Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2004 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2004

Enhancing Personalized Indexing with XML

Shin-Jeng Lin
Le Moyne College

Enhancing Personalized Indexing with XML

Shin-jeng Lin
Le Moyne College
linsj@lemoyne.edu

ABSTRACT

Studies show that revisiting Web pages constitutes a significant segment of Web navigation and information retrieval. To facilitate more efficient re-visitation of Web pages, Personalized Indexing (PI) is developing as a tool that enables a user to index and categorize the Web pages of interest in his/ her own terms. Furthermore, it allows a user to dynamically re-group their information collection tailored to his/ her information problems at hand. However, currently PI does not reflect the fact that personal knowledge is cumulative, analogical, and continuous, as there is no granularity of indexed terms. The conceptual relationships among index terms, including synonyms, homonyms, and semantic relations, are not recognized by PI. This paper will explore how XML and its family technology can be utilized to help construct a personal vocabulary management mechanism to supplement PI. It concludes with future research directions to meet the newly raised usability concerns.

Keywords

XML, information seeking, personalized indexing, indexing, navigation

INTRODUCTION

Motivation of the study

Studies show that a user revisits Web pages very frequently (McKenzie and Cockburn, 2001) and that a user tends to have a difficult time in orienting himself/ herself to the pages he/ she visited before (Keller, Wolfe, Chen, Rabinowitz, and Mathe, 1997; Abrams and Baecker, 1997). As a consequence, a user who cannot return to Web pages effectively is likely to experience frustration in navigating the information space and thus fail to find useful and/or update information.

Commercial Web browsers support returning to Web pages with functions such as “Bookmark” in Netscape and “Favorites” in Internet Explorer. Although these features seem useful helping to revisit Web pages, many users are still unsatisfied with them.

Purposes of Personalized Indexing

To facilitate a user to revisit Web pages more efficiently, (Lin, 2001, 2002) developed Personalized Indexing (PI) as a tool that extends the indexing function from the system to the user. It enables a user to index the Web pages that interest him/ her in his/ her own terms and organize them using the index terms as category labels. User indexing actually is very similar to user annotation in the sense that a user is commenting on the Web pages per se. The key difference is that index terms are required to adhere to a more strict syntax form than annotation text. For example, a user of PI use terms rather than sentences to represent the aspects of Web pages that are relevant to his/ her information problems, and use comma to separate index terms.

Furthermore, PI allows a user to dynamically re-group his/ her information collection tailored to the information problems at hand by re-specifying index terms as new category labels. For example, a user types “New York, Museum” to represent a Web page concerning museums in New York, “San Francisco, restaurants” to represent another page concerning restaurant information in San Francisco, “New York, restaurants” for another page. Later on, he/ she can group by “New York, San Francisco” to find that he/ she has two pages for New York and one for San Francisco or group by “restaurants, museums” to learn that he/ she has more information about restaurants than museums, but there is only one piece of information on restaurants for each city. Sample screen shots of indexing a Web page and its result are available in figure 1 and 2.

In addition to the group-by function, as described in the previous paragraph, PI also provides two additional organizational functions. First is a document-centric approach, which removes all the hierarchy of index terms and simply lists them as the way a user originally indexed his/ her interesting Web pages. Each Web page only appears once for a combination of index terms. Second is an indexing-term-centric approach. It lists one index term at a time and has all the corresponding Web pages underneath. A Web page can appear multiple times for different index terms. All the optional approaches together are to provide divergent choices for a user to better organize his/ her information collection, help him/ her self remember what each Web page is about, and find the pages he/ she intends to return faster. A user can switch from one organization approach to another, depending on his/ her information needs.

