
Stanford Center for Biomedical Informatics Research
Stanford University
Stanford, CA 94305
{noy,tudorache}@stanford.edu

Abstract

As knowledge engineering moves to the (Semantic) Web, ontologies become dynamic products of collaborative development rather than artifacts produced in a closed environment of a single research group. We examine today's large collaborative ontology-development projects–in particular in the domain of biomedicine–and outline some requirements for the tools to support this enterprise. We then present our initial prototype of Collaborative Protégé–an extension to the Protégé ontology-editing environment that enables distributed users to develop ontologies collaboratively and that provides an integrated platform for discussions.

Collaborative Ontology Development in Biomedicine

The biomedical community has embraced ontologies probably more than any other discipline. From the implementation of hospital information systems to the organization of experimental data for bioinformatics research, developers now identify the key issue to be the manner in which salient concepts are labeled and defined, and ultimately used computationally. With this embracement, there comes the next challenge, however: Ontologies and terminologies become so large, diverse, and specialized that it is often impossible for any single centralized group to develop them effectively. Indeed, the following projects represent just some of the most visible ontology-engineering initiatives that are incorporating community participation as a key element in their development work.

The Gene Ontology (GO) is probably one of the more prominent examples of an ontology that is a product of a collaborative process (Hartel *et al.* 2005). GO provides terminology for consistent description of gene products in different model-organism databases in terms of their associated biological processes, cellular components, and molecular functions in a species-independent manner. Members of the GO community constantly suggest new terms for this ontology. Three full-time curators examine the suggestions and incorporate them into GO on a continual basis.

The International Classification of Diseases (ICD)[1] is a public global standard to organize and classify information about diseases and related health problems. The World Health Organization (WHO) plans three major shifts for the upcoming 11^{th} revision of ICD (ICD-11): First, the ICD will represent clinical knowledge explicitly in machine-processable form. WHO plans to use an ontological approach, formalizing the definitions of each clinical entity and organizing the terms in a semantically meaningful way. Second, WHO will open the process of ICD revision to a wide community of experts. Topic Advisory Groups (TAGs) will serve as planning and coordinating bodies for specific areas of medicine, such as Oncology, Mental Health, and Communicable Diseases. Each TAG will support several international working groups and an additional corps of field testers who will use on-line tools to evaluate the evolving ontology and to generate proposals for revisions and enhancements. Third, ICD-11 will include direct linkages to terms in other standardized terminologies, such as SNOMED-CT.

The National Cancer Institutes Thesaurus (NCI Thesaurus) is a biomedical reference ontology that covers areas of basic cancer biology, translational science, and clinical oncology developed at the NCI Center for Bioinformatics (NCICB) (Fragoso *et al.* 2004; Sioutos *et al.* 2007). Currently, the NCI Thesaurus is used to index documents, to support the NCI Cancer portal,[2] as the terminology source for a number of applications such as a NCI Drug Dictionary,[3] and for annotation of metadata in the Cancer Bioinformatics Grid (caBIG).[4] Recently, the NCICB has launched a new terminology product, Biomedical Grid Terminology (BiomedGT), to support the needs of its NCICB partners. This new terminology restructures the NCI Thesaurus to facilitate terminology federation and open content development. The goal of BiomedGT is to empower the wider biomedical research community to participate directly and collaboratively in extending and refining the terminology on which they depend.

The Ontology for Biomedical Investigations (OBI),[5] a product of the OBI Consortium, is a federated ontology being developed collaboratively. OBI describes biological and

[1] `http://www.who.int/classifications/icd/en/`

[2] `http://cancer.gov`

[3] `http://www.cancer.gov/drugdictionary/`

[4] `http://cabig.nci.nih.gov`

[5] `http://obi.sourceforge.net/consortium/index.php`

medical experiments and clinical trials, including a set of universal terms that are applicable across different biological and technological domains, and a set of domain-specific terms relevant only to particular disciplines. The goal of the OBI Consortium is to support the consistent annotation of data from biomedical experiments, regardless of the particular field of study. The OBI Consortium has more than 40 active curators, each responsible for a particular scientific community (e.g., cellular assays, clinical investigations, neuroinformatics, immunology).

