DOCUMENT RESUME

ED 427 689 IR 019 240

AUTHOR Budzik, Jay; Hammond, Kristian; Marlow, Cameron; Scheinkman, Andrei

TITLE Anticipating Information Needs: Everyday Applications as Interfaces to Internet Information Resources.

PUB DATE 1998-11-00

NOTE 9p.; In: WebNet 98 World Conference of the WWW, Internet, and Intranet Proceedings (3rd, Orlando, FL, November 7-12, 1998); see IR 019 231. Figures may not reproduce clearly.

PUB TYPE Reports - Descriptive (141) -- Speeches/Meeting Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.

DESCRIPTORS Artificial Intelligence; *Computer Interfaces; *Computer System Design; *Information Management; *Information Retrieval; *Information Systems; Man Machine Systems; Relevance (Information Retrieval); *User Needs (Information); World Wide Web

IDENTIFIERS Filters; Prototypes; *Query Processing; Ranking; Search Engines

ABSTRACT

This paper outlines work on a class of systems called Personal Information Management Assistants (PIMAs). PIMAs observe user interaction with everyday applications and use these observations to anticipate a user's information needs. They then automatically fulfill these needs by accessing Internet information sources, filtering the results, and presenting them to the user. They allow everyday applications to serve as interfaces for traditional information retrieval systems (e.g., Internet search engines). The paper presents preliminary work on an architecture for this class of systems and progress in implementing such a system, including finding relevant pages (i.e., query construction and information filtering) and exploiting structural clues. Preliminary results and directions for future work are also discussed. Two tables present output of a query generated from a page on Java standardization and results of clustering responses; three figures show the PIMA architecture and examples of displays suggesting relevant World Wide Web pages and images. (Author/AEF)

Anticipating Information Needs: Everyday Applications as Interfaces to Internet Information Resources

Jay Budzik, Kristian Hammond, Cameron Marlow, and Andrei Scheinkman
Intelligent Information Laboratory
The Institute for the Learning Sciences
Northwestern University
1890 Maple Ave.
Evanston, IL 60201 USA
{budzik, hammond, camarlow, andrei}@ils.nwu.edu
http://infolab.ils.nwu.edu/

Abstract: We outline work on a class of systems called *Personal Information Management Assistants* (PIMAs). PIMAs observe user interaction with everyday applications, and use these observations to anticipate a user's information needs. They then automatically fulfill these needs by accessing Internet information sources, filtering the results, and presenting them to the user. Essentially, they allow everyday applications to serve as interfaces for traditional information retrieval systems. In this paper, we present our preliminary work on an architecture for this class of systems, and our progress implementing such a system. Finally, we discuss our preliminary results and survey directions for future work.

1. Motivation and Introduction

In recent years, we have experienced an explosion in the amount of information available online. Unfortunately, tools that allow users to access this information are still rudimentary. Users are often forced to express their information needs in Boolean query languages, or fill out a complicated form. More, systems often offer results that are unnecessarily redundant and poor in quality—partly because the user is unable to specify his needs in terms of a query well enough, and partly because of the nature of the software servicing his query. Some intelligent systems allow users to pose their information needs in the form of a question [Burke, et al., 1997] [Kulyukin, et al., 1998]. Others allow users to search by example [Hammond, et al., 1994]. Nonetheless, these kinds of systems still require the user to make his information needs explicit to the system. Thus, while Internet search engines provide a first step at solving this information access problem, most of them not only fail to produce good results reliably, but are also hard to use. Systems that answer questions or allow users to search by example provide a solution to part of this problem, yet remain inconvenient.

In response to the problems posed by the current state of information retrieval systems, we are working on a class of systems we call *Personal Information Management Assistants* (PIMAs). PIMAs observe user interaction with everyday applications, and use these observations to anticipate a user's information needs. They then automatically fulfill these needs by accessing traditional information retrieval systems (e.g., Internet search engines), filtering the results, and presenting them to the user. Essentially, they allow everyday applications to serve as interfaces for traditional information systems, paving the way for us to remove the notion of query from information systems altogether.

In this paper, we present our preliminary work on an architecture for this class of systems, and our progress implementing such a system. Finally, we discuss our preliminary results and survey directions for future work.

