Expert Retrieval for Computer Message Systems

by
Edward A. Fox

June 1986

TR-86-13

EXPERT RETRIEVAL FOR COMPUTER MESSAGE SYSTEMS

Edward A. Fox
Dept. of Computer Science, Virginia Tech
Blacksburg, VA 24061

ABSTRACT

This paper describes how information storage and retrieval (IS&R) and artificial intelligence (AI) methods can be integrated with modern computers and networks to provide access for broad classes of users to archives of electronic: mail, digests, and bulletin boards. A status report is given on the COmposite Document Expert/extended/effective Retrieval project, designed to employ multiple communicating computers for free text analysis, indexing, and retrieval.

The aim of CODER is to make IS&R systems more intelligent, so that a greater share of the information present in society can become directly accessible to casual users. With the worldwide proliferation of computer based message systems, large numbers of heterogeneous messages aid the immediate communication needs of users. However, lack of adequate software has limited access to the accumulation of information in message archives. The first application of CODER is to allow search through roughly four years of AIList Digest issues distributed over the ARPANET. The approach is to automatically analyze, categorize, and index each message so that the resulting knowledge representation can be searched by users submitting either natural language or Boolean queries. A variety of interfaces will support different styles of browsing and search, guided by a knowledge base describing each person who is allowed access to the system. In order to provide a fine grain of rapid human-computer interaction, CODER will integrate document feedback, term relation, and browsing approaches according to a rule-based paradigm.

The CODER system has an entry/analysis/indexing subsystem, a user interface/search sysbsystem, and a central "spine" including two external knowledge bases: a lexicon based upon machine readable English dictionaries, and a document database including text and surrogate representations. Both subsystems are composed of a community of "experts," a blackboard, and a coordinating strategist. Details of the document-type expert are included to illustrate the approach.

CR Categories and Subject Descriptors: C.2.3 **[Computer Communication Networks]:** Network Operations; H.3.1 **[Information Storage and Retrieval]:** Content Analysis and Indexing; H.3.3 **[Information Storage and Retrieval]:** Information Search and Retrieval; H.3.4 **[Information Storage and Retrieval]:** Systems and Software; H.4.3 **[Information Systems Applications]:** Communications Applications – *electronic mail;* I.2.1 **[Artificial Intelligence]:** Applications and Expert Systems

General Terms: Design, Human Factors

Additional Key Words and Phrases: Composite documents, computer based message systems, user interface, document analysis

INTRODUCTION

The COmposite Document Expert/extended/effective Retrieval project is a multi-year effort to determine the utility of artificial intelligence (AI) methods for improving the effectiveness and efficiency of information storage and retrieval systems [2]. Of particular interest is the handling of "composite documents" which are commonly found in the context of computer based message systems [3]. These documents vary widely in length and content, and may have a complex structure which is often difficult to recognize. As a consequence, issues of storage representation and subsequent retrieval are particularly important to consider [4].

Since the late 1970's, many AI practicioners have investigated the use of "expert systems" for a variety of practical applications, and certain methods employed in developing expert systems have been refined [5]. One early expert system for document retrieval assisted a search intermediary interacting with a large number of online commercially available databases [6]. A more recent system allows users to construct elaborate queries that are actually sets of rules for detecting suitable documents [7]. Current work at the University of Massachusetts at Amherst, however, is the closest effort to CODER [8]. While there are many similarities, CODER is more focused on: natural language analysis, utilization of a comprehensive lexicon, building a variety of complex document representations, working in a distributed environment, handling composite documents, and employing logic programming techniques.

This report describes the application domain of computer based message systems, and provides an overview of the CODER project. The design and architecture of CODER are discussed, with particular attention given to the user interface. Some specific details are provided regarding the CODER expert for classifying documents as to type. Finally, plans for further development of the CODER system are summarized.

IS&R FOR COMPUTER BASED MESSAGE SYSTEMS

Computer based message systems are collections of computers connected through some type of physical network or networks, which have software to enable interchange of messages. Most commonly, electronic mail messages are transmitted from one user's mail handling program (e.g., mailer or user agent (UA)), to routines for decoding and "munging" addresses, to a local mail transport agent (MTA), out through one or more networks via a succession of MTAs, and eventually to the recipient's UA [9]. Small to moderate size files can be transferred as mail messages in this way. In addition, a number of news, bulletin board, or electronic digest services are available to effect broadcast modes of transmission; AIList Digest deals with AI, and IRList Digest focuses on information retrieval as well as other related information science topics [10].