BIRNLex[6] is a controlled terminology for annotation of data resourcessuch as structural and functional image datafor the Biomedical Informatics Research Network (BIRN). BIRNLex provides terms, utilized by BIRN investigators in the context of their research, covering neuroanatomy, species designations, behavioral and cognitive processes, subject information, experimental practice and design, and associated elements of primary data provenance required for large-scale data integration across disparate experimental studies. Many of the terms are drawn from other terminologies and ontologies, such as the Unified Medical Language System (UMLS), the Foundational Model of Anatomy (FMA), NeuroNames, and GO.

We have described just a few of the many ongoing projects that employ different forms of collaborative and distributed ontology and terminology development. Other ontology-development projects that currently depend heavily on community-based development include **RadLex**,[7] an effort to develop a standard terminology for radiology, sponsored by the Radiological Society of North America, **BioPAX**,[8] a data exchange format for biological pathway data, and the **Phenotype and Trait Ontology (PATO)**,[9] an ontology describing phenotypic qualities. Many other ontologies, such as the **Foundational Model of Anatomy (FMA)** (Rosse & Mejino 2004), do not represent community efforts, but require close coordination among a small group of developers. These projects need support for workflows similar to those of the large, community-based efforts described above. Finally, **OBO Foundry** (Smith *et al.* 2007) is a community effort to develop and publish a collection of core, reference ontologies in the biomedical domain. Among the ontologies we mentioned, GO, OBI, and PATO are included in OBO Foundry. Developers can submit their ontologies to the OBO Foundry if they accept a set of guiding principles. These principles include the requirements that the ontology content be open, that the ontology has a plurality of users, that relations in the ontology are unambiguously defined, that the ontology has a clearly delineated subject matter, ensuring that ontologies in the OBO Foundry do not have overlapping content. One of the key principles of OBO Foundry is that developers must provide procedures for user feedback and must commit to working collaboratively with other participants of the OBO Foundry, thus opening the evolution of their ontology to the input and contributions of other members of the community.

Features of Collaborative Development

The projects that we have discussed are very diverse. The *size* of each ontology ranges from several hundred terms for OBI to tens of thousands of terms for the NCI Thesaurus, FMA, and GO. The *number of contributors* also varies: only several editors contributed to the development of the 80,000 classes in the FMA and there are only three curators who can make any changes to GO. However, dozens of researchers suggest new terms for GO. The ICD-11 and BiomedGT development will be open to hundreds of editors in widely distributed locations. Projects such as OBI, BIRNLex, and RadLex each have several dozen active contributors; approximately 20 content curators participate in CDE content meetings at NCI. Contributors in these projects have very different roles: In some cases, a large number of users can make proposals for changes or new terms, but only a few people can make the actual changes to the ontology (e.g., BiomedGT, OBI, GO); in other cases, the community of contributors is smaller and most of them have direct editing rights (e.g., FMA, BIRNLex). Some projects are beginning to identify a number of roles for their participants, defining a more explicit workflow and different levels of review.

The projects that we have highlighted in the previous section, use a variety of tools to support their collaborative efforts. These tools, however, are not designed specifically for ontology development, and therefore do not integrate naturally with the structured nature of the process.

Discussion tools comprise mostly mailing lists and message boards (as used by OBO Foundry, OBI, GO, BIRNLex, and many others). Whereas these forums provide some archiving capability, the content of the discussion is not linked specifically to the ontology itself. It is often difficult for the ontology authors to find and access the related ontology content several months later. The discussions, the alternatives considered, and the ultimate design rationale are separate from the terms to which they refer and are not accessible from the ontology itself. Moreover, the community cannot easily peruse the ontology to get an overview of those portions of the ontologies that are under active discussion or development as opposed to those that appear stable and less likely to change. Face-to-face meetings are essential to many of the collaborative projects (e.g., OBI, RadLex). In these meetings, participants discuss the modeling issues and identify new areas for development. The minutes of the meetings are recorded in text files and wiki pages, but the discussions do not become accessible in structured form as part of the ontology metadata. In general, an interested person has to travel to one of the face-to-face meetings to have any inkling for the areas of design contention. Some modeling efforts, such as that for the HL7 Reference Information Model (RIM), have acquired the unfortunate reputation that frequent travel and participation in physical meetings offers the only mechanism for potential adopters to learn about design decisions to know how to put the model to use.

