Aiding Knowledge Capture by Searching for Extensions of Knowledge Models

David B. Leake, Ana Maguitman, and Thomas Reichherzer
Computer Science Department
Indiana University, Lindley Hall 215
Bloomington, IN 47405, U.S.A.
{leake, anmaguit, treichhe}@cs.indiana.edu

Alberto J. Cañas, Marco Carvalho, Marco Arguedas, Sofia Brenes, and Tom Eskridge
The Institute for Human and Machine Cognition
40 S. Alcaniz St., Pensacola, FL 32501, U.S.A.
{acanas,mcarvalho,marguedas,sbrenes,teskridge}@ihmc.us

ABSTRACT
Electronic concept mapping tools empower experts to play an active role in the knowledge capture process, and provide a medium for building richly connected multimedia *knowledge models*—sets of linked concept maps and resources about a particular domain. Knowledge models are intended to be used as a means for sharing knowledge among humans, not as carefully-crafted knowledge bases upon which machines will be performing inference. However, users must still confront the questions of what to include in a concept map and which concept maps to include in a knowledge model. This paper describes ongoing research on methods to provide content-based support to users as they extend concept maps by adding concepts and propositions, and as they select topics for new maps. The goal is to provide scaffolding for experts as they build their own concept maps, link their maps to others', and decide how to extend their knowledge models. The paper presents three approaches which start from a concept map under construction and mine related information—both from prior concept maps, and from the web—to propose information to aid the user's knowledge capture and knowledge construction. The paper begins with a brief summary of the concept mapping process and the CmapTools concept mapping software. It then presents three types of implemented suggesters, to suggest concepts, propositions, concept maps, and new topics to aid experts using the CmapTools, and describes preliminary experiments to assess their performance. It closes with a discussion of next steps for testing and refining these methods.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Abstracting methods, Indexing methods*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods; I.2.6 [**Artificial Intelligence**]: Learning—*Knowledge Acquisition*

General Terms
Algorithms, Design, Documentation, Human Factors

Keywords
Case-based reasoning, concept mapping, context, knowledge acquisition tools, knowledge engineering and modeling methodologies, knowledge management environments, retrieval

1. INTRODUCTION
Successfully capturing and sharing expert knowledge depends on the ability to elucidate expert knowledge and to represent it in a form supporting examination by others. In light of the difficulties in capturing knowledge through traditional knowledge engineering processes, there is considerable interest in facilitating the knowledge capture process, in particular through methods that allow more direct and natural interactions between system and expert (e.g., [13]). The Institute for Human and Machine Cognition (IHMC) at the University of West Florida has led a multiyear initiative on tools to empower experts to directly construct knowledge models, through the process of *concept mapping* [24]. Concept maps represent concepts and relationships in a two-dimensional graphical form, with nodes representing concepts, connected by links representing propositions. Concept maps provide a simple, informal representation that has been used by people of many ages, from elementary school children to NASA scientists. As a knowledge capture method, concept mapping is appealing for its simplicity—experts can construct knowledge models of their domains directly, or actively participate in assisted knowledge elicitation processes—and for facilitating knowledge sharing, by producing representations that are easily understood by others.

To support capture and sharing of concept-map-based knowledge models, the IHMC has developed CmapTools, electronic concept mapping tools which support interactive concept map generation, linking of concepts with other concept maps and multimedia resources, and distributed concept map access. A recent effort addresses how to provide content-based support: to help experts to identify useful concepts, propositions, concept maps, and multimedia resources to include in a concept map under construction, and even which topics may be appropriate to include in a new concept map. The goal is to provide scaffolding for experts building their own concept maps, consulting and critiquing prior concept maps, linking their own concept maps to others', and deciding how to extend their knowledge models. This paper presents ongoing research on three approaches being investigated as part of that larger project.

Each of the three systems presented here takes as its starting point a concept map under construction, and proposes information to aid the user's knowledge capture and knowledge construction. Because the software does not enforce that every concept–link–concept form a coherent proposition, the quality of the constructed map taken as starting point depends on the map-builder. Although most users build well-structured maps, in the worst case, a user may build models in which sequences of concepts and nodes simply form English sentences with arbitrary divisions among nodes and links. The algorithms we present degenerate for such maps, but give good performance for a wide range of concept maps tested in practice.

One of our systems is targeted towards using the propositions reflected by the maps, summaries of the maps' structure, and contextual information to guide searches through libraries of prior concept maps. Two others mine the web to suggest terms or topics that are different from, but related to, those in a current concept map. In so doing, they help both to capture expert knowledge and to suggest new candidate areas in which knowledge models should be developed.

The paper begins with a brief summary of the concept mapping process and the CmapTools. It then presents three types of implemented suggesters, to suggest concepts, propositions, concept maps, and new topics to aid experts as they extend partial knowledge models, and describes preliminary experiments to assess their performance. It closes with a discussion of next steps for testing and refining these methods.

2. CONCEPT MAPS AND THE IHMC CMAPTOOLS

Concept maps, developed by Novak (1984), are tools for organizing, representing and sharing knowledge. Specifically, they are designed to tap into a person's cognitive structure and externalize concepts and propositions. A concept map is a two-dimensional graphical representation of a set of concepts connected by directed arcs encoding propositions in the form of simplified sentences, such that the interrelation-

ships among them are evident (see Figure 1). The vertical axis tends to express a hierarchical framework for the concepts. More general, inclusive concepts tend to be found at the highest levels, with progressively more specific, less inclusive concepts arranged below them. These maps emphasize the most general concepts by linking them to supporting ideas with propositions. A *knowledge model* of a domain is typically arranged as a hierarchical structure of concept maps with maps containing more general concepts toward the top and more detailed maps lower in the hierarchy.

