Cognitive Process as a Basis for Intelligent Retrieval Systems Design

Information Processing & Management Vol. 27, No. 5, pp. 405–432, 1991
Printed in Great Britain.

0306-4573/91 $3.00 + .00

COGNITIVE PROCESS AS A BASIS FOR INTELLIGENT RETRIEVAL SYSTEMS DESIGN

HSINCHUN CHEN
MIS Department, University of Arizona, Tucson, Arizona 85721, U.S.A.

and

VASANT DHAR
IS Department, New York University, New York, NY 10003, U.S.A.

(*Received* 3 *December* 1990; *accepted in final form* 20 *December* 1990)

Abstract—Two studies were conducted to investigate the cognitive processes involved in online document-based information retrieval. These studies led to the development of five computational models of online document retrieval. These models were then incorporated into the design of an "intelligent" document-based retrieval system. Following a discussion of this system, we discuss the broader implications of our research for the design of information retrieval systems.

1. INTRODUCTION

The cognitive processes involved in human problem formulation and problem solving have been the focus of studies for researchers in the areas of Cognitive Psychology and Artificial Intelligence for the last two decades. Empirical research on the way human problem solvers address simpler questions such as cryptarithmetic problems and games has provided the foundation for information processing theory [8]. A large body of research in recent years has concentrated on problem solving in real world domains such as program debugging [9], mathematical programming [10], physics problems [11][12], and conceptual data modeling [13]. Significant findings pertaining to the representation of knowledge, the identification of effective problem solving strategies, and the nature of expertise in complex task environments have been derived.

Ramaprasad recently proposed applying the cognitive process research approach to information systems design. As he argued in [14]:

> Cognitive process focuses on specific influences on a person's cognitive information processing. Its focus is micro, on the elements of a person's cognitive information processing. These are, for example, perceiving and recognizing stimuli, remembering and searching information, inducing rules, recognizing patterns, formulating concepts, and applying all these in sensing, formulating, and solving problems. . . . To provide useful input to the designers it will be necessary to focus the research upon cognitive processes—the influence of strategies and structures upon cognitive information processing.

In this article, we present the results of applying such an approach to the problem of online document-based information retrieval. Specifically, we conducted two empirical studies investigating the cognitive processes involved during online document-based retrieval. We identified the searchers' and the information specialists' process models during information retrieval and then used these process models to construct an "intelligent" document-based system.

In Section 2, we summarize prior research. We then describe in detail the research design and method used in our empirical studies. The results of these studies, which are presented as a taxonomy of five search strategies, are described in Section 3. In Section 4, we report the design of a document-based system that rests upon these process models. We summarize our system evaluation in Section 5. The implications of our research to the design of document-based retrieval systems are presented in the last section.

2. DOCUMENT-BASED INFORMATION RETRIEVAL

While archival information sources are becoming increasingly computerized, access to such information is often difficult. This is due in large part to the indeterminism involved in the process by which documents are indexed and to the latitude searchers have in expressing a query.

In this research we focused on computerized *document-based systems.* These systems store data, usually in text format, and retrieve it upon user request [15]. Swanson and Culnan provided the following definition for document-based systems [16]:

> An information system may be said to be a *document-based system* when it is based primarily upon a store or collection of documents, rather than a store or collection of structured data. . . . A *document* is an ordered set of recorded visual images constructed so as to communicate as a whole. Examples of documents are: this article, a book, a motion picture film, a bank check, an engineering drawing, and a musical score.

A typical document-based retrieval system consists of a database of documents, a classification scheme to index the documents, and an online system to access the documents.

2.1 *Search uncertainty*

Search uncertainty is the primary source of problems in information retrieval. Search uncertainty arises because searchers have latitude in choosing *terms* to express a query and the *search strategies* they employ in acquiring information. Search terms cause two problems. First, an exact match between the searcher's terms and the index terms is unlikely. This is referred to as the *term matching* problem. Second, the search terms used may not in fact truthfully represent what the searcher is really looking for. This is referred to as the problem of *query articulation.*

In addition to the considerable latitude involved in the terms a searcher may employ to describe a subject area, the approach to carrying out the search may also vary. This approach or process of performing the search is generally referred to in this article as the *search strategy.* It is also referred to as the *process model of information retrieval.*

In prior research, the role of search strategy in aiding online search has been well recognized. Fenichel [17], in summarizing the results of several online search studies, concluded that for both experienced and inexperienced searchers the major problems of information retrieval were with search strategy, not search mechanics. A substantial group of experienced searchers perform simple searches that make use of few of the capabilities of online systems.

