AN EXPERT ASSISTANT FOR DOCUMENT RETRIEVAL

W. Bruce Croft
Roger H. Thompson

COINS Technical Report 85-5

Department of Computer and Information Science
University of Massachusetts
Amherst, Massachusetts 01003, USA

TABLE OF CONTENTS

1.0 CHARACTERIZATION OF THE PROBLEM

1.1 Typical Systems

Document retrieval systems are designed to retrieve documents or references to documents that meet a person's need for information. The commercially available systems allow a user to find documents by specifying a query that is a Boolean combination (using AND, OR, and NOT) of index terms. The terms are assigned either by professional indexers from a list prepared by experts ("controlled" terms), or are derived automatically from the titles and the abstracts of the documents ("free" terms). The query acts as a filter, selecting documents whose characteristics exactly match the specification.

Some of the assumptions made in the design of current commercial systems are that the user has a definite idea of what he is looking for, and that he has expressed this need precisely in the vocabulary of index terms allowed by the system. This is usually not the case. Either the user has only a general idea of what he wants, cannot properly reduce his need to appropriate index terms, or both. In any case, the results produced by the system can be less than adequate. The effectiveness of these systems depends on a number of factors. Some of these factors relate to the user and some relate to the system.

1.2 Improvements

Most of the research in document retrieval systems has focused on developing better document representatives and models for document retrieval. A document representative is derived by the *indexing process* from either the document or its abstract; a query is derived using the same process from a statement of the user's information need. A typical indexing process proceeds as follows. The text is processed to remove what are called *stop words*. These consist of words such as conjunctions, demonstratives, question words, and other words which are so frequent that they are not good content discriminators. The remaining words are stripped of their suffixes. The stemmed words become the *index terms* of the documents.

Along with these index terms, counts are kept of the number of times they occur in the document.

To find documents in the system a *search technique* is used. The statistically based techniques [VanR79, Salt83a] use the information derived in the indexing process to compute a similarity value or a probability for each document, specifying how likely it is to be relevant to the query, and then ranks them accordingly. An example of a similarity measure is Dice's coefficient [VanR79], which is a normalized measure of how many terms are shared by a query and a document.
This is computed as

$$2 \cdot (| A \cap B | / | A | + | B |)$$

where A and B are documents (or a document and a query) and $| . |$ is the number of terms. Probabilistic models rank documents according to their probability of relevance. One form of the probabilistic model [Crof79] giving a ranking function $g(X)$ is computed by the following,

$$g(X) = \Sigma\, c_i x_i + C,$$

where X is a binary vector of index terms such that $X = (x_1, x_2, ..., x_n)$ and $x_i = 0$ or 1 signifying the absence or presence of the ith index term. The factor c_i (called the *inverse document frequency* weight [Spar77]) can be approximated by,

$$\log(\max(n_i) \;/\; n_i),$$

where n_i is the frequency of a term, and $\max(n_j)$ is the number of occurrences of the most frequent term in the document collection. C is a constant that is assumed to the same for all documents in the collection, but varies from query to query. The summation is over all query terms. Statistical techniques have a number of advantages compared to conventional Boolean systems. Ranked lists of documents are easy for users to evaluate. The query specification in a statistical system is straightforward; it is typically a list of terms with no Boolean relationships between them. Furthermore, statistical techniques help the user to revise the initial query through a process called *relevance feedback* [Salt83a]. Relevance feedback uses the individuals' evaluations of the retrieved documents to adjust the estimates of what relevant and non-relevant documents look like. The system then does another retrieval, producing a new list of documents. Formulating queries as simple term lists can lead to a loss of information, though, since terms may be related to other terms. These relationships can be specified by the Boolean operators. Terms that are ORed together generally represent synonyms, and those ANDed together are similar to phrases. More sophisticated probability models that are based on term dependence can also capture these relationships [Harp78,Crof85].

The statistical methods have been shown to be more effective, in general, than the methods used in commercial systems [Salt83a]. Effectiveness is determined by comparing the documents retrieved to known relevant documents using standard test collections. Two effectiveness measures are *recall* and *precision*.[1] The statistical methods are efficient because of their simple indexing and matching techniques, and can be used with large collections. Furthermore, these techniques are subject independent and can be used with little modification on collections containing documents from a wide variety of subjects.

The commercially available systems, as well as the statistically based techniques, contain a major impediment to their effectiveness. They respond to each query and each user in the same way. No provision is made to use more than just a superficial indication of the user's information need. Furthermore, these systems have no way to incorporate any background information, which is relevant to the search but not directly part of the need. Once the query formulation of the user's need is obtained, the system is confined to using a single retrieval technique. Besides the limited system response, the user has other problems to contend with. Almost all of the systems are designed to work on hardcopy terminals. This limits the interaction to a line-oriented style and makes the response quite slow. Furthermore, the online assistance provided with these systems can be primitive. Many of the systems have a command language explanation aid similar to the

---

[1] Recall is defined as the number of relevant documents retrieved divided by the total number of documents relevant for a query. Precision is the number of relevant documents retrieved divided by the number of retrieved documents.

VAX/VMS help facility. Online vocabulary help is often limited to listing terms which are alphabetically near to a term in question. Finally, to get the most complete results possible, the user may have to use several systems, which means he has to learn the intricacies of several command languages.

1.3 The Usual Solution

The usual solution to the problem of getting effective results with commercial document retrieval systems is to employ people called *search intermediaries* [Barr77]. Generally, intermediaries are not experts in the subject area being searched. They are experts at using the retrieval systems. The intermediary's task can be divided into a number of steps.

1. Get from the user a characterization of his information need, either by interviewing him or by having him write it down.
2. Select the appropriate database.
3. Reduce the need to a query appropriate to the database being searched.
4. Use the system to find the documents.
5. Present the documents to the user for his evaluation.
6. Go back to the first or second steps, if necessary, to revise the approach, and repeat until the user is satisfied.

To accomplish his job, the intermediary must be both a negotiator and a strategist. Negotiation describes the interaction between the user and the intermediary during the interview. As he conducts the interview, the intermediary learns about the user's need, and the user is helped to define his need more precisely. While they refine the expression of the need, each one gains knowledge about the other which will help them with further interaction. The intermediary's knowledge guides him in effectively presenting information to the user. For instance, if the user is a novice with retrieval systems, the intermediary will provide more explanation of what he is doing and what information provided by the system is important. If the user determines that the intermediary has some knowledge of the subject area, then he would not have to describe as much domain knowledge.

The strategist aspect describes the intermediary's interaction with the retrieval system. From experience, intermediaries develop heuristics to manipulate the query to get the best possible results. Some examples of these search heuristics are described by Bates [Bate79].

However, there are disadvantages with using the services of a search intermediary. The main one is that the person with the information need is not using the system. The intermediary, who does the actual search, has only a limited knowledge of the domain being searched. It would be impossible for the searcher, who has the knowledge of the domain, to describe all relevant information to the intermediary. The preferred interaction has the intermediary doing the searching with the user present to help with the subject knowledge [Henr80, p.97]. This situation, however, slows the intermediary down, and drives up the cost of the search which is often a factor to be considered. Also, depending on the personalities involved, difficulties can arise between the searcher and the intermediary making the search impossible or, at least, less than effective. Another consideration is scheduling. A search intermediary can be occupied with other people, and is seldom available after hours or on the weekend. These times may be when the searcher wants to do his work.

1.4 Another Solution

Another solution to the difficulty of using document retrieval systems is to make the system more flexible and responsive. Recent research indicates that flexibility is a major factor in improving the effectiveness and usability of these systems [Oddy77, Crof84a]. Flexibility can be incorporated into a number of aspects of these systems such as:

- **Query Formulation** - By giving the user more assistance as he begins to formulate a query, and by letting him browse the document collection, he can develop a more precise query, thereby retrieving relevant information more quickly.
- **Retrieval Techniques** - Previous work [Crof84] indicates that, although one retrieval technique is the most effective over a large number of queries of diverse types, another technique may work substantially better for a specific

query. If a system could select the best retrieval technique for a specific query, the performance of the system would be significantly improved.

- **Interface Design** - Currently available CRT terminals and personal computers have far more power to display information in an effective way than the line-oriented terminals of the past. For example, in the area of text editing, use of video terminals with greater display capability has increased the usability of text editors significantly [Meyr82a,b].

By making the system more flexible, the need for a human intermediary can be removed. In fact, the flexibility needed to make the system more responsive could be obtained by embedding the intermediary's expert knowledge in the system itself.

The next section reviews recent research directed at increasing the effectiveness and usability of document retrieval systems. Some work in user modeling is examined. The third section summarizes research issues. The last section presents the design of an expert assistant which address the problems stated here.

2.0 PREVIOUS RESEARCH AND RELATED WORK

Recently there have been a number of systems designed to improve the usability and effectiveness of document retrieval systems. Many of the systems attempt to use artificial intelligence techniques to accomplish these goals. The two areas from which these techniques are drawn are natural language processing (NLP) and expert systems.

The main use of NLP techniques has been to to derive better representations of the content of both documents and queries. This work has focused on capturing word relationships that are missed by the statistical techniques. Salton [Salt68] used purely syntactic methods to derive higher level relationships. The experimental results show that these methods are less effective than the statistical methods of the time. The work on FASIT [Dill83], which represents some later work along this line, shows more promise.

Most NLP techniques require extensive use of semantic information. This information must be incorporated into systems manually. Consequently, many of the successful NLP systems were developed to work in highly constrained domains. An example of this is the program SHRDLU by Winograd [Wins77]. Its domain of discourse is limited to a simulated one-armed robot manipulating colored blocks on a table. A more recent example is UC (UNIX[2] Consultant) by Wilensky, Arens, and Chin [Wile84]. This system is a help facility front-end, designed to assist the naive user with the UNIX operating system. These constraints limit the amount of semantic information required. Document retrieval systems, on the other hand, operate in relatively unconstrained environments. The massive amount of semantic information needed to operate in these kinds of environments and effort needed to use it make NLP techniques generally unacceptable.

[2] UNIX is a trademark of AT&T Bell Laboratories

Spark Jones [Spar83] recently has done work using NLP techniques with simplified semantic information. Her main focus was using these techniques to generate term variants to enhance the representation of a user's need. The results of the research, so far, have shown slightly better recall performance. Even though these techniques show promise, there are still major efficiency problems to be addressed. Some of the information derived using NLP techniques could be obtained more easily by asking the user to state relationships between terms. For example, he could specify a relationship between "Information" and "Management" to form "Information Management". This type of user-specified domain knowledge will be discussed in a later section.