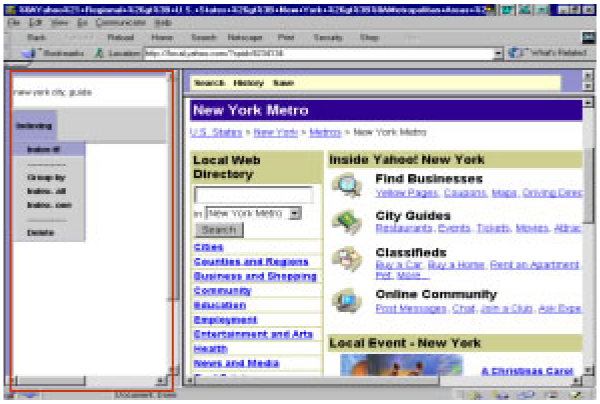
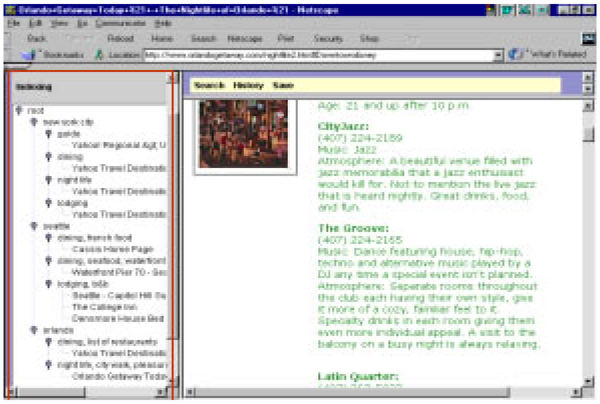


Figure 1: Screen shot of Using the "Indexing" Function



Age: 21 and up after 10 p.m.

CityJazz:
(407) 224-2159
Music: Jazz
Atmosphere: A beautiful venue filled with jazz memorabilia that a jazz enthusiast would kill for. Not to mention the live jazz that is heard nightly. Great drinks, food, and fun.

The Groove:
(407) 224-2165
Music: Dance featuring house, hip-hop, techno and alternative music played by a DJ any time a special event isn't planned.
Atmosphere: Separate rooms throughout the club each having their own style, give it more of a cozy, familiar feel to it. Specialty drinks in each room giving them even more individual appeal. A visit to the balcony on a busy night is always relaxing.

Latin Quarter:

Figure 2: Snap shot of using the "Group By" Function

In a study that evaluates Personalized Indexing, Lin (2001) finds that most subjects preferred PI over Bookmarks/ Favorites mainly for the reasons of representation, visualization, and organization capabilities, three of the four weaknesses of Bookmarks/ Favorites that Abrams, Baecker and Chignell (1997) pointed out. By representation, Abrams et. al. (1997) argues that the descriptors of pages in Bookmarks/ Favorites should be informative for a user to recognize them. PI was perceived as easier to return because a user can better remember what the Web pages in Bookmarks/ Favorites are about since he/ she can use his/ her own familiar terms to represent the Web pages. By visualization, Abrams, et al. (1997) suggests that Bookmarks/ Favorites must provide ways of *systematically* visualizing large numbers of bookmarks to aid retrieval. PI was favored because its index terms can also help overview and visualize the types of information found for a specific task and the number of Web pages for each type, as different index terms represent different aspects of a Web page. By organization, Abrams, et al. (1997) means that Bookmarks/ Favorites should be organized based on usage patterns so a user can easily save time and efforts in tuning his/ her archives. PI was perceived as more flexible than Bookmarks/ Favorites because the group-by function of PI helps efficiently reclassify the indexed Web pages into different clusters by specifying new criteria for organization. The dynamic organization capability of PI (i.e., changing the organization scheme for the entire information collection in one command) particularly separates itself from Bookmarks/ Favorites and the likes.

Limitations of Personalized Indexing

Nevertheless, PI is yet to live up to its full potential. The theory behind the design of PI is called MISE (Lin 2000, 2001, 2002), which postulates that the knowledge of a searcher is cumulative, continuous and analogical (Schutz & Luckmann, 1973; Marchionini, 1995); the current knowledge state of a searcher evolves from the previous search experience and will continue to evolve into the future. However, while the prototype of PI asks a user to use index terms to explicitly represent his/ her knowledge about the Web pages in interests, all index terms are independent of each other. The conceptual association among index terms for a specific Web page is not applied to other Web pages unless a user manually does so. That is, the conceptual and linguistic relationships among index terms, known or unknown to a user, are not captured.

The conceptual and linguistic relationships among index terms have been discussed in many indexing literature as the variables that would affect the indexing outcome and thus retrieval effectiveness. These indexing variables, including semantic relations, homonyms, synonyms, stop words, and stemming, can still be applied to s user who used PI, since he/ she is concerned with the indexing activity, not indexers. The loss or lack of information regarding the conceptual and linguistic relationships among index terms reflects the gap between the MISE theory and the system feature based on the MISE theory and likely would hamper the usability of PI.