[6] http://xwiki.nbirn.net/xwiki/bin/view/+BIRN-OTF-Public/About+BIRNLex

[7] http://radlex.org

[8] http://www.biopax.org/

[9] http://www.bioontology.org/wiki/index.php/PATO:Main_Page

Projects employ a variety of *synchronization and editing mechanisms* for new versions. For example, OBI and GO use systems for version management of software code (SVN and CVS) to maintain the versions, to enable active editors to check in new versions with their changes, and to find differences between versions. Developers in these projects use informal means to agree on who can edit which part of the ontology, so that their edits do not conflict with one another. Because systems such as SVN and CVS were not designed for versioning ontologies, they are cumbersome to use for this purpose. For instance, *diff* services to compare versions of software code assume the use of linear text files, and will fail when used for ontologies, which may be serialized in a variety of ways; ontology developers require a structural or semantic diff (Noy & Musen 2004). Many ontologies are very large and are not modularized. Thus, if curators want to lock out a version, they have to lock out the whole ontology, while others cannot edit it. The modularization of large ontologies in an efficient and practical manner is an active area of research, but no practical solutions exist yet (Seidenberg & Rector 2006).

Recently, the *wiki software* that drives well-known applications such as Wikipedia has gained enormous popularity as a way of soliciting *community participation and feedback*. For example, a platform known as LexWiki currently is at the core of community-based development of BiomedGT. LexWiki is based on Semantic MedaWiki,[10] and enables users to browse an ontology or terminology and to make comments or propose changes to (usually text-based) definitions. In BiomedGT, for example, the LexWiki software stores proposals for ontology revisions as annotations to the ontology; the NCI curators who have the privileges to make changes then open this annotated ontology in Protégé and perform the actual edits there. Wikis provide a natural forum for discussions, and the provenance information for suggested changes is easy to archive. Wikis also enable programmatic extensions, and developers have added capabilities such as class hierarchy browsing, autocompletion, and other features (e.g., see the extensions developed for the Halo project).

Wikis, however, are not intended for ontology development and users cannot easily edit class definitions using this kind of framework. For example, in BiomedGT, curators must switch to Protégé to make the actual changes. This approach, with two unrelated access points to edit an ontology (Protégé and LexWiki), requires additional synchronization, lest users comment on classes in the wiki that already have been changed in Protégé. It is difficult for a wiki environment to support ontology editing directly, since text-based wikis cannot perform even simple semantic checks on the data being entered. Furthermore, each implementation of a wiki environment (such as with BiomedGT) supports a specific workflow and a set of user roles, lacking the flexibility and customizability needed to support the wide variety of paradigms that we observe in development of biomedical ontologies today. Wikis offer excellent platforms for soliciting unstructured feedback. Tools that support a more structured development and that can be custom-tailored to the workflows of specific projects are needed when actual editing needs to be performed (also collaboratively). Developers require a single platform that integrates the collection of community feedback, supports alternative workflows for ontology development and curation, and facilitates collaborative work.

This discussion highlights several key points: First, the trend for opening the development of biomedical ontologies and terminologies to some community of experts is very pronounced, and few, if any, prominent biomedical ontologies are developed in a closed environment any more. Second, different projects have different protocols and workflows, processes for allocating work and for auditing progress, roles for developers, and mechanisms for reaching consensus. Third, none of the tools that ontology developers currently use can support this type of development, leaving the discussions, rationale, and provenance information inaccessible from the final products.

