AUTOMATED INFORMATION RETRIEVAL USING CLIPS

Rodney Doyle Raines III
United States Coast Guard
Department of Computer Science
Cal Poly, San Luis Obispo

James Lewis Beug
Department of Computer Science
Cal Poly, San Luis Obispo

Abstract. Expert systems have considerable potential to assist computer users in managing the large volume of information available to them. One possible use of an expert system is to model the information retrieval interests of a human user and then make recommendations to the user as to articles of interest. At Cal Poly, a prototype expert system written in CLIPS serves as an Automated Information Retrieval System (AIRS). AIRS monitors a user's reading preferences, develops a profile of the user, and then evaluates items returned from the information base. When prompted by the user, AIRS returns a list of items of interest to the user. In order to minimize the impact on system resources, AIRS is designed to run in the background during periods of light system use.

THE INFORMATION RETRIEVAL PROBLEM

Introduction

A potentially important source of information available to both industrial and university researchers is electronic news. Like its printed counterparts, electronic news consists of a set of articles on various topics. Apart from the obvious difference of medium of distribution, electronic news differs from printed news in that it covers a wide range of topics, has typically non-professional contributors and a much greater volume of information. The volume of electronic news at Cal Poly is estimated to be over 50 megaBytes per day. Data gathered by the Network Measurement Project at the DEC Western Research Laboratory in Palo Alto (Reid 1991), California, show the following for a sample of 675 news sites for July 1991.

Average traffic per day (*megabytes*):	12.189
Average traffic per day (*messages*) :	5674

Clearly, a single user cannot read all the electronic news available, nor in fact are all articles relevant to a given user. Furthermore, much of what is distributed via electronic news is redundant.

Reading electronic news

In order to make use of the news, the user must have a strategy for selecting a relevant subset of what is available each day. The news itself is partitioned into news groups consisting of articles usually related to some common topic, such as artificial intelligence. Current news readers (programs for reading the news) provide several strategies:

- Articles may be read from selected news groups in an order determined by the user.
- Articles within a selected news group can be read in the order received.
- Articles which are related (a thread) may be read in order.

Additionally, a user may choose to display a list of article headers or subject lines, choosing articles to be read from that list. In any case, typically the user is responsible for determining which articles are relevant to his or her information needs.

The Problem

The major problem with these approaches is that the user will have to read more articles in selected groups than necessary to find information of interest (low relevance) and may miss information in news groups not typically read (low precision). In fact, electronic news shares many of the same problems which have been addressed by both the information storage and retrieval (IS&R) and library communities. Typically however, IS&R systems are expected to retrieve either exactly (neither fewer or more) the relevant documents published previously or, in the case of current awareness systems currently. For current awareness systems, an information specialist will prepare a profile of user interests, which is then used to route selected journals and/or articles to the reader. Complicating the issue is the fact that news articles tend to be very time sensitive and granular. Electronic news is time sensitive in content, news tends to deal with current issues which change rapidly, as well as space, news is not retained in readily accessable storage at Cal Poly for more than a week. The work described in this paper consists of using a CLIPS based expert system for automating information retrieval. Such a system should outperform the unaided reader, be adaptive to his or her changing information needs, and be extensible to other information systems such as electronic mail or messaging systems.

Related Work

Improving the performance of information retrieval systems is an area of research which falls into two general areas: improving the quality of the stored information and improving the performance of the user. Artificial Intelligence techniques have been used in both areas. Efforts to improve the quality of stored information can be categorized by three main approaches:

- Representing documents and search terms as an associative network
- Using natural language techniques to select index items
- Building a knowledge base from the document's contents

The system that Wyle and Frei have developed for managing network news is an example of improving the quality of stored information. They assume a passive user and perform algorithmic information gathering, filtering and dissemination (Wyle&Frei 89). This system lacks the sophistication provided by using an expert system to interact with the users profiles as we have developed.

Much of the use of expert systems in information processing has concentrated on determining a user's needs and then directing the appropriate information to the user. Gauch has recently developed an expert system which assists a user in searching through full-text knowledge-bases. Her system works with syntactically and semantically unprocessed text and uses a domain independent search strategy to present passages to a user in decreasing order of relevancy (Gauch 91). The system also does not deal with the entire information base; instead it sends out selected queries to database servers acting as an intermediary for the user.