Using expert systems to develop effective interfaces seems to be a promising research direction for document retrieval systems [Spar83]. One of the major aspects of expert systems is the use of production rules for encoding knowledge [Barr81]. This has a number of benefits:

- **Modularity** - When system is being developed, individual rules may be added, deleted, or changed without grossly affecting the entire system.
- **Flexibility** - The system selects its next action based on the current context and not on a fixed control cycle.
- **Inferential Capability** - The system can reason about its actions and explain to the user why it took a particular action. This aspect may be particularly important in overcoming the view that document retrieval systems are hard to understand and use.

The rest of this section reviews specific research that has been done to alleviate the difficulties of using document retrieval systems and other related research in expert systems.

2.1 CONIT / EXPERT

CONIT (Connector for Networked Information Transfer) [Marc83] and EXPERT [Yip81] are two closely related systems. The CONIT system is a solution to the problem caused by the number of different document retrieval systems available. It presents a virtual interface to multiple heterogeneous databases with a single command language that is meant to be easy to use. By having a command language that is simple to understand and use, the need for a search intermediary is eliminated. The system also attempts to make the search for documents less expensive by using either the cheapest or least loaded path to the databases.

The control of the system uses a rule-based approach. The interpreter is customized for this particular application to handle multiple input streams for multiple users. The rules translate the simplified command language into the commands appropriate to the system to which the user is connected via CONIT. They also handle all the necessary protocols to get the user logged onto a system.

An alternative system, arising from this research on the multiple database problem, is EXPERT. It also is a rule-based system. The purpose of this system is to relieve the user of having to use any command language and to assist him with his query formulation. The need for a command language is removed by using a menu-style interface. The system builds a query by developing concepts, which are unions of index terms, and intersecting them. It prints out the query in a columnar manner, with each column representing a concept, and in each column are the terms forming the concept. Query reformulation is done very simply. For example, it suggests that a user narrow his concepts when the query retrieves more than ten times the number of documents the user thought he should find. It does not provide any analysis of the query to suggest which concepts or terms within the concepts are responsible for the results. EXPERT also does not currently take advantage of any of the usual search aids, like a thesaurus, that are commonly available.

2.2 IR-NLI

IR-NLI [Guid83] is another system designed to be a front end for the available commercial systems. The main goal of this work is to apply natural language processing techniques to generate a query from a written statement of the problem and a structured dialogue with the user.

The system analyzes the information provided by the user to plan a strategy that will be effective for the problem. The strategies are sequences of steps to find terms for the query and then to use the query to retrieve documents. These steps are called *information search tactics* and are heuristics which have been developed by experience. They are discussed in detail by Bates [Bate79]. Selection of these steps is controlled by a strategy generation module, which is a ruled-based pattern matcher. This strategy generator operates on an internal representation of the query generated by the natural language analysis module. This representation is not only a model of the user's need, but also contains some information about the user, such as interest in recall or precision and output format.

The main feature of the system is the use of domain knowledge to help in the generation of strategies. This system knowledge is encoded in what is essentially a highly structured thesaurus. The search tactics are different ways to take advantage of this knowledge, and it is an extension of what is provided manually with most systems.

The natural language processing portion of the system is customized for the application of document retrieval. It strips out the words that are terms in the domain knowledge base and groups them into related concepts. The other words are either used to determine the relationships between these or are thrown away.

This system, like the previous one, is designed to improve the existing commercial systems. The design seems reasonable given the constraints of current systems. It does not retain knowledge about the user or the need from session to session, so it treats every user the same way. It also puts the control of the interaction entirely in the hands of the system.

Although this system contains a number of interesting ideas, it is not apparent how much of the system has been implemented. Consequently, the effect of these ideas on retrieval has not been reported.

2.3 THOMAS

Some interesting research to extend the capabilities of IR systems was the work on the program THOMAS [Oddy77]. In this work, Oddy was investigating an alternative to the traditional form of query formulation. The research was motivated by questioning the assumption in IR experiments that the user has a precise idea of his problem but may have difficulty in selecting the right terms to model it as a query. It appears that this is not the case. The user changes the idea of what he is looking for during the search. Therefore, the traditional query formulation process needs to be modified.

THOMAS provides a different way of retrieving information by using a model of the user's information need to guide its selection of documents. The program, via a structured dialogue, allows the user to browse the document collection. To begin, the user must enter some information to locate a starting point in the document collection. The program responds by presenting a document representative for user judgment. The user has a number of options in this judgment phase. He may judge on the relevance of a document or the document's descriptor terms, or authors, or all of these. Using these evaluations, the system updates its model of the user's search and presents another document based on the new model. The model changes as the system gathers information from the user.

The system works flexibly. If the model has changed substantially since the initial part of the dialogue, and some of the documents judged relevant early on appear inconsistent with the current model, the system will present them for re-evaluation. The user may decide that these documents are no longer relevant, reflecting the view of what he now wants.

To support this kind of interaction, the document collection is organized as a network. The nodes of the network are documents, authors, and index terms. The links connect documents to their index terms and to their authors. The model of the user's need is a subgraph of this network. This model reflects both the positive and negative evaluations of specific items by adding weights to the links.

A document is selected for presentation by measuring its association to the model of the user's need. This association measure is the ratio of the number of associated items to the total number of associated items known. It is, basically, a simple matching function. For example, if a user has seen one document and has judged it relevant, the next document selected for presentation will be the document that shares the most terms and authors with the first one. Items that have been negatively judged are not counted. A document that has items that are negatively judged will not be excluded; it will just have a lower association measure.

The program works effectively, but it does have some disadvantages. After some time using the program, the user may have a definite idea of he wants. At this point the model could be used as a formal query, but the program cannot take advantage of this; it has no mechanisms for performing any formal search techniques. The only retrieval technique it has is browsing. The network provides enough structure to support the browsing style of retrieval, but it does not capture similarity information which is easily derivable statistically from the documents and terms.

2.4 Anomalous States of Knowledge

The research by Belkin, Oddy, and Brooks [Belk83a,b] represents the initial stages of work to build a document retrieval system based on a cognitive view of information. It proposes a different model of the user's information need and a new retrieval technique based on that model. This information need is called an Anomalous State of Knowledge (ASK). It is called this because the user has a gap or an anomaly in what he knows. The request is an attempt to fill the gap or resolve the anomaly. The retrieval mechanism attempts to find documents that are similar to what the user knows, as represented by the query, but will supply the missing information.

The input to the system is either a written statement by the user, or a transcribed interview with him, that states his information need. A simple language processor takes this need statement and identifies the terms and the relationships between them. These relationships may be weak, medium, or strong depending on their proximity in a sentence or paragraph. This information is converted into a graph with the terms as nodes and the relationships as links. The graph is further reduced by collapsing certain structural features into supernodes. For example, a cluster of terms that are linked by strong links form one kind of supernode and, clusters which are linked by medium links form another. The result is a structural representation of a user's information need.

Representation for documents is the same. Retrieval is accomplished by comparing these representations. It is not necessarily a best match selection in that a document may match part of its structure with the query and be selected for retrieval.

Although retrieval effectiveness of the system has yet to be reported, this research contains some important ideas. First, the structural representation captures information about the relationships between terms that is missed in both the Boolean and term list formulation of a query. The term list query representation carries no user specified relationships at all. The Boolean query formulation does have some of this information, but it does not have the same gradation of relatedness -- terms are either related or not related. Terms associated by the OR to form concepts could be considered supernodes.

Another interesting result comes from surveying users about how well they thought the graphical representation represented their need and asking authors how well it represented the content of their document abstracts. In general, the people were satisfied with the representations. Some thought a number of terms were too weakly associated; some thought a number were too strongly associated. As these were processed only for a design evaluation, no feedback was taken to refine the representations before using for retrieval. The dissatisfaction with the association strengths, as the authors say, indicates a need for some revision of the association algorithm. In the system to be described, these representations could be presented to the users for their evaluation or adjustment.

2.5 Distributed Problem Treatment

Some other recent work by Belkin and others [Belk83c] deals with the design of an Information Provision Mechanism (IPM). This design proceeded from an abstract analysis of what a mechanism should contain, in order to provide ideal assistance to users in managing their information problems. They indicate that an IPM requires a minimum of ten functions to be able to provide effective assistance for management of information problems. These are:

1. **Problem State** - Determine the position of the user in the problem management process.
2. **Problem Mode** - Determine the appropriate information provision mechanism. Some are, fact retrieval, advice, role reinforcement.
3. **User Model** - Generate description of the user's type, intentions, beliefs, etc..
4. **Problem Description** - Generate description of problem structure, problem type, context, etc..
5. **Dialogue Model** - Determine the appropriate dialogue mode for the immediate situation. For example, use a natural language dialogue or a graphical interaction.
6. **Relevant World Builder** - Choose and apply appropriate retrieval strategies, if any, to the IPM's world model.
7. **Response Generator** - Determine the propositional structure of a response to the user which is appropriate to the immediate situation.
8. **Input Analyst** - Convert input from the user into structures appropriate for the functions.
9. **Output Generator** - Convert internal response structures into those appropriate to the user.
10. **Explanation** - Describe the IPM's operation, capabilities, etc. to the user at at appropriate points in the dialogue.

The validity of these functions is supported by empirical evidence obtained from the analysis of dialogues between users and search intermediaries in the information retrieval context [Belk83d, Broo85]. These interviews also show that the interaction shifts in and out of different areas of focus in a loosely structured fashion and does not proceed in a step by step manner. This, plus the fact that these functions share substantial amounts of information, leads them to propose a Hearsay-II style architecture for the IPM.

2.6 GRUNDY

The work by Rich on GRUNDY [Rich79] explores the use of user models to help systems adapt their actions for individuals. The test domain was a library system that provided reading recommendations.

The system gains knowledge about the users by asking them for a few words to describe themselves, or, if that fails, asking them to answer a few questions. From this information, it forms a model of the user. The model is composed of a number of stereotypes arranged hierarchically from the general to specific. The most general stereotype is Any_Person; examples of the more specific are a Protestant, a Radical, or a Sports_Person. These stereotypes represent the assumptions the system makes about a person as it learns. Some of these assumptions may be valid and some not.

The stereotypes are composed of items called facets, each of which consists of a name, a value, and a confidence rating. For example, one facet is called Tolerates_Violence; it is measured on a scale from -5, which means no toleration, to +5, which means great toleration. All the confidence ratings are measured from 0 to 1000 for computational simplicity, and represent 0% to 100%. The more general stereotypes have lower confidence ratings on the values, reflecting that not much is known with certainty about the user. As more becomes known about the user, more facets are applied and the confidence ratings increase.