Study goal

Thus, the goal of this study is to explore the possibility of closing the gaps between the design of PI and its underlying theory by allowing a user to construct his/ her own knowledge networks to support indexing useful Web pages efficiently. The personalized knowledge network, representing a user's knowledge of conceptual and linguistic relationships among the index terms, is a vocabulary management mechanism that can supplement the manipulation of a user's indexing schema for organizing Web pages. For example, a user can type in a broader term as the group-by criterion instead of all the related narrower terms. The organization of Web pages can also become more effective and make more sense, now that Web pages with synonym index terms can be grouped together.

In the second section of this paper, I will briefly explain what XML is and why and how XML can be applied to improve PI with a vocabulary management capacity. The third section describes how a user would index Web pages and engage in vocabulary management. In the fourth section, I will put everything together and propose a possible architecture. The fifth section raises usability concerns as vocabulary management is introduced to enhance PI. Finally, I will conclude with the discussion of the challenges ahead for PI and directions for future studies.

CONSTRUCTING PERSONALIZED KNOWLEDGE NETWORK WITH XML

What is XML

XML, standing for Extensible Markup Language (W3C), has a family of technology supporting XML documents, which use markup languages to enrich documents with a format that can help machines recognize meanings of documents. The XML family of technology includes document type definition (DTD), extensible stylesheet language transformation (XSLT), XPath, and document object model (DOM). DTD is to define the structure of a document by describing elements in the document, their attributes and relationships. XSLT enables the separation of the style layer from the structure layer of a document. XPath provides methods to queries data elements in XML documents. DOM is a standard for parsers to parse XML data elements that have been represented as a hierarchal tree structure in computer memory.

Why XML could help

XML documents are the text with additional custom-made markup tags. The markup tags can be used to enable the system understand the meaning of text. Because tags can be custom-made rather than pre-defined, the system can create tags as necessary. In addition, the standard Application Programming Interface (API) of the XML technology, such as DOM, that helps manipulation of XML documents more easily, would allow PI to enhance its organizational capabilities.

How XML could help

When applied to PI, the indexing scheme of a user can be represented as an XML document. Each index term is a data element in an XML document.

DTD can be used to keep the XML-represented indexing scheme in check with the vocabulary and grammar the machine can understand. XSLT along with DOM and XPath can be used to control the display of the indexing scheme when a user attempts to organize his/ her information collection in PI. For example, PI can use XSLT and XPath to filter and show only the index terms and their corresponding Web pages that satisfy the organization criteria specified by a user.

PI will maintain a vocabulary management sheet for each user, which is a XML document that represents a user's indexing scheme taking into account all the indexing variables.

Below, I will describe how XML can be applied to enhance the indexing activity by representing the key indexing variables. For each indexing variable, I will start with the rational for the need to consider that indexing variable in the context of PI, and proceed with sample DTD and XML document examples. In all the examples, the XML data element of each index term will be called 'index_term'.

Semantic Relation

According to Craven (1997), semantic relations include hierarchical relations and non-hierarchical relations. Hierarchical relations mean that conceptually, one term could be broader (e.g., writing tools) than the other (e.g., pens). Non-hierarchical relations indicate that the conceptual relations exist between terms even though one term is not necessarily broader than the other; for example, pens versus calligraphy.

Without the support for hierarchical relations, a user could not group index terms as categories; in other words, an index term cannot include other index terms. For example, a user would not be able to use "destination" to represent a collection of prospective vacation cities. When performing the "group-by" function, a user would have to tediously enter each individual prospective city instead of simply the term "destination."

The solution is to treat the regular index terms as broader terms and use <narrower_term> as the data element to represent index terms that are conceptually narrower in the separate vocabulary management XML document.

DTD

```
<!ELEMENT index_terms (index_term+)>
<!ELEMENT index_term (#PCDATA | broader_term |  narrower_term)*>
<!ELEMENT broader_term (#PCDATA)>
<!ELEMENT narrower_term (#PCDATA)>
```

XML

```
<index_terms>
  <index_term>
        <broader_term>Destination</broader_term>
        <narrower_term>New York</narrower_term>
        <narrower_term>San Francisco</narrower_term>
        <narrower_term>Orlando</narrower_term>
  </index_term>
</index_terms>
```

Non-hierarchical relations concern with the index terms that are topically relevant to each other but one is not necessarily conceptually broader or narrower than the other, for example, painters versus paint brushes, computers versus software. This could be useful for query expansion when a user tries to search against, rather than browse, his/ her personal information space confined in PI.