CmapTools, developed by the IHMC, is a suite of publicly-available software tools for knowledge acquisition, construction, and sharing (http://cmap.coginst.uwf.edu/). The software, used in over 150 countries, facilitates construction and sharing of knowledge models based on concept maps, and also enables the use of concept maps to serve as the browsing interface to a domain of knowledge. The tools facilitate the linking of a concept to other concept maps, pictures, images, audio/video clips, text documents, Web pages, etc., enabling users to navigate to relevant resources by browsing through concept maps.

Concept maps capture "informal" knowledge models: Although nodes and links can be seen as encoding propositions, they are not represented in a formal logic, and have no associated formal semantics. However, they provide a concise representation of information for human use, providing a representation between that of traditional representations—which are hard to capture and require intervention by knowledge engineers—and text—which may obscure the structure of a domain. Concept maps are used by people of all ages, from elementary school children to NASA Scientists. More important, experts are able to construct knowledge models of their domain without the need for a knowledge engineer's intervention, or to actively participate in the knowledge elicitation if a knowledge engineer leads the process.

The CmapTools client provides a simple point-and-click interface to build new concept maps. Users can construct new concepts by double-clicking into a concept map window and entering the name of the concept into the appearing text field. They can then link two concepts by clicking on the arrow button of a selected concept and dragging the displayed arrow to a target concept or the background of the concept map for creating a link to a new concept. When the link has been constructed, users can specify the label of the link. Users can link concept maps and other multi-media resources to concepts using menu options or a drag-and-drop interface.

The CmapTools and the accompanying knowledge elicitation methodology have been used successfully for capturing, representing and sharing expertise in a variety of domains. Applications include a nuclear cardiology expert system [14]; a prototype system to provide performance support and just-in-time training to fleet Naval electronics technicians [7]; a knowledge preservation model on launch ve-
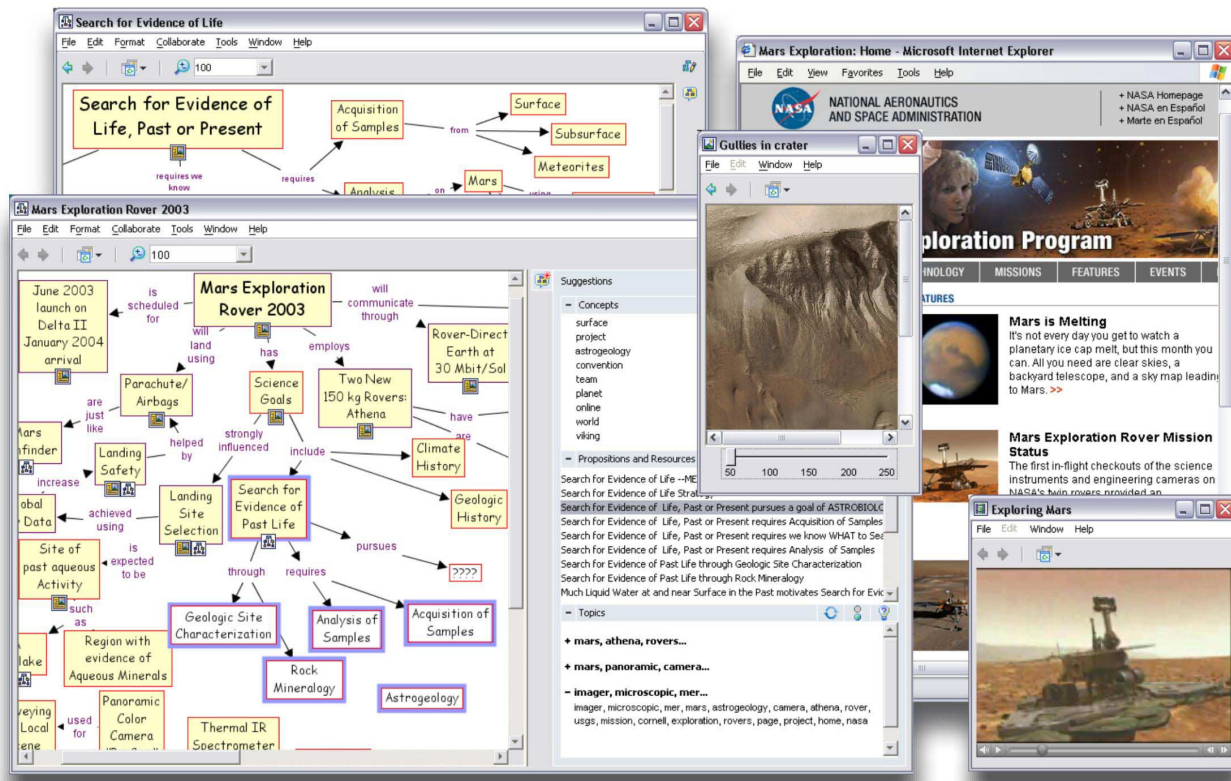
Figure 1: Portion of a Knowledge Model with Associated Suggestions.

hicle systems integration at NASA, a large-scale knowledge modeling effort to demonstrate the feasibility of eliciting and representing local meteorological knowledge undertaken at the Naval Training Meteorology and Oceanographic Facility at Pensacola Naval Air Station [17], and a large multimedia knowledge model on Mars (http://www.cmex.arc.nasa.gov), constructed entirely by a NASA scientist, without the participation of knowledge engineers.

