COE: Tools for Collaborative Ontology Development and Reuse

Pat Hayes, Thomas C. Eskridge,
Thomas Reichherzer, Raul Saavedra
Institute for Human & Machine Cognition
40 S. Alcaniz Street
Pensacola, FL, 32502, USA
{phayes,teskridge,treichhe,rsaavedra}@ihmc.us

Mala Mehrotra, Dmitri Bobrovnikoff
Pragati Synergetic Research
914 Liberty Court
Cupertino, CA, 95014, USA
{mm,dmitri}@pragati-inc.com

Keywords: Knowledge Management, Ontologies, Search and Retrieval, Visualization

Abstract

The meaningful integration and comprehension of data from many different sources poses a difficult challenge for the Intelligence Community. Establishing the underlying semantics of that data and developing semantics for new or specialized scenarios is a key part of the solution to that challenge. The emerging Semantic Web technologies address this challenge by developing an 'open network' of formally described concepts, in which the publication of a formal ontology allows other users to re-use concepts when describing new knowledge. This vision requires that the tasks of knowledge entry and review are conducted in an environment providing direct, intuitive access to the formally defined meanings of existing concepts, and tools to determine relationships between concepts which may have been composed in isolation from each other. We outline this vision and describe an initial implementation of a software suite, Collaborative Ontology Environment (COE), which uses concept mapping and cluster analysis to provide an ontology composition, comprehension, and reuse framework.

1. Introduction

The Intelligence Community is faced with the challenge of integrating data from numerous different sources at varying levels of resolution, and comprehending the importance of that data with respect to well-understood threat scenarios as well as one-off events (Semy *et al.* 2004). It is not possible to surmount this challenge simply with more databases, or standardized schemas on top of existing databases (Berners-Lee *et al.* 2001), because that does not address the need for developing a collaborative agreement on semantic meaning of the data, or provide tools that permit operators using or developing this shared meaning to understand the extent and limitations of their efforts.

The Semantic Web (henceforth SW) envisions a planet-wide network of content expressed in formal ontologies. It provides a growing set of standards (RDF, OWL, SWRL) and a growing body of available software for realizing this vision (Berners-Lee *et al.* 2001). The high-level nature of ontology content, and the universal use of Web conventions (URIs, XML) together provide new opportunities for formal and semi-formal knowledge from a wide variety of sources and formats, ranging from free-text archives to tabular databases, to be used together effectively in a distributed information network.

In order to be fully effective, such a network of formally expressed content must have many conceptual connections between its components. It is not practicable to impose a single uniform *conceptual* framework on all users of an entire network and this is not necessary for successful integration. The SW vision might be called 'voluntary syndication' of knowledge composition: a distributed network of linked mini-ontologies integrated into a connected web by their re-use of concepts; and composed by subject-matter experts or small teams, rather than professional ontology engineers. (Although there is no *requirement* of small size, simple considerations of cost, both of composition and network transfer, suggest that small ontologies will outnumber large ones.) This notion of 'linking' is central: we mean simply that these pieces of formalized knowledge will *re-use the same concepts*, and thereby achieve interoperability simply by, as it were, being written in the same language.

Conventional ontology-authoring tools such as Protégé (Noy *et al.* 2001) were developed to help designers of large, often proprietary, ontologies that are intended for use by specialists in highly technical domains. Such ontologies are analogous to large pieces of

