Collaborative knowledge capture in ontologies

Conference Paper · January 2005

DOI: 10.1145/1088622.1088641 · Source: DBLP

6 authors, including:

Patrick Hayes
Florida Institute for Human and Machine Cognition
157 PUBLICATIONS **8,934** CITATIONS

Thomas C Eskridge
Florida Institute of Technology
103 PUBLICATIONS **1,031** CITATIONS

Thomas Reichherzer
University of West Florida
63 PUBLICATIONS **1,034** CITATIONS

Mala Mehrotra
securboration
18 PUBLICATIONS **153** CITATIONS

Collaborative Knowledge Capture in Ontologies

Pat Hayes, Thomas C. Eskridge, Raul Saavedra
IHMC
Pensacola, FL 32502, USA
{phayes, teskridge, rsaavedra}@ihmc.us

Thomas Reichherzer
Indiana University
Bloomington, IN 47405, USA
treichhe@cs.indiana.edu

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

ABSTRACT

This paper describes a new environment, COE, for capturing and formally representing expert knowledge for use in the Semantic Web. COE exploits the ease of use and rapid knowledge construction capabilities of the CmapTools concept mapping system and extends them to support the import and export of formal, machine-interpretable knowledge representations, such as OWL, across multiple ontologies. Pragati's Exposé tool suite complements COE's ontology construction, browsing and navigation features by providing cluster-based search capabilities that expose existing reusable concepts relevant to the user's focus of attention.

Categories and Subject Descriptors
I.2.4 Knowledge Representation Formalisms and Methods – *representation languages, semantic networks.*

Keywords
concept maps / diagrams, knowledge management, ontologies, information retrieval.

INTRODUCTION
The construction of Concept Maps [19] is a proven method for explicating and communicating domain knowledge [15]. CmapTools [3] has been developed over the previous decade as an intuitive, human-centered computer interface for creating and managing concept maps. Several large knowledge capture efforts have used CmapTools with great success [5]. These efforts have demonstrated the effectiveness of the elicitation methodology to create conventional concept maps from domain experts' knowledge. However, the concept maps built using these methodologies and tools are "informal" representations that are meant to communicate knowledge between people, and are not sufficiently formally specified to be used by automated reasoners.

The recent deployment of the semantic web (SW), a set of web-based standards, notably RDF and OWL, for representing ontological content, requires tools which enable human composers to rapidly capture knowledge in formally exact frameworks, making use of already 'published' concepts in other Web ontologies wherever possible. This "distributed syndication" model of knowledge capture, which is integral to the success of the 'open network' Semantic Web, requires tools which provide structured access to concept relationships in published formal ontologies and allow human subject-matter experts to easily compose formal knowledge.

The body of this paper describes a prototype of the knowledge capture tools which will be required to facilitate the Semantic Web, implemented as a tool called COE (Collaborative Ontology Environment), which combines an OWL ontology viewing/editing environment based on concept maps with a suite of cluster-based knowledge re-use tools developed by Pragati. COE can display any OWL or RDF ontology as a readable concept map, supports intuitive editing and construction of these 'ontology maps', allows users to rapidly locate related concepts in published SW ontologies, and outputs legal OWL/RDF/XML. The goal is to enable rapid and intuitive capture of machine interpretable knowledge by combining navigation, comprehension, selection and construction of knowledge in a single collaborative environment with an intuitive GUI.

KNOWLEDGE ELICITATION WITH CONCEPT MAPS
Concept maps are collections of propositions, which can be seen as simplified natural language sentences, and are commonly displayed as a two-dimensional network of labeled nodes and links. Several groups have studied the use of concept maps for knowledge elicitation, both as an external representation of the topic being discussed, and as an organizational method for knowledge elicitation efforts. Ford *et al.* [11] used the term *knowledge models* to denote groups of interwoven concept maps and associated resources. Knowledge models have been developed using CmapTools [3] for several large institutional memory and expert knowledge preservation tasks, including launch vehicle systems integration [5], mesoscale weather forecasting [13], Thai fabric design, Mars exploration [2], and nuclear power air effluent analysis [6]. Knowledge acquisition using concept maps is also very efficient, with produc-

K-CAP'05, October 2-5, 2005, Banff, Canada.

tion averaging two useful propositions generated per session minute [13].