The SCISOR project at MIT (Jacobs&Rau 91) has successfully demonstrated using natural language processing techniques to extract relevant information from online financial news. Much of this success can be attributed to its narrow domain which belies a lack of extensibility.

The LENS program at MIT sorts and prioritizes E-mail, based on importance and urgency. This system also suggests a course of action to the user and can automatically respond to some messages (Robinson 91). The initial implementation of the Automated Information Retrieval System (AIRS), running under *arn - adaptive read news*, concentrates on determining the user's needs and then directing information to the user. Later iterations will attempt to improve the quality of stored information.

AUTOMATED INFORMATION RETRIEVAL SYSTEM (AIRS) PROTOTYPE

To assist a user in sorting through the tremendous volume of information available at a computer center, we are developing an intelligent information management system. The heart of this management system is an Automated Information Retrieval System (AIRS). We chose to develop this system using a rapid prototyping methodology. At the onset, we established several objectives for the project. Our general objectives were to develop an environment for research into: the use of artificial intelligence in information retrieval; the application of traditional software engineering techniques to expert system development and the exploration of distributed computing in an expert system environment. More specific objectives are:

Programming Objectives

- Scalability to large databases
- Timely implementation
- Code reuseability
- Modularity
- Exportability to other computers
- Extensibility to other applications

User Interface Objectives

- Ease of use
- Transpancy to the user

To support these objectives, we chose CLIPS as the programming language. As a rule-based system, CLIPS readily adapts to a rapid prototype. An additional advantage is that the prototype need not be disposed of after the requirements specifications have been validated. Instead, the expert system can be reused after the prototype is finished, meeting yet another objective. CLIPS readily lends itself to modularity and scalability In addition, it is easy to learn. The availability of the source code, coupled with the wide base of computers already using CLIPS, makes it easily exportable to other computer systems.

The NeXT computer (NeXTstations), was selected for the initial implementation. A NeXTcomputer running the User Interface Builder provides a good rapid prototyping platform for Graphic User Interfaces (GUI's), which meets another of our objectives. The built-in primitives within the Mach operating system provide tools for distributed computing and easily allow for CLIPS to be run in a client/server configuration. Perhaps most important, the NeXT is very reasonably priced, providing more bang for the research buck.

AUTOMATED INFORMATION RETRIEVAL SYSTEM (AIRS) DESIGN

Overview

The Automated Information Retrieval System (AIRS) is just one component of a newsreader system. All articles are stored in a hierarchal file structure which we refer to as the newsbase. A user retrieves articles from the newsbase using arn - adaptive read news. Within arn, there are three main components:

- The Newsreader *(tass)*
- The search engine *(lqtext)*
- The Automated Retrieval System *(AIRS)*

An overview of the newsreader system is provided in Figure 1. The newsreader, *tass,* is a thread-based newsreader available via the Internet (Skrenta 1990). *tass,* like most newsreaders, allows a user to select newsgroups of interest, and then to browse through the newsgroups for articles of interest. The necessity of selecting newsgroups of interest is due to the sheer volume of news which must be otherwise eliminated. In addition to reading by newsgroups, *tass* allows a user to consider news threads, which are articles grouped by themes, within a newsgroup. Articles of interest contained in newsgroups not selected are never considered.