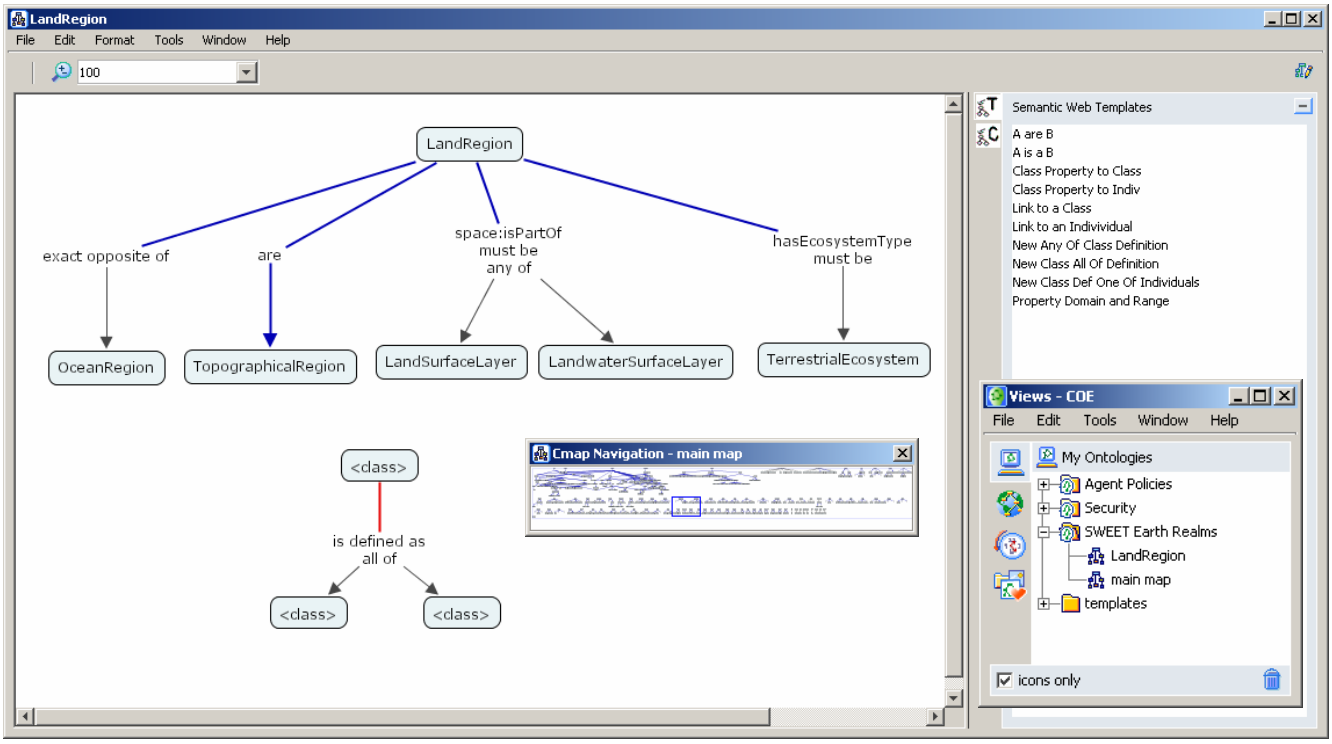

Figure 1: COE concept map interface with template side panel.

software, and their development proceeds similarly, with sharply distinct populations of developers and users. Such 'knowledge models' are often intended to provide checks on correctness or conformity of data. Although SW technology can, and will, make use of such highly developed knowledge models, the role of ontology on the SW undergoes a fundamental change: rather than input to specialized data handlers, it is seen as a form of public markup. The concepts are spread all over the network, and achieve their power from the extent of their re-use, and linkages to other concepts on the distributed network. If my OWL ontology uses a concept from your OWL ontology, then an agent (human or software) using my ontology is also able to extract relevant content from yours, and use them together to draw conclusions (by applying any inference scheme which conforms to the OWL semantics (Patel-Schneider *et al.* 2004)). This not only creates an opportunity for ontology authors, but also an obligation on them to re-use concepts in ways that are consistent with their expression in those original ontologies. To achieve this, new knowledge exploration and composition tools are needed. The body of this paper describes such a prototype system, COE, based on technology that has been successfully applied in education and training applications (Cañas *et al.* 2004).

COE is a suite of Java-based tools for displaying and editing OWL content in the form of simple node-link diagrams called *concept maps* (Cmaps), translating these from and into the OWL/RDF/XML interchange notation, searching through ontologies to locate potentially useful concepts, and locating clusters of contextually relevant concepts in existing ontologies. The software is freely available for testing and can be downloaded from http://homam.ihmc.us.