In recent years there has been an explosive growth in networks supporting electronic mail and other services. The oldest packet switched network, ARPANET, has split several times and is now being further extended through the creation of NSFNET by the National Science Foundation, which hopes to thereby stimulate use of supercomputers. BITNET, aimed at the educational community, has spread to Europe and the Far East. UUCPNET, connecting mostly systems that run the UNIX™ operating system, has thousands of machines throughout the world. CSNET, serving the computer science community, has grown from a handful to roughly 200 sites, several of which are interested in enhancing networking software. In February 1986 representatives of each of these non-commercial networks met and agreed to adopt standards for electronic addresses that will greatly facilitate intra- as well as inter-network communication [11].

There are three principal areas where information storage and retrieval techniques can be employed to aid the operation of these computer based message systems. These are: filing, filtering, and finding of either messages or names and addresses. Specifically, items can be filed (e.g., indexed) for later retrieval, compared against profiles for people or for topics in order to rank or set priorities or allow discarding of superfluous materials, and eventually retrieved. While existing UAs have some of these abilities, they generally do not employ any sophisticated retrieval methods. One of the best UAs in terms of Boolean retrieval capabilities is the now obsolete Hermes system [12]. In BITNET, a simple system for limited search through electronic mail digests has recently be announced [13]. For CSNET, a name server with a variety of uses but with a somewhat awkward command syntax has been available for several years [14] and an "information server" has recently been added.

One of the reasons for developing CODER is to validate the use of advanced retrieval methods for network applications. While testing with data from the CSNET name server is planned, of even greater interest is the filing of electronic mail digests and finding of relevant passages in the resulting heterogeneous collection. In this regard, all issues of the AIList Digest, distributed over the ARPANET since March 1983, have been captured for analysis as composite documents.

In order to properly analyze and represent the messages found in each digest, it is necessary to have a fitting model for documents. In the office domain, the Office Document Architecture model has been proposed [15]. In building a hardware/software system for analyzing and formatting documents, Peels et al. provide grammars for three "information streams" that are typically present [16]. Croft and Pezarro focus on the retrieval problems associated with handling general office documents [17]. More closely related to the situation in networks, however, is the work of Babatz and Bogen, where embedding and referring characteristics of messages are highlighted and where the "kernel" of a message can be identified and compared with other kernels [18]. Given one of these models, however, it is still a difficult problem to analyze complex composite documents.

OVERVIEW OF CODER

The CODER system is a large system involving, at times, roughly a dozen developers. It can be viewed from many different perspectives. The subsections below will give a brief overview of some of the more important aspects, especially those relating to computer message systems and user interfaces.

Design

A summary of the design issues and decisions made in developing CODER is given in [19]. However, there are a number of key points that should be made at this time in the current context. First, there is the matter of perspective on retrieval. Based on the investigation described in [20], several observations can be made:

1) Depending on the user's background and training, there is value in supporting vector, probabilistic, and extended Boolean retrieval methods.
2) Feedback is an important way to improve each of the above-mentioned query formulation approaches. It should be possible to make further improvements in feedback if graphical interfaces are employed that allow the user to comment on relevance of a part of a single document.
3) When a variety of different types of information is available about documents and their relationships, retrieval can be improved if the contribution of each part is intelligently considered.
4) Since relatively good performance is possible when computations are based on statistics, even better performance should result if a finer case-by-case analysis is employed, based on knowledge about individual documents, users, and words.
5) More detailed information about lexemes and phrases should be of value in word sense disambiguation and in feedback query construction.

Another key design issue in developing CODER is that of human-computer interaction. It is assumed that a rapid interactive dialog is most appropriate [21]. The overall goal is for each system user to be able to browse, search, perform feedback, manipulate sets of query terms – all based on up-to-the-moment information. The interface must also be intelligent – in that characteristics of each user, obtained by inquiry or through an observation/learning process, should be stored in a user model knowledge base.

Implicit in having an intelligent adaptable interface is the requirement of code modularity. This attribute is particularly important since a number of people are developing different parts of CODER. With modular code, future experimental studies will be more easily conducted. Modularity is achieved in CODER as follows:

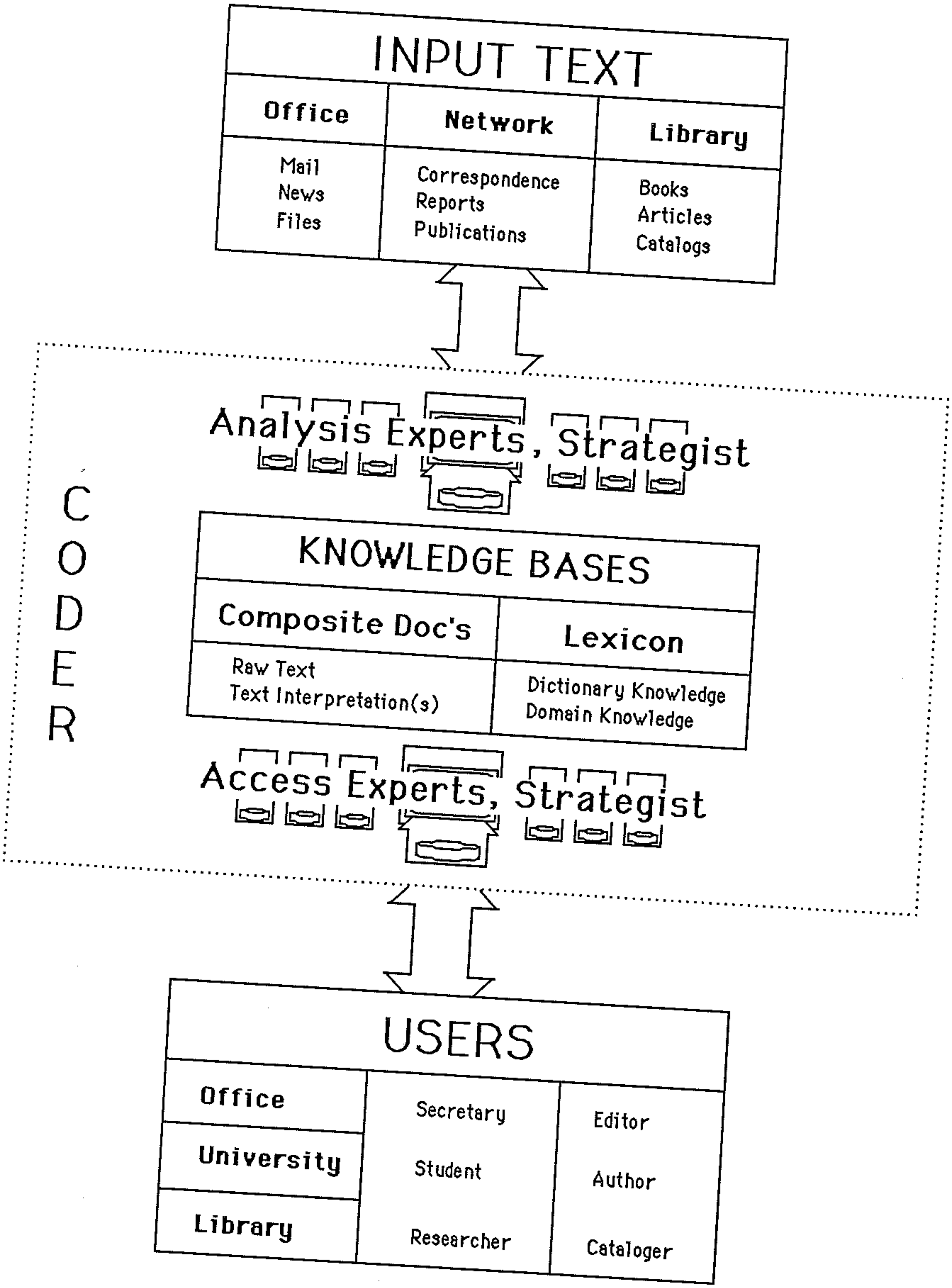
1) The system is in reality a collection of processes, possibly running on more than one computer, which communicate through message passing. Many of the processes are small expert systems; this parallels the simulation set up by Belkin et al. in [22].
2) When processes communicate between machines, the current DoD standard protocols, TCP/IP, will be used [23]. On a single processor, pipes can be employed as provided by the UNIX™ environment presently used.
3) Processes interacting with users or large databases, where special services and/or performance is needed, can employ the C++ language, which adds classes and other tools encouraging object oriented programming to the C language [24].
4) Expert modules each have separate declarative knowledge needed for a particular type of problem. Developing these experts is best done in Prolog [25]. Since large databases are involved, a special version of Prolog adapted to rapidly process tens of millions of facts is employed [26]. Using Prolog supports a rule based approach to capturing expertise [27].
5) Knowledge is also modular in form. Both frames and relations are supported and can be arbitrarily nested and intermixed. By supporting simple structures with straightforward inheritance rules, reasonable efficiency and clear semantics result, in similar fashion to the ARGON system [28].

Of special importance in the design of CODER is the availability of a comprehensive lexicon. Evens and Smith demonstrated the value of a good lexicon to support question answering and other AI tasks [29]. Several machine readable dictionaries have been made available in recent years [30], so it was decided to obtain as many as possible to use in lexicon construction. Since natural language analysis is one of the tasks CODER is to perform, it was considered necessary to obtain a lexicon comparable or larger than that used in such comprehensive systems as LSP [31]. While it is to be expected that any parser would use the lexicon, regardless of what grammar and interpretation scheme is employed, such an approach does not preclude use of fast, relatively robust parsers (e.g., [32]).