3. AIDING MODEL EXTENSION

An initiative is now under way to develop methods to aid the user during concept map construction. These aids are designed in response to observations of snags which may arise during concept mapping. During concept mapping, users often stop and wonder what other concepts they should add to the concept map they are working on, frequently spending time looking for the right word to use in a concept or linking phrase; they search for other concept maps that may be relevant to the one they are constructing, and they search through the Web looking for additional material that could help them enhance their maps. The following sections describe three methods developed to address these issues.

3.1 Suggester for Concepts

The goal of the *concept suggester*, developed at the Institute for Human and Machine Cognition, is to facilitate concept map construction by proactively searching and suggesting

new concepts to the user [6]. In contrast to the *proposition and resource suggester*, discussed in the next section, the concept suggester proposes collections of terms, each of them representing a concept that is novel (i.e., not contained in the current map) but potentially relevant. This can (1) help the user to remember familiar concepts that might otherwise be forgotten, and (2) give the user the opportunity to further explore and understand new and potentially relevant concepts.

The search for relevant concepts is a two-step process. The system first searches the Web for documents related to the current map [9], to cache a collection of documents that can then be mined, in the next step, for the concepts to be suggested. The state of the map under construction is continuously monitored for significant changes that could trigger a new search for concepts to be added to the cache (Figure 2 shows the process). Significant changes in the map are defined as any modifications of the root, the hubs, or the authority nodes [8]. Such modifications may affect the relevance of cached documents to the current context, thus requiring the system to launch a new search.

A search process starts with a request for concept suggestions sent from the CmapTools client to a search server. All the processing occurs at the server side, avoiding any additional processing load on the client or client use of addi-
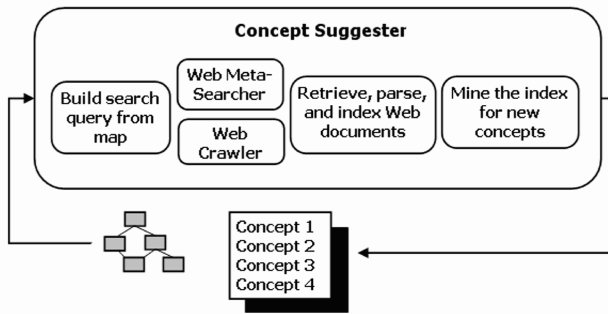
Figure 2: The process for searching new concepts.

tional network bandwidth. At the server side, the map is converted into a text query for a meta-search engine/crawler to retrieve additional documents that will be added to the database, and the database is searched for documents that are relevant to the context of the map. For performance reasons, this search process takes place in parallel, allowing for a timely response to the search request while still supporting database updates for future requests.

The subset of relevant documents retrieved from the database is then searched for potential concept suggestions. Our current approach to extracting relevant concepts starts by searching the documents for concepts that are already in the map. Each time a concept is found in a document, all the neighboring words are saved in a temporary table as potential suggestions. Neighboring words are defined as the non-stop words in the document within a fixed distance threshold (currently 3 words) of the concept term. After searching for all the map's concepts in all the documents we have a large collection of terms that are, at some level, neighbors of the map's concepts in the text. A frequency analysis is then applied to rank these terms and determine the subset for the suggester to display.

Preliminary experiments [6] with the concept suggester show promising results. Working with a trace of human concept map construction, the experiments measured the number of suggestions that would have been used at some point by 7 subjects building maps about electricity. This was done by analyzing the history of the map and comparing the concepts introduced in certain stages of the map construction with the concepts suggested in previous stages. The results have shown conservative matching rates around 49%, for a suggester presenting 15 suggestions at a time.

3.2 Suggester for Propositions, Concept Maps, and Multimedia Resources

Previously-built knowledge models, shared from other users, may help suggest propositions to consider and concept maps to consult while constructing a new concept map. To provide suggestions from prior knowledge models, the proposition suggester, developed at Indiana University, applies techniques inspired by case-based reasoning [19, 20].

3.2.1 Concept Maps as Cases
The concept maps of various users are considered as case-bases of their concept-mapping activity, with each concept map considered to be a separate case. When a new user wants to "extend" a concept—add a new connected concept—the system views prior concept maps including the original concept as examples of how that concept was extended in the past.

In the current implementation, case libraries are compiled periodically from concept maps on the CmapTools servers and clients, generating case representations from raw concept maps and indexing new concept map cases. Each case stores information about a map's content, its structure, and links to other concept maps and resources that are attached to its nodes. This information is necessary to generate suggestions in the form of propositions, concepts, and relevant multi-media resources that may be helpful in extending and annotating new concept maps.

3.2.2 Building an Index from Concept Maps
Central to any case-based approach are techniques for indexing—characterizing when cases are likely to be useful in the future. Our system guides retrieval based on a category index computed from the concept map library. The index organizes concept maps into a hierarchical structure of categories, each containing a set of concept maps involving correlated concepts. More tightly coupled clusters of concept maps appear towards the bottom of the hierarchical structure, and more loosely coupled clusters towards the top. For each category, the index maintains references to the original concept maps and a cluster representative, generated from concept maps in the category to serve as a prototype. The cluster representative is used to determine if a new concept map is related to the maps in a category.