2. Concept Mapping and Ontologies

Concept maps (Novak 1977; Novak and Gowin, 1984) are collections of propositions, which can be seen as simplified natural language sentences, displayed as a two-dimensional network of labeled nodes and links. Concept maps are "informal" representations that facilitate knowledge capture for human examination and sharing; they have been primarily used in educational and training settings. CmapTools (Cañas *et al.* 2004), on which COE is based, allows users to build knowledge models consisting of sets of interconnected concept maps annotated with material such as text documents, diagrams, and video clips. It provides rich, searchable, browsable knowledge models available for navigation and collaboration across geographically-distant sites. It supports both synchronous and asynchronous methods for collaborative map development. CmapTools have been downloaded to approximately 30,000 institutions worldwide and have been used as a tool for knowledge acquisition and as a GUI to expertise-capturing software (Briggs *et al.* 2004; Coffey J. W. 1999; Clark *et al.* 2001). The success of the CmapTools interface stems from its flexibility and ease of use by subject-matter experts and its ability to represent complex networks of relationships in ways that

foster understanding and permit rapid visual inspection.

COE is comprised of a set of Java modules that extend CmapTools with features that are particularly oriented towards the expression of formalized SW ontologies, while retaining and extending the human-oriented advantages of the concept map interface. The purpose is to give domain experts the ability to construct, share, and examine Web ontologies. The system supports importing, editing, and storing of OWL Ontologies in the form of concept maps, exporting concept maps to OWL/XML files, searching for concepts and properties in existing concept maps and Web ontologies, and examining the different roles which concepts play in other ontologies, which gives composers the ability to choose appropriate concepts for their ontologies.

2.1. Representing Ontologies as Cmaps

To bridge the gap between the informal nature of concept maps and the formal, machine-readable Web ontology languages, COE uses a set of conventions and guidelines that enables users to construct syntactically valid Web ontologies using the concept-mapping interface. These conventions retain as far as possible an intuitive reading of the concept map while faithfully capturing the precision of the OWL syntax, and are based on a few basic ideas (which make them easy to learn). English words and phrases are used as far as possible, and we avoid the 'mathematical logic' terminology that pervades the OWL documentation. For example, the fact that land regions are topographical regions would be represented in OWL-XML syntax by saying that *LandRegion* is a 'subclass' of *TopographicalRegion*; in COE, as shown in Figure 1, it is rendered using a link labeled '*are*' from the subject to the object of the sentence, mirroring the syntax of a simple English sentence and avoiding the (logically accurate but conceptually jarring) use of the '*subclass*' terminology.

Similarly, the fact that all parents of humans must be human (itself an illustration of the kind of mundane fact that must be rendered explicitly in a formal ontology) would be rendered in OWL syntax by creating a restriction class on the property *hasParent* to the class Human, and then asserting that Human is a subclass of it. In COE such a restriction is written as a single link (constructed by a single mouse operation) with a special '*must be*' label (selected by a single motion from a pulldown menu). It can be read directly as slightly broken English to be a kind of 'note' attached to the category. Figure 1 illustrates two restrictions involving the classes *LandRegion*, *LandSurfaceLayer*, *LandwaterSurfaceLayer* and *TerrestrialEcosystem* and the properties *isPartOf* and *hasEcosystemType*.

Some OWL constructions do not transcribe into anything remotely resembling idiomatic English, and so are rendered in COE using graphical or labeling conventions. As a side effect, we have found that importing OWL ontologies into COE often makes their essential nature quickly apparent. It is easy to distinguish ontologies which are largely taxonomic, since COE displays the subclass links with a distinctive blue color and connects them together into a tree or graph. Ontologies which are less concerned with classes and more with relationships (called *properties* in OWL) have many dotted links, arising from COE's display of domain and range information. Strict concept definitions (which are quite rare in OWL, for technical reasons) are clearly marked by red links (see Figure 1); and so on. COE's visual layout also draws attention to 'missing' information, which is regrettably common in OWL formalizations, such as unspecified class names or missing domain and range information.

Space does not allow a detailed description of the COE conventions, which can be obtained from the website. Some general aspects of the design are important, however.

First, it reflects a basic idea which is easy to grasp and which it uses consistently: properties always label links, classes and individual things label nodes (the various OWL categories of 'thing' are indicated by the style of the node rendering.) Properties of a property (that it is transitive, such as *ancestorOf*, or symmetric, such as *brother*) are indicated by extra tags on the link label, and OWL restrictions on a property are indicated by predefined phrases such as *must be*, *can be*, *exactly one*, *at least 3* that are attached to the property name as part of the link label. This link/node conceptual discipline is notably absent from the XML syntax for OWL, which treats all entities similarly.