2. Overview and Architecture

One of the main insights driving our work is that information-seeking behavior, such as posing a query to a search engine, is *goal-directed* behavior. In this view, posing a query to a search engine is a step in a plan that satisfies the goal of finding information about a certain topic. Given that finding information is usually in service of some goal, we can construct a library of *information-consumption scripts* (using "script" in the sense of [Schank & Abelson, 1977]) associated with satisfying a user's goals. Scripts are knowledge structures that house information about highly routine situations. In an appropriate context, they serve the purpose of making strong predictions about situations and sequences of events. For a PIMA, knowledge of a user's information-consumption scripts means the ability to anticipate information-seeking goals and the ability to automatically fulfill them.

The second observation we bring to this problem allows us to readily apply the above understanding technique. The observation is that standard information systems, the documents themselves, and the environments in which they are produced, consumed, and otherwise manipulated are highly structured and regular. To mention just a few of these regularities:

1. Within the document, gross structures such as headings, paragraphs, and titles are prevalent and have a well-known semantics.
2. Documents have a gross morphological form (e.g., letter, newspaper article, invitation, memo, etc.), that corresponds directly to the document's function.
3. Everyday computer applications serve a particular and easily attributable function.
4. Everyday computer applications have well-formed interaction semantics.
5. Information retrieval systems have an interface that is arguably easier for a computer to use than a human.
6. Information retrieval systems are computer programs that generate regular output.

The above structure and regularity, as well as the semantics associated with the regularity of this world make it particularly amiable for a computer program. This is not all that surprising—after all, the environment, in fact, *is* a collection of computer programs. What is interesting, however, is that because this particular part of the world is so strongly structured, human behavior within it is also highly structured and regular. It follows that this regularity in the environment (in terms of computer programs, documents, and information systems) and in user behavior should be known and used by a PIMA to inform the task of understanding that behavior.

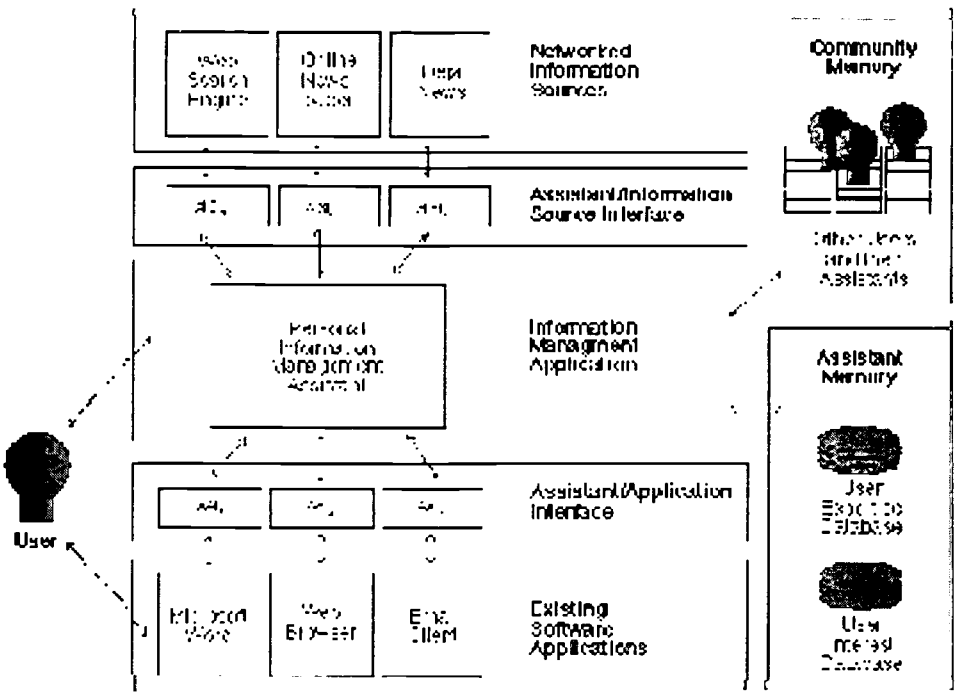

Figure 1: PIMA Architecture

We have built a prototype PIMA that observes user interaction with everyday applications (e.g., Netscape Navigator, Microsoft Internet Explorer, and Microsoft Word), and, using a very preliminary knowledge of information-consumption scripts, is able to anticipate a user's information needs. It then attempts to automatically fulfill them using common Internet information resources.