User Interfaces

As can be seen in Figure 1, CODER is targeted as a system to support a variety of types of input and a diversity of users. Typical data sources would be offices, networks, or libraries, each of

FIG. 1: Data and Users



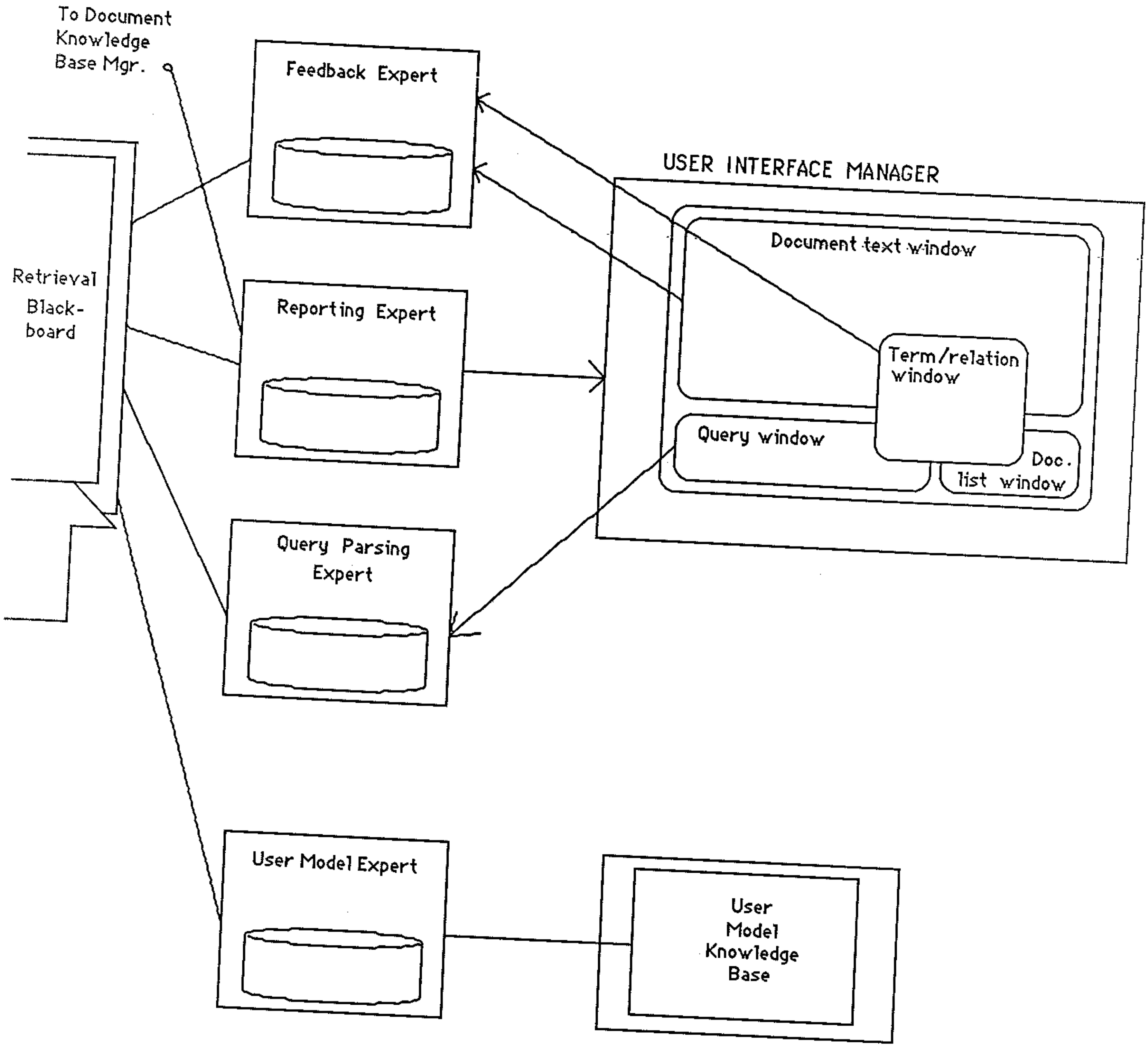
which has its own breakdown of document types. Users, taken from a general office, part of a university, or from the community which inhabits a library, would represent many different occupations. To help each individual user locate relevant materials, CODER has passive knowledge bases and active experts or strategists.

CODER is made up of a central spine (which includes the lexicon and document knowledge base), and two subsystems. Users may occasionally interact with the analysis subsystem, especially to add new entries to the collection. However, they will typically spend most of their time conducting searches with the aid of the retrieval subsystem (see Figure 2). Control of the retrieval function rests with a Retrieval Strategist, which monitors the Retrieval Blackboard, the entity that coordinates message passing among the various retrieval experts. Communication with the user is controlled by the User Interface Manager, which in turn interacts, possibly through the blackboard, with the several experts involved in query submission, display of results, and feedback.