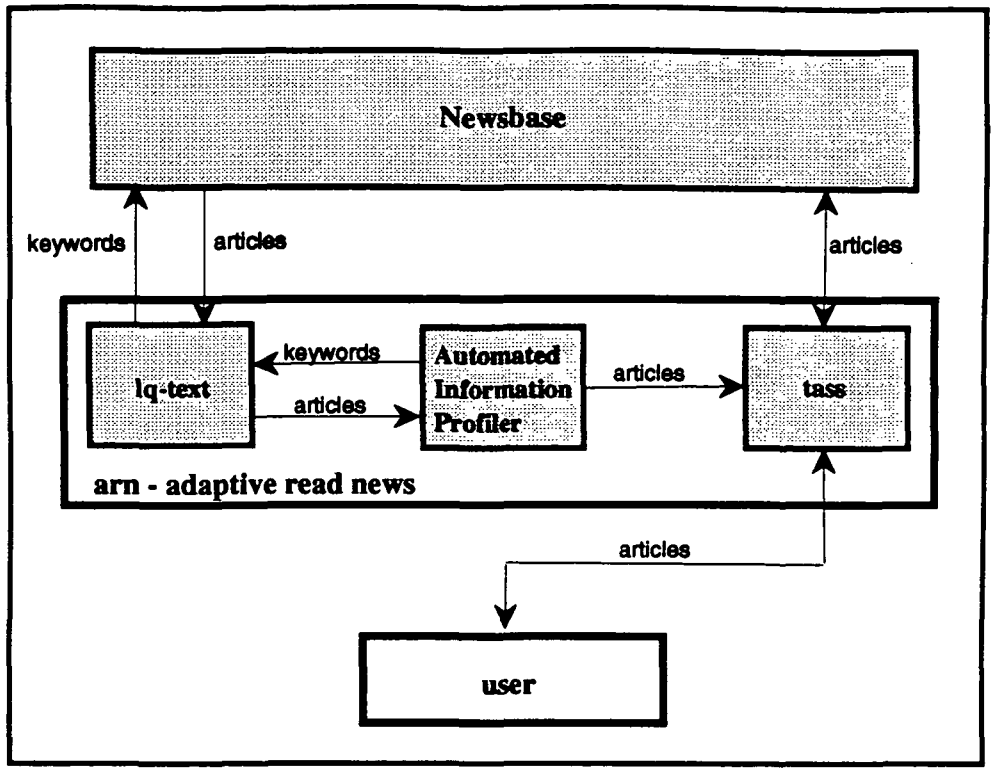

Figure 1. An overview of the newsreader system

To support a user who may need to locate information of interest in newsgroups other than those normally selected, a search engine component has been added to the newsreader system. The search engine component selected for the prototype newsreader is *LQ-text* (Quam 1990). Although not a true Boolean search engine, *LQ-text* allows for both keyword and keyphrase searches of a database. Using the search engine component, a user can compose a search of the newsbase and attempt to locate news of interest. This method of searching the newsbase is very labor intensive and does not lend itself to browsing. To alleviate this, an expert system component has been added to the newsreader, to compose and evaluate searches of the newsbase. This is the Automated Information Retrieval System (AIRS) component.

AIRS is the key component of the newsreader system. Written in the CLIPS expert system shell, it develops a profile of the user's news interests and disinterests. The profile is developed by monitoring the user's reading preferences at several levels. At the highest level, groups read by a user, both past and present, are considered by AIRS. Newsgroup information is obtained by examining the .newsrc file created by the newsreader. Historical information is kept in the .newsclps file. Information as to threads which have been read provide information as to the interests of the user, and threads which are "killed" provide information as to the disinterests of the user. Articles which are read or killed provide even more specific information as to the user's interests. Finally, a history of user-generated searches over the newsbase provides very detailed information as to the interests of the user. The various files containing this information relating to the user are parsed by AIRS, and a profile of user's interests, established by examining keywords, is created.

Once the user profile is created, the newsbase is searched for information which matches the user profile. This is done by sending queries to the search engine. The search tactics heuristics are used to improve query performance. Articles of interest returned by the search engine are evaluated and if necessary revised and sent back to the search engine. Once the information search process is complete, the findings are evaluated, ordered and presented to the user for reading via the newsreader. By using the newsreader, the user can continue to indicate their interests and the profile is continuously updated.

Rule Base Design

As discussed by Mehrotra and Johnson, we agree that rule base design is an important factor in the scalability of an expert system. Their work in using an expert system to automate the grouping of rules within a rule base is interesting and useful (Mehrotra&Johnson 1990). We propose, however, that rule base design and rule groupings should be established in the design phase of the expert system development to provide maximum benefit from the rule groupings. With this in mind we designed the AIRS rule base.