All of the stereotypes invoked in the course of the interaction with the user are summarized in the user synopsis. This synopsis is what the system remembers about the user from session to session. It contains all the facets applied so far along with the stereotype name that caused them to be applied.

GRUNDY organizes its knowledge hierarchically on three levels. The first level is composed of dialogue specific information. In this system, this knowledge has been reduced to a few simple switches that record things such as if any books have been recommended yet. The second level is user specific knowledge represented by the user synopsis. The last level is what the system knows about users in general. This information potentially may apply to any user, but not necessarily to all users.

This organization allows the system to keep knowledge where it is appropriate. This is important since the knowledge gained from the user interaction may be relevant to any level. Some information may be relevant only to the current session; some may affect the confidence ratings of a stereotype, in which case it is relevant to the user synopsis too.

The specific facets in the stereotypes and how books are represented and recommended are not of great relevance to document retrieval. The attributes which describe the books correspond directly to the facets of the stereotypes which describe the users. Recommendations are made by matching the books to the stereotypes. This is similar in nature to what occurs in a document retrieval system, with the stereotypes corresponding to the queries and the books to the documents. The intent of using these modeling techniques is to tailor the system's interaction to the user and not to find documents which match the user.

Rich also describes how the user modeling methods can be applied to other domains. Specifically, she describes how a model might be applied to two mathematical systems, MACSYMA [Macs77] and Eurisko [Len77]. The operation of these systems depends on different factors about the user than those used in book recommendation. The factors might include such things as:

- Mathematical background -
  Is he a college freshman learning integration or is he a graduate student in engineering?

- Mathematical speciality -
  Is he a specialist in group theory or a statistician?
- Experience with computers -
  Has he never used one before or is he a "hacker"?
- What kind experience with computers -
  Has he programmed in BASIC and FORTRAN or Lisp and Prolog?

Factors similar to these can be derived for the document retrieval environment, so that these modeling techniques can be applied to make retrieval systems more responsive to the end users.

2.7 Associative Search Network

This research by Croft and Thompson [Crof84a] was directed at improving the effectiveness of statistically based systems. It incorporated several search techniques into one system using a control mechanism called an Associative Search Network (ASN) and a network representation of document contents. The ASN is capable of learning what action to take for a given set of environmental inputs (context), which in this case were characteristics of the query. The selection criteria are either positively or negatively reinforced by an evaluative feedback that measures the success of the selected action. For a full discussion of ASNs see [Bart81a,b].

The ASN was successful in learning what was the best overall search technique for the test collections used. However, it could not learn what was the best technique for a given query. This is attributable to two factors. First, there were no query features that correlated with a good choice of search technique. Second, it was difficult to obtain an accurate evaluation function. Although it did not perform as expected in selecting search techniques, the ASN may be useful in learning a user's preferences in such areas as interaction style.

The network consisted of document nodes and term nodes linked together. This representation supported multiple search techniques, and is the basis for the representation used in the system to be described.

2.8 Other Systems

Shoval's Work.

This system [Shov81] uses a knowledge representation that adds a number of relationships between terms that are not normally found in thesaurus information. The usual thesaurus links indicate hierarchical or synonymous relationships. Shoval adds two more called generator links and model links. The generator links define phrases or as he calls them "multiword concepts". The link ties together words such as "Information" and "Retrieval", which by themselves are general, but when combined are more specific. In other words, these links define phrases. The model links tie together terms that are related in some sense that is not captured by the normal thesaurus links. They are descriptive of a broader term. Shoval's examples are terms related to "Business," which are "organizational structure" and "functional areas." This type of semantic net organization for the thesaurus may be important in organizing domain knowledge obtained from the user.

Shoval presents heuristics for using the links in the thesaurus. They consist of expanding terms using the links to include new but related concepts. These new concepts are evaluated by the user. A positive evaluation will direct the system to continue along the current path. A negative evaluation will stop growth along a path, and the system will expand a term along a new path. The user may also request an explanation of why a concept was presented, in which case the system displays the links back to the original term. This allows the user to make use of the expert knowledge of the system to refine his query.

RUBRIC.

RUBRIC (RUle Based Retrieval of Information by Computer) is a full text document retrieval system based on expert system techniques [McCu83]. Queries are composed of sets of production rules allowing the user to define concepts which specify his retrieval request.

At the bottom level are "text reference expressions" which are occurrences of words in some syntactic unit. Some examples of these units are adjacent words, the same sentence, or the same paragraph. These are combined by the rules to form higher level concepts. Attached to the rules are certainty factors indicating how confident the user is that the higher level concept is represented by the lower level ones. These concept rules essentially define user specific semantic links. They are flexible, allowing the user to grade the strength of associations. For example, a person could define synonyms by the rule:

If word_a *Then* word_b *with* cf=0.9,

or define what Shoval calls a "model" link by the rule:

If "Business" *Then* "organizational structure" *with* cf=0.4.

Meno-Tutor.

Meno-Tutor [Wool84] is an experimental intelligent tutoring system. The basic thrust of the research was the design and implementation of a generic tutoring system. It uses a combination of expert knowledge in the area of discourse, models of student skill and misconceptions, and a tutoring component to probe student's knowledge of a subject area and to correct ant misconceptions he has about it.

The contribution to our work is derived from the tutoring component. This component consists of a number of states organized hierarchically into three levels; the lower levels refine the actions of the higher ones. The states which at each level define the form and the content of the interaction. The discourse is driven out by passing through the different states. The default or *context independent* path through these states follows their structure as an OR-graph with the states being the nodes and the default paths being the links. Besides these links, there are other paths through the states which are called *context dependent* transitions. These alternate paths are taken when meta-rules fire, indicating that the discourse ought to go a different direction to keep in line with the responses give by the student.

CALIBAN.

The CALIBAN system by Frei and Jauslin [Frei83] demonstrates the power of using a high resolution screen and a mouse pointing device as the interface hardware for a retrieval system. This system, like traditional menu based systems, rids the user of having to use a command language, but it gives him a better picture of where he is and what he is able to do. He can traverse a menu tree using the mouse to get to the service he needs. High resolution screens combined with multiple display windows, allows the system to display more and varied types of information. This reduces the amount of system information the user must keep in his mind, allowing him to concentrate more on solving his problem.

One difficulty with this system, which may be a fault of the underlying hardware, is that it has only a non-overlapping three window display. Much more information could be accommodated by using multiple overlapping windows that appear as needed.

POISE.

POISE [Crof84c] is designed to be an intelligent assistant for tool-based systems. The domain in which these ideas are being tested is the automated office environment. POISE supports tasks that involve a sequence of tool invocations. One of these tools may be a document retrieval system, but in the office environment this is limited to the filing and retrieval of office documents. Consequently, POISE deals with this task in much less detail than an expert search assistant would. Of interest, though, is the mechanism that POISE uses to recognize tasks initiated by the user. This mechanism may be useful in determining what search technique to apply to a specific query.

Pollitt's Work.

The system developed by Pollitt [Poll83] is a touch screen based aid for use in the selection of cancer treatment documents. The goal of this system was to increase the use of a document database by implementing an effective and easy to use interface. One problem which this system overcame was the difficulty in typing terms that appear in cancer treatment literature. The vocabulary in this area contains many rather long and easily misspelled words like otorhinolaryngologic. Merely having to touch them reduces the amount of frustration considerably.

The system automatically generates a query from the user's actions. The presentation is organized as a group of frames that define the context of the system. As the user selects an alternative, the control rules update the display and change the context.

The retrieval performance using the queries composed by this system was quite close to that of using queries composed by a human intermediary. They were about the same in recall. This performance similarity is attributable to the highly structured and narrow nature of the subject. Both the intermediary and the system rely on the same thesaurus for term selection.

EUREKA.

EUREKA [Burk79] is a full text retrieval system. Documents are found by queries composed of a Boolean combination of words, phrases, or both. Words in the queries can contain a match-any character at the beginning or end. Using the match-any character at the end of a word provides stem searching capability. Every word in a document appears as an index in the inverted file. A posting file is also kept for each word. EUREKA has two elements that are of interest: a user specific thesaurus, and context specific vocabulary assistance.

Because of the design of EUREKA, there are no assigned index terms for forming document representatives. Consequently, there is no system-wide thesaurus available to help the user select the proper search terms. To overcome this lack, the system lets the user build up a list of synonyms in his own local thesaurus. This thesaurus is consulted while a query is being processed to make use of the user's knowledge.

The context specific vocabulary assistance is a post-search aid to help the user sort through a set of retrieved documents. The set may be large if the user is imprecise in formulating his query. The system presents a histogram of word frequencies of the documents that the user retrieved, or a list of words that are used in many documents or have high average frequencies. From these lists he can select words to refine the retrieved set.

The most interesting aspect of these aids is that they are focused on the user's problem. The user thesaurus is an attempt to capture knowledge that is specific to the user and use it to increase the effectiveness of the system.

3.0 SUMMARY OF RESEARCH ISSUES

The expert assistant will integrate artificial intelligence (AI) and information retrieval (IR) techniques to produce a more effective and usable document retrieval system. Specifically, detailed knowledge of the users and their information needs will be used to increase the system's effectiveness. With the system to be described, a number of research issues will be considered. These issues can be divided into two major considerations, modeling and control. Both of these considerations are involved with a number of the system's aspects. Three aspects that are of primary intrest are: 1) interacting with the user, 2) obtaining an accurate characterization of the user and his need, and 3) responding to the need in the most effective way. The modeling and control considerations applied to these system aspects raise a number of questions, such as:

- Can knowledge about human intermediaries' search techniques and query formulation be effectively used to control the system's actions?
- Are the formal search techniques needed or will the most effective retrieval be done during the database browsing that is part of the query formulation process?
- What is the most effective way of presenting information to the user?
- What knowledge about the user's domain expertise is needed?
- How can domain knowledge provided by the user be incorporated into the system?
- What query-independent aspects of the user are useful to know?
- How can knowledge about different categories of users be used?
- What information about a document should be kept by the system and how should it be organized?
- What knowledge is needed to control the activation of the search techniques?

4.0 SYSTEM DESCRIPTION

The system is an expert assistant that provides information and tools to help a user formulate a query that specifies his information need, and provides a number of search techniques for retrieving documents to meet that need. At the outset of his search, the user may have only a general idea of the information he wants, and may be unsure of how to specify it. The system assists the user by gaining knowledge about him and his need, and uses this knowledge to guide the presentation of information for query refinement. While this interaction occurs, both the user and the system refine and expand what they know. The user gets a better idea of what his query should be; the system gets a better idea of both the user and the need. When the information need has been made sufficiently clear, the system will select the most effective search technique to retrieve documents for user evaluation.