Given the requirements that it must observe several applications and that it must also use multiple information resources, we have adopted the five-tiered architecture depicted in [Fig. 1]. The user interacts with their everyday applications (shown at the bottom of the diagram), and the information management application in the middle. Through a series of adapters, the assistant application communicates with the existing software applications through the operating system's IPC facilities. The assistant then interprets user behavior in these applications, and constructs a query, which it sends off to information sources at the top. It collects the results, and applies information-filtering heuristics that allow it to present the user a concise, high-quality list of suggestions. These suggestions are presented in a window for the user to browse. Eventually, we plan to give our PIMA a memory of user interests and expertise (c.f. [Budzik & Hammond, 1998]), as well as the ability to communicate with other users' assistants, in order to personalize and improve the quality of the results.

3. Implementation

Currently, our PIMA observes user interaction in unmodified versions of Microsoft Internet Explorer and Microsoft Word, as well as a modified version of Mozilla (Netscape's free-source version of Navigator). The PIMA communicates with Microsoft Word and Internet Explorer through COM (PC only), and with Mozilla through BSD sockets (UNIX and PC). We designed our architecture with the idea that application interfaces should be the only OS-dependent components. We implemented the assistant application in Java, for maximum portability and ease of development. These design decisions afford us the ability to extend the PIMAs field of observation relatively easily, without having to change the core application code.

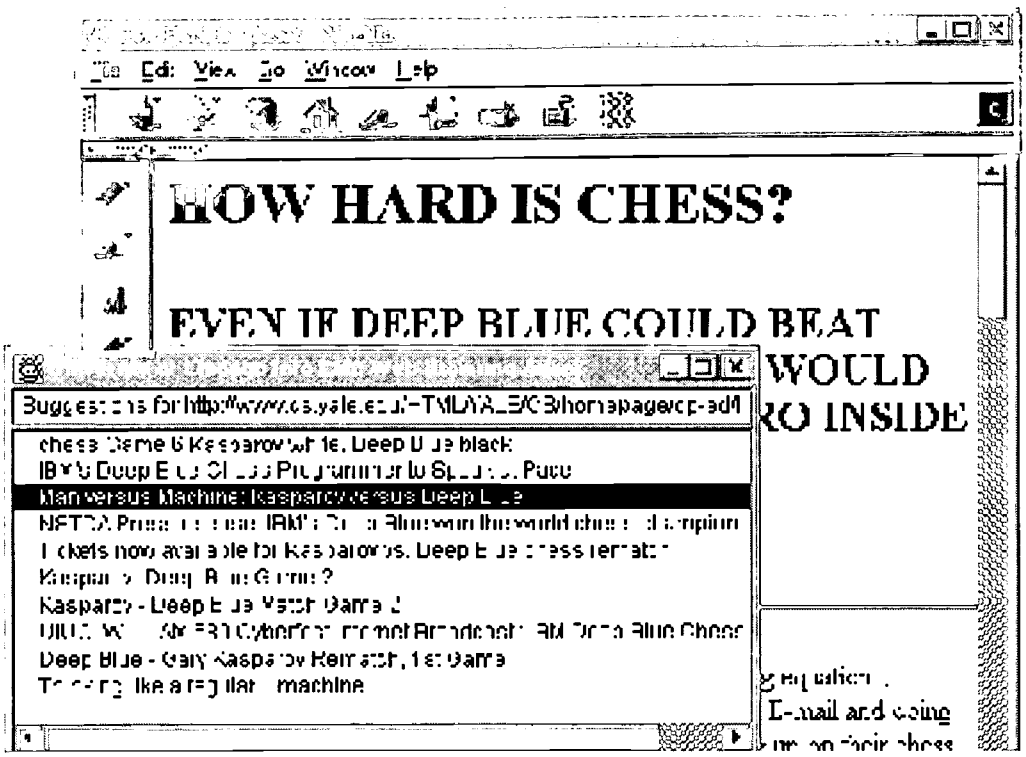

Figure 2: Suggesting Relevant Web Pages

3.1 Finding Relevant Pages

The simplest of the information-consumption scripts we have identified is associated with finding related web pages. The FIND-RELATED-PAGES script is composed of the following basic steps.