User Profiling Rules
- parse files related to the user - adaptive keyword filter - develop user profile
Search Tactics Rules
- formulate queries - evaluate queries - rank articles for user consideration
Command and Control Rules
- distribute search engine workload over network - provide control elements to rules - interface with newsreader

Table 1. Rule base Structure

The AIRS rule base is organized into three general classes of rules. This organization was chosen not only to facilitate rule writing, but also to improve the extensibility of the rule base to other applications. For example, the first of these classes are User Profiling Rules, rules which develop a profile of the user. As discussed in the project overview, there are several files created by the newsreader which contain information about the user's news interests. The User Profiling Rules access these files, extract and filter the

information, then build the user profile. These rules, as initially implemented, appear to be very specific to the news domain. However, one would expect most information about a computer user to be stored in a file. Given this assumption, then any profiling of a user will be a process of accessing files, extracting and then filtering the information. It is expected then, with minimal modification, these rules could be extended to point at a different type of file, extract and filter that information and then build a profile of a user's interests in the new area.

Table 1 depicts the structure of the rulebase and shows some of the types of rules which may be found within a class. Rules in the search tactics class develop and evaluated the queries which are sent to the search engine for processing. The heuristics for developing and evaluating these queries have been previously validated and draw heavily upon work in library science (Bates 1979). A similar search tactics rulebase implemented in OPS5 has provided an excellent example, and many ideas for the search tactics rules (Gauch 1991). Rules within the search tactics class are intended to be immediately extensible to any type of query generation. The only modification required would be format changes due to using different search engines.

The third class of rules are the control and interface rules. These rules provide structure and execution order to the rulebase, and act as an interface to the operating system.

CLIPS Client/Server

Several of the modules of the newsreader system are anticipated to consume considerable system resources. The search engine will initially be very I/O bound while it constructs a reverse index, and then it is anticipated to swing over to being a heavily cpu bound process. AIRS places a heavy load on system memory resources as well as I/O due to the need to parse many files and then perform the appropriate pattern matching functions. To compensate for this, AIRS takes advantage of the Mach Operating system, and runs as a client/server based distributed system. Figure 3 depicts the current configuration of the *arn - adaptive news reader* with the AIRS component.

A client which serves as a triggering device is initiated by a rsh shell and an accompaning shell script. Once the client ensures that the parameters for low use time and minimal load have been met it sends the server which is blocked on receive. Our production model will migrate several of these functions to the chrontab but for prototype implementation it was simpler to use the rsh/client combination. When the clips server receives the message, it fetches the AIRS rule base, and begins profiling the user. Currently it is anticipated that AIRS will run on only one machine on the network. The AIRS component itself, when it has a query to send to the search engine, will send a remote procedure call to a machine on the net, to instantiate the search engine and run the query. As the search engine is the heaviest drain on system resources, the expert system will sequence through the various available resources, assigning a job to each, continuing to sequence through the resources until all of the jobs have been assigned. In the interest of efficiency, the platform running the AIRS component will not receive a search engine job. Developing a set of rules which would examine use and load of the the various platforms on the net was considered and rejected. By the time the evaluation was made and the job was assigned, the situation could have changed significantly, invalidating the initial findings.