Requirements for Tool Support

Having analyzed the different aspects of collaborative development in a number of ongoing projects, we now summarize the features that such projects would require from the tools that support collaborative development of ontologies. In addition to analyzing the projects described in the previous section, the list below was also informed by requests from the Protégé users and the results of the Collaborative Knowledge Construction challenge at WWW'07 (Noy, Chugh, & Alani 2008).

Tools for discussion and reaching consensus: Almost by definition, an ontology is an artifact that requires its authors to reach consensus. At the same time, our experience demonstrates that developing an ontology is not a straightforward task and the developers can disagree on the best way to model concepts in the ontology or, in fact, on which concepts to model. Thus, tools that support discussion, such as forums and chats are essential. Furthermore, such tools must be tightly integrated with the ontology editor itself. For example, users should be able to associates their notes and discussions with specific ontology components or changes to these components; a mention of a class in a note should be linked to the class view itself, so that one can easily see what the note author is talking about; and so on.

Provenance information: When an ontology is developed in a closed collaboration of a handful of developers, it may not matter which one of them has created or edited a particular concept. The story changes drastically when the pool of developers grows larger and more open: the history of each concepts and information on who changed it and why and when becomes critical. The ability to keep track of this information and to present it to the users is almost a make-or-break requirement for tools for collaborative development.

[10] `http://semantic-mediawiki.org`

Ways to establish trust and credibility: A related issue is the ability to establish credibility of different users, in particular in the more open settings. There are different mechanisms for establishing trust that range from explicit assignment of authority (e.g., specific editors can be designated as trusted sources) to more dynamic ways of building a web of trust.

Views geared towards users with different levels of expertise: In an ontology-developent project, not every user will be satisfied with just having access to a class hierarchy, just as not every user will want to see an OWL expression. Almost any collaborative setting requires different views that vary in their complexity and content. At one end can be very simple presentation, similar to wikis that so many users are already comfortable with; at the other end are interfaces that support the full power of expressive ontology languages.

Access control: We often hear from our users that develop ontologies collaboratively in large groups that one of the features that all ontology-development tools largely lack today is access control: for the most part, any user with writing privileges can edit anything in an ontology. It should be possible to have more fine-grained control, particularly in the development of large ontologies. For example, users with expertise in an area represented by some part of an ontology should be able to edit that part, but may only be able to browse other parts or link to them. Because in some ontologies concept definitions are intertwined and a change in one part can affect definitions in another part, making such separation is far from trivial.

User roles: An extension of various access-control policies is a more detailed model of user roles. Many ontology-develoment projects today in fact maintain separation between what different users can do: For instance, some users can make proposal for changes but not make the changes themselves; others can comment on these proposal, but not put out new ones; another group of users can make decisions based on the discussions and affect the changes in the ontology itself.

Workflow support: Many collaborative development projects have specific workflows associated with making changes. A workflow specification may include different tasks that editors are charged with; the process for proposing a change and reaching consensus; roles that different users play, and so on. We are only beginning to understand different workflow models that collaborative ontology development requires. Flexible support for these workflows must be an integral part of tools for collaborative development.

Collaborative Protégé

We started addressing some of the challenges listed in the previous section in Collaborative Protégé—an extension to the Protégé ontology-editing environment that supports collaboration.[11]