1. Summarize the document in terms of a few words.
2. Pose a query to a search engine using these words.
3. Sift through the results, searching for ones that are actually related.

It is applied when the assistant anticipates the user wants more information in the subject area of the document he currently manipulating. For the two Web browsers, the PIMA recognizes when a user navigates to a new web site, either by clicking on a link, or by explicitly opening a URL. In Microsoft Word, it recognizes when a user has opened a document or changed it significantly.

There are essentially two processes associated with retrieving relevant documents: query construction and information filtering. [Fig. 2] demonstrates the user interface associated with the result of these two processes applied in sequence.

3.1.1 Query Construction

In order to retrieve relevant sites, the PIMA constructs a query based on the contents of the current web page, and sends it off to AltaVista [1]. To construct a query, the PIMA uses three techniques to decide on which words should be included: a standard stop list and two heuristics for rating the importance of a word.

Sun Speaks Out On Java Standardization	http://www.html.co.nz/ news/110605.htm
Java Standardization	http://aidu.cs.nthu.edu.tw/ java/JavaSoft/www.javasoft.com/aboutJava/standardization
Java Standardization Update – SnapShot	http://www.psgroup.com/snapshot/1997/ss109706.htm
Informal XML API Standardization for Java	http://xml.datachannel.com/ xml/dev/XAPIJ1p0.html
International Organization For Standardization Gives Java The Nod	http://techweb1.web.cerf.net/ wire/news/1997/11/1118java.html
Java Standardization	http://java.sun.com:81/ aboutJava/standardization/index.html
Sun Moves Java Standardization Forward	http://techweb4.web.cerf.net/ wire/news/1997/09/0922standard.html
Java Standardization	http://java.sun.com/aboutJava/standardization/index.html
Java Standardization	http://www.javasoft.com/ aboutJava/standardization/index.html
Informal XML API Standardization for Java	http://www.datachannel.com/xml/dev/Commonality.html
Java Standardization	http://www.intel.se/design/news/javastand.htm
The impact of Java standardization	http://www.idg.net/ new_docids/find/java/suns/standardization/ developers/submitter/approval/ affects/new_docid_9-48305.html
Java Standardization - A Whitepaper	http://java.sun.com/aboutJava/standardization/javastd.html

Table 1: Output of a query generated from a page on Java standardization

[1] http://altavista.digital.com/

Sun Speaks Out On Java Standardization	http://www.html.co.nz/news/110605.htm
Java Standardization	http://aidu.cs.nthu.edu.tw/java/JavaSoft/ www.javasoft.com/aboutJava/standardization/, http://java.sun.com:81/aboutJava/ standardization/index.html, http://java.sun.com/aboutJava/standardization/index.html, http://www.javasoft.com/aboutJava/ standardization/index.html, http://www.intel.se/design/news/javastand.htm, http://www.idg.net/new_docids/find/java/ suns/standardization/developers/submitter/ approval/affects/new_docid_9-48305.html, http://java.sun.com/aboutJava/standardization/javastd.html,
Java Standardization Update - SnapShot	http://www.psgroup.com/snapshot/1997/ss109706.htm,
Informal XML API Standardization for Java	http://xml.datachannel.com/xml/dev/XAPIJ1p0.html, http://www.datachannel.com/xml/dev/Commonality.html,
International Organization For Standardization Gives Java The Nod	http://techweb1.web.cerf.net/wire/news/ 1997/11/1118java.html,
Sun Moves Java Standardization Forward	http://techweb4.web.cerf.net/ wire/news/1997/09/0922standard.html,
change nothing - JavaWorld - October 19	http://www.javaworld.com/javaworld/ jw-10-1997/jw-10-iso.html,

Table 2: Results of clustering responses

The first heuristic is that words at the top of the page tend to be more important than the words at the bottom. The second is words that occur with high frequency (that are not in the stop list) in a document are usually representative of the document. The terms with the top 20 weights form the query that is sent to AltaVista.