Additionally, when using non-hierarchical relations to enumerate different aspects of an object, it could fasten the retrieval of Web pages in PI. For example, a user can list "sight seeing," "cruise", "accommodation", "transportation" as different aspects of a vacation experience. These index terms can remind a user of what he/ she would have to search to plan a vacation.

DTD

```
<!ELEMENT index_terms (index_term+)>
<!ELEMENT index_term (#PCDATA, relation_term*)>
<!ELEMENT relation_term (#PCDATA)>
```

XML

```
<index_terms>
  <index_term>Vacation
        <relation_term>Sight seeing</relation_term>
        <relation_term>Cruise</relation_term>
        <relation_term>Musuem</relation_term>
  </index_term>
</index_terms>
```

A user can also use the set concepts (e.g., union, intersect and minus) to explore the similarity or differences between two or more clusters of information in PI, if he/ she has declared one additional pair of non-hierarchical relations, "vacation" to “destination”, plus the hierarchical relation example above (e.g., “destination” to different prospective places). For example, when a user asks PI to compare what kinds of information he/ she has gathered about one prospective vacation place but not about another, PI can use the minus set operator to compare the non-hierarchical relation index terms (e.g., aspects of vacation) between two prospective vacation places. Below is the example of such an XML vocabulary management sheet.

DTD

```
<!ELEMENT index_terms (index_term+)>
<!ELEMENT index_term (#PCDATA, relation_term*)>
<!ELEMENT relation_term (#PCDATA, broader_term*, narrower_term*)>
<!ELEMENT broader_term (#PCDATA)>
<!ELEMENT narrower_term (#PCDATA)>
```

XML

```
<index_terms>
   <index_term>Vacation
         <relation_term>
                  <broader_term>Destination</broader_term>
                  <narrower_term>New York</narrower_term>
                  <narrower_term>San Francisco</narrower_term>
                  <narrower_term>Orlando</narrower_term>
         </relation_term>
   </index_term>
</index_terms>
```

Synonyms

PI also needs to allow a user to specify synonyms so it can group indexed Web pages as a user intended, such as in a situation where many different terms can mean the same thing.

For example, a user could represent a Web page pertaining to prospective vacation spots sometimes by using the term, “destinations”, and some other times, the term, “city.” When a user specifies “destination” and “city” as synonyms, PI will be able to group information indexed with these two terms in the same folder. Moreover, synonyms are likely to be valid only under certain circumstance or for certain information tasks. For instance, “city” might not mean “destination” at all for the information tasks of an economic research paper on comparison of several cities. Therefore, additional attributes to the synonym index term for indicating the context is necessary.

DTD

```
<!ELEMENT index_terms (index_term+)>
<!ELEMENT index_term (#PCDATA, synonym_term*)>
<!ELEMENT synonym_term (#PCDATA)>
<!ATTLIST synonym_term context CDATA #IMPLIED>
```

XML

```
<index_terms>
   <index_term>Destinations
         <synonym_term context=”vacation”>City</synonym_term>
   </index_term>
</index_terms>
```

Homonyms

In the same way as synonyms, PI needs to support a user specifying homonyms in his/ her knowledge network to avoid ambiguity in organizing or classifying information collections. Especially when a user uses PI to collect information for multiple domains or tasks, many index terms could have different meanings for different domains, where a domain is a subject of knowledge. For example, “country” could be an ambiguous index term. “Country” could mean a nation or a rural area, depending on the contextual use. To improve the organizational capability, PI will need to allow a user to specify domains or meanings for his/ her homonym index terms. Before performing any organization function with the criteria that involve defined homonym, PI will prompt a user for clarifying the domain or meaning of those criteria.

DTD

```
<!ELEMENT index_terms (index_term+)>
<!ELEMENT index_term (#PCDATA, homonym_term*)>
<!ELEMENT homonym_term (homonym_domain* | homonym_meaning*)>
```

XML

```
<index_terms>
  <index_term>Country
        <homonym_term>
            <homonym_domain>Travel</homonym_domain>
            <homonym_meaning>Nation</homonym_meaning>
            <homonym_meaning>Rural</homonym_meaning>
        </homonym_term>
  </index_term>
</index_terms>
```

The indexing variables that does not require XML programming

Stop words (e.g., the, a, she, good) could add noise in matching index terms and corresponding documents, as in regular information retrieval. Most information systems would automatically remove stop words before matching queries and documents.