Second, the automatic layout algorithm embedded in COE is critical to the tool's utility. It has been carefully adapted to the display of OWL ontologies, which are impossibly 'tangled' (i.e., non-planar) when considered as abstract graphs. The COE layout uses heuristics to determine when it would be good to 'clone' a node, i.e., to split off part of the graph and display it as a separate piece of the concept map. It also uses heuristics to guess which parts of the OWL file are best grouped together into a single tree. The CmapTools software allows for search, rapid navigation and image zooming through large ontology maps using text-based matching on concept names. We have found that the conceptualization of an ontology as something displayed on a surface, and the associated metaphors of *moving* and *looking more closely* are powerful user aids in organizing and comprehending large bodies of information.

2.2. Templates for Composing Ontologies

Users can construct ontologies by direct manipulation of the CmapTools interface, but for several of the OWL constructs, this requires a number of steps to make the appropriate linkages and set the box and line styles according to convention. To ease this, COE provides templates for commonly used OWL structures such as Union, Intersection, and Restriction. An exam-

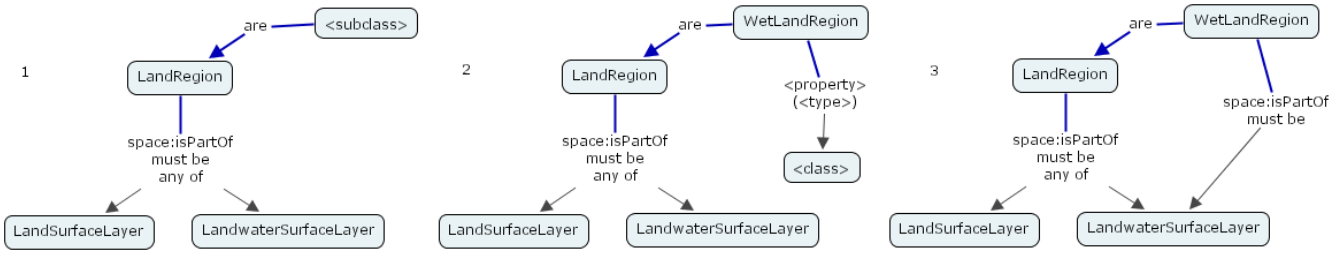

Figure 2: Using templates to create *WetLandRegion*.

ple of a template, for the OWL *intersectionOf* restriction–intuitively expressed by the words 'all of'–is shown in Figure 1. The text enclosed in brackets is meant for user replacement, and will generate an error at export time if left in the ontology map by mistake. Users can drag and drop these templates from the side panel and onto an existing node, replacing the root element of the template, or directly onto the ontology map canvas.

For example, Figure 2 shows the incremental construction of a subclass of *LandRegion* called *WetLandRegion*. Step one shows the result of dragging the *A is B* template (which simply has the nodes <*subclass*> *are* <*class*>) and dropping it on the existing *LandRegion* node. In step 2, the user types in the new subclass name, *WetLandRegion*, and then drags the *Class property to Class* template and drops it on *WetLandRegion*. The relation *space:isPartOf* is typed in, and the *must be* modifier is selected from a pulldown menu located to the right of the relation typing area (not shown). Finally, we drag the template element labeled <*class*> to the existing node labeled *LandwaterSurfaceLayer*, which merges the two nodes, yielding the final result in rectangle 3. It is much quicker to do than it is to describe.

3. Cluster-Based Vicinity Concepts

COE incorporates a cluster-based *vicinity concepts view* that complements COE's definitional view by showing concepts from existing ontologies that are relevant to the COE user's current focus. The view is powered by Pragati, Inc.'s Multi-Viewpoint Clustering Analysis (MVP-CA) software (Mehrotra *et al.* 2002), an instance of which is maintained on a Web server at IHMC. MVP-CA is an integrated suite of cluster-based cognitive assistance tools based around the core capability of grouping concepts that occur in similar contexts, but for which direct formal relationships do not necessarily exist (Mehrotra 2002). These *vicinity concepts* aid the user's comprehension of existing ontologies and lead to "fortuitous" reuse opportunities—even when the source ontologies are unfamiliar to the user.

3.1. Finding Relevant Concepts

MVP-CA's clustering engine applies a hierarchical, agglomerative, clustering algorithm to group similar OWL concepts into the clusters that underlie the vicinity concepts view. A simple wizard-based interface enables users to cluster ontologies with minimal knowledge of the underlying mechanism. For advanced users who require finer-grained control, detailed parsing and clustering parameters are available. An analysis subsystem provides multiple views of the clusters, and can use statistics- and heuristics-based algorithms to rank and filter them based on their probable value to the user.