The system is more than a conventional retrieval system with enhanced help facilities. Most systems have simple command explanation facilities and limited term selection assistance, both of which are invoked only at the user's request. The expert assistant takes an active role, and uses deeper knowledge to guide its assistance. Using the knowledge it has acquired about the user and his need, plus knowledge about the document collection, the system provides focused assistance. The system can recognize when the user needs help and offers it, although he is not obliged to accept it. Furthermore, the expert assistant can offer explanations of its actions at a level appropriate to the user.

4.1 System Overview

The system is divided into three major components, the Interface Manager (IM), the System Experts (SEs), and the Knowledge Base (KB). Figure 1 shows the overall structure of the system. The SEs are a collection of function specific experts which give the system the capability to assist the user in clarifying and expressing his information need and to retrieve documents likely to be relevant to that need. The KB is the repository of all information collected by the system. The IM contains

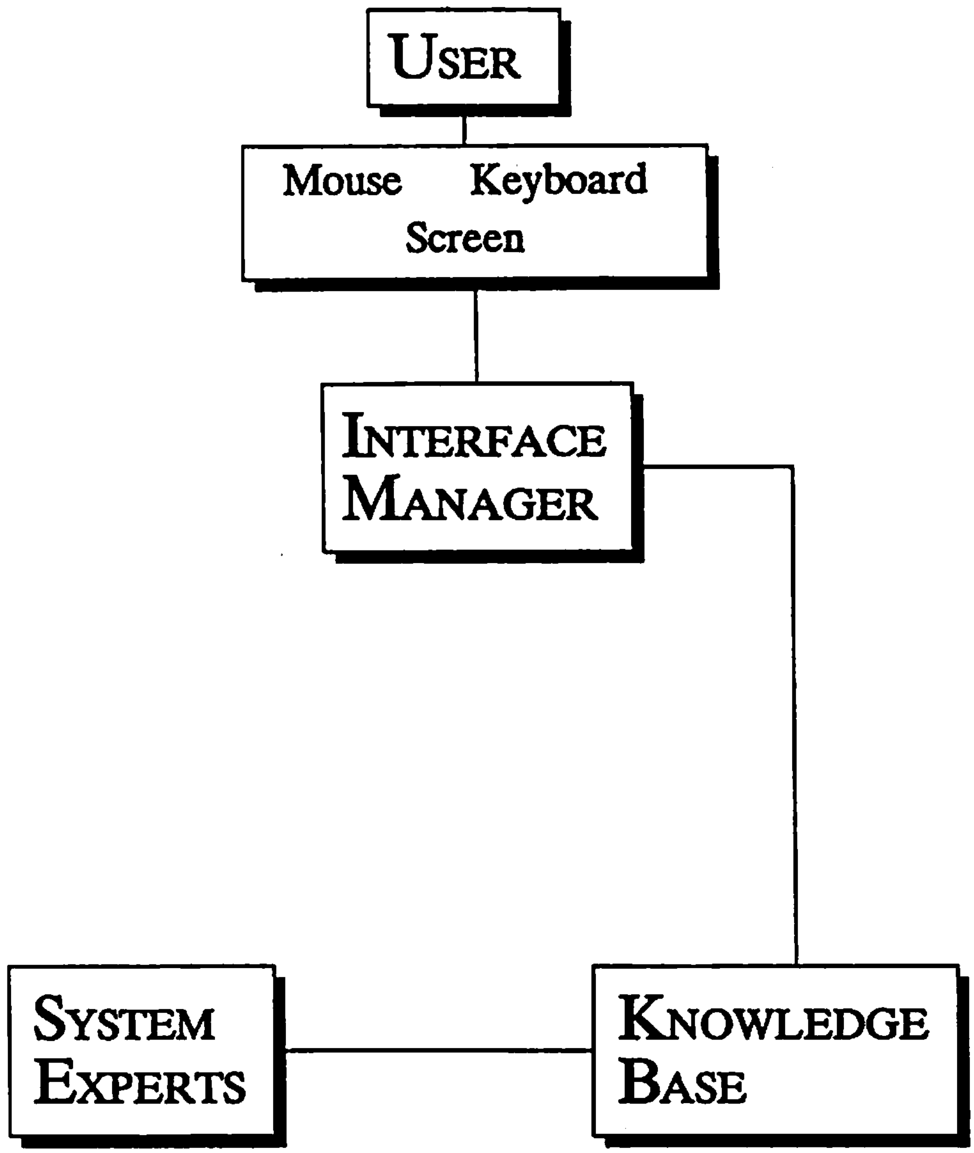

Figure 1: Structural View of the System

the knowledge needed to display and collect information in an appropriate form for a given device and a given user. The following sections examine the major components in more detail.

4.2 System Operation

The system's organization is derived from the Hearsay-II speech understanding system [Erma80]. The system experts view a blackboard, called the Short Term Memory, in the Knowledge Base and perform various actions depending on the current context. The experts also use information from the other part of the KB, the Long Term Memory. These two parts of the KB represent what information applies to the specific session in progress and that which the system knows in general. These models will be described further in a later section.

There are seven experts in the system, which are:

1. Browsing Expert (BE)
2. Explainer (EXP)
3. Thesaurus Expert (ThE)
4. Request Model Builder (RMB)
5. User Model Builder (UMB)
6. Search Controller (SC)
7. Natural Language Expert (NLE)

The specific actions of the experts will be discussed in a later section. Basically, they fall into two groups: those with which the system initiates actions (3–7) and those with which the user initiates actions (1,2). The first group operates, generally, by asking the user for information, which they then use to build or modify models. The second group allows the user to interrogate the system for specific information about its knowledge base and its operation.

The Scheduler is responsible for coordinating the activities of the system experts. In general, the experts have two kind of activities. They can evaluate the state of the system to decide if they have any actions to perform or they can carry

SYSTEM COMPONENTS

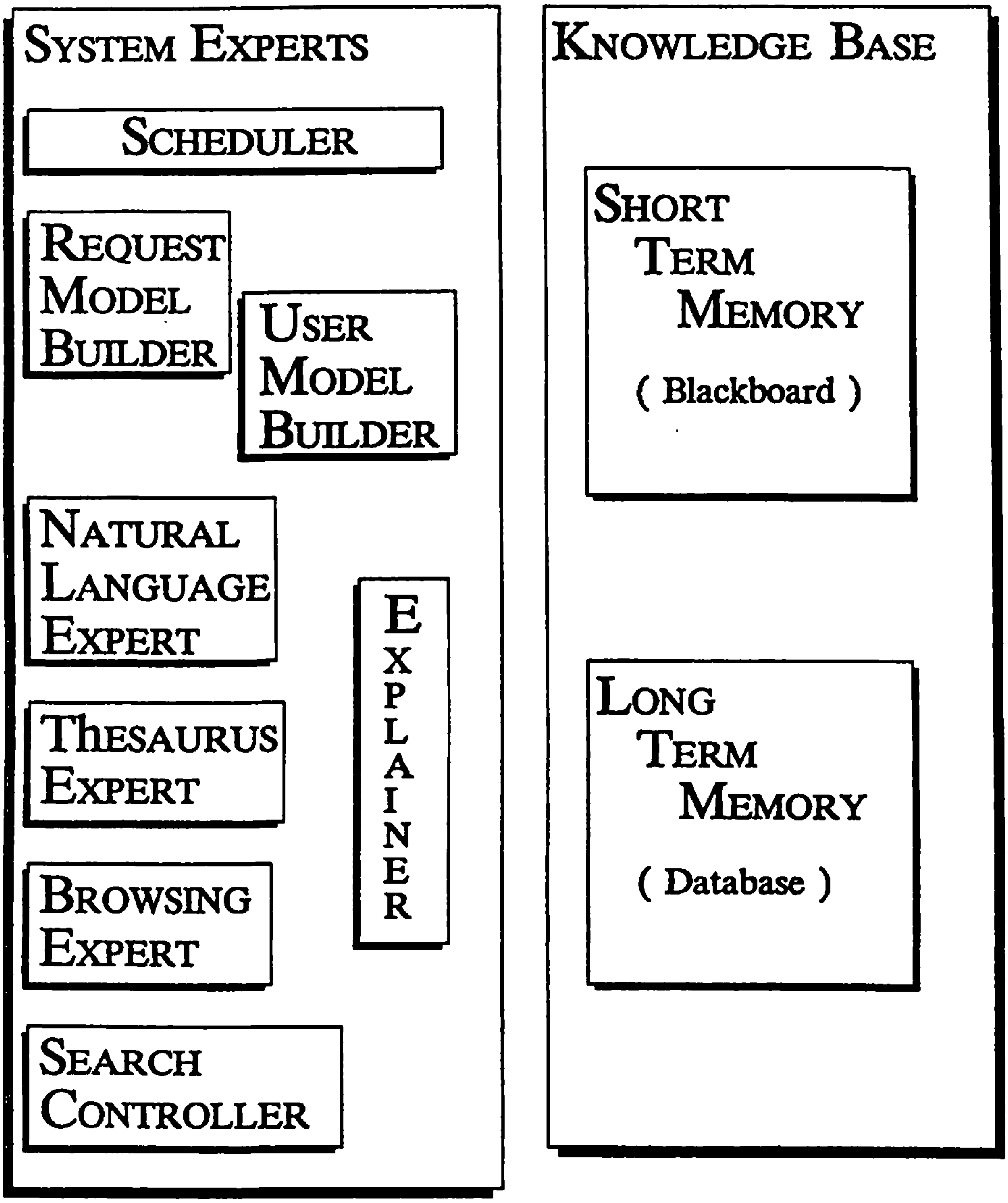

Figure 2: System Components

out the actions they have selected. The former activity does not cause any conflict with other experts, other than competing for computing resources. The latter activity is likely to change the state of the system and affect the actions other experts want to carry out. The Scheduler avoids this conflict by assigning priorities to the experts. These priorities change as a retrieval session proceeds.

To coordinate the activities of the experts the Scheduler uses a basic plan which is represented as an Augmented Transition Network (ATN). The states in this network are organized hierarchically and represent goals which must be reached to complete the retrieval process. Associated with each state is code that the Scheduler uses to monitor the progress of satisfying that goal state. The top level goal state is **Conduct Retrieval Session.** This represents the beginning and ending of a retrieval session. To accomplish this goal, three things must be done which define the next states. These are **Characterize the User**, **Determine the Information Need**, and **Search for Relevant Documents.** These, in turn, are refined further. **Characterize the User** is split into two substates to determine the user's system experience and the user's level of knowledge. In first of these, the system asks the user about what kind of general computer experience or what sort of information retrieval system experience he has. The second state assesses the user's level of knowledge in the subject area of interest. In both of these states the UMB and the EXP will have high priority. In contrast, the BE initially will not be allowed to run. The system must capture some knowledge about the user and must have some starting point before he can be allowed to explore the document collection.