PI currently relies on a user to avoid using stop words by encouraging him/ her to use "keywords" to index Web pages. In order to enhance the organizational capability of PI, it is important for the system to remove stop words in the index terms automatically, before performing any organization functions. A user will have control to add, update and delete the list of stop words.

Index terms can be either in singular or plural forms. They can also be in different tenses or in the forms of verbs versus nouns. The system will not be able to recognize that index terms represented in different forms actually refer to the same thing unless the index terms are stemmed before it performs any of the organizational functions. Thus, stemming is a strongly desirable feature to vocabulary control and a user should have different levels of stemming options. For example, a more conservative approach is to only consider singular versus plural forms, whereas a more liberal approach is to treat terms in verbs or nouns the same.

Both removing stop words and stemming index terms can be achieved by non-XML programming as these indexing variables actually will not need a user's input. PI should automate these tasks case-insensitively whenever a user indexes Web pages or organizes his/ her information collection within PI.

APPLYING VOCABUARY MANAGEMENT TO THE INDEXING SCHEME

A menu-driven and form-based graphical user interface as well as a minor requirement on syntax are combined to support the user indexing and vocabulary management (i.e., construction of personal knowledge network) activities, which can be engaged together or at different times. The indexing syntax requirement that a user must know is to separate different index terms with comma. When engaging in both activities at the same time, a user can simply enter the terms, then highlight those which require vocabulary management and choose from the menu for the option of hierarchical relations, non-hierarchical relations, synonyms or homonyms. Afterwards, he/ she can follow the prompt to enter the associative terms in the newly emerging form-based text fields.

For example, a user could enter "New York, San Francisco, Restaurant" as the index terms. He/ she then proceeds to highlight "New York, San Francisco" and then chooses the option of "broader hierarchical relations" from the menu bar. The PI interface will dynamically generate a new form text field underneath the text field for the original index terms, within which a user can enter "destination." In this case, a user indicates (a) that the Web page contains information pertaining to

New York, San Francisco and Restaurant, and (b) that destination is a broader term than New York and San Francisco. A user can also type in directly “destination/ (New York, San Francisco), Restaurant” as index terms and the PI will follow its internal syntax rules to parse and update the index scheme and vocabulary management files simultaneously.

By a similar token, a user could type in “sight seeing cruise, museum, New York” to indicate the topicality of the Web page. A user can then highlight “sight seeing, cruise, museum”, select the “non-hierarchical relations” from the menu bar, and enter “vacation” in the dynamically generated new text field. Or, a user can enter the index terms with required syntax directly, like “vacation: (sight seeing, cruise, museum), New York.” Either case, he/ she is indicating that sight seeing, cruise, museum are all for the task of vacation planning.

To specify synonyms and homonyms, a user can enter “New York, lodging.” To indicate that the Web page contains information pertaining to accommodation or lodging in New York, which is a city rather than a state, a user can highlight “lodging,” select the “synonyms” option and type in “accommodation,” and follow to highlight “New York,” select the “homonyms” and key in “city.” “New York (city), lodging | accommodation” is the acceptable shortcut.

When engaging in vocabulary management after the indexing activity, a user can select the vocabulary management option from the menu bar and follow to a window where he/ she can specify conceptual relations among index terms, similar to the examples demonstrated. When doing so, all the Web pages that have the same index terms will be updated when vocabulary management is applied or index terms are revised.

Once a user has specified conceptual relations in his/ her vocabulary management set-up, he/ she can simplify the organizational procedure of PI, which becomes more powerful at the same time. For example, as described earlier, a user can type in “destination,” instead of each individual prospective vacation place, as the criterion to organize information collection around the places that have been defined as destination. A user can also use set operators to compare the strength of information collection for different cities.

ARCHITECTURE

The architecture of PI consists of two components: back-end proxy Web server and front-end scripting programs.