As illustrated in the figures, node labels in OWL ontologies are typically composite terms formed by concatenating English words. Although not mandated by any formal standards, this is a widely used convention for producing intuitively readable labels for complex concepts. MVP-CA exploits this common practice by decomposing concept labels into text fragments that are primary inputs to its clustering system.

When a COE user queries the vicinity concepts view for a concept of interest, a web services-based query server returns a set of *focus clusters* that are relevant to the query. A client-side component, running in the COE environment, renders the focus clusters as a set of Euler diagrams. The user can interact with the diagrams by moving, zooming, and sending information about concepts to other areas of the COE interface.

To determine which clusters are relevant to a query, the system employs a two-stage search algorithm. In the first stage, it finds concepts that textually match the query, based on simple substring comparisons. The second stage finds the focus clusters in which the matching concepts have *stabilized*—that is, the smallest clusters that fully contain the matching concepts.

Each diagram contains two regions, one nested inside the other. The outer region shows those concepts that stabilized in the focus cluster; the inner region shows the concepts that stabilized in sub-clusters of the focus cluster. Due to the hierarchical nature of the clustering algorithm, all concepts in the inner region are also stable with respect to the focus cluster.

3.2. Case Studies

When a user issues a vicinity concepts query, the system searches all clusters in the query server's repository. These clusters may be based on multiple ontologies, or on combinations of ontologies. For example, in Figure 3a, the query term *region* finds clusters from

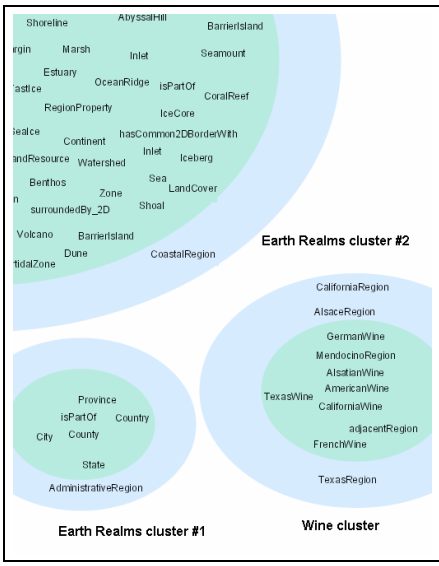

Figure 3a: Vicinity concepts for *region*.

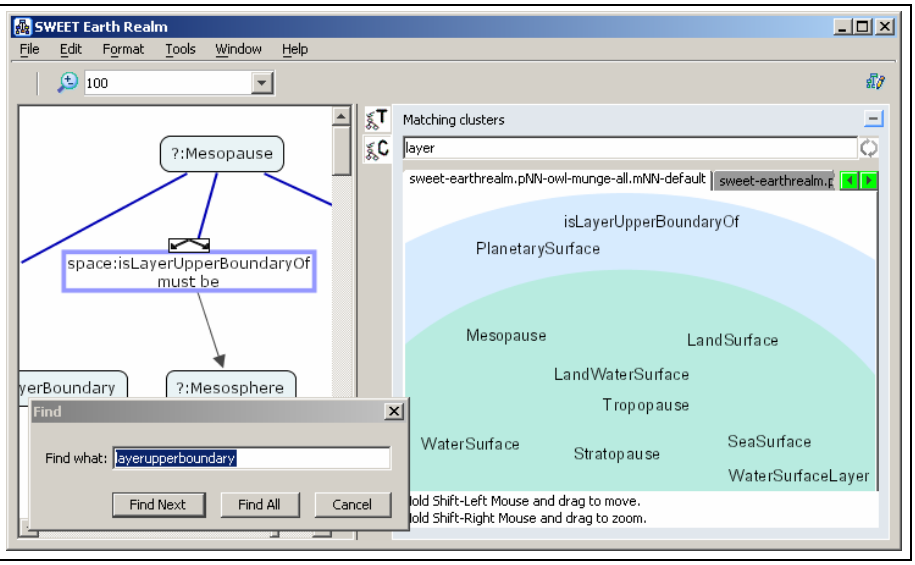

Figure 3b: Ontological and vicinity views displayed together.

both the Wine sample ontology and JPL's SWEET Earth Realms[1] ontology. In this case, the clusters from the two ontologies are quite different, which is unsurprising given the minimal overlap between the domains. The Wine cluster contains concepts relating to the types of wines produced in different regions. The two clusters from Earth Realms show different aspects of *region* within that ontology: the first cluster groups region-related concepts based on administrative boundaries, while the second, which contains the first cluster but is broader in scope, includes geographical features such as volcanoes and beaches.