3.1.2 Information Filtering

Because the results returned from AltaVista are often redundant, containing copies of the same page or similar pages from the same server, the PIMA must filter the results so as not to *add* to a user's feeling of information overload [Maes, 1994]. If these similarities are not accounted for, some of the more interesting pages returned by AltaVista may be missed. Moreover, we constantly face the risk of annoying the user instead of helping him. As a result, we actively attempt to reduce the amount of irrelevant information presented, and in doing so address some of the problems associated with a constantly updating interfaces (like [Lieberman, 1995]). To this end, we have designed our prototype to collect search engine results and cluster similar pages, displaying a single representative from each cluster for the user to browse.

For this task, we currently use three pieces of information that AltaVista returns for each document: the document's title, its URL, and the date on which it was last modified. For each of these pieces we have a heuristic similarity metric. The similarity of two titles is represented numerically by the percentage of words they share in common; two URLs are compared by examining their host, port and closeness in terms of directory structure; and two dates are judged by the number of days separating them. The combination of these similarity metrics is sufficient to determine the uniqueness of the documents returned.

[Tab. 1] shows a typical response from AltaVista generated by a query posed in response to a page on Java Standardization (we have deleted several long ones for brevity). Notice there are a number of mirror sites, as well as logical duplicates (they may have different URLs, but they are the same file). [Tab. 2]

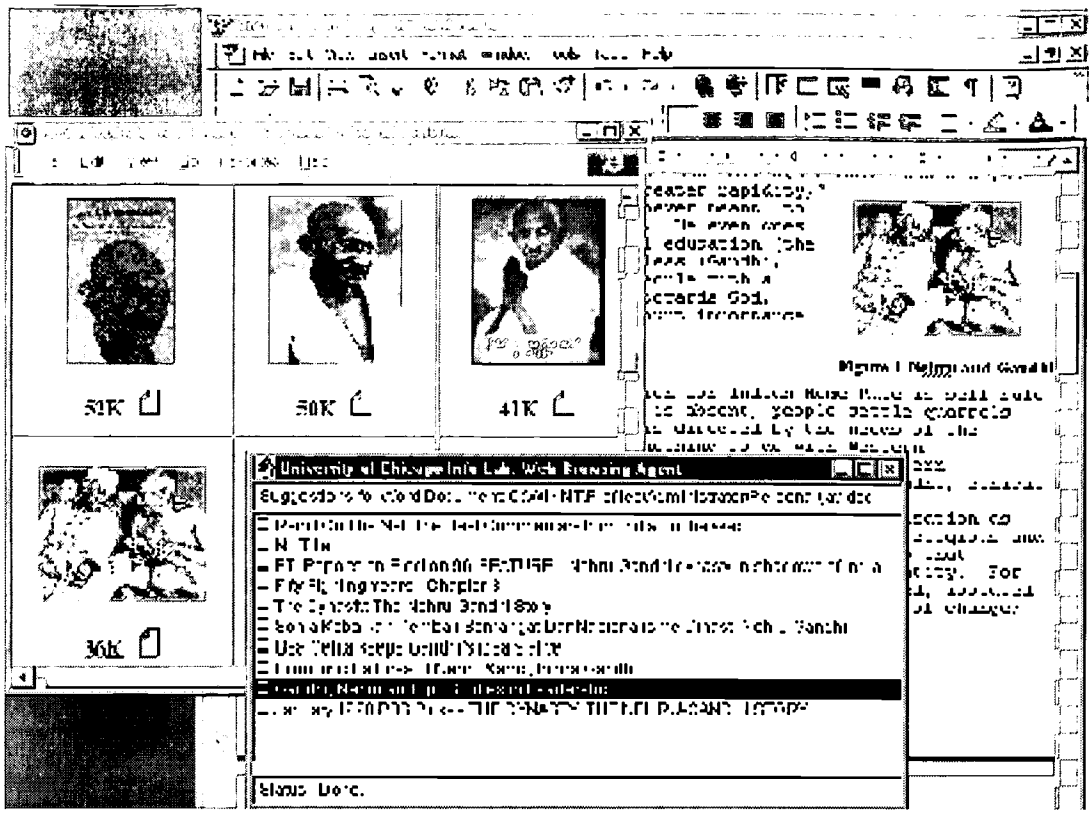

Figure 3: PIMA Prototype - Suggesting Images

shows these URLs after clustering. Instead of presenting the user with 20 sites, we present him with 10, effectively removing the duplicates and mirrors.