The success of these knowledge capture methodologies seems to depend on several factors. One is the freedom provided by the lack of formal syntactic restrictions. Users take advantage of this to explore, manipulate and gradually crystallize their thinking while constructing concept maps. In contrast, in order to produce machine-useable formalized ontology content, it is usually necessary to either learn, and cleave to, highly unnatural syntactic conventions, or else to work within a tightly controlled 'fill-in-the-blanks' style user interface. The COE interface was designed to maintain the compositional freedom of the CmapTools GUI while still providing the ability to output strictly correct OWL/RDF/XML syntax suitable for input to mechanical processors.

The graphical nature of the CmapTools GUI, in contrast to text-based GUIs, does not require users to maintain a connected mental model of the domain: it allows users to rely on spatial and geometric layout to segregate concepts into meaningful 'chunks' which can be rapidly selected, copied and pasted, and which can be arranged to form meaningful and memorable visual patterns. The COE GUI follows this practice. Even thought the visual arrangement has no formal meaning, it provides a key memory and compositional aid, and in particular allows for rapid and efficient communication between collaborating users. COE further utilizes the graphical nature of the interface by using distinctive graphical conventions (color, highlighting) to 'mark' significant aspects of the formal ontology content.

DEVELOPING FORMAL REPRESENTATIONS OF EXPERT KNOWLEDGE

The Semantic Web envisions a planet-wide network of content expressed in formal ontologies [1]. A growing set of emerging standards (RDF, OWL [17], SWRL) and an increasing repertoire of available software systems for realizing this vision are being deployed. The wide use of these and other Web conventions (URIs, XML) will soon 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 open network, and this is not necessary for successful integration. The SW vision might be called a *distributed voluntary syndication* model of knowledge composition: a distributed network of publicly readable, linked mini-ontologies integrated into a connected web by the re-use of concepts; and composed by subject-matter experts or small teams, rather than professional ontology engineers. Hence the 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 overlapping micro-languages.

Conventional ontology-authoring tools such as Protégé [20] were developed to help designers of large, often proprietary, ontologies intended for use by specialists in highly technical domains. Such specialized ontologies are analogous to large pieces of software, and have a similar development process. Although the SW will make use of such highly developed ontologies, the role of ontologies on the SW is already undergoing a fundamental change: rather than being 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 an OWL ontology uses a concept from a different OWL ontology, then an agent (human or software) using one ontology is also able to extract relevant content from the other, and use them together to draw conclusions (by applying any inference scheme which conforms to the OWL semantics [21]). This creates an opportunity for ontology authors to build their own ontologies, but also an obligation on them to re-use concepts from other ontologies in ways that are consistent with their expression in those original ontologies.

To achieve success of the voluntary syndication vision, new knowledge exploration and composition tools are needed. A Web-oriented knowledge capture tool must be more than simply a user interface to an ontology editor; it must in addition provide intuitive mechanisms for *locating* appropriate formalized concepts in previously published Web ontologies. This requires web searching, of course, but also an ability to rapidly identify appropriate and inappropriate sample concepts when SW ontologies are found. COE provides a useful visual interface (see Figure 1), but checking the intuitive content of concepts in large, or in many, ontologies is still a major cognitive burden. The *vicinity concepts* display supported by the clustering process, described below, is intended to facilitate such searches and decisions by arranging concepts into groups based on proximity and structural similarity found in their containing SW formalizations.

The preferred method for acquiring formalized knowledge in COE is to capture expert knowledge either through direct manipulation of formal knowledge structures (using templates, described below), or to start informally with concept maps, and later translate them into a formalized knowledge representation such as first-order logic or description logic. Currently, our choice of formal knowledge representation language is OWL, and through COE's knowledge elicitation process we generate "ontology maps". Thus, the ontology maps can be described as – indeed, are – both concept maps and graphical representations of groups of OWL description logic axioms. (Although COE is currently focused on OWL, the approach is not limited to any particular formal representation language; we are also exploring others.)

This methodology for generating ontology maps takes full advantage of the synchronous and asynchronous collaboration facilities of CmapTools, enabling group-based development of expert ontologies. Figure 1 shows an annotation on the *LandRegion* concept, which act as "sticky notes" allowing collaborating users to comment on the ontology during construction. Hyperlinks to other ontology maps are attached to the *OceanRegion* and *LandSurfaceLayer* concepts. A discussion thread (an email based discussion group) is shown on the *LandwaterSurfaceLayer* concept.