Figure 3b shows the complementary nature of the collaboration between COE and Pragati's tool suite. A portion of COE's definitional view for the Earth Realms ontology appears in the left panel; the right panel shows vicinity concepts for one cluster found by the query *layer*. COE's definitional view provides the user with the ontological details necessary to define and modify OWL concepts. The vicinity concepts view, on the other hand, exposes a number of contextually related concepts, including *SeaSurface*, *WaterSurface*, *LandSurface*, *Mesopause*, *Stratopause*, and *Topopause,* that aid the COE user's understanding and present potential reuse opportunities.

The system selected the cluster shown because, based on a simple substring match, *isLayerUpperBoundaryOf* meets the user's *layer* query. However, the vicinity concepts shown are far from the results of a simple grep operation—the concepts clustered together due to the aggregate similarity of their definitions, not necessarily or exclusively because they all reference *isLayerUpperBoundaryOf*. Some of the concepts in the cluster, e.g., *Mesopause* (shown in the COE view) and *WaterSurface* do define restrictions on the *isLayerUpperBoundaryOf* property. However, others, e.g., *WaterSurfaceLayer*, do not directly reference *isLayerUpperBoundaryOf* at all. *WaterSurfaceLayer* appears in this cluster because, among other characteristics, it *isAdjacentTo WaterSurface*. Inter-concept similarity may be due to similar restriction declarations, parallel disjointness relationships, inverses, etc., or, usually, combinations thereof.

By exposing contextually related conceptual modules in the system, clustering reveals intuitive information about the knowledge base that formal analysis cannot easily achieve. Through an intelligent, user-directed search, the vicinity concepts view reduces information overload for ontology developers, significantly lowering the knowledge entry barrier. Most critically for our point, it provides a unique way for users to access concepts in existing network content and gain insight into their ontological utility.

4. Related and Future Work

The promise of the Semantic Web has sparked considerable interest in tools to aid the construction of ontologies (Clark *et al.* 2001; Davies *et al.* 2003; Farquhar *et al.* 1997). A prominent example is the Protégé knowledge acquisition tool built at Stanford University (Noy *et al.* 2001). Since its inception in the mid-1980s, the tool has undergone several revisions, offering today a variety of plug-ins to extend its capabilities. Similar in spirit to our work are the visualization plug-ins for Protégé such as ezOWL or Jambalaya (ezOWL 2004; Ernst, N.A. and Storey, M.-A. 2003). These interfaces are closely related to graphical software modeling tools such as the Unified Modeling Language (UML) which use a specific set of graphical notations to represent a design.

[1] The SWEET Earth Realm ontology can be found at http://sweet.jpl.nasa.gov/sweet/earthrealm.owl

Our system differs from these software tools in that it provides an abstraction from the intricate details encoded in Web ontologies, allowing users to focus on the conceptual nature of the ontology design, while at the same time providing the ability to export strict OWL/XML content. The abstraction and the simplicity of the concept-mapping interface make our tool accessible to a broad group of individuals without the need for extensive training; and it is thoroughly integrated with the current generation of Web standards.

By developing improved cluster filtering algorithms, we hope to further reduce information overload. Additionally, we plan to implement a semi-automated cluster categorization system that will enable users to direct their cluster-based queries with greater precision.

5. Summary

The Semantic Web vision requires tools to allow users with different technical backgrounds to collaborate in the construction of distributed knowledge bases. COE is a prototype of such a tool, combining an intuitive graphical user interface based on concept maps that facilitate ontology construction and understanding with sophisticated cluster concept analysis to aid the search for relevant concepts. All the components of COE have been used successfully in related areas, and we are confident that this basic design will become the preferred technique for composing knowledge intended to be accessed from, and contribute to, a distributed information Web.