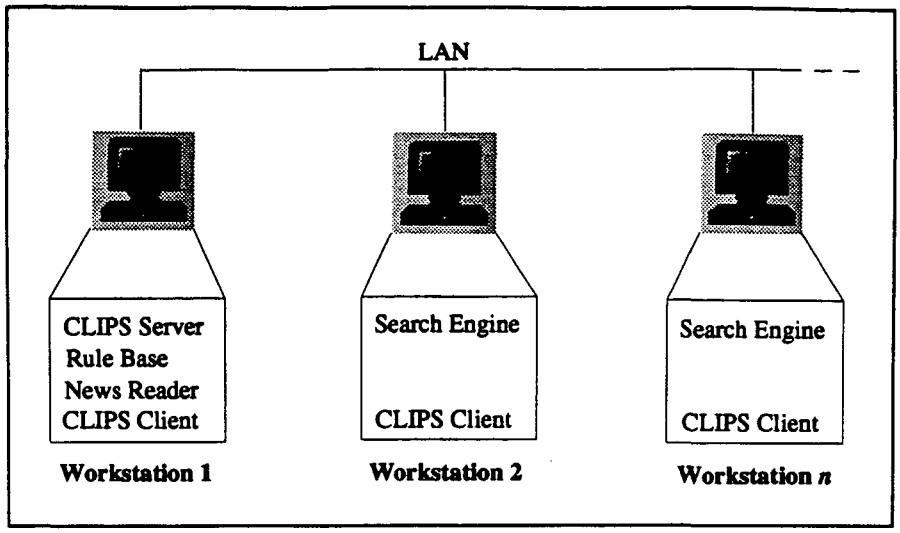

Figure 3. Newsreader configuration as a distributed network

AUTOMATED INFORMATION RETRIEVAL SYSTEM METHODOLOGY

Automated information retrieval methology in AIRS consists of developing a profile of the user, searching the newsbase for articles of interest, evaluating the articles returned, if necessary continuing the search, and finally ranking the articles and presenting them to the user via the newsreader. Figure 4 is a depiction of this process using a Petri Net. We find Petri Nets to be an excellent tool for modeling the behavior of an expert system and plan on validating and expanding upon previous work in this field (Etessami&Hura 1991). The first step in the AIRS process is user profiling.

User Profiling

User profiles are developed with keywords located in several files which are modified while a user reads the news. The newsreader creates two files which profile the newsreading habits of a user. The first of these is the .tindx file. The .tindx file is a linked list of news article "threads." A thread is an article with all of the responses generated by the article. Threads provide themes of interest within a newsgroup. The original *tass* newsreader has been modified to create a second file, the .keykill file. This file contains a reference to articles and threads which have been killed. This indicates a lack of interest in a news item or theme. A third file which indicates a user's interest is called the .keysrceng file and is created by the search engine. This file contains information pertaining to previous keyword searches over the newsbase which have been conducted by the user.

The profile of the user's interest is constructed using a weighted keyword scheme. First all three files are parsed and piped through a stop-list. The stop-list is an adaptive filter which eliminates keywords with a low discrimination ability. These words are kept in a .keyfilter file. Initially the .keyfilterfile is populated with a list of words which either are non-discriminate because of the English usage or are non-discriminate because of