COE extends this notion of collaboration further: it provides collaborative capabilities for establishing the underlying semantics of the elicited knowledge domain in a formal manner. COE enables direct, intuitive access to the formally defined meanings of existing SW concepts, and the concept clustering view exposes different types of relationships across related concepts. These latter relationships may not be explicitly indicated, as the concept formulations may have occurred in isolation from each other and over a period of time.

Visualizing Ontologies with CmapTools

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).

The basic insight behind the design mirrors the description-logic foundations of the OWL language itself, which can be viewed as providing 'packaged' pieces of logical content representing commonly occurring quantifier and connective uses. COE in turn deals with 'chunks' of OWL content (see the 'navigation window' in figure 1) which can be given a natural English-like or graphical rendering. COE conventions avoid the 'mathematical logic' terminology that pervades the OWL documentation. For example,

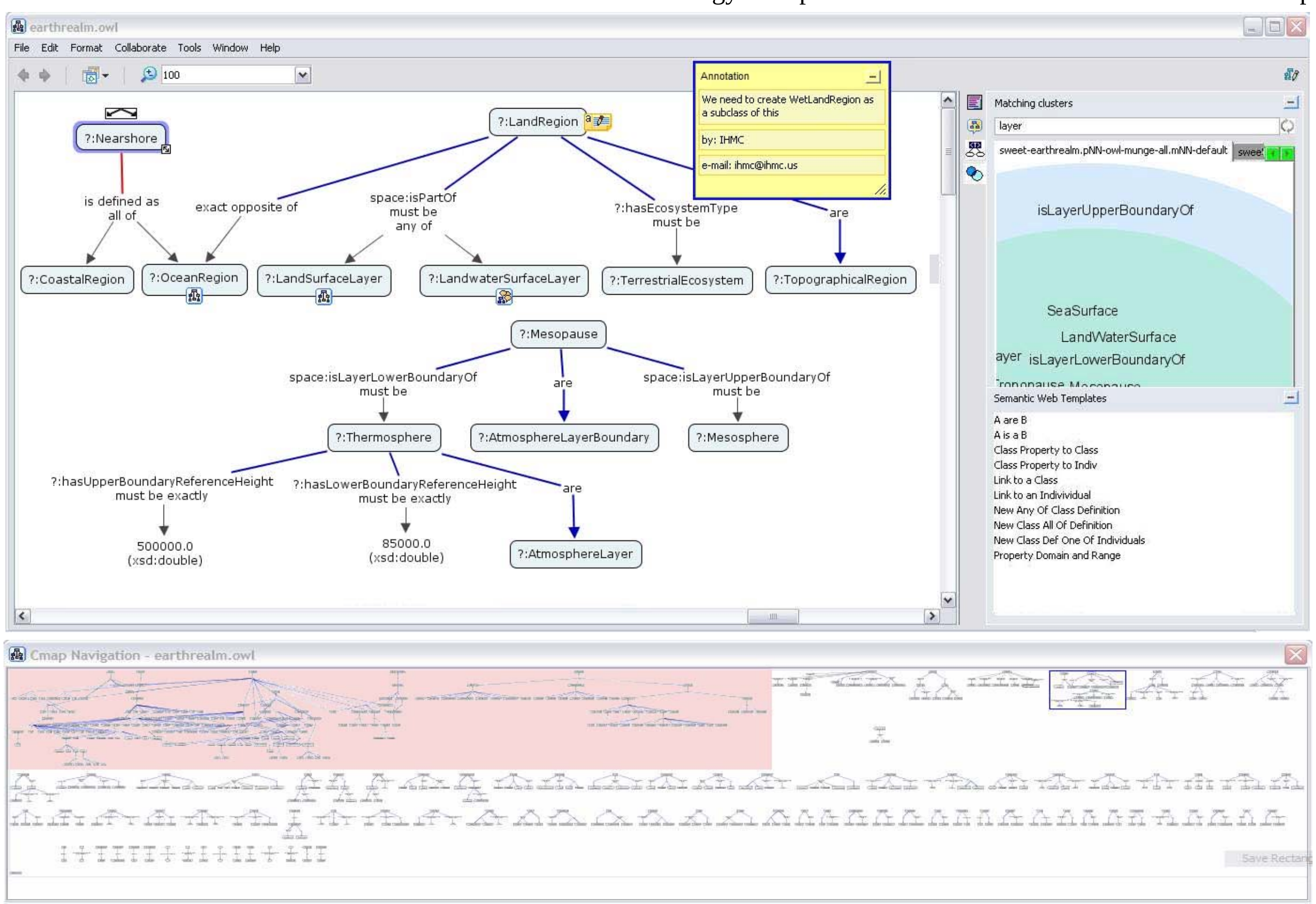

Figure 1. COE ontology map interface. Links to other ontology maps are shown as boxes under *OceanRegion* and *LandSurface-Layer*. A discussion thread is on the *LandwaterSurfaceLayer* concept, and an annotation on the *LandRegion* concept. A cluster of vicinity concepts, and list of available templates, are shown in windows on the right-hand side panel. The complete ontology layout is shown in the navigation window at the bottom. The class hierarchy is shaded; the area being edited is indicated by the box, which can be dragged to move the main view.

the fact that land regions are topographical regions would be represented in OWL-XML syntax by saying that *LandRegion* is an 'owl: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.

The COE treatment of OWL restrictions is more significant. Much OWL content is defined using restriction classes, each of which requires an elaborate piece of syntax in OWL/XML, involving arcane technical terminology such as owl:onProperty and owl:allValuesFrom. In COE, every restriction is written as a single link, constructed by a single mouse operation, labeled with the property name and additional predefined phrases such as *must be*, *can be*, *exactly one*, *at least 3* attached to the property name as part of the link label. Importantly for maintaining the concept-map 'feel' of the GUI, this 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*.

OWL constructions which do not transcribe into idiomatic English, such as the fact that a property is transitive or that two properties are inverses of each other, are rendered in COE using graphical or labeling conventions, and the same conventions are used to draw other, related distinctions. For example, a thick blue arc, used to render the '*are*' links, always indicates a necessary but not sufficient condition. Although such 'meaningful' use of graphical conventions is not recommended by conventional concept mapping methodology, we have found that importing OWL ontologies into COE often makes their essential nature vividly apparent to a quick glance. It is easy to distinguish ontologies which are largely taxonomic, since the subclass