Acknowledgments

We gratefully acknowledge the contributions of our colleagues Alberto Cañas and the CmapTools developer team, and helpful suggestions by Steve D. Cook. This research was supported in part by DARPA and the Department of Defense.

References

Berners-Lee, T., Hendler, J., and Lassila, O. 2001. The semantic web. Scientific American, 284(5):34-43.

Blythe, J., Kim, J., Ramachandran, S., and Gil, Y. 2001. An integrated environment for knowledge acquisition. In *Proc. 6th Intl. Conf. on Intelligent User Interfaces*, pp. 13-20. ACM Press.

Briggs, G., Shamma, D., Cañas, A. J., Carff, R., Scargle, J., and Novak, J. D. 2004. Concept maps applied to Mars exploration public outreach. In *Proc. First Intl. Conf. on Concept Mapping, Volume 1*, pp. 109–116, U. of Navarra.

Cañas, A. J.; Hill, G.; Carff, R.; Suri, N.; Lott, J.; Eskridge, T.; Gómez, G.; Arroyo, M.; and Carvajal, R. 2004. CmapTools: A knowledge modeling and sharing environment. In *Proc. First Intl. Conf. on Concept Mapping, Volume 1*, pp. 125–133, U. of Navarra.

Clark, P., Thompson, J., Barker, K., Porter, B., Chaudhri, V., Rodriguez, A., Thomere, J., Mishra, S., Gil, Y., Hayes, P., and Reichherzer, T. Knowledge entry as the graphical assembly of components. 2001. In *Proc. Intl. Conf. on Knowledge Capture*, pp. 22-29. ACM Press.

Coffey, J. W. 1999. Institutional memory preservation at NASA Glenn Research Center. Unpublished technical report, NASA Glenn Research Center, Cleveland, OH, USA.

Davies, J., Duke, A., and Sure, Y. 2003. OntoShare: a knowledge management environment for virtual communities of practice. In *Proc. Intl. Conf. on Knowledge Capture*, pages 20-27. ACM Press.

Ernst, N. A., Storey, M.-A. 2003. A Preliminary Analysis of Visualization Requirements in Knowledge Engineering Tools. Unpublished Technical Report, University of Victoria, Victoria, Canada.

ezOWL: Visual OWL Editor Plugin for Protégé. Web page http://iweb.etri.re.kr/ezowl/.

Farquhar, A., Fikes, R., and Rice, J. 1997. The Ontolingua server: A tool for collaborative ontology construction. Int. J. of Human-Computer Studies, 46(6):707-727.

Gil, Y. 1994. Knowledge refinement in a reflective architecture. In *Twelfth Natl. Conf. on A.I.*. AAAI Press.

McGuinness, D., and Harmelen, F. 2004 *OWL Web Ontology Language Overview*. W3C recommendation, World Wide Web Consortium, 2004 (http://www.w3.org/TR/owl-features/).

Mehrotra, M. 2002. Ontology Analysis for the Semantic Web. In *AAAI-02 Workshop on Ontologies and the Semantic Web*. AAAI Press.

Mehrotra, M. and Bobrovnikoff, D. 2002. Multi-ViewPoint Clustering Analysis Tool. In *Proc. Eighteenth National Conference on Artificial Intelligence*, pp 1006-1007. AAAI Press.

Novak, J. and Gowin, D. B. 1984. *Learning How to Learn*. Cambridge University Press.

Novak, J. 1977. *A Theory of Education*. Ithaca, Illinois, Cornell University Press.

Noy, N. F., Sintek, M., Decker, S., Crubezy, M., Fergerson, R. W., and Musen, M. A. 2001. Creating Semantic Web Contents with Protege-2000. *IEEE Intelligent Systems* 16(2):60-71.

Patel-Schneider, P. F., Hayes, P, Horrocks, I. 2004. *OWL Web Ontology Language Semantics and Abstract Syntax*. W3C recommendation, World Wide Web Consortium, 2004 (http://www.w3.org/TR/owl-semantics/).

Semy, S.K, Hetherington-Young, K.N., Frey, S.E. 2004. Ontology Engineering: An Application Perspective. Unpublished research report. The MITRE Corporation, August 2004. (http://www.mitre.org/work/tech_papers/tech_papers_04/04_0847/04_0847.pdf)