The algorithm for compiling concept maps into a hierarchical, tree-like structure uses an agglomerative approach, starting from a set of initial clusters referring to a single concept map each. It then repeatedly merges the clusters whose cluster representatives are closest to each other (by the criterion described below), making each merged cluster the parent of the clusters that were merged. This is continued until all clusters have been merged or the similarities measured between the cluster representatives fall below a pre-set threshold, suggesting that the concept maps from different clusters have little in common and should remain distinct.

During the merging process, the algorithm prunes the tree so that only groups of maps above a minimum size form the nodes of the tree. This affects the depth and the width of the category index's hierarchical structure, ultimately reducing the storage requirements, as well as reducing the number of comparisons needed during retrieval to find the category of a new concept map. The end result is a tree with nodes as categories. The leaf nodes in the tree form a partition of the concept map library.

Determining concept map similarity using a vector-space model: Concept map similarity is computed from a vector representation of the concept maps. This representation is similar to the popular term-frequency vector with inverse-document frequency adjustment (TF-IDF), but takes advantage of the structure of concept maps to adjust term weights, based on structural and topological clues to concept importance. In concept maps, more general concepts are typically found at the top of the map while more inclusive concepts are located at the bottom. The system adjusts weights accordingly, assigning higher weights to keywords from top concepts and lower weights to keywords from concepts located at the bottom of a map. In addition, the system considers the number of outgoing and incoming links to a concept node, strengthening the weightings of keywords in nodes for concepts with many connections to other concepts in the map. In our current approach, the link labels in concept maps are ignored, but using their information is a topic for future research.

The similarity metric is defined as follows. For each concept map C_j of a library of maps $\mathcal{L}$, let $freq_{ijk}$ be the raw frequency of keyword i in the label of concept k of concept map C_j. Assume concept k has n outgoing links and m incoming links and is d steps distant from the top node of the map. The system computes the weight of keyword i of concept k in map C_j as

$$w_{ijk} = freq_{ijk} \cdot (\alpha n + \beta m) \cdot (1/(d+1))^{1/\delta}.$$

Propositions in the map are encoded as concept-link-concept triples, where the link is outgoing from the first concept, and incoming to the second. The parameters α and β are used to weight keywords differently according to their concept's position in the triple. The parameter δ adjusts the decay of weighting as keywords appear lower in a concept map. The decay is slower for $\delta > 1$ and faster when $0 < \delta \le 1$.

The total weight of keyword i in C_j is the sum of all weights w_{ijk} for all concepts k in map C_j. This weight is normalized using the largest keyword weight in concept map C_j and adjusted using the inverse document frequency (a concept map is considered to be a single document) for keyword i.

Similarity between the vectors for two concept maps is calculated by the cosine measure. This has a desirable property for centroid-based clustering: When using centroids as the cluster representatives, the inner product of a document with the centroid is the average similarity between the document and all documents in the cluster [18]. The clustering algorithm determines the similarity matrix between the cluster centroids, selects the two clusters with most similar representatives as the clusters to merge, and computes the cluster representative of the new cluster as the average weighted sum of the two most similar centroids from the similarity matrix. The clustering process produces a hierarchical categorization indexing all concept maps in the concept map library.

We have tested this approach on sample sets of concept maps. We envision that, for an application of this approach in CmapTools, indices and case libraries would be compiled periodically by the individual concept map servers and then uploaded to a designated index server. This server would be responsible for merging the different indices into a combined index and responding to queries from clients for relevant suggestions. The combined index could include several disjunct category hierarchies if the individual hierarchies are dissimilar.

3.2.3 *Generating and rating suggestions*
Users can actively initiate search for new concepts or multimedia resources by selecting the concepts for which extensions are sought, or can rely on the system to monitor concepts being added to the concept map and proactively suggest propositions or annotations.

Retrieving relevant suggestions: Whether in user-driven or proactive mode, the suggester converts the map in progress to a term vector as described above and extracts keywords from the concepts selected by the user or the suggester. The keywords of the selected concepts and the vector representation form a query, processed locally by the client and remotely by a designated index server. While the keywords are used to look up specific suggestions in a case, the term vector serves as a context in the search for suggestions. The vector is used to perform a binary search for the best-fitting category starting from the top of the relevant hierarchies in the combined index and going towards the bottom.

Adjusting search specificity: By adjusting a slider, users can control how far the retrieval algorithm descends in the hierarchy tree to search for related concept maps. The further it descends, the fewer maps it finds, but those found are more closely related to the map in progress. This allows users to control how broad or narrow a search should be performed. Once a set of related maps has been identified, they are examined to find suggestions for concepts to extend the current map and to suggest resources linked to relevant nodes in the retrieved map.

Ranking retrieved cases: For ranking suggestions extracted from a concept map library, we have implemented a keyword correlation metric, based on the distances between concepts within a concept map. The inter-keyword distance is the number of links between the concepts from which the keywords are extracted, with distance zero for keywords appearing in the same concept and infinite distance for keywords from concepts that are not connected. To increase efficiency for ranking, the algorithm that compiles the category index also pre-computes and stores the distance-based keyword correlations for the lower categories in the tree-structured hierarchy (in order to reduce disk and memory space requirements for storing the keyword correlations, we pre-compute the correlation matrix only for the smaller categories that appear in the lower level of the hierarchy). Formally, for each pair of keywords (i, j) of a concept map C

in a category χ we compute $D_C(i,j)$ as 1 plus the minimum number of links between concepts containing i and j. The distance-based correlation between i and j is determined by

$$M_\chi(i,j) = \frac{2}{|\Theta_i| + |\Theta_j|} \times \sum_{C \in (\Theta_i \cap \Theta_j)} \frac{1}{D_C(i,j)}$$

where Θ_i and Θ_j are the set of maps in χ containing keywords i and j respectively. The final values of the computed correlations range between 0 and 1 with 0 indicating no correlation and 1 indicating maximum correlation.