At present, two different C versions of the User Interface Manager have been developed, one for the SUN Microsystems Inc. SUN-2/170 workstation on which the lexicon is being constructed, and the other using the CURSES package to make it adaptable to a variety of terminal types. In both cases, a searcher can examine and manipulate entries in at least four windows, for: a list of retrieved and ranked documents, the text of the selected document, the original query, and information about terms and their relationships. These interfaces will be further developed, and others provided as well, when additional types of interaction devices become available.

Very little controlled experimentation has been done with interfaces of this type. Most of the researchers studying user interfaces for retrieval systems have assumed use of Boolean searches, where no ranking of output and no automatic feedback is employed [33]. The TOPIC system, which uses word-expert parsing methods to try to create a hierarchical topic structure from document texts, something that might be done by CODER, is another example of a system with a radically different range of information that could be displayed [34]. Alternatively, if the aim of the display is to effect a type of passage retrieval [35] by presenting the original text of a document, then use of highlighting (e.g., inverse video) by the system, and user responses of turning on (or off) highlighting in selected areas, are most appropriate. If the focus is on question answering [36], text generation rather than elaborate display is the priority. Finally, if the focus is on browsing, or extending a query through term associations, a stimulating environment similar to the dynamic encyclopedia of Weyer & Borning [37] is desired. Based on studies by Katzer et al. [38], it appears that best performance should result from searching which merges the results of a variety of approaches and representation schemes.

FIG. 2: Retrieval Subsystem: User Interface



Architecture

In addition to the user interface component of CODER mentioned above, there are a number of other distinctive architectural features of the system that must be described. For additional detail, the reader is referred to [39].

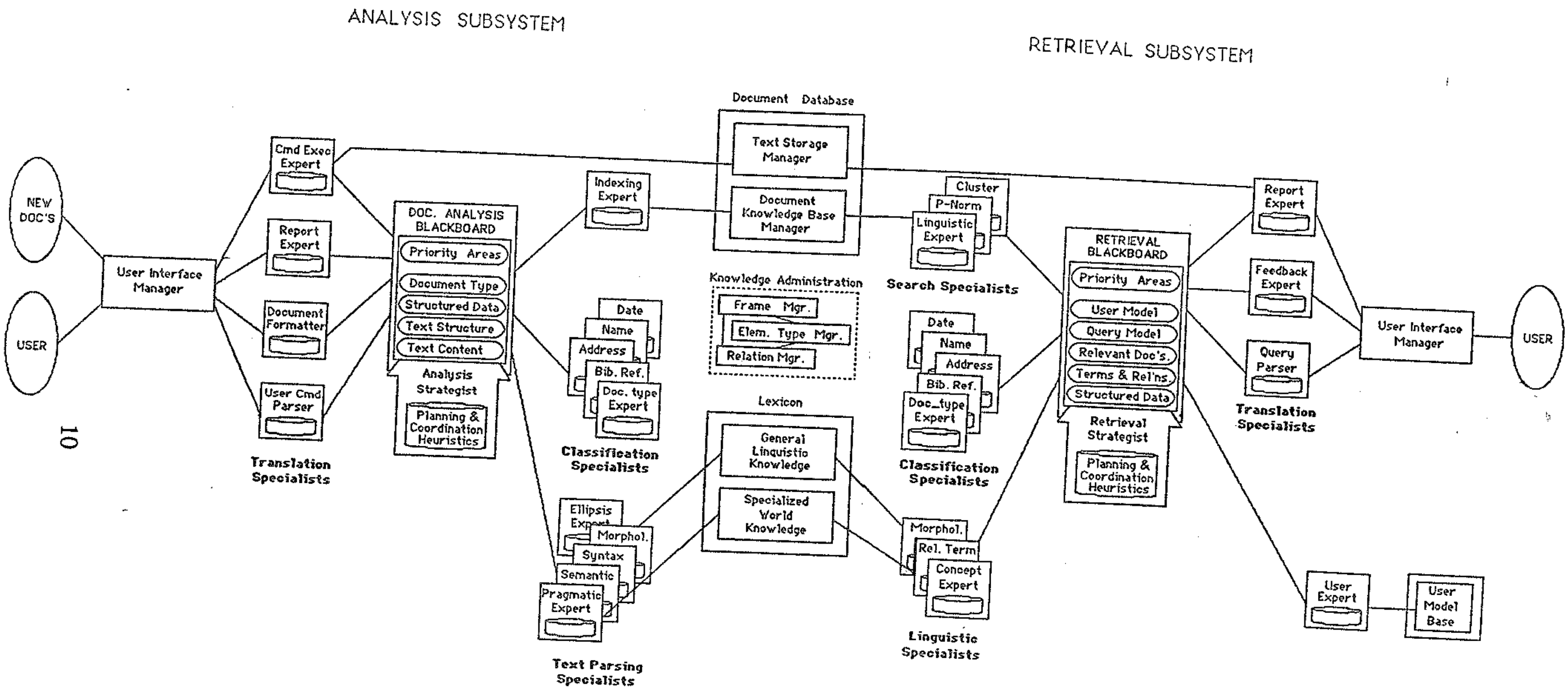
CODER is in reality two systems interacting through the Spine and its knowledge bases (see Figure 3). Both the analysis and retrieval subsystems are implemented as a community of experts and managers communicating with the aid of a blackboard. Blackboards have prove to be very useful in systems like Hearsay-II, which performs a task in many ways similar to that of CODER [40]. For CODER, a transactional model of the blackboard [41] is particularly appropriate due to the use of numerous (possibly changing) data sources. The active strategist attached to each blackboard has a generic part to handle communication and scheduling tasks, as well as specialized knowledge relating to the desirable interactions of experts it controls.