The goal state of **Determining the Information Need** changes the focus of the interaction from the user himself to the user's need. The first subgoal state is to get the **Search Request** which includes getting initial statement of need in one form or another and refining it if need be. This kind of activity gives high priority to the NLE, RMB, and the ThE. The subsequent subgoal is to acquire **Domain Knowledge** in the area of his request. The information gained here helps the system confirm or revise the estimation of the user's knowledge level.

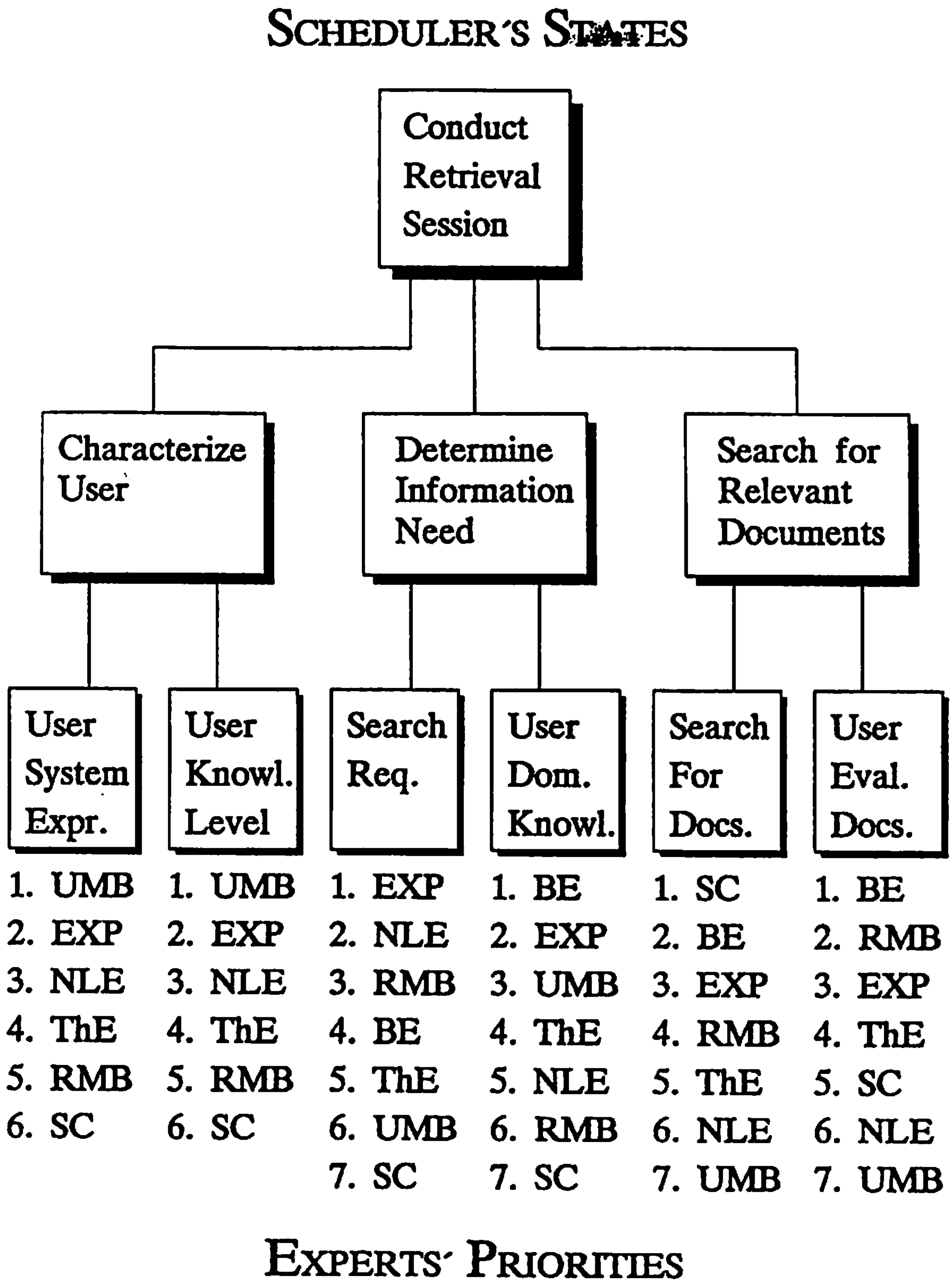

Figure 3: Structure of the Scheduler

The third state of **Search for Relevant Documents** is split in to a **Search For Documents** state and an **Evaluate Documents** state. The former gives priority to the SC and the BE. The second gives a higher priority to RMB. Figure 3 summarizes the basic structure of the Scheduler.

Once the priorities are established, the Scheduler selects which expert will run in the following way. First, all experts that request to evaluate their state are allowed to do so. Those that desire to perform an action signal their intention to do so. Then, the expert with the highest priority is allowed to run.

This algorithm provides more information for the state code than simply running down the priority list evaluating each expert and firing the first one capable of performing an action. With the former algorithm the monitoring code can detect when an expert continually has actions to perform and never gets to do them. This may indicate that the system is not in the proper state and should change.

4.3 The Knowledge Base

The Knowledge Base of the system is organized hierarchically as two memories, the Long Term Memory (LTM) and the Short Term Memory (STM). The LTM forms the permanent knowledge of the system and contains a number of models. Some of these models represent default information which the system uses at the start of a session. The Short Term Model (STM) forms the session specific information that is collected as a session proceeds. Organizing the system's information on these two levels allows it to be more versatile by giving it the means to adapt to different situations and users.

The Long Term Memory.

The Long Term Memory consists of the document collection and related statistical information, the user models, and the user stereotypes. To support the varied actions of the system, the document collection must be appropriately organized. A network provides a structure which has the advantages of a number of file organizations, supports a variety of formal search methods [Crof83a], and facilitates browsing. There are four kinds of nodes in this network; these are: documents, authors, concepts, and journals. A document node contains the title, publication information, and a unique identifier. An author node simply contains the author's name. A concept node represents semantic information. The simplest type of concept node represents a single word and has a textual representation, making them equivalent to the index terms in standard retrieval systems (and will simply be called "terms"). At higher levels, these concepts represent phrases. The journal node represents a specific issue of a journal and contains the name, date, volume and number, and page range. The journal information is useful in browsing as many journals present issues focused on a particular topic.

The links connecting the nodes are of two types: representation links and association links. The representation links connect a document node to the document, author, and term nodes forming a document representative. The association links connect documents to other documents or concepts to other concepts that are associated in some way. Two association links (Doc-Doc C & D) based on citation information are tentatively included; Mansur discusses the usefulness of these links [Mans80]. The following is a summary of the links in the network.

- **Doc-Doc.**
  (A). **Citation Link** (Cites) A representation link connecting a document to all the documents that it refers to or refers to it. This link is directed from the *citing* document to the *cited* document.

  (B). **Nearest Neighbor** (NN) An association link which connects documents of similar content; the content of the document being indicated by index terms. Links between documents are determined using a nearest neighbor algorithm [Crof83]. This algorithm produces a link to only the most similar document that is above a similarity value threshold. The similarities are computed using Dice's coefficient [VanR79]. Each link is weighted with this value indicating the strength of the association. This is not a two-way link, since the nearest neighbor relationship is not necessarily symmetric.

  (C). **Bibliographic Coupling** (BC) An association link based on bibliographic coupling. This link is analogous to B but uses references (doc-doc link A) instead of terms. It indicates that two documents are strongly associated if their bibliographies are similar.

(D). **Co-citation** (CC) An association link based on co-citations in other documents. If Doc_A and Doc_B are both cited in Doc_C, then Doc_A and Doc_B are co-cited. It measures the association of documents by their use in the literature. In the absence of citation information, this information can be partially derived from author co-citations

- **Doc-Term.**
  A representation link which connects a document to the term concepts in it. With this link, counts of a term's occurance in the document are kept.

- **Doc-Author.**
  A representation link connecting a document to its authors.

- **Doc-Journal.**
  Connects a particular journal issue to the documents it contains.

- **Concept-Concept.**
  (A). An association link based on the co-occurrence of term concepts in documents. This link is analogous to the first doc-doc link and is determined using the same nearest neighbor algorithm. These links are used in term dependence models of retrieval.

  (B). The association of concepts based on semantic information. These links represent different semantic relationships and the certainty value associateds with them. The types of semantic links are *instance/is-a*, *generator*, and *synonym*. This information could possibly come from concept associations made by experts in a field or from sources like the Computing Reviews classification scheme [ACM82], which defines keywords and concepts for organizing computing literature. Therefore, in time, some of the user-specific domain knowledge could migrate into the global system knowledge.

These links are implemented as relations in a database system with the nodes and weights being the attributes. For example, the nearest neighbor links are represented as: NN(Doc_A, Doc_B, Wgt). The use of relations to store the network gives greater flexibility than implementing it directly using a graph data stucture.

The second part of the LTM is the user models. These represent the system's knowledge about specific users from previous sessions. A user model consists of three parts, the user's domain knowledge, the user's interaction summary, and the previous request models. The user's domain knowledge is acquired from interaction with the system. It is usually obtained as a result of answers to questions posed by the system. For example, a user enters a list of words and the system responds by asking him to state any relationships between any of them or the ThE asks him for a confirmation of a concept.

The system can disregard the domain knowledge from previous sessions, if the user is searching in an area where it does not apply. There is also a possibility of information from the domain knowledge of experts being added to the LTM as general semantic information.

The interaction summary is a record of how the user has interacted with the system. It is derived from the system journal contained in the STM, and contains information on what facilities the user uses and how often. The summary also records his search statement preferences, for example, whether he enters search requests as a Boolean combination of terms or as a natural language statement.

The third part of the LTM consists of the user stereotypes. These represent what the system knows about users in general and are used primarily to guide interaction. These stereotypes categorize the user depending on how familiar he is with the use of the system and how familiar he is with the subject area he is searching. The particular profile for a session is selected using knowledge from the user's domain knowledge and from the need model. Consider a user that has used the system five times previously to get documents in a particular subject area, and who is currently looking for documents in different area. His profile would indicate that he is not a novice with the system, but not an expert, and that he is in a new subject area. This characterization might cause the system to offer more help in selecting terms appropriate to the unfamiliar domain, rather than offering assistance on system use. There are a number of other stereotypes that can be included in the system's repertoire. These include interest in high recall, interest in high precision, and interest in recent articles.

The Short Term Memory.