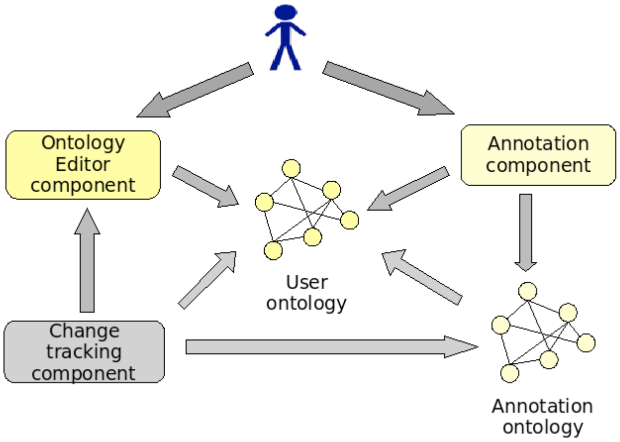


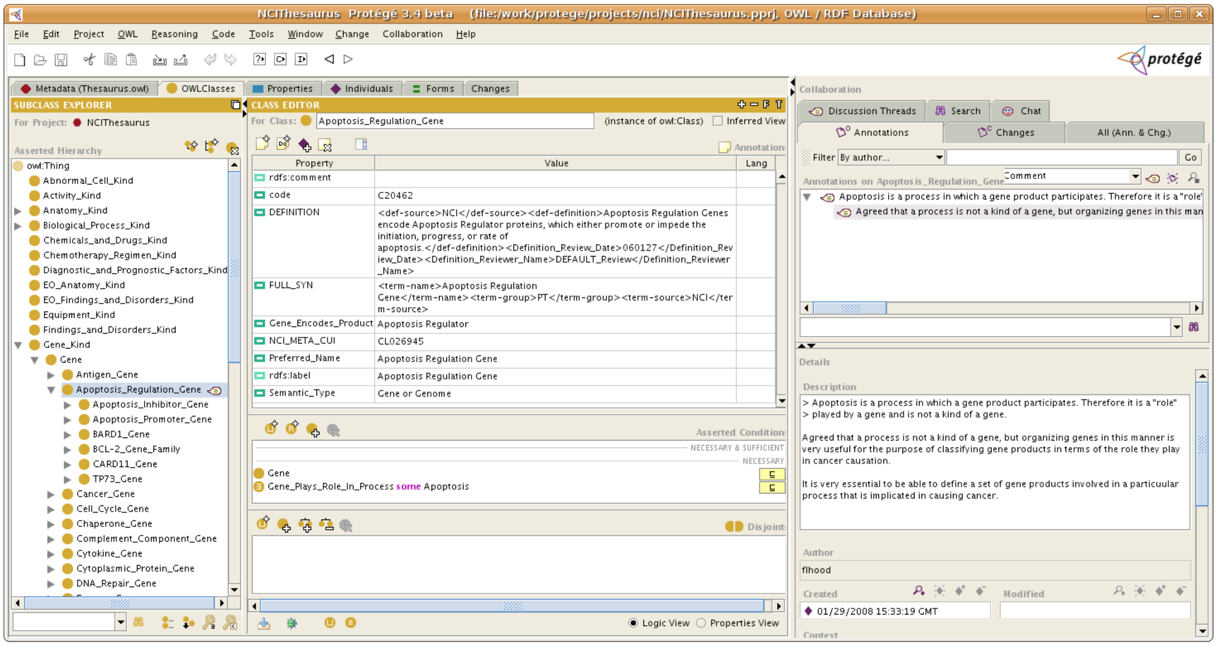
Figure 2: Core components of the Collaborative Protégé architecture that support the annotation of ontology components and of changes in the ontology.

The Protégé system is an open-source ontology editor and knowledge-base framework developed by Stanford Medical Informatics. It supports a number of knowledge-representation formalisms, from frame-based representation to RDF and OWL. Protégé can be used in a client–server mode where multiple users edit the same ontology simultaneously; when one user makes a change, others see it immediately. We developed the prototype of Collaborative Protégé as an extension to this mode. In Collaborative Protégé, users can comment on ontology components, discuss changes, and interactively reach consensus on modeling decisions. We implement these features by supporting the association of annotations to any component of the ontology or to any change that occurs in the ontology. The tool also provides support for different visualizations of the annotations, which users can customize by specifying different filtering criteria (Figure 1).