links have a distinctive blue color and are connected into a single subgraph by the layout algorithm. Ontologies which are less concerned with classes and more with relationships (called *properties* in OWL) have many dotted links, arising from COE's convention displaying domain and range information. Strict concept definitions (which are quite rare in OWL, but important when they do occur) are clearly marked by red links; 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. These advantages were serependitious, rather than by design, but they have proved to be important in practice. The value of this capability is that finding an appropriate concept to re-use will often require determining how that concept is being used in the ontology in which it was originally defined, and in other ontologies that use it. COE's simple visual layout greatly facilitates making this determination.

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, the interface maintains a consistent 'style': properties always label links; classes, literals and individuals label nodes, with distinctive style renderings for the various OWL categories of 'thing'[1]. 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' (non-planar) when considered as abstract graphs. The COE layout uses heuristics to determine when to 'clone' a node, i.e., to split off part of the graph and display it as a separate piece of the concept map, in order to produce a tidier display. (The heuristics are based on reducing multiple incoming links, but they take special account of subclass taxonomies.) It also uses heuristics to guess which parts of the OWL file are best grouped together into a single tree, and to produce a minimally tangled overall layout, tending to result in small, readable 'chunks' of content.

Third, the interface completely suppresses many nodes and arcs that would be visible in a naïve rendering of the RDF graph and represent 'obvious' information which would normally never be mentioned by human users but are required by formal reasoners. This greatly diminishes the perceptual burden on human readers of ontology content. On a randomly chosen set of 21 published OWL ontologies, ranging in size from 11 to 12,260 RDF triples, the COE interface layout heuristics reduced the number of nodes by an average of 48% and number of links by 67% compared to a 'raw' representation of the OWL/RDF content.

Templates for rapid knowledge construction

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 language-specific templates for commonly used OWL structures, shown in Figure 2. The language templates illustrate the graphical representation for each OWL construct. All text in brackets (such as "<class>" or "<property>") is meant to be replace by the user, either by selecting the link or node and directly entering text, or by a drag and drop operation that either merges the generic node with and existing node, or by dropping the template on an existing node. When this is done, the root element of the template is replaced by the text in the existing node. Templates can also be dropped directly onto the ontology map canvas, where they appear as extensions to the concept map containing the bracketed

[1] This applies to OWL-DL. COE will correctly import OWL-Full ontologies, but may render some classes as individuals.