As mentioned earlier, CODER employs relations and frames to build up knowledge representations. A variety of schemes have been proposed for use in information retrieval systems [42]. Frames are particularly valuable as structures to organize and bring together information about various objects, as well as to specify control schemes [43]. Representing temporal information is exceedingly difficult in free text situations [44], but can be somewhat controlled through the uses of frames and of relationships among frames. Use of such higher level structures raises the level of representationship from atomistic terms to more meaningful "concepts" [45]. Many document texts are really elaborations of standard "schema" [46] which may appear at various levels and can be represented with the help of frames.

As can be seen in Figure 3, CODER has a large number of components, each responsible for a critical part of the system's operation. This is illustrated at the level of a single component in the following section.

CODER DOCUMENT TYPE EXPERT

Based on a preliminary analysis of roughly 1000 AIList messages, a variety of observations were made regarding heuristics needed for automatic analysis. Following subsections have partial information to summarize some of the features observed, give a taxonomy of message intent, report on message structure, provide rules for message type identification, and present a high level algorithm for message analysis.



Overview of the CODER System

Figure 3

Important Features

Length: varies from 1 sentence to several pages. The number of lines is a good indicator of message type. Thresholds TOP and BOT are needed as well to indicate how many lines at the top and at the bottom are to be specially scrutinized for clues.

Retransmitted?: did this originate on AIList or was it forwarded from another bulletin board or network? Clues are constructs like

[Forwarded from the <institution name> bboard by <name>]

References to: may cite date or person. Often have indented fields or lines looking like

>lines of text

Bibliographic and other citations: can check by looking up names of journals and publishers in reference lists.

Intent (partial list)

1) Announce a forthcoming event.
   - a- Seminar, conference, course
   - b- Call for papers
2) Information exchange
   - a- Request for information
   - b- Provide information
   - c- Offer to provide information on request
   - d- Correct a factual error in previous msg
3) News
   - a- New AI product, source
   - b- Survey of articles/journals/books
   - c- Reprint or summarize article
   - d- Report on past conference or symposium
4) Work in progress / exchange of ideas
   - a- Detailed seminar announcement
   - b- Project abstracts
   - c- Presenting an algorithm, idea or opinion
   - d- Discussion after one of above
5) Humor - joke, anecdote, parody

Message Structure

Each Digest has a table of contents. On the left are "Header1" fields which are one or two word high level classes, such as "Seminar." On the right, in Header2, are longer phrases, which typically copy or summarize the Subject or other key part of a message that follows.

Each message has a header section with fields: Date, From, Subject, and possibly some others such as Reply-to. The content of the Subject field is herein called Header3. Following is the message body. There are often subheads in the body, identified by surrounding blank lines, or lines of equal signs or hyphens, or hyphens on either side of the subhead text. Some subheads may be in all caps, or labelled and followed with a colon as in “Location:”

Heuristics for Identifying a Common Message Type

CALL FOR PAPERS: Header1 as “Conference” or “Symposium”- necessary. Header3 of "Call" is sufficient. “Call for Papers” as subhead is sufficient. Common subheads are: Conference title, Sponsoring organization, Place, Date. When “Conference” appears in Header1 then submission requirements in top essential region of the message is sufficient. Key phrases indicating this are: (must/should) be (submitted/sent/received); must (send/include) <n> copies; no later than <date>; to <person's name and address>.

Basic Analysis Algorithm

- assign the correct header pair to message
- count lines and identify key top and bottom areas
- scan each line for special indicating phrases
- fill in slots in all frames suggested as suitable

STATUS AND FUTURE PLANS

The CODER Project has simultaneously focused attention on a multitude of questions relating to intelligent processing of composite documents. A lexicon has been constructed from one full English dictionary with more than 80,000 headwords and can be loaded as a large Prolog fact base. Two user interfaces have been programmed in C. System design has been completed and important predicates have been specified.