When the suggestions have been generated, we use pairs of keywords (i,j) where i is extracted from the potential suggestions and j is extracted from the selected concepts of the map in progress to compute a rank. The rank of a suggestion is determined by the subset of pairs resulting in the highest correlation value. Among all the potential suggestions we display only the n most relevant suggestions sorted by their rank. The value of n can be changed by the user.

3.2.4 Initial evaluation of indexing performance

The algorithms for generating category indices and compiling case libraries have been implemented, and we are evaluating the quality of retrieval of suggestions. Our initial experiment focused on the performance of the indexing and retrieval algorithms. Several tests were conducted on two data sets. The first set contained three knowledge models on overlapping topics, respectively comprising 93 concept maps from the Mars 2001 library, 9 concept maps on the NASA Centaur Rocket System, and 14 maps on a meteorology project. The former two knowledge models were created by experts from NASA. The second data set contained two knowledge models, but on dissimilar topics, with 14 maps on AI topics and 17 concept maps on water and glaciers. The experiment was designed to determine whether (1) similar projects would be merged into a single hierarchy of categories while dissimilar projects be kept separate and (2) indexed maps would be located in the assigned category. Table 1 summarizes the results from the experiment. For both data sets, we tested different parameter settings for the initial size of a leaf node in the clustering algorithm, resulting in different clustering structures. The second column of the table shows the resulting number of leaf nodes (i.e., partitions) of the concept map library. The third column shows the percentage of maps that couldn't be located after the maps had been indexed and assigned to categories. Regardless of the different parameter settings for clustering, the algorithm computed a single category hierarchy for the first data set, and two separate hierarchies for the second. Thus the algorithm correctly determined that the models of the first set share common concepts, while the models of the second set have nothing or little in common.

For the maps of the first set that could not be located in the index, resulting in an erroneous classification, we determined that the maps still correlate with the maps in the incorrectly selected category. The correlation value ranged

	Tests	# Partitions	Error Rate
First Data Set	1	12	2.57%
	2	6	0.86%
	3	5	1.72%
Second Data Set	1	9	0%
	2	4	0%

Table 1: Results from an automatic categorization.

from 14 to 38% using the cosine measure and comparing a map with its best matching map in the category selected by the retrieval process (for the correct category, the correlation value is 100%). We also determined that the closest shared parent category in the hierarchy between incorrectly selected category and the correct category is–except for one case–at most one distant in the tree, meaning that the wrong category was selected in the final step of the lookup process. This is encouraging for the performance of the retrieval system, because this means that (in our tests) the original category would always be found if the user broadened the search to include a single additional level.

3.3 Suggester for Relevant Topics

Suggestions from previous concept maps are useful for elaborating new maps, but cannot help to extend the knowledge model beyond information that has already been captured in the concept map libraries. Another suggester, EXTENDER (EXtensive Topic Extender from New Data Exploring Relationships), developed at Indiana University, identifies and suggests novel topics that the expert may wish to include in the knowledge model.

The World Wide Web provides a rich source of information that may suggest useful topics; the key issue for the user is finding the right information. Commonly-available access methods fall into two main classes. The first class relies on web search engines (e.g., Google), and requires the user to explicitly specify information needs in the form of a query. A second approach uses directory services (such as the dmoz open directory project), requiring the user to manually traverse a hierarchy of topics. Both approaches have disadvantages. Methods based on querying web search engines usually result in large amounts of unclassified information, with filtering and classification left to the user. Methods that rely on the user navigating directories of topics access only a small subset of the relevant information, due to the slow and painstaking work currently required for the manual classifications on which web directories depend. Finally, a major disadvantage of both approaches is that they require users to take the initiative to search, and to know what they seek. Our goal in topic suggestion is to automatically identify topics of potential interest, that the user may not have thought of— thus proactively providing support by anticipating the user's needs.

Our basic approach is to use information automatically extracted from the current map to guide mining the web for

relevant information. EXTENDER addresses the challenges of proactively and unobtrusively providing the knowledge modeler with a set of novel but relevant topics. As opposed to manually constructed topics selected in light of a particular theme, the topics generated by EXTENDER result from automatic processes involving web mining and clustering. Hence, we refer to them as *artificial topics*.

3.3.1 The Nature of EXTENDER's Artificial Topics

Topics are commonly defined as pieces of data that have been grouped together as a result of having a common theme. Documents and terms have dual roles in describing topics, and while it is normal to describe topics as collections of related documents here we will treat topics as cohesive sets of terms. Topics are first presented to the user as suggestions consisting of a small collection of terms. These suggestions enclose, for each topic, a ranked list of constituent web pages together with their descriptions and URLs.

EXTENDER's artificial topics are produced by an iterative process which takes a knowledge model as input and mines the web to find topics related to the initial model. At each step, the information found is clustered and incrementally used to guide further search, resulting in a sequence of generations of artificial topics. Our current implementation of EXTENDER utilizes a document centered clustering technique to construct new generations of topics (i.e., sets of similar web pages are grouped to form coherent topics), but each topic is also specified by a set of weighted terms. The terms that characterize an artificial topic are a combination of new terms originating from recent web search results, and preexisting terms carried along in the sequence of generations of topics, starting from the initial model. EXTENDER attempts to diversify terms during initial generations and to focus at the end, after processing several sequences of generations of topics. The exploration and exploitation of new-found topics is controlled by using a diversity factor to adjust the weights of novel terms, and a focus factor to regulate the preservation of existing terms inherited from earlier generations of topics.