3.2 Exploiting Structural Cues: An Example

Of course, the more abstract goal is to find relevant resources, not simply related pages. To demonstrate both the feasibility of extending the domain of resources exploited by our architecture, as well as power of this paradigm of interaction, we have extended the PIMA's functionality in Microsoft Word to include the ability to recognize when a user inserts a caption. The script associated with this situation suggests a different class of information-consumption behaviors we can anticipate: those that are dependent on the structural context of the active document. The FIND-RELATED-IMAGES script is applied when the user has inserted a caption with no image to fill it (and probably others—we make no claims about being exhaustive, here). It contains the following steps:

1. Summarize the desired picture in terms of a few words.
2. Send these words off to an image search engine.
3. Sift through the results and choose the best one.

In response to the above situation, the PIMA applies this script and predicts the user will require an image. It then sends off a query to Arriba Vista [2] a commercial implementation of WebSeer [Frankel, et al., 1997], an image search engine. The PIMA constructs the query using a piece of knowledge about the structure of documents, in general: that the caption of an image is a verbal summary of that image. Hence the query it constructs is simply the conjunction of the stop-listed terms in the caption. The results from the image search are presented in a web browser window, from which the user can drag-and-drop the images into Microsoft Word (see [Fig. 3]).

[2] http://www.arribavista.com/

4. Related Work

Software that recommends web pages and learns user preferences has been the intense focus of recent research in Artificial Intelligence and Information Retrieval. A few good examples of related work follow. Letizia [Lieberman, 1995] is an agent that recommends web pages by compiling a profile and doing lookahead search in the locus of the current web page. ParaSite [Spertus, 1998] is a system that suggests relevant web pages using link topology. WebMate [Chen & Sycara, 1998] keeps track of user interests and uses them to extend queries to search engines. Maxims [Lashkari, et al., 1994] learns relationships between application events and user interaction patterns in an email package and uses them to predict and carry out common tasks. The Remembrance Agent [Rhodes & Starner, 1996] suggests related documents you've written as you compose a new document. The Shop Bot [Doorenbos, et al., 1997] queries various commercial sites for price information to aid the user in the task of comparison-shopping. Finally, Alexa [3] is a company dedicated to recommending web pages, and is responsible for the "Find Related Pages" button in the new Netscape Navigator.

Our approach differs from the above and contributes in several important ways:

1. *User behavior modeling.* User behavior is modeled and can be used to inform all of the assistant's tasks.
2. *Search is automatic and distributed.* The goal of finding relevant documents is predicted and carried out automatically using an extensive, dynamic chunk of the web. We do not require a pre-compiled index of related documents—only access to standard web search engines.
3. *Search is directed.* Search is directed and constrained by the content and the structure of the document at hand.
4. *Results are post-processed.* By applying several simple, low-cost web page similarity heuristics we were able to improve the quality of suggestions dramatically.

5. Directions for Future Research

Our initial experiment suggests that the combination our heuristics for query generation and for response clustering produce high quality, on point suggestions. Our hypothesis is that this is due to the fact that the query generation algorithm we apply to documents roughly mirrors the process of document indexing, and that the clustering heuristics are effective. While our initial results are promising, the system has much room for improvement.

Most obviously, our library of scripts is very sparse. Augmenting it so it understands more user/application interactions (and thus is able to anticipate more kinds of information needs) will be of primary concern. Tied to this is the fact that the PIMA has a very rudimentary notion of document structure. As it stands, the query construction algorithm ignores all but the most obvious structure of the documents it uses. Applying heuristics to improve the query construction algorithm based on document structure will not only improve query construction, but it will also afford the assistant the opportunity to direct its search for recommendations based on that structure. Queries frequently include terms that are of little information value to vector space information retrieval systems like AltaVista. Composing a table of term frequencies from a random sample of web documents and using this table to negatively weight terms with very high frequencies will increase the number of "quality" query terms sent to information sources. As a further improvement, we plan on adding support for more information resources and developing a vocabulary for expressing the kind of information available, as well as a means by which the assistant can be made aware of new information resources as well as filter suggestions in a task-directed way.