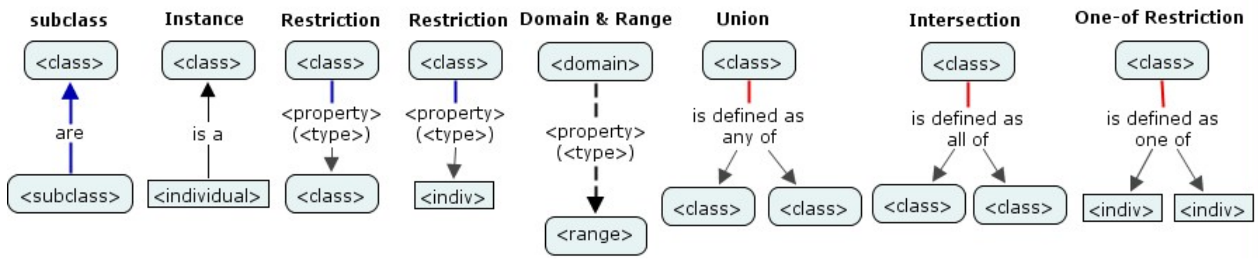

Figure 2. OWL-specific language templates.

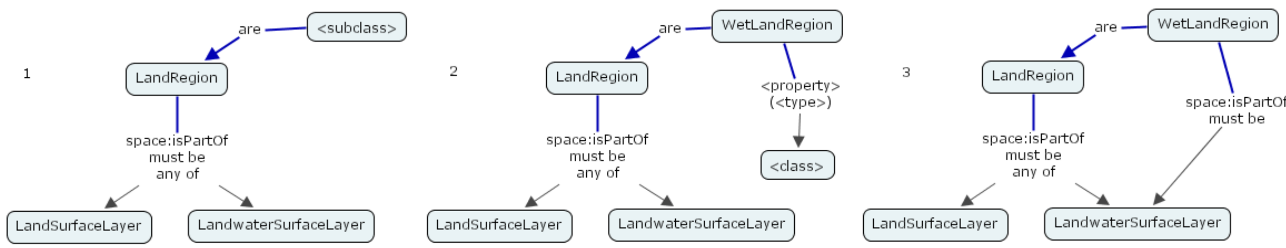

Figure 3. Using templates to construct the concept *WetLandRegion.*.

'reminder' phrases. The "subpropertyof" and "inverseof" properties of properties are constructed directly by dragging a link line between property labels.

For example, Figure 3 shows the incremental construction of a subclass of *LandRegion* called *WetLandRegion*. Step one shows the result of dragging the **A is B** template (the leftmost template in Figure 2) 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 shown on the right side of the figure. Users familiar with concept mapping can easily learn to do such operations very quickly.

CAPTURING THE "RIGHT" CONCEPT

A key issue in the development of any ontology is the determination to re-use an existing concept or to create a new one specifically for the ontology under development. COE provides several tools to assist the user in making this decision. First, it provides the ability to import and browse other ontologies using COE's graphical conventions. This depicts the context in which concepts under review are defined. Second, it provides a simple wildcard text-based search mechanism to search through all previously imported ontologies for concepts with specific names. Searching for the concept *region* will return a number of hits from different ontologies, matching concepts such as *LandRegion* and *WetlandRegion.* Finally, COE provides access to a structural clustering plug-in that exposes similarities between concepts that do not necessarily contain the search string. Access to this tool is provided through cluster-based *vicinity diagrams*, generated through a web-based service from the Expozé's tool suite. Expozé is an integrated suite of cluster-based cognitive assistance tools based around the core capability of grouping together concepts which have structural similarities in their formal descriptions.

Cluster-Based Vicinity Diagrams

Given a query focus term from COE's clustering interface, COE can display clusters of related concepts created by analyzing multiple SW ontologies (the window, visible in Figure 1, provides for zooming, scrolling and navigation between clusters). The clusters are produced off-line by Pragati's proprietary clustering software, Multi-Viewpoint Clustering Analysis (MVP-CA)[18], and are stored at the IHMC's COE server repository[2]. A cluster contains concepts from multiple ontologies that have similar formal descriptions. These concepts often do not have direct formal relationships defined across them in the ontology; these are implicit connections found by noting structural

[2] This web-service architecture was designed in part to provide open access to the clustering results without compromising the proprietary nature of the Expozé tool suite.