3.3.2 Issues and Methods

EXTENDER's algorithm exploits the structure and content of an "in-progress" concept map for automatic query formation. As pointed out previously, concept maps posses the desirable property of having a rich structure. Automatic techniques that analyze the map structure [8, 21] are used to rate concepts from the starting set of maps. These concepts are used to incrementally search for new topics with the purpose of extending the knowledge model under construction. Irrelevant information is filtered by contrasting the search results with the search context. The search context is initially defined using the knowledge model under construction, and is then progressively updated as the focus shifts through a connected series of topics. Cohesive topics are generated by clustering the results returned by the web mining process. The diversity/focus factor is adjusted to favor exploration

during initial stages and exploitation at the end of the process.

For concept map extension, EXTENDER addresses the problems associated to directly querying search engines or manually navigating directory services. EXTENDER operates proactively and unobtrusively, and profits from the search context to filter irrelevant data and return mostly related information. Although it can sometimes benefit from information available in directory services, it does not depend on the pre-existence of related topics in the directories. It can dynamically generate, in real time, a specialized taxonomy of artificial topics. In the following we outline the algorithm used to generate this taxonomy.

3.3.3 EXTENDER Algorithm

EXTENDER takes as input a concept map or small collection of concept maps about a certain domain, and generates artificial topics as follows:

Step 1: Apply topological analysis to transform concept maps into a term vector and generate initial corpus.

Step 2: Combine weighted terms to produce the first generation of artificial topics.

Step 3: Repeat steps 4–10 until the final generation of topics.

Step 4: Use diversity/focus factor to define a similarity threshold (higher diversity corresponds to a lower similarity threshold).

Step 5: Use artificial topics to define a context for search.

Step 6: Use artificial topics to generate queries for a web search engine.

Step 7: Use the search context and similarity threshold to filter irrelevant results.

Step 8: Identify the most relevant novel keywords and update the corpus.

Step 9: Use the diversity/focus factor to integrate returned results with prior information and complete the term-web page matrix (the integration multiplies the weight of novel keywords (prior keywords) by the diversity factor (focus factor).

Step 10: Apply term clustering to the term-web page matrix to obtain new generation of artificial topics.

Step 11: Return the final generation of artificial topics.

Because one of the goals of this system is to generate topics in a reasonable amount of time, only the information that is readily accessible as the result of a web search is used when generating terms to describe topics. Rather than processing the complete content of web pages, EXTENDER processes the "snippet" returned by Google, a text excerpt from the page summarizing the context in which the search terms occur. The system generates a set of weighted keywords from the snippet, URL domain, title, and, when available, the open directory project category name of the search results.

3.3.4 Evaluation

The usefulness of EXTENDER's suggestions is difficult to assess in a controlled way, because the cohesiveness and usefulness of topic suggestions is highly subjective. In order to perform an objective test of whether the system was able to generate artificial topics with content similar to hand crafted ones, we performed an automatic evaluation of the system using the Mars 2001 knowledge model. The top-level concept map from the knowledge model (chosen as the most basic) was used by EXTENDER as the starting point (representing the map under construction) and EXTENDER was used to produce a collection of artificial topics, without access to any of the other maps in the knowledge model. We then determined the overlap between EXTENDER's topics and those that an expert had actually chosen to include in the knowledge model. This was measured by two rate functions. Given a set T of terms in a topic and a set C of terms in a target expert concept map, $Rate_1$ measures the proportion of terms in an artificial topic that are actually part of a target concept map, by:

$$Rate_1(T,C) = \frac{|T \cap C|}{|T|}.$$

$Rate_2$ represents the proportion of novel terms (terms not in the starting map) in an artificial topic that are also part of a target concept map. Consider the set O, containing the terms of the originating concept map. If $T \subset O$, then no novel terms were generated and $Rate_2$ returns 0. Otherwise,

$$Rate_2(T,C,O) = \frac{|(T \cap C) - O|}{|T - O|}.$$

While $Rate_1$ shows the degree to which an artificial topic resembles a given map, $Rate_2$ shows the proportion of *novel* terms in an artificial topic that are also part of a target map, but are not in the starting model.

In our experiment, the originating concept map was the top map in the hierarchy in the Mars 2001 knowledge model. With three iterations, EXTENDER produced 19 artificial topics, each containing 20 terms. In table 2, we report the values of $Rate_1$ and $Rate_2$ for 10 randomly-selected concept maps of the Mars 2001 knowledge model. In the table, we only present six of the artificial topics, each characterized by its 5 terms ranked by the system as most relevant.

The observation that different topics are similar to different maps in the Mars 2001 knowledge model is encouraging: it suggests that, while EXTENDER preserved the general theme of the originating concept map, it truly created diverse topics. If a topic and some expert map (other than the originating map) are highly similar, then we have a good reason to believe that the topic is a valuable suggestion, because the information that is provided by the topic is new but highly relevant to a topic the expert chose to include. In many cases no concept map in the pre-existing expert knowledge model used for this evaluation is similar to some of the generated topics. This does not necessarily mean these artificial topics are not valuable—they may suggest useful extensions of the Mars 2001 knowledge model—but in the context of our evaluation, the usefulness of these topics remains indeterminate.