Finally, our prototype is reactive in the strictest sense—it has no memory, and knows nothing about what the user prefers. Giving our PIMA the ability to learn user preferences and leverage this knowledge as it attempts to anticipate information needs, select appropriate sources, and filter results from those responses is sure to improve the quality of suggestions dramatically. Clearly there is much more to be done.

---

[3] http://www.alexa.com/

6. Conclusion

In summary, we have outlined several major problems associated with contemporary information access paradigms. In response, we presented an architecture for a class of systems we call *Personal Information Management Systems*. These systems observe user interactions with everyday applications, anticipate information needs, and automatically fulfill them using Internet information sources. Essentially, they turn everyday applications into intelligent user interfaces for conventional information retrieval systems. We presented our initial work on a prototype of this kind of system, some related work, and closed with directions for future research.

7. References

[Budzik & Hammond, 1998] Budzik, J.; Hammond, K. 1998. Learning for Question and Text Classification. In *Proc. Learning for Text Categorization Workshop.* AAAI Technical Report WS-98-05.

[Burke, et al., 1997] Burke, R.; Hammond, K.; Kulyukin, V.; Lytinen, S.; Tomuro N.; Schoenberg, S. 1997. *Question Answering from Frequently-Asked Question Files: Experiences with the FAQ Finder System.* Technical Report TR-97-05, The University of Chicago, Department of Computer Science.

[Chen & Sycara, 1998] Chen, L; Sycara, K. 1998. WebMate: A Personal Agent for Browsing and Searching. In *Proc. Autonomous Agents '98.*

[Doorenbos, et al., 1997] Doorenbos, R.; Etzioni, O.; Weld, D. 1997. A Scalable Comparison-Shopping Agent for the World-Wide Web. In *Proc. Autonomous Agents '97.*

[Frankel, et al., 1996] Frankel, C.; Swain, M.; Athitsos, V. 1996. *WebSeer: An Image Search Engine for the World Wide Web.* Technical Report TR-96-14, The University of Chicago, Department of Computer Science.

[Hammond, et al., 1994] Hammond, K.; Burke, R.; Schmitt, K. 1994. A Case-Based Approach to Knowledge Navigation. In *Working Papers of the AAAI '94 Workshop on Multi-Media and Artificial Intelligence.*

[Kulyukin, et al., 1998] Kulyukin, V.; Hammond, K.; Burke R. 1998 Answering Questions for an Organization Online. In *Proc. AAAI '98.*

[Lashkari, et al., 1994] Lashkari, Y.; Metral, M.; Maes, P. 1994. Collaborative Interface Agents. In *Proc. AAAI '94.*

[Lieberman, 1995], Lieberman, H. 1995. Letizia: An Agent That Assists Web Browsing. In *Proc.IJCAI '95*

[Maes, 1994], Maes, P. 1994. Agents that Reduce Work and Information Overload. In *Communications of the ACM* 37(7).

[Schank & Abelson, 1977] Schank, R.; and Abelson R. 1977. *Scripts, Plans, Goals and Understanding.* New Jersey: Lawrence Earlbaum Associates.

[Spertus, 1998] Spertus, E. 1998. ParaSite. In *Working Papers of the AAAI '98 Workshop on Recommender Systems.*

Acknowledgements

The authors thank the Department of Computer Science at The University of Chicago for their support during their transition to Northwestern. The first author thanks Janos Simon (of the same department) for reminding him why he's doing AI.

U.S. Department of Education
Office of Educational Research and Improvement (OERI)
National Library of Education (NLE)
Educational Resources Information Center (ERIC)

NOTICE

REPRODUCTION BASIS

This document is covered by a signed "Reproduction Release (Blanket) form (on file within the ERIC system), encompassing all or classes of documents from its source organization and, therefore, does not require a "Specific Document" Release form.

This document is Federally-funded, or carries its own permission to reproduce, or is otherwise in the public domain and, therefore, may be reproduced by ERIC without a signed Reproduction Release form (either "Specific Document" or "Blanket").