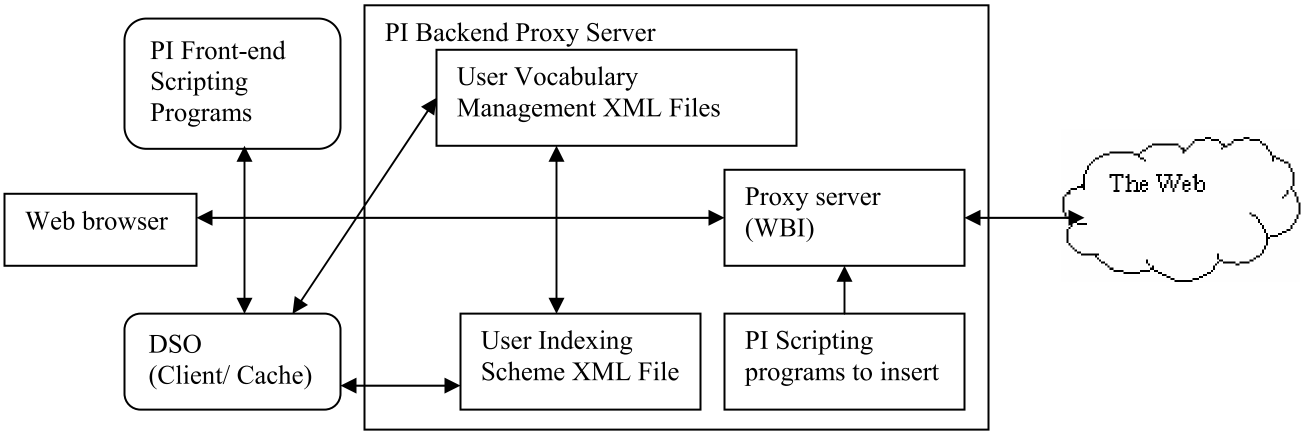

PI Front-end Scripting Programs
PI Backend Proxy Server
User Vocabulary Management XML Files
Web browser
Proxy server (WBI)
The Web
DSO (Client/ Cache)
User Indexing Scheme XML File
PI Scripting programs to insert

Figure 3: Architecture of PI

The back-end proxy server extends the IBM WBI 4.4 technology (Barrett and Maglio, 1998, 1999) with additional customized functionalities atop the core functionalities of the WBI proxy server. Implemented with agent technology, WBI can mediate the HTTP stream for observation and alteration, running either on the user’s client machine or on a server that many users access (Barrett and Maglio, 1998, 1999). The customized functionalities for PI will be to insert the PI front-end scripting programs to the Web pages requested by a user and to maintain index scheme and vocabulary management files, both of which are in XML.

The scripting programs, passed along with the HTML pages a user requests, will interact with a user with their functionalities, including user indexing, three organizing methods (i.e., group-by, document-centric listing, indexing-term centric listing), maintaining the knowledge network with vocabulary management, and viewing the index scheme. Technically, the client-side scripting programs consist of the modules for removing stop words and for stemming index terms. The program will also use the data binding technique to communicate and update index scheme files and vocabulary management files. The data binding technique can cache the resulting data on the client computer. The ability to cache the data on the client makes it possible to dynamically manipulate the data without additional server hits. The link can be made in both directions, so data binding can be used to both display content on the client and update content on the server (Edwards, 1997). Both index scheme and vocabulary management files are treated as data source objects. The scripts will interact with the data source objects to load the index scheme and allows a user to update the index scheme and engage in vocabulary management.

DISCUSSION

There are several issues arising when imposing the indexing responsibility on a user. These issues mainly concern usability of PI. PI likely requires more cognitive and physical efforts from a user, compared to traditional Bookmarks and Favorites. Lin (2001) conducted a lab-controlled, single-task, 3-successive-search-session study in evaluating the original PI that did not support vocabulary control. He found that most of 20 subjects perceived that it was fairly easy to index Web pages by typing index terms, although a few said that it was hard to think of index terms at times and that indexing could become tedious. In a real-life setting, a user is likely to gather information for multiple tasks over a long period of times. Remained to be answered are these questions, to what extent is a user willing to manually index Web pages and build and manage his/ her personal knowledge vocabulary, and whether somewhat increasing effort from a user can be paid off by the effectiveness and efficiency benefits of PI?

To reduce a user's effort, PI could build-in a default, starting-point, and generic vocabulary management sheet and then allow a user to customize the vocabulary management sheet as he/ she becomes more comfortable with the system. In addition, PI can experiment with the artificial intelligence techniques, such as text mining and natural language processing, to discover the conceptual and linguistic relations of index terms and make appropriate suggestions to a user.