3.4 Integrated Suggestion Presentation

To integrate suggestions from the three suggesters and present them in a convenient form, we have designed a panel for CmapTools that collects suggestions from each suggester. The panel is attached to the side of a concept map and becomes visible only when the user decides to open it; otherwise, an unobtrusive signal lets users know when suggestions have arrived if the panel is closed. Figure 1 depicts a partial knowledge model and the side panel with associated suggestions. Controls allow users to enable or disable particular suggesters, to request an update on the presented suggestions, and to request additional suggestions of a given type.

4. DISCUSSION AND FUTURE WORK

Initial tests and evaluations of all three suggesters are encouraging. We intend to both make them available for trials by a larger user base, and to perform controlled human-subjects experiments to assess their usefulness. We also see refinements to the methods themselves. Currently, our three suggesters rely on exact term matching; we are now investigating the use of Wordnet or similar electronic lexical databases to enable the suggesters discover a wider range of relevant concepts, propositions and topics by using information on synonyms. In addition, this will allow EXTENDER to better filter non-novel information, and to make a more informed ranking of suggestions.

Parsing concept and proposition labels and assembling them, to generate suggestions in English sentences, has proven to be a non-trivial issue. Likewise, additional use of NLP techniques could improve the extraction of useful query information, and could also help in using information from link labels.

To directly aid concept map extension, we envision a drag-and-drop interface to allow users to integrate suggestions directly from a concept map's suggestion panel into the map, with the system highlighting possible locations for the new concept.

5. RELATED WORK

Several suggester systems have been developed to offer users context-sensitive assistance during task performance. For example, the Remembrance Agent [26] monitors what the user is reading or editing to retrieve related user documents; Watson [5] examines the user's current document to retrieve relevant online material. Other tools monitor user browsing activity to identify relevant web pages (e.g., [1, 2, 22, 23]). These systems are similar to ours in attempting to provide users with context-relevant information, but differ in not attempting to extract concepts, propositions, or topics as our suggesters do.

Target Concept Maps and most relevant terms of six Artificial Topics	Topic 1: technology penetration revolutionary systems protection		Topic 2: air dry composition nitrogen oxygen		Topic 3: entry atmospheric system esa level		Topic 4: landing sites specific nasa mars		Topic 5: mars lowell history percival planet		Topic 6: fluvial history erosion activity glaciers	
Originating Concept Map	0.1	0	0	0	**0.25**	0	**0.4**	0	0.15	0	**0.3**	0
Climate History	0	0	0.15	0.15	0.15	0.2	0.05	0.08	0.1	0.05	**0.7**	**0.72**
Deep Access	**0.75**	**0.72**	0.05	0.05	0.2	0.2	0.15	0.08	0.1	0.05	0.1	0
Earth's Atmosphere	0	0	**0.5**	**0.5**	0	0	0	0	0	0	0.1	0.07
Geologic History	0	0	0	0	0	0	0.05	0.08	0.1	0.05	**0.3**	0.28
History of Water	0	0	0	0	0.1	0.13	0	0	0.05	0	**0.5**	**0.5**
Landers	0.2	0.16	0.1	0.1	**0.7**	**0.6**	**0.55**	**0.41**	0.05	0	0.15	0.14
Myth and Science Fiction	0.05	0	0	0	0.05	0	0.1	0	**0.7**	**0.7**	0.05	0
Pathfinder	0.05	0	0.05	0.05	**0.25**	0.2	0.15	0	0.1	0	0.05	0.07
Rovers	**0.35**	**0.27**	0	0	**0.55**	**0.46**	**0.5**	**0.33**	0.1	0	0.2	0.14

Table 2: Relationships of Artificial Topics to Expert Concept Maps.

Topic-driven web exploration programs help maintain web portals on certain topics, by seeking and indexing information on the web (e.g., [10, 3]). The approach of these systems differs in that EXTENDER does not involve pre-crawling and indexing, and is not aimed at generating extensive topic information, but instead attempts to dynamically generate samples of topics that will serve as basic hints to the knowledge modeler.

Systems to facilitate topic exploration (e.g., [16, 28]) are similar to EXTENDER in the sense that they cluster search results into topically-coherent groups. However, these tools are based on a browsing interface with explicit user intervention. Another difference is that the ultimate goal of these systems is to generate more specific topics, while the goal of EXTENDER is to create diversity.

A number of systems provide interfaces for editing and browsing knowledge bases and ontologies (e.g., [27, 25, 12]). These use standardized languages and require the intervention of programmers or knowledge engineers. Systems in a similar spirit to our work, aiming to enable flexible knowledge acquisition without the mediation of knowledge engineers includes EXPECT [15, 4] and SHAKEN [11], which is based on a graphical interface and a library of reusable components. In contrast to the informal nature of concept map representations, however, SHAKEN aims to capture first-order logic descriptions of concepts.

6. CONCLUSION

This paper has described an approach to supporting experts as they build their own knowledge models of a domain by concept mapping. It builds on electronic concept mapping tools, which provide the functionality to generate concept maps, link them to other maps, and perform distributed access. In order to aid experts at extending their knowledge models, we have developed three suggesters to aid in selecting the content to include in a map. These suggesters mine the web, or prior concept maps, to suggest concepts, propositions, resources, and even new topics to include in the knowledge model. This provides scaffolding for the expert's knowledge capture process, as well as for the expert constructing new knowledge during the concept map construction process. By mining the web, they provide a tremendous resource for the knowledge modeling process. All three suggesters are now robust prototypes, and results are encouraging for their performance. We see these as steps towards a library of suggesters to aid this knowledge capture process.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symposium on Information Gathering*, pages 6–12. AAAI Press. Menlo Park, CA, 1995.