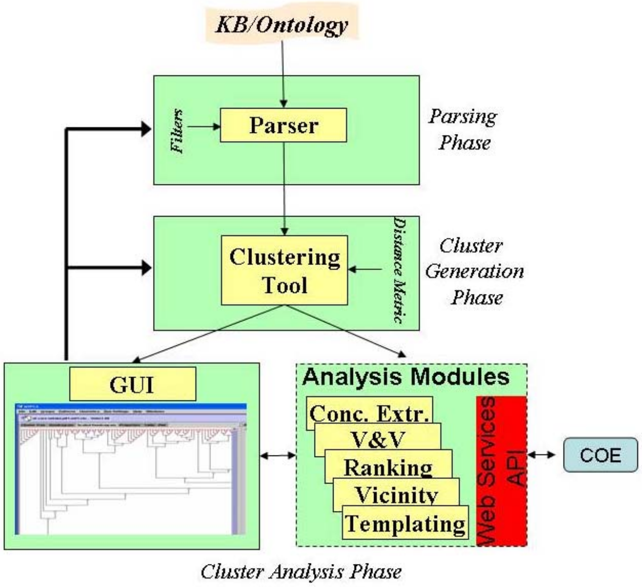

Figure 4. Clustering Architecture.

similarities between the formal axioms which contain them and, recursively, the similarity of terms surrounding the definitions of the concepts [17][18]. Exposing such *vicinity concepts* to the user provides a quick, intuitive "peek" into likely similarity of intended meanings of concepts in different ontologies and aids exploratory ontology building, and (even with the limited scope of the current implementation) can lead to "fortuitous" re-use opportunities.

The data-flow diagram for the Expozé tool suite is shown in Figure 4. In the Parsing Phase an interpreter parses a knowledge-base or an ontology and transforms it into an internal form required by the clustering tool. In the Cluster Generation Phase the focus is on generating clusters through an agglomerative clustering algorithm. Various adaptive syntactic and semantics-based measures are used at this stage to capture the similarity between axioms. They depend on the structural characteristics of the ontology and the class of problem-solving knowledge they embody. For example, classification systems have different dependencies across axioms from a monitoring system. By defining different types of heuristic measures for a given input, multiple clusterings may be generated.

The clustering algorithm starts with each axiom as a cluster. At each step of the algorithm, two clusters which are the most "similar" are merged together to form a new cluster. This pattern of mergings forms a hierarchy of clusters from the single-member rule clusters to a cluster containing all the rules. During the merging process, clustering may cause a concept term to localize in the newly-formed group. When this happens, the concept term is flagged as having become 'stable' with respect to this group.

When a COE user issues a query for a concept of interest, the web services-based query server returns a set of relevant *focus clusters* from the Cluster Analysis Phase. These clusters are based on the stable groups for the query term. 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.

Euler diagrams are generated from the stable concepts in the clusters. Concepts that stabilized in the cluster with the query concept form the outer region of the Euler diagram. Concepts that have stabilized earlier form the inner region of the Euler diagram. Intuitively, the concepts in the inner region are more "tightly bound", than the concepts in the outer cluster.

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 Figure 5 shows clusters for the query term *region* from both the Wine ontology and JPL's SWEET Earth Realms ontology. In this case, the clusters are quite different, which is unsurprising given the

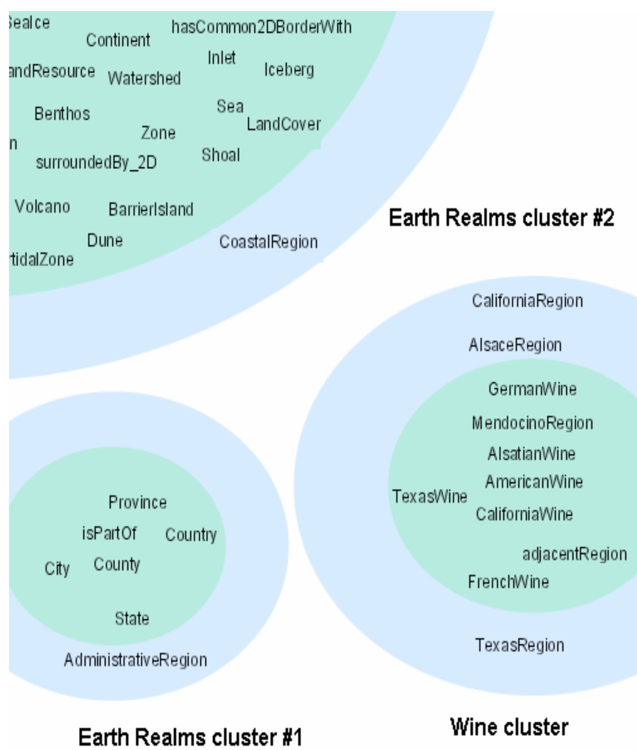

Figure 5. Vicinity concepts for *region*.