The main functionality provided by the Collaborative Protégé prototype are:

- Annotation of ontology elements, such as classes, properties, individuals
- Annotation of ontology changes, such as class creation, deletion, renaming, etc.
- Support for change proposals and voting of proposals
- Support for filtering of existing annotations
- Support for searching of annotations based on simple or complex criteria
- Support for discussion threads
- Support for chat

[11] http://protege.stanford.edu/doc/collab-protege/



owl:Thing
- Abnormal_Cell_Kind
- Activity_Kind
- Anatomy_Kind
- Biological_Process_Kind
- Chemicals_and_Drugs_Kind
- Chemotherapy_Regimen_Kind
- Diagnostic_and_Prognostic_Factors_Kind
- EO_Anatomy_Kind
- EO_Findings_and_Disorders_Kind
- Equipment_Kind
- Findings_and_Disorders_Kind
- Gene_Kind
  - Gene
    - Antigen_Gene
    - Apoptosis_Regulation_Gene
      - Apoptosis_Inhibitor_Gene
      - Apoptosis_Promoter_Gene
      - BARD1_Gene
      - BCL-2_Gene_Family
      - CARD11_Gene
      - TP73_Gene
    - Cancer_Gene
    - Cell_Cycle_Gene
    - Chaperone_Gene
    - Complement_Component_Gene
    - Cytokine_Gene
    - Cytoplasmic_Protein_Gene
    - DNA_Repair_Gene

For Class: Apoptosis_Regulation_Gene

Property	Value	Lang
rdfs:comment		
code	C20462	
DEFINITION	<def-source>NCI</def-source><def-definition>Apoptosis Regulation Genes encode Apoptosis Regulator proteins, which either promote or impede the initiation, progress, or rate of apoptosis.</def-definition><Definition_Review_Date>060127</Definition_Review_Date><Definition_Reviewer_Name>DEFAULT_Review</Definition_Reviewer_Name>	
FULL_SYN	<term-name>Apoptosis Regulation Gene</term-name><term-group>PT</term-group><term-source>NCI</term-source>	
Gene_Encodes_Product	Apoptosis Regulator	
NCI_META_CUI	CL026945	
Preferred_Name	Apoptosis Regulation Gene	
rdfs:label	Apoptosis Regulation Gene	
Semantic_Type	Gene or Genome	

Gene ⊑
Gene_Plays_Role_In_Process some Apoptosis ⊑

- Apoptosis is a process in which a gene product participates. Therefore it is a "role'
  - Agreed that a process is not a kind of a gene, but organizing genes in this man

> Apoptosis is a process in which a gene product participates. Therefore it is a "role"
> played by a gene and is not a kind of a gene.

Agreed that a process is not a kind of a gene, but organizing genes in this manner is very useful for the purpose of classifying gene products in terms of the role they play in cancer causation.

It is very essential to be able to define a set of gene products involved in a particuular process that is implicated in causing cancer.

Author: flhood

Created: 01/29/2008 15:33:19 GMT

Figure 1: The Protégé user interface, with the Collaborative Protégé plug-in. This screen capture shows the Classes tab, in which the user edits and browses the classes that describe a domain ontology–here the NCI Thesaurus. The left panel shows the class tree; the middle panel displays the form for entering and viewing the description of the selected class (Apoptosis_Regulation_Gene), as a collection of attributes; the right column shows the discussion among users about this class. Other tabs include: the Properties tab, for creating and editing properties; the Forms tab, for customizing the widgets and layout of knowledge-entry forms; the Individuals tab for creating instances of classes and entering particular property values for those instances; and any other special-purpose "plug-in" tab that the user may want to use.

Another important feature that is currently only experimental is the support of ratings for existing user annotations. This feature enables the implementation of different web-of-trust algorithms.

Figure 2 shows a diagram of the core components of the system that support collaborative development of ontologies and specifically the annotation process. The user interacts with the *Ontology Editor Component* and the *Annotation Component*. The editing component is provided by the underlying Protégé system. The *Annotation Component* allows the user to annotate ontology components, such as classes, properties and individuals, as well as ontology changes, such as class creation or deletion, with annotation types defined in the *Annotation ontology* (Noy *et al.* 2006).

The *Annotation ontology* is a RDF(S) ontology that provides the structure for the annotation types supported by the tool. The annotation types are extensions of the Annotea (Kahan & Koivunen 2001) annotations and contain concepts such as Comment, Advice, Example, etc. User annotations are stored as instances of the predefined annotation classes and can be used for annotating both ontology components as well as ontology changes.