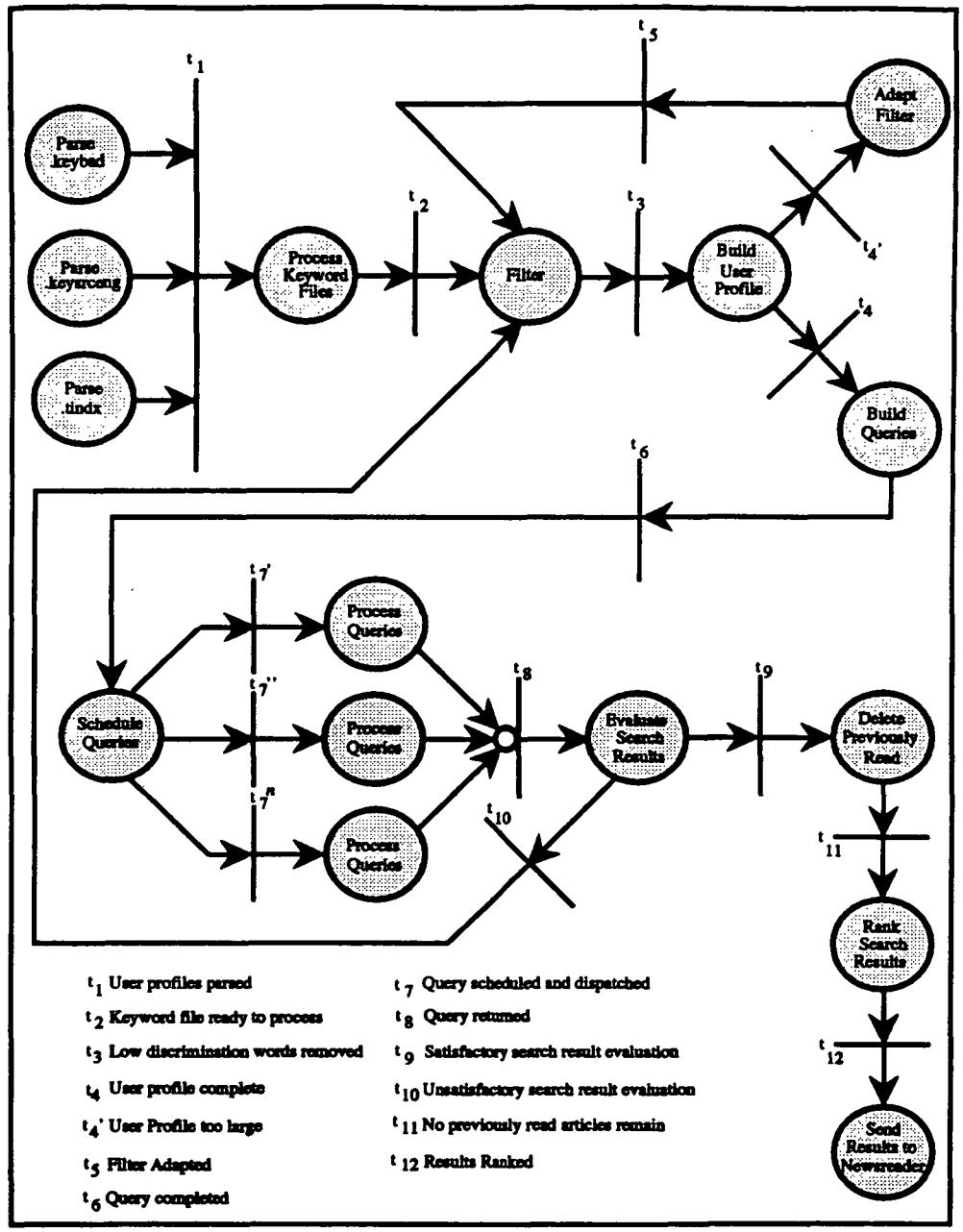

Figure 4. Petri Net representation of the AIRS prototype.

their relation to the news. Table 2 provides a sample of some of these words. It is interesting to note that many words fall into both catagories. The filter is adaptive in that as AIRS encounters words with low discrimination ability (which correspond to numerous instances of the word) the word is added to the .keyfilter file. The number of instances of a word which can exist without sending it to the .keyfilter file is the.keyfilter cutoff variable and may be adjusted if production is too high. Punctuation and contractions are also filtered out at this point.

Sample Non-discriminating Words	
English Usage	News Usage
if while the she at it he and	Date Subject Organization Message From Re: Lines Keyword Path

Table 2. Sample Non-discriminating Words

After an initial keyword list is developed, a weighting and elimination scheme is used to develop the profile of the user. In the initial prototype, a simple algorithm is used. Keywords from the .tindx file are given a weight of 1. Keywords from the .keykill file are given a weight of -2 and keywords from the .keysrceng file are given a weight of +2. The weights are given at each occurence in the file. The user profile is calculated by summing up the weights and ranking the keywords. A cutoff score is determined and keywords above the cutoff are sent off to the search engine.

Query generation and evaluation

The AIRS component prepares a query to send to the search engine after the user profile has been developed. This query is in the form of a single keyword search. Although the lqtext based search engine is capable of locating word combinations, this has been left to future iterations of the AIRS component. AIRS uses a round robin-scheduler to determine which computer on the network will receive the search for processing. The search query is bundled by the expert system into a remote procedure call (rpc) and sent to the appropriate network resource. When the query is received by a computer, the search is initiated and the results are returned to the platform running the expert system. The results are then collated, duplicate are articles deleted, and production is evaluated. If too few articles are returned, the total cutoff score will be lowered and new searches will be initiated. If too many articles are returned, the .keyfilter cutoff variable will be lowered, the.keyfilter file adjusted, and the articles returned will be re-evaluated. If an appropriate number of articles are returned, they will be examined to ensure that they have not already been read. Articles which the user has never seen will be ranked from most to least interesting and sent to the newsreader module for reading by the user.