The Short Term Memory contains the information which the system uses during a specific session. This information comes from the user as he makes selections and enters text, and it comes from the system as the experts retrieve and post data from the LTM which is relevant to the current session. It is organized as a blackboard [Wins84] on which the experts can post and view information. The models which are found on the blackboard are described in the sections describing the experts which build them. It also serves as the communication path to the Interface Manager. At the end of a session, the information in the STM is integrated into the permanent information about the user in the Long Term Memory. Figure 4 summarizes the information contained in the STM.

4.4 The System Experts

Browsing Expert.

The Browsing Expert provides an informal way for the user to search for documents and to examine other parts of the Long Term Memory. The user can begin browsing anytime he is examining a node (document, concept, author, or journal). A person, generally, will use the facility at the outset of a search, if he has an incomplete or vague idea of what information he wants. The other time when the user is likely to browse is when the formal search techniques have failed to retrieve any relevant documents.

The expert works by displaying the content of a node in one window and a structural display of its *neighborhood* in another. The user can then evaluate the components of the node or elect to follow any of the displayed links. While the information is displayed the BE also shows its recommendation of what link to follow.

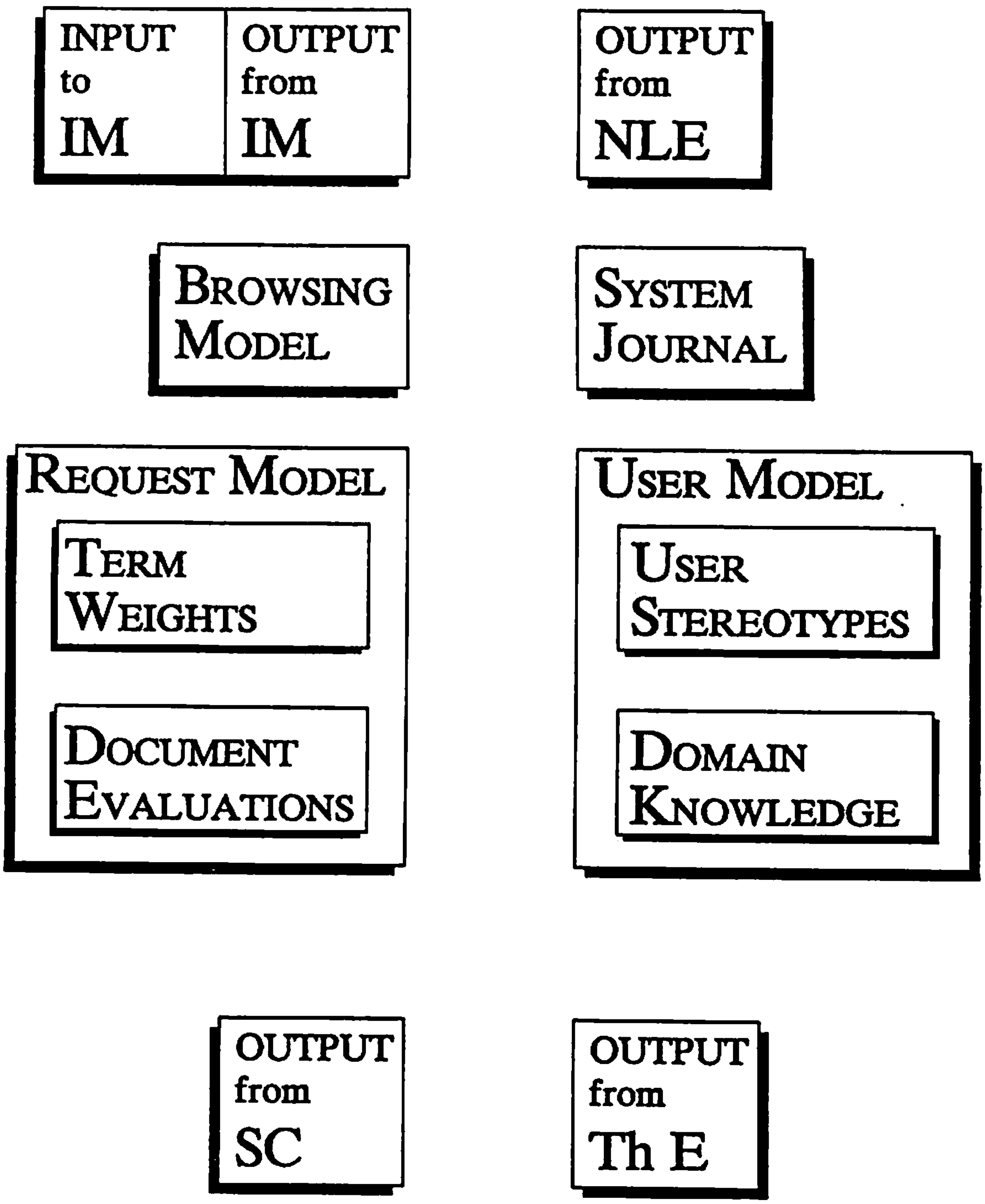

Figure 4: The Short Term Memory

One problem for the expert is deciding how much of the neighborhood to display. A document representation can contain 30 or 40 terms plus a potentially large number of links to other documents. A term can be found in hundreds of documents. This much information would easily saturate a window based display. The solution to this depends on what type of node is being examined and depends on the user's preference. The following sections explain the basic methods used by the BE to decide how much to show of the neighborhood surrounding a node.

Document Methods

Structural Display:

a. From the current document node expand the display using nearest neighbor (NN) links *originating from* it.

b. Expand the display using NN links *terminating at* the current document node. If there are more than three links, display the three most similar and display a fourth link tagged with the number of remaining NN links.

c. If there are any cocitation or bibliographic coupling links show them. If there are more than three, follow the same procedure as in step b.

Document Representation Display:

This display shows the components of a document representation. The top portion shows the title, the authors, plus publication data. The bottom portion contains the terms found in the document and its reference list. The abstract is not immediately displayed but is available to see on request, which when requested appears in another window. The user may evaluate the entire document and each part of it. If a positive evaluation is made of any component and this component is shared with any of the documents displayed on the screen, a node appears in the stuctural display with links to the documents it occurs in. Once the document and its components have been evaluated by the user, the expert makes a recommendation on what path to follow. It does this by counting the number of links from the current document to any of the displayed ones and selects the one with greatest number.

Term Methods

In the previous discussion the user was already viewing documents so the expert worked to move the user to other documents. In this case, with the user entering terms as the initial input, the expert still tries to direct the user to documents.

Structural Display:

In each term node displayed the textual representation is shown.

a. Expand the display using NN links *originating from* the term.

b. Expand the display using NN links *terminating at* the term. If there are more than three, select the three with the greatest similarity and display those. Show a link tagged with the number of remaining links.

c. Expand the display with *narrower than* semantic links.

d. Expand the display with *synonym* semantic links.

e. On each displayed term, mark the number of documents in which it occurs. If a term occurs in five or less documents, show the document nodes.

f. If any concept is valid as a result of all its constituent parts being displayed, display that concept.

Term Representation Display:

The representation display of a term is quite simple. It consists of its frequency in the document collection, the various connections it has to other terms with semantic links, and what concepts it is a member of along with its importance to that concept.

Journals and Authors

Browsing using journals or authors as a starting point is straightforward. With a journal, the documents which make up the issue are displayed. With authors, the documents written by that author are listed in order of date with most recent first, or listed by citation count.

While the user is browsing through the collection, his evaluations are taken and posted in the Short Term Memory for use by the other experts. Many of these evaluations are particularly useful to the Request Model Builder.

It is possible that the user may choose to explore the document collection and its associated information in a manner not consistant with the general goal of the Browsing Expert. This is permissible; the user simply ignores the Browsing Expert's recommendations. The displays can be expanded in ways not included in the general methods by selecting a link from the representation display which is not currently shown in the structural display. For example, a user may want to explore the vocabulary of the system so he decides to go up to a concept that is connected by an *is-a* link from the node he is currently examining.

Explainer.

The Explainer gives the user the means to interrogate the system about its actions, its models, and its use. The first of these capabilities is performed in the tradtional manner of the MYCIN system [Shor76]. The reasons for an action are shown by displaying the chain of rules that lead up to it. The source for this information is the system journal in the STM. This is a log of all the actions taken by the system for a particular session. The second capability simply allows the user to view the contents of any of the models, and if the user has sufficient experience with the system, may alter their contents.

The third capability is implemented by displays of examples tailored to the various aspects of the system. It is also designed to work in conjunction with the UMB and Scheduler to help obviate difficulties which may arise due to ignorance of system operation. For example, say a user indicates that he is interested in a precision oriented search and has provided search terms which are general in nature, as indicated by a high frequency in the document collection. The UMB will detect the mismatch between the user's intent and the information provided by him, causing it to request the Explainer to display the appropriate information.

Thesaurus Expert.

The goal of the Thesaurus Expert is to establish the presence of higher level concepts in the user's request. These concepts are used to find additional terms which will help make the information request more specific.

The expert takes the terms supplied by the user and searches either the user's domain knowledge or the system's database for related concepts. The user domain knowledge and the semantic link in the system database form the thesaurus information for the session. For example, the user has supplied the terms *statistical, retrieval,* and *model.* In the thesaurus the concept *probabilistic retrieval model* exists. The expert will immediatedly find *retrievel* and *model.* Having two of the three words the expert will assign a high degree of confidence to the concept being relevant to the users information need. However, the thesaurus may also contain the concepts *Boolean retrieval model* and *vector retrieval model.* In which case, the expert at this stage would consider these as equally likely to be relevant to the information need. The expert will look for evidence to establish a ranking according to how likely they are to be relevant to the information need.

To do this, the expert looks for links to the missing concept terms from the remaining user supplied terms. Here, it looks for any links from *probabilistic, Boolean,* and *vector* to the remaining term *statistical.* In this context, there may be a term-term synonym link between *probabilistic* and *statistical.* With this information the expert can conclude that *probabilistic information retrieval* is the concept the user is probably interested in. The expert then presents this concept first to the user for his evaluation. The others having no other supporting terms are presented next.

Request Model Builder.

The purpose of the Request Model Builder is to construct and maintain a model of the users information need. Sources of information for the model are evaluations by the user of terms, documents, authors, and concepts plus statistical information about the document collection. This information is posted in the Short

Term Memory by the experts responsible for its collection, with the exception of the statistical information. For example, the user may be browsing and evaluates a document as relevant. The Browsing Expert posts this infomation for the RMB to use. A request model consists of a number of elements, the first of which is document evaluations. These evaluations are simple assesments of relevance: relevant, not relevant, and no decision. They are used by the RMB in computing weights for the terms in the request model.