[2] T. Bauer and D. Leake. Using document access sequences to recommend customized information. *IEEE Intelligent Systems*, 17(6):27–33, 2002.

[3] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *International Conference on Research and Development in Information Retrieval (SIGIR '98)*, pages 104–111. ACM Press. New York, 1998.

[4] J. Blythe, J. Kim, S. Ramachandran, and Y. Gil. An integrated environment for knowledge acquisition. In *International Conference on Intelligent User Interfaces (IUI '01)*, pages 13–20. AAAI Press. Menlo Park, CA, 2001.

[5] J. Budzik, K. Hammond, and L. Birnbaum. Information access in context. *Knowledge based systems*, 14(1–2):37–53, 2001.

[6] A. Cañas, M. Carvalho, and M. Arguedas. Mining the web to suggest concepts during concept mapping:

Preliminary results. In *XIII Simpósio Brasileiro de Informática na Educação*, SBIE UNISINOS, 2002.

[7] A. Cañas, J. Coffey, T. Reichherzer, G. Hill, N. Suri, R. Carff, T. Mitrovich, and D. Eberle. A performance support system with embedded training for electronics technicians. In *Proceedings of the Eleventh Annual Florida Artificial Intelligence Research Symposium*, pages 79–83. AAAI Press, 1998.

[8] A. Cañas, D. Leake, and A. Maguitman. Combining concept mapping with CBR: Experience-based support for knowledge modeling. In *Proceedings of the Fourteenth International Florida Artificial Intelligence Research Society Conference*, pages 286–290. AAAI Press, 2001.

[9] M. Carvalho, R. Hewett, and A. Cañas. Enhancing web searches from concept map-based knowledge models. In *Proceedings of the SCI Conference*, Orlando, Florida, 2001.

[10] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource list compilation by analyzing hyperlink structure and associated text. *Computer Networks*, 30(1–7):65–74, 1998.

[11] P. Clark, J. Thompson, K. Barker, B. Porter, V. Chaudhri, A. Rodriguez, J. Thomere, S. Mishra, Y. Gil, P. Hayes, and T. Reichherzer. Knowledge entry as the graphical assembly of components. In *International Conference on Knowledge capture (K-Cap '01)*, pages 22–29. ACM Press. New York, 2001.

[12] A. Farquhar, R. Fikes, and J. Rice. The ontolingua server: A tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46(6):707–727, 1997.

[13] K. Forbus, J. Usher, and V. Chapman. Sketching for military course of action diagrams. In *Proceedings of the 2003 international conference on Intelligent user interfaces*, pages 61–68. ACM Press, New York, 2003.

[14] K. M. Ford, J. W. Coffey, A. J. Cañas, E. J. Andrews, and C. W. Turner. Diagnosis and explanation by a nuclear cardiology expert system. *International Journal of Expert Systems*, 9:499–506, 1996.

[15] Y. Gil. Knowledge refinement in a reflective architecture. In *Twelfth National Conference on Artificial Intelligence (AAAI '94)*, pages 520–526. AAAI Press. Menlo Park, CA, 1994.

[16] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 76–84. ACM Press, New York, 1996.

[17] R. R. Hoffman, J. W. Coffey, K. M. Ford, and M.-J. Carnot. 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, 2001.

[18] G. Karypis and E.-H. Han. Concept indexing: A fast dimensionality reduction algorithm with applications to document retrieval and categorization. Computer science department TR-00-0016, University of Minnesota, 2000.

[19] J. Kolodner. *Case-Based Reasoning*. Morgan Kaufmann, San Mateo, CA, 1993.

[20] D. Leake, editor. *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press/MIT Press, Menlo Park, CA, 1996.

[21] D. Leake, A. Maguitman, and A. Cañas. Assessing conceptual similarity to support concept mapping. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 168–172. AAAI Press, Menlo Park, 2002.

[22] H. Lieberman. Letizia: An agent that assists web browsing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 924–929, San Mateo, 1995. Morgan Kaufmann.

[23] S. E. Middleton, D. C. D. Roure, and N. R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. In *International Conference on Knowledge Capture (K-Cap '01)*, pages 100–107. ACM Press. New York, 2001.

[24] J. Novak and D. Gowin. *Learning How to Learn*. Cambridge University Press, New York, 1984.

[25] S. M. Paley, J. D. Lowrance, and P. D. Karp. A generic knowledge-base browser and editor. In *Ninth Conference on Innovative Applications of Artificial Intelligence (IAAI '97)*, pages 1045–1051. AAAI Press. Menlo Park, CA, 1997.

[26] B. Rhodes and T. Starner. The remembrance agent: A continuously running automated information retrieval system. In *First International Conference on The Practical Application of Intelligent Agents and Multi Agent Technology (PAAM '96)*, pages 487–495. Practical Application Company. Blackpool, UK, 1996.

[27] L. G. Terveen and D. A. Wroblewski. A collaborative interface for editing large knowledge bases. In *Eight National Conference on Artificial Intelligence (AAAI '90)*, pages 491–496. AAAI Press. Menlo Park, CA, 1990.

[28] O. Zamir and O. Etzioni. Grouper: a dynamic clustering interface to web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1361–1374, 1999.