Work is proceeding to develop grammars like that built by Ahlswede (for describing the dictionary definitions of adjectives [47]); these will lead to further analysis of the dictionaries to identify additional lexical/semantic relationships.

It is expected that during the later part of 1986 an initial version of CODER will be complete, enabling comparisons to be made with the performance of the SMART retrieval system on the AIList test collection. It is expected that CODER will continue to be used as a teaching and research tool for many years to come.

NOTES

(1) This work was supported by the NSF under grant IST-8418877 and by Virginia's Center for Innovative Technology under grant INF-85-016.

(2) Edward A. Fox, "Composite Document Extended Retrieval: An Overview," *Res. & Dev. in Inf. Ret., Eighth Annual Int. ACM SIGIR Conf.*, Montreal, (June 1985) 42–55.

(3) Edward A. Fox, "Composite Document Search in a Networked Society," *Telecommunications & Networking: supplying the missing link,* Collected papers of the Amer. Soc. for Inf. Sci. 14th Mid-Year Meeting, (May 19-22, 1985) Fiche 11.

(4) Edward A. Fox, "Analysis and Retrieval of Composite Documents," *ASIS '85, Proc. 48th ASIS Annual Meeting,* (Oct. 1985) 54-58.

(5) F. Hayes-Roth, D.A. Waterman, and D.B. Lenat, eds. *Building Expert Systems,* (Reading, MA: Addison-Wesley, 1983).

(6) Man-Kam Yip, "An Expert System for Document Retrieval," (M.S. Thesis. M.I.T., 1979).

(7) R.M. Tong, et al., "A Rule-Based Approach to Information Retrieval: Some Results and Comments," *Proc. AAAI-83,* 1983.

(8) R.H. Thompson and W.B. Croft, "An Expert System for Document Retrieval," *Proc. Expert Systems in Gov. Symp.*, IEEE, (Oct. 1985) 448-456.

(9) T.H. Myer, "Standards for Global Messaging: A Progress Report," *J. Telecommunication Networks*, 2(4) (Winter 1983).

(10) IRList Digest can be subscribed to by sending a letter or message to this author.

(11) Craig Partridge, "Report from the Internet NIC on Domains," *CSNET Forum*, 2(2) (Feb. 1986).

(12) C.D. Mooers, "The Hermes Guide," (Report No. 4995, BBN, Inc., Aug. 1982).

(13) Harry Nussbacher, "Announcement of Expanded Internet Database Services," *IRList Digest,* 1(15) (Oct. 1985).

(14) M. Solomon, L.H. Landweber, and D. Neuhengen, "The CSNET Name Server," *Computer Networks 6,* (North–Holland, 1982).

(15) W. Horak, and G. Kronert, "An Object-Oriented Office Document Architecture Model for Processing and Interchange of Documents," *Proc. Second ACM-SIGOA Conf. on Office Information Systems* , (June 25-27, 1984) 152-160.

(16) Arno J.H.M. Peels, Norbert J.M. Janssen, and Wop Nawijn, "Document Architecture and Text Formatting," *ACM Trans. Office Inf. Sys.*, 3(4) (Oct. 1985) 347-369.

(17) W.B. Croft and M.T. Pezarro, "Text Retrieval Techniques for the Automated Office" *Office Information Systems,* ed. by N. Naffah, (Amsterdam: North- Holland, 1982) 565-576.

(18) R. Babatz and M. Bogen, "Semantic Relations in Message Handling Systems: Referable Documents," *Proc. IFIP WG 6.5 Symposium,* (Sept. 1985).

(19) E.A. Fox and R.K. France, "A Knowledge-Based System for Composite Document Analysis and Retrieval: Design Issues in the CODER Project," (TR-86-6, Virginia Tech Dept. of Comp. Sci., March 1986).

(20) Edward A. Fox, "Extending the Boolean and Vector Space Models of Information Retrieval with P-Norm Queries and Multiple Concept Types," (Dissertation, Cornell University, University Microfilms Int., Ann Arbor MI, Aug. 1983).

(21) R.N. Oddy, "Information Retrieval Through Man-Machine Dialogue," *J. Doc.*, 33(1) (March 1977) 1-14.

(22) N.J. Belkin, R.D. Hennings, and T. Seeger, "Simulation of a Distributed Expert-Based Information Provision Mechanism," *Inf. Tech.: Res. Dev. Applications,* 3(3) (1984) 122-141.