The second element in the request model is the term weights. Three weights are kept for each term. The first is the frequency of a term in the user's statement of information need. Second, is a user supplied emphasis weight, which can be captured in a variety of ways. For example, in response to a list of documents posted by the SC, the user may evaluate a document as relevant and at the same time single out certain terms as particularly interesting. By doing this, the emphasis on these terms is increased. The third weight is computed using the probabilistic retieval model. The details of this model are presented by van Rijsbergen [vanR79].

User Model Builder.

The UMB is responsible for building and maintaining models that describe various aspects of the user. These models are the user's domain knowledge and the user's stereotypes. The domain knowledge is the same as that used by the ThE. In fact, the domain knowledge may be a subset of the system's information differing only in the values of the certainties. This reflects that while the system may have many concepts, some are more important than others for a particular user.

The use of user-specific domain knowledge is vital for effective system response of the user. Consider two people, one who is primarily interested in database management systems, and the other who is primarily interested in document retrieval systems. Both these areas are considered to be part of *Information Management*, yet they differ in the way they use terms such as *Model*, and *Operation*. Figure 5 is a representation of a portion of the domain knowledge of these two users.

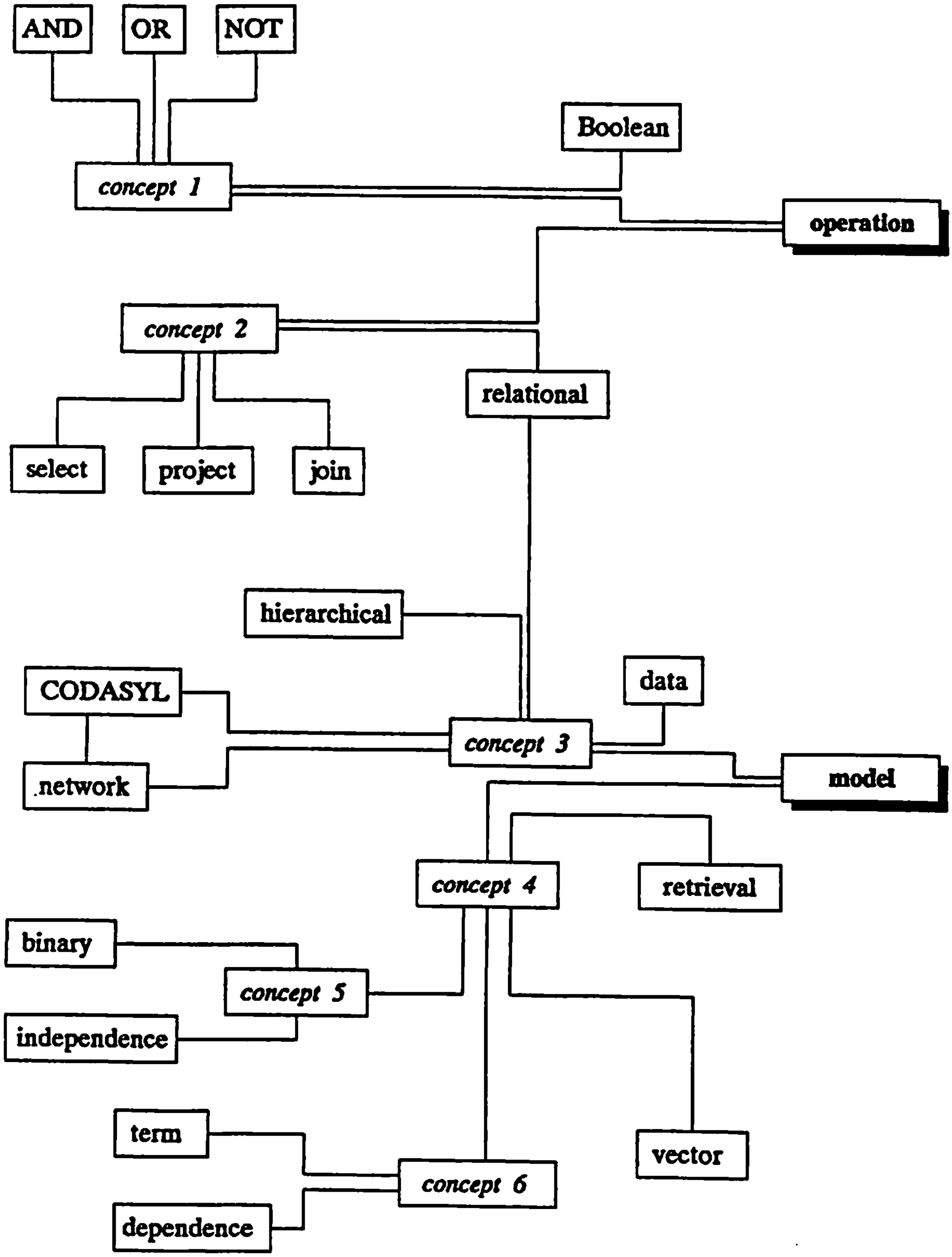

Figure 2: User Domain Knowledge

The domain knowledge of the database person would contain a generator link associating *Model* to *Data* making the concept *Data Model*, This concept would be linked by instance links to *Hierarchical, Network and CODASYL* (these two are linked by a synonym link) and *Relational*. *Operation* would be linked to *Relational* forming the *Relational Operation* concept. *Select, Project*, and *Join* are connected to this by instance links as more specific terms describing possible operations.

The person that is interested in document retrieval would interpret the two terms differently. *Model* would be linked to *Retrieval* forming a concept, which is linked to *Vector, Binary Independence*, and *Term Dependence* as specific instances. The concept *Query Operation* might be linked to the *Boolean Operation* concept which is connected to *AND, OR*, and *NOT* as specific terms.

This last example suggests another point. The user's domain knowledge, might contain words that are not found in the document collection. Words like AND, OR, and NOT are found on the stop-word list used by the NLE, so they are not included in the document representative. The user, however, may use these words in expressing the concept of *Boolean Operators*. By retaining these in the user's domain knowledge model, they can be translated to an existing term that captures a concept that would otherwise be missed.

The user domain knowledge also helps the system focus its search efforts. Since the document collection is not partitioned into smaller subject-specific databases, a term can be used in a wide variety of contexts. If the person, who was interested in databases, in the previous example was searching in the database field, the system could use its acquired information of the user's domain knowledge to limit the search.

Another element contained in the domain knowledge model of the user is his opinion of the various journals. This information also can be useful in selecting documents. A user may be aware of only a few of the sources of articles in the collection. By selecting articles from outside his usual sources, the system may present discussions of a subject he has not seen before. This knowledge can also act as a filter. For example, a user may not want to look at any article from a source that is not refereed, which he would indicate by a low opinion rating. Similarly, the model contains the user's opinions about authors.

The UMB also determines what stereotypes apply to the particular session by asking the user a few questions which will identify him. For example:

How would you describe your expertise in the subject of your request:

A. Novice

B. Advanced

C. Expert

Depending on the catagorization the user makes of himself, the UMB will activate an appropriate stereotype. The UMB will also watch the interaction to check that its characterizations are accurate. It may be that the user has indicated that he is an expert but only judges as relevant very general documents and ignores more specific documents.

Search Controller.

The function of the Search Controller is the selection and application of formal search techniques to retrieve documents. Its purpose is to give the system the capability to use standard retrieval techniques in addition to the document collection browsing provided by the Browsing Expert. By having this capability, the system's effectiveness is significantly enhanced. These techniques are based on different theoretical models which include:

- Binary Independence Model [Robe76, Crof83]
  This may be used when the user supplies a textual statement of need or a list of terms he feels adequately describes his need.
- Term Dependence Model [Harp78, Yu83]
  If strong statistical relationships exist between the terms in the request model, along with other types of relationships this model is appropriate.
- Cluster Searching [Crof80]
  This may be selected if the user requires high precision results or other techniques have failed.

- Extended Boolean Model [Salt83b]
  The user enters a Boolean query.
- Citation Searching [Garf79].
  This technique of search may be used when the user is trying to trace the history of an idea or the origin of an experimental procedure.

The controller looks at the information in the request model to decide if it can activate one of its formal search techniques. It is possible that in the course of a session the SC may never get sufficient information to use any of them. When it has enough information, it activates a search using one its search techniques. It is also possible that the request model may indicate that more than one can be used. In this case, the SC may start a search using the one it decides is the most appropriate. If it cannot decide on one, it may use any applicable technique. When the search is completed, the results, consisting of a list of documents, are posted in the STM along with the reasons (i.e. what rules were used) why the particular search technique was used.

Natural Language Expert.

The NLE is responsible to transform a natural language statement of information need into the internal form used by the system. This is accomplished by using the same indexing process [Port80] which produces the document representations used in the system. The output is a list of term numbers and their frequency in the need statement. This list is posted in the Short Term Memory.

4.5 The Interface Manager

While the System Experts are responsible for *what* information is to be displayed, the Interface Manager is responsible for *how* the information is displayed and collected. The kind of information it must display is varied. Some examples are:

- Document Text -
  The abstract of the document.

- Document Representatives -
  This can be some or all of the information which is kept about a particular document.
- Views of the document collection -
  Given a document, the user may want to see what other documents are related to it by any of the links.
- Views of the user's information need -
  The system could present the user's need model and ask him to adjust it or ask him question about it.

How it displays this information depends on the user's preference. To determine what this is, the IM consults the user model in the Short Term Memory.

An example of the kind of interaction the IM must support is specification of concept relationships. One user may specify a relationship between "Information" and "Management" by using a functional notation such as relate("Information","Management") or by using a Lisp-like notation such as (relate 'information 'management). Another may prefer a more graphical approach. He can select a type of relationship from a list, and then draw a line from a visible node marked "Inform-" to a visible node marked "Manag-". The "-" markings indicate the user is working with stems rather than with complete words.

An example of using a multiple window display is browsing using the links in the document collection part of the LTM. The display might consist of three windows, one with a structural view of the collection, another with document representatives, and the third with document abstracts. The user can move from node to node in the structural view using the pointing device to select the node he wants to view. He is not restricted to selecting nodes that are linked to the one where he is currently located. When the user selects a displayed node the representative and the abstract appear in the appropriate windows. Since a document may have more links emanating from it than can be reasonably displayed, the user can select the type of links that are shown in the structural view.

5.0 REFERENCES

ACM82 "New Computing Reviews Classification Scheme"
ACM Computing Reviews, 23; 10-48, 1982.

Barr77 Barraclough, E.D.
"On-line searching in information retrieval",
Journal of Documentation, 33; 220-238, 1977.