CONCLUSION

Many information tasks require dynamically grouping of the evolving information collections for information seeking, retrieval, organization, comparison and synthesis, based on the information problems at hand. For example, message boards can be enhanced with such functionality to better trace threads. Email archives can be better organized to search for relevant emails. Project management can be improved for retrieving relevant documents and comparing projects in a shorter amount of time and a less amount of efforts. While many prior studies (e.g., Roussinov and Chen, 1999, 2001; Roussinov et. al. 2001) have focused on machine automation in clustering information objects and even labeling the clusters, Lin (2001) argued that highly uncertain information tasks that emphasize learning over resolving problems per se should particularly put a user in the center of information tasks. PI is an interface designed to support a user playing a more active role in engaging in such information tasks.

This paper explores the possibility of using XML as a mechanism for vocabulary management to supplement PI. The next stage of the study will center on implementation and evaluation. Even though XML seems conceptually feasible and technically promising to enhance PI with the vocabulary management capability, it will have to answer the usability questions regarding the required effort from a user. Long-term evaluation of how a user interacts with PI is needed to study the effect of PI and how to improve the interface for better usability. Meanwhile, it is also a prominent research topic to pursue an ideal balance between the system-centered assistance with artificial intelligence techniques and the user-centered control and self-efficacy in order to support information tasks that evolve swiftly over time.

REFERENCES

1. Abrams, D., Baecker, R.M., and Chignell, M. (1998) Information Archiving with Bookmarks: Personal Web Space Construction and Organization, In *Proceedings of the CHI '98 Conference,* ACM Press, 41-48.
2. Barrett, R. and Maglio, P. P. (1998) Intermediaries: New places for producing and manipulating web content. In *Proceedings of the Seventh International World Wide Web Conference (WWW7)*, Brisbane, Australia.

3. Barrett, R. and Maglio, P. P. (1999) Intermediaries: An approach to manipulating information streams. *IBM Systems Journal*, 38, 629-641
4. Craven (1997) *Thesaurus Construction*. Available at http://instruct.uwo.ca/gplis/677/thesaur/main00.htm
5. Edwards, M. (1997) 1,001 Ways to Get Input from Web Users. Available at http://msdn.microsoft.com/library/en-us/dnscrpt/html/1001ways.asp
6. Keller, R. M., Wolfe, S. R., Chen, J. R., Rabinowitz, J. L., and Mathe, N. A. (1997) Bookmarking Service for Organizing and Sharing URLs. In *Proceedings of the Sixth International World Wide Web Conference*, Santa Clara, CA.
7. Lin, S. (2001) *Modeling and supporting multiple information seeking episodes over the Web*, Unpublished doctoral dissertation, Rutgers University.
8. Lin, S. J. (2002) Design space of Personalized Indexing: Enhancing successive Web searches for transmuting information problems. In *Proceedings of the 2002 Americas Conference on Information Systems,* Dallas, TX.
9. Lin, S. J. and Belkin, N. J. (2000) Modeling Multiple Information Seeking Episodes, *Proceeding of the 63rd ASIS Annual Meeting,* Chicago, IL, 133-147.
10. Marchionini, G. (1995) *Information seeking in electronic environments*. Cambridge University Press, New York NY.
11. McKenzie, B., and Cockburn, A. (2001) An Empirical Analysis of Web Page Revisitation. *34th Hawaiian International Conference on System Sciences, HICSS34 (CD ROM), Maui, Hawaii,* IEEE Computer Society Press.
12. Roussinov, D. and Chen, H. (1999) Document Clustering For Electronic Meetings: An Experimental Comparison Of Two Techniques, *Decision Support Systems,* 27, 1-2, 67-79.
13. Roussinov, D., and Chen, H. (2001) Information Navigation on the Web by Clustering and Summarizing Query Results, *Information Processing and Management*, 37, 6, 789-816.
14. Roussinov, D., Crowston, K., Nilan, M., Kwasnik, B., Cai, J. and Liu, X. (2001) Genre Based Navigation on the Web, *Hawaii International Conference on System Sciences (HICSS-34)*. Island of Maui.
15. Schutz, A. and Luckmann, T. (1973) *The structures of the life-world*. Translated by R. M. Zaner & H. T. Engelhadt, Jr. Evanston, IL: Northwestern University Press.
16. W3C (World Wide Web Consortium (W3C). Extensible Markup Language. Available at http://www.w3.org/XML/