(23) Samuel J. Leffler, Robert S. Fabry, and William N. Joy, "A 4.2BSD Interprocess Communication Primer," *ULTRIX-32 Supplementary Documents,* 3 (March 1984).

(24) Bjarne Stroustrup, *The C++ Programming Language,* (Reading, MA: Addison-Wesley 1985).

(25) A.R. Helm, Kimbal Marriott, and Catherine Lassez, "Prolog for Expert Systems: An Evaluation," *Proc. of Expert Systems in Government Symp.*, (October 1985) 284-293.

(26) L. Naish, *MU-Prolog 3.1db Reference Manual.* (Melbourne Univ., 1984).

(27) F. Hayes-Roth, "Rule-Based Systems," *Commun. ACM,* 28(9) (Sept. 1985) 921-932.

(28) P.F. Patel-Schneider, R.J. Brachman, and H.J. Levesque, "ARGON: Knowledge Representation meets Information Retrieval," (Fairchild Technical Report No. 654, FLAIR Technical Report No. 29, Sept. 1984).

(29) M.W. Evens and R.N. Smith, "A Lexicon for a Computer Question-Answering System," *Am. J. Comp. Ling.*, (1979) Microfiche 83.

(30) R.A. Amsler, "Machine-Readable Dictionaries," *ARIST*, 19 (1984) 161-209.

(31) C. White, "The Linguistic String Project Dictionary for Automatic Text Analysis," *Workshop on Machine Readable Dictionaries*, SRI, Menlo Park, CA, May 1983.

(32) G. DeJong, "An Overview of the FRUMP System," *Strategies for Natural Language Processing*, ed. by Wendy G. Lehnert and Martin H. Ringle, (Hillsdale, NJ: Lawrence Erlbaum Assoc., 1982) 149-176.

(33) Christine L. Borgman, "Psychological Research in Human-Computer Interaction," *ARIST*, 19 (1984) 33-64.

(34) U. Hahn and U.Reimer, "Heuristic Text Parsing in 'Topic': Methodological Issues in a Knowledge-based Text Condensation System," *Representation and Exchange of Knowledge as a Basis of Information Processes*, ed. by Hans J. Dietschmann, (New York: North-Holland, 1984) 143-163.

(35) J. O'Connor, "Answer-Passage Retrieval by Text Searching," *J. Am. Soc. Inf. Sci.*, 31(4) (1980) 227-239.

(36) W.G. Lehnert, *The Process of Question Answering: A Computer Simulation of Cognition,* (Hillsdale, NJ: Lawrence Erlbaum Assoc., 1978).

(37) S.A. Weyer, and A.H. Borning, "A Prototype Electronic Encyclopedia," *ACM Trans. on Office Information Systems,* 3(1) (January 1984) 63-68.

(38) J. Katzer, et. al., "A Study of the Overlap Among Document Representations," (Syracuse Univ. Sch. of Info. Studies, 1982).

(39) E.A. Fox and R.K. France, "Architecture of a Distributed Expert System for Composite Document Entry, Analysis, Representation, and Retrieval," Proceedings *Third Annual Comp. Sci. Symp. on Knowledge-Based Systems: Theory and Applications,* (March 31-April 1, 1986, to appear).

(40) L.D. Erman, F. Hayes-Roth, V.R. Lesser, and D.R. Reddy, "The Hearsay-II Speech-Understanding System: Integrating Knowledge to Resolve Uncertainty," *ACM Comp. Surveys,* 12 (1980) 213-253.

(41) R.J. Ensor, and J.D. Gabbe, "Transactional Blackboards," *Proc. of IJCAI '85,* (August 1985) 340-344.

(42) L.C. Smith and A. J. Warner, "A Taxonomy of Representations in Information Retrieval System Design," *Representation and Exchange of Knowledge as a Basis of Information Processes,* ed. by Hans J. Dietschmann, (New York: North-Holland, 1984) 31-49.

(43) Richard Fikes and Tom Kehler, "The Role of Frame-Based Representation in Reasoning," *Commun. ACM,* 28(9) (Sept. 1985) 904-920.

(44) G.P. Zarri, "An Outline of the Representation and Use of Temporal Data in the RESEDA System," *Inf. Tech.: Res. Dev. Applications,* 2(2/3) (July 1983) 89-108.

(45) R.C. Schank, J.L. Kolodner, and G. DeJong, "Conceptual Information Retrieval," *Information Retrieval Research,* ed. R.N. Oddy et al., (London: Butterworths, 1981).

(46) R.F. Simmons, *Computations from the English.* (Englewood Cliffs, NJ: Prentice-Hall, 1984).

(47) T. E. Ahlswede, "A Linguistic String Grammar of Adjective Definitions from *Webster's Seventh Collegiate Dictionary*," *Humans and Machines,* S. Williams (ed.), (1983) 101-127.