The *Change tracking component* is responsible for intercepting the user actions in the GUI and creating change annotations attached to the changed ontology components. Change annotation types are defined in the *Annotation ontology*.

The system supports discussion threads by allowing the users to reply to the comments of other users. This feature is realized by a flexible representation of annotations, which can themselves be annotated. The system supports the rating and voting of proposals, which are represented as annotation types in the *Annotation ontology*.

Other components of the system are the *searching* and the *filtering* components, which are crucial in dealing with large bodies of annotations. The filtering component will be used in future versions of the system to create user-defined views of an ontology based on user preferences.

Future Work

We continue to develop Collaborative Protégé actively. Our future plans include implementation of a web client for Collaborative Protégé, an infrastructure to enable users to custom-tailor their display, a "MyPage" facility to provide summaries of the activities that have occurred since the last time the user has logged into the system, notifications of changes for concepts for which the user registered, and so on.

We are also working on implementing a flexible workflow infrastructure: just as Protégé itself generates knowledge-acquisition tools based on ontology definition, we will have Collaborative Protégé generate custom-tailored tools that support specific collaborative workflows based on the users' descriptions of their processes. For example, a project can describe the different roles that users play, actions that users with these roles can perform, specific protocols for reaching consensus, making changes, initiating discussion, and so on. Them for example, depending on the role of the user who logs into the system, certain menus and button will be enabled or disabled. The system will notify the users if it expects certain actions from them. The users will be able to review their pending tasks, or to see the status of the tasks that they assigned to other users. Such flexible and customizable support requires the definition of a workflow model for collaborative ontology development and integration of this model with other components such as change and annotations representations.

References

Fragoso, G.; de Coronado, S.; Haber, M.; Hartel, F.; and Wright, L. 2004. Overview and utilization of the nci thesaurus. *Comparative and Functional Genomics* 5(8):648–654.

Hartel, F. W.; Coronado, S. d.; Dionne, R.; Fragoso, G.; and Golbeck, J. 2005. Modeling a description logic vocabulary for cancer research. *Journal of Biomedical Informatics* 38(2):114–129.

Kahan, J., and Koivunen, M.-R. 2001. Annotea: an open RDF infrastructure for shared web annotations. In *Proceedings of the 10th International World Wide Web Conference*, 623–632.

Noy, N. F., and Musen, M. A. 2004. Ontology versioning in an ontology-management framework. *IEEE Intelligent Systems* in press.

Noy, N. F.; Chugh, A.; Liu, W.; and Musen, M. A. 2006. A framework for ontology evolution in collaborative environments. In *Fifth International Semantic Web Conference, ISWC*, volume LNCS 4273. Athens, GA: Springer.

Noy, N. F.; Chugh, A.; and Alani, H. 2008. The ckc challenge: Exploring tools for collaborative knowledge construction. *IEEE Intelligent Systems* 23(1):64–68.

Rosse, C., and Mejino, J. L. V. 2004. A reference ontology for bioinformatics: The foundational model of anatomy. *Journal of Biomedical Informatics.*

Seidenberg, J., and Rector, A. 2006. Web ontology segmentation: Analysis, classification and use. In *15th International World Wide Web Conference*.

Sioutos, N.; de Coronado, S.; Haber, M.; Hartel, F.; Shaiu, W.; and Wright, L. 2007. NCI Thesaurus: A semantic model integrating cancer-related clinical and molecular information. *Journal of Biomedical Informatics* 40(1):30–43.

Smith, B.; Ashburner, M.; Rosse, C.; Bard, J.; Bug, W.; Ceusters, W.; Goldberg, L. J.; Eilbeck, K.; Ireland, A.; Mungall, C. J.; Leontis, N.; Rocca-Serra, P.; Ruttenberg, A.; Sansone, S. A.; Scheuermann, R. H.; Shah, N.; Whetzel, P. L.; and Lewis, S. 2007. The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. *Nature Biotechnology* 25(11):1251–5.