RESULTS

Most of our initial objectives have been met in the prototype newsreader. Implementation time was exceptionally fast: a single-reader, single-machine configuration was prototyped in less than 1 programmer month. This is particularly significant given that the expert system programmer had no CLIPS or other rule-based programming

experience. As we expected, we have had no difficulty porting CLIPS to other computers. In fact, as is commonly done at Cal Poly, most of our rules were developed on IBM PC based machines, at home, and ran with no changes when transferred to the NeXT computers at school. Compiling a CLIPS client/server for the project was particularly easy, as was developing the interface to the Mach Operating System.

Use of *arn - adaptive read news* with the AIRS component is intended to supplement the normal reading habits of the user. Accordingly, an increase in news reader efficiency is predicted as readers will not only be able to continue their normal reading, but they will also benefit from the information provided by the expert system. Our newsreader/ expert system combination should provide a user with access to a significantly greater portion of the newsbase yet still reduce the amount of time a user spends reading the news. The second area of performance we consider is that of the traditional information retrieval measures of recall and precision. The actual performance of the expert system in terms of speed and efficiency are not considered in the evaluation of the prototype. As the prototype was designed to run in background during non-peak use periods, the performance of the expert system was not deemed critical to the prototype. In later production models, this will become an important issue.

Future Plans

Refinement of the Automated Information Retrieval System will continue using a stepwise development methodology. Current plans call for an upgrade in the search engine to full boolean search capability. Expansion and integration of more complex search tactics will continue, as will the effort to further distribute the system to improve efficiency. In its current form, the AIRS component relies solely on keyword-based heuristics. Additional user profiling heuristics will be added in later iterations. Distributed computing issues will be considered in hopes of improving performance. And as stated in our objectives, we intend to port the arn newsreader to other computer systems, including IBM PC's and Sun workstations.

The development of the Automated Information Retrieval System and *arn -adaptive read news* is the initial phase of an advanced information retrieval project, NewsClips, an intelligent system for the Automated Filtering of Time-Sensitive Information. NewsClips is envisioned as a distributed bulletin board system, running in a client/server configuration and written in CLIPS. The NewsClips system will evaluate incoming information in real time and, based on previously determined user profiles, will discard information of no value. It is intended as a testbed to determine whether sufficient confidence in an automated system can be achieved to allow it to discard information deemed of little value without human intervention.

Acknowledgements

This project is funded in part by the TRW Foundation. We would like to acknowledge the assistance of the Computer Systems Lab and the Advanced Workstations Lab at Cal Poly, and the following individuals for their assistance; Don Erickson, Ellen Clary, Peter Ogilvie, and Ding Yan.

References

(Bates 1979)
Bates, Marcia. "Information Search Tactics." *Journal of the American Society for Information Science* 30 (July 1979): 205-214.

(Etessami&Hura 1991)
Etessami, Farhad S. and Hura, Gurdeep S. "Knowledge Net Shell (KNS): Petri Net Based Development Tool for expert systems." *Microelectrtonics and Reliability* 31 4 (April 1991): 793-812.

(Gauch 1991)
Gauch, Susan Evalyn. "An expert system for Searching in Full-Text." Ph.D diss., University of North Carolina, 1991.

(Jacobs&Rau 1991)
Jacobs, Paul S. and Rau, Lisa F. "SCISOR: Extracting Information from On-line News." *Communications of the ACM* 33 (November 1990): 88 - 97.

(Mehrotra&Johnson 1990)
Mehrotra, Mala and Johnson, Sally C. "Rule Groupings in Expert Systems." *First CLIPS Conference Proceedings.* Houston: NASA, 1990, 274-285.

(Quam 1990)
Quam, Liam. "Lq-text"

(Reid 1991)
Reid, Brian. "Usenet Readership summary report for Jul 91." Unpublished electronic news article, August 1991.

(Robinson 1991)
Robinson, Mike. "Through a Lens Smartly." *BYTE* 16 (May 1991): 177-187.

(Skrenta 1990)
Skrenta, Rick. "Tass, rtass - Visual threaded Usenet news reader".

(Wyle&Frei 1991)
Wyle, M. F. and Frei, H. P. "Retrieving Highly Dynamic, Widely Distributed Information" *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval.* Association for Computing Machinery: New York, 1989 pp 108-115.