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 1 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. Users can click on the concept in the vicinity concepts view to open the ontology map in which the concept is defined, to investigate the concept further.

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 much more than 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 combinations thereof.

RELATED AND FUTURE WORK

The promise of the Semantic Web has sparked considerable interest in tools to aid the construction of ontologies ([4], [7], [10]). A prominent example is the Protégé knowledge acquisition tool built at Stanford University [12]. 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 ([9], [8]). 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.

Our system differs from these software tools chiefly by providing the concept map-based user interface, and in its emphasis on locating existing concepts and structurally informed Web navigation.

COE is capable of importing and rendering ontologies containing only up to approximately 12,000 RDF triples. This limitation is due to memory problems resulting from large-scale ontologies being imported into a single window which must hold all of the Java data structure necessary for representing and rendering the map in memory; but such large concept maps are not suitable for human inspection in any case. Future versions will partition the imported ontologies into separate concept maps each containing a subset of the original, organized into 'ontology folders'. Work in this direction has already begun.

Currently, templates represent OWL language idioms, but the same mechanism can provide more generally commonly occurring knowledge structures as named concept map fragments. Many domain-specific templates, which can be used as "short-cuts" for knowledge authoring, have been exposed by Expozé's clustering software. These general-purpose templates could then be used to enable the rapid specification of instances of particular concepts, processes, or events. We plan to integrate this capability in a future version of COE. We also intend to incorporate improved cluster filtering algorithms, to further reduce information overload in the number of vicinity concepts being exposed to the user. Additionally, we plan to implement a semi-automated cluster categorization system that will enable users to direct their cluster-based queries with greater precision.

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. Pragati's authors also acknowledge the support from the Office of Naval Research SBIR program for development of some of the Expozé tools. This research was supported in part by DARPA and the Department of Defense. The COE software is available for testing and can be downloaded from http://coe.ihmc.us.

REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] Coffey, J.W., Eskridge, T.C., Sanchez, D.P. (2004). A case study in knowledge elicitation for institutional memory preservation using concept maps. In *Proc. First Intl. Conf. on Concept Mapping, Volume 1*, pp. 151-157, U. of Navarra.

[7] 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.

[8] Ernst, N. A., Storey, M.-A. 2003. A Preliminary Analysis of Visualization Requirements in Knowledge Engineering Tools, Unpublished Technical Report, University of Victoria, Canada. <http://www.neilernst.net/docs/pubs/ernst-survey.pdf>

[9] ezOWL: Visual OWL Editor Plugin for Protégé. Web page <http://iweb.etri.re.kr/ezowl/>.

[10] 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.

[11] Ford, K.M. & Adams-Webber, J.R. 1991. Knowledge acquisition and constructivist epistemology. In R.R. Hoffman (Ed.), *The Psychology of Expertise: Cognitive Research and Empirical AI.* pp. 121-136. New York: Springer-Verlag.

[12] Gennari, J., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., Tu, S. W. 2002. The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. *International J. of Human-Computer Studies*, 58, 1, 89-123

[13] Hoffman, R. R., Coffey, J. W., Ford, K. M., and Carnot, M. J. (2001). Storm-LK: A human-centered knowledge model for weather forecasting. In *Proceedings of the 45th Annual Meeting of the Human Factors and Ergonomics Society*, Minneapolis, MN, USA.

[14] Hoffman, R. R., Shadbolt, N, Burton, A. M., & Klein, G. A. 1995. Eliciting knowledge from experts: A methodological analysis. Organizational Behavior and Human Decision Processes, 62(2) 129-158.

[15] Hoffman, R. R., & Woods, D. D. 2000. Studying cognitive systems in context. *Human Factors*, 42, 1-7.

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

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

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

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

[20] 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.

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