Barr81 Barr, A., Feigenbaum, E.A. (Editors)
The Handbook of Artificial Intelligence, Vol.1, William Kaufmann Inc., Los Altos, CA., 1981.

Bart81a Barto, A.G., Sutton, R.S., and Brouwer, P.S..
"Associative Search Network: A Reinforcement Learning Associative Memory",
Biological Cybernetics, 42 (1981), 201 - 211.

Bart81b Barto, A.G., Sutton, R.S..
"Landmark Learning: An Illustration Of Associative Search",
Biological Cybernetics, 42 (1981), 1 - 8.

Bate79 Bates, M.J.
"Idea Tactics",
Journal of American Society for Information Science, 30; 280-289, 1979.

Belk82a Belkin, N.J., Oddy, R.N., Brooks, H.M.
"ASK for information retrieval: Part I. background and theory",
Journal of Documentation, 38; 61-71, 1982.

Belk82b Belkin, N.J., Oddy, R.N., Brooks, H.M.
"ASK for information retrieval: Part II. results of a design study",
Journal of Documentation, 38; 145-164, 1982.

Belk82c Belkin, N.J., Seeger, T., Wersig, G.
"Distributed expert problem treatment as a model for the analysis and design of information provision mechanisms"
Journal of Information Science, 5, 153-167, 1983.

5.0 REFERENCES

ACM82 "New Computing Reviews Classification Scheme"
ACM Computing Reviews, 23; 10-48, 1982.

Barr77 Barraclough, E.D.
"On-line searching in information retrieval",
Journal of Documentation, 33; 220-238, 1977.

Barr81 Barr, A., Feigenbaum, E.A. (Editors)
The Handbook of Artificial Intelligence, Vol.1, William Kaufmann Inc., Los Altos, CA., 1981.

Bart81a Barto, A.G., Sutton, R.S., and Brouwer, P.S..
"Associative Search Network: A Reinforcement Learning Associative Memory",
Biological Cybernetics, 42 (1981), 201 - 211.

Bart81b Barto, A.G., Sutton, R.S..
"Landmark Learning: An Illustration Of Associative Search",
Biological Cybernetics, 42 (1981), 1 - 8.

Bate79 Bates, M.J.
"Idea Tactics",
Journal of American Society for Information Science, 30; 280-289, 1979.

Belk82a Belkin, N.J., Oddy, R.N., Brooks, H.M.
"ASK for information retrieval: Part I. background and theory",
Journal of Documentation, 38; 61-71, 1982.

Belk82b Belkin, N.J., Oddy, R.N., Brooks, H.M.
"ASK for information retrieval: Part II. results of a design study",
Journal of Documentation, 38; 145-164, 1982.

Belk82c Belkin, N.J., Seeger, T., Wersig, G.
"Distributed expert problem treatment as a model for the analysis and design of information provision mechanisms"
Journal of Information Science, 5, 153-167, 1983.

Broo84 Brooks, H.M., Belkin, N.J.
"Using discourse analysis for the design of information retrieval interaction mechanisms"
Proceedings of the ACM SIGIR Conference, SIGIR Forum, 17; 31-47, 1983.

Broo85 Brooks, H.M., Daniels, P.J., Belkin, N.J.
"Using problem structures for driving human-computer dialogs"
Proceedings of RIAO85 Symposium, (to appear), 1985.

Burk79 Burket, T.G., Emrath, P., Kuck, D.J.
"The Use of Vocabulary Files for On-Line Information Retrieval,
Information Processing and Management, 15; 281-289, 1979.

Crof80 Croft, W.B.
"A model of cluster searching based on classification",
Information Systems, 5; 189-195, 1980.

Crof83a Croft, W.B.
"Experiments with representation in a document retrieval system",
Information Technology, 2; 1-22, 1983.

Crof83b Croft, W.B.
"Applications for information retrieval techniques in the office",
Proceedings of the ACM SIGIR Conference, SIGIR Forum, 17; 18-23, 1983.

Crof84a Croft, W.B., Thompson, R.
"The use of adaptive mechanisms for selection of search strategies in document retrieval systems",
Proceedings of the third joint BCS and ACM Symposium on Research and Development in Information Retrieval, 95-110, Cambridge University Press, Cambridge, 1984.

Crof84b Croft, W.B., Parenty, T.J.
"A comparison of a network structure and a database system used for document retrieval",
Technical Report, Computer and Information Science Department, University of Massachusetts, 1984.

Crof84c Croft, W.B., Lefkowitz, L.
"Task support in an office system",
ACM Transactions on Office Information Systems, 2, 197-212, 1984.

Crof85 Croft, W.B.
"Boolean queries and probabilistic models of information retrieval"
Technical Report, Computer and Information Science Department, University of Massachusetts, 1985.

Date81 Date, C.J.
An introduction to database systems.,
Third Edition, Addison-Wesley, Reading MA, 1981.

Dill83 Dillon, M., Gray, A.S.
"FASIT: A Fully Automatic Syntactically Based Indexing System",
Journal of the American Society for Information Science, 34; 99-108, 1983.

Erma80 Erman, L.D., Hayes-Roth, F., Lesser, V.R., Reddy, D.R.
"The Hearsay-II speech-understanding system: integrating knowledge to resolve uncertainty"
ACM Computing Surveys, 12; 213-253, 1980.

Frei83 Frei, H.P., Jauslin, J.F.
"Graphical presentation of information and services: A user-oriented interface",
Information Technology, 2; 23-42, 1983.

Garf79 Garfield, E.
Citation Searching,
John Wiley and Sons, 1978.

Guid83 Guida, G., Tasso, C.
"IR-NLI: An expert natural language interface to online data bases",
Proceedings of the Conference on Applied Natural Language Processing, Santa Monica, California, 31-38, 1983 (Published by the Association for Computational Linguistics).

Harp78 Harper, D.J., Van Rijsbergen, C.J.
"An evaluation of feedback in document retrieval using co-occurrence data",
Journal of Documentation, 34; 189-216, 1978.

Henr80 Henry, W.M., Leigh, J.A., Tedd, L.A., Williams, P.W.
Online Searching – An Introduction.,
Butterworths, London, 1980.

Len77 Lenat, D.B.
"The Ubiquity of Discovery",
Artificial Intelligence, 9; 257-285, 1977.

Macs77 MACSYMA.
MACSYMA Reference Manual,
Technical Report, M.I.T., 1975.

Mans80 Mansur, O.
"On Selection and Combining of Relevance Indicators",
Information Processing and Management, 16; 139-153, 1980.

Marc83 Marcus, R.S.
"An experimental comparison of the effectiveness of computers and humans as search intermediaries",
Journal of the American Society for Information Science, 34; 381-404, 1983.

McCu83 McCune, B.P., Tong, R.M., Dean, J.S., Shapiro, D.G.
Technical Report 1018-1, Advanced Information & Decision Systems, Mountain View, CA, 1983.

Meyr82a Meyrowitz, N., Van Dam, A.
"Interactive Editing Systems: Part I",
ACM Computing Surveys, 14; 321-352, 1982.

Meyr82a Meyrowitz, N., Van Dam, A.
"Interactive Editing Systems: Part II"
ACM Computing Surveys, 14; 353-415, 1982.

Oddy77 Oddy, R.N.
"Information retrieval through man-machine dialogue",
Journal of Documentation, 33; 1-14, 1977.

Poll83 Pollitt, S.E.
"End user touch searching for cancer therapy literature - a rule based approach", *Proceedings of the ACM SIGIR Conference, SIGIR Forum*, 17; 136-145, 1983.

Rich79 Rich, E.
"Building and exploiting user models",
Ph.D. Thesis, Technical Report CMU-CS-79-119, Carnegie-Mellon University, 1979.

VanR79 Van Rijsbergen, C.J.
Information Retrieval,
2nd Edition, Butterworths, London, 1979.

VanR80 Van Rijsbergen, C.J., Robertson, S.E., Porter, M.F.
"New models in probabilistic information retrieval",
British Library Research and Development Report 5587, Computer Laboratory, University of Cambridge, England, 1980.

Robe76 Robertson, S.E., Sparck Jones, K.
"Relevance weighting of search terms",
Journal of the American Society for Information Science, 23; 129-146, 1976.

Salt83a Salton, G., McGill, M.J.
Introduction to Modern Information Retrieval,
McGraw-Hill, New York, 1983.

Salt83b Salton, G., Fox, E.A., Wu, H.
"Extended Boolean information retrieval",
Communications of the ACM, 26; 1022-1036, 1983.

Shor76 Shortliffe, E.H.
MYCIN: Computer-based medical consultations,
Elsevier, New York, 1976.

Shov81 Shoval, P.
"Expert/consultation system for a retrieval data-base with semantic network of concepts",
Proceedings of the ACM SIGIR Conference, SIGIR Forum, 16; 145-149, 1981.

Spar72 Sparck Jones, K.
"A statistical interpretation of term specificity and its application to retrieval",
Journal of Documentation, 28; 11-20, 1972.

Spar77
Spark Jones, K., Bates, R.G.
"Research on automatic Indexing 1974-6"
2 Vols., Computer Laboratory, University of Cambridge, UK, 1977.

Spar83 Sparck Jones, K.
"Intelligent retrieval",
Proceedings of Informatics 7, Aslib, London, 136-142, 1983.

Vigi83 Vigil, P.J.
"The Psychology of Online Searching",
Journal of The American Society of Information Science, 34; 281-287, 1983.

Wins77 Winston, P.H.
Artificial Intelligence,
First Edition, Addison-Wesley, Reading MA, 1977.

Wins84 Winston, P.H.
Artificial Intelligence,
Second Edition with Corrections, Addison-Wesley, Reading MA, 1984.

Wile84 Wilensky, R., Arens, Y., Chin, D.
"Talking to UNIX in English: An Overview of UC"
Communications of the ACM, 27; 574-593, 1984.

Wool84 Woolf, B.P.
"Context dependent planning in a machine tutor"
Ph.D. Thesis, COINS Technical Report 84-21, University of Massachusetts, 1984.

Yip81 Yip, M-K.
"An expert system for document retrieval",
M.S. Thesis, Department of Electrical Engineering and Computer Science, M.I.T., 1981.

Yu83 Yu, C.T., Buckley, D., Lam, K., Salton, G.
"A generalized term dependence model in information retrieval",
Information Technology, 2; 129-154, 1983.

Errata

p. 13 line 16, Belk83 -> Belk82
p. 15 line 1 , Belk83c -> Belk83
p. 15 line 30, Belk83d -> Broo83
p. 47 line 25, Belk82c -> Belk83
p. 48 line 2 , Broo84 -> Broo83
p. 49 line 19, Garf79 -> Garf78
p. 50 line 6 , 1975 -> 1977