In a card catalog study by Tagliacozzo [18], two strategies for searching were identified: a "self-reliant" style where searchers generate their own search terms and a "catalog-oriented" style where searchers use the terms found in the card catalog.

Lancaster classified the search strategy in terms of the critical decision points faced during an online search. Two types of decision points occur during search: a decision to react to unfavorable results and a decision to revise search logic [19].

Bourne identified two search strategies. In the "building-block" strategy, one enters various terms as separate search statements. After the search results are derived, one combines all search statements into a single final statement using the Boolean operator, AND. This strategy contrasts with the "pearl-growing" strategy, in which one initially searches on a few specific terms to retrieve some citations. These citations are then examined carefully for new candidate search terms to be added to the subsequent searches [20]. These two strategies were also verified in Palmer's study [21].

Bates identified 29 *search tactics* [22]. A *search tactic* is a move or maneuver made to further a search. It can be considered as the basic constituent of a search strategy. She grouped these tactics into four categories: monitoring tactics, file structure tactics, search formulation tactics, and terms tactics.

Smith, Shute, and Galdes, after analyzing the discourses between an expert information intermediary and 17 real information seekers, identified 19 *knowledge-based search tactics* [23]. *Knowledge-based search tactics* are defined as the search tactics that result in a change in the topic of interest. Such a taxonomy, however, does not identify the cognitive processes responsible for generating such changes. It simply characterizes the changes in terms of their end results.

2.2 *Research design and method*

Research setting. Prior research has provided a sound basis for understanding the retrieval process. However, in order to identify the cognitive processes involved during online document retrieval, in-depth, empirical, cognitive-psychology based investigation is needed. We conducted two empirical studies of this nature. Data collection techniques included think-aloud protocols, tape recordings of the interactions, interviews, and questionnaires.

The first study observed the interactions between searchers and reference librarians using a retrieval system (34 subjects). These subjects were patrons of the New York University (NYU) library. The second study, on the other hand, studied the interactions between searchers and a retrieval system without the assistance of information specialists (30 subjects). Both studies involved the use of the searchers' real queries. These subjects were mainly NYU students, ranging from freshmen to Ph.D. candidates and were mostly business majors.

Subjects were asked to perform a search for documents on a topic of their choosing. The queries were restricted to "subject-based search" (instead of search involving a known item such as author or title). The authors tape-recorded each subject's initial query at the beginning of a search and acted as passive observers, providing no clues or assistance to the searchers. Searchers were asked to think aloud during their search and the think-aloud protocols were tape-recorded. Occasionally, the authors prompted the subjects to think out loud. Subjects' interactions with the retrieval system were also logged. After the search, an interview and a questionnaire, which were aimed at eliciting the subject's background and comments, were administered. In the interview, we also asked the subjects to summarize their approaches (to searching) and any questions or suggestions they might have. Each search lasted between 5 minutes and 45 minutes, with a median at about 25 minutes. The interviews, questionnaires, think-aloud protocols, and log files were then analyzed. The think-aloud protocols and log files, in particular, were extremely useful for identifying the search process.

Most interactions involved the use of the NYU online catalog system, Bobcat, which lists over 600,000 catalog records, including all new materials purchased after 1973 and many older items previously listed in the card catalog; journals are not listed under Bobcat. The system provides seven search options, namely, title search, author search, combination of author and title search, subject search, number search, keyword search, and Boolean search. These options are available in most online catalog systems. The subjects' familiarity with the system varied widely: A few searchers had never used the system before and some expert searchers knew the system's functionality very well. Most of the subjects, however, were occasional searchers. They used the system about once or twice a month.

Data analysis. The technique we used to analyze the collected data was *protocol analysis*, which involves detailed, qualitative analysis of the think-aloud protocols and logs. Protocol analysis may require categorizing the protocols and logs according to the processes that could have generated them [24]. Various information processing concepts and tools, such as *problem space* and *Problem Behavior Graph*, are important for identifying these processes.

In order to solve a task, a person needs to create an internal representation of the task. This internal representation can be modeled as a *problem space*. A *problem space* can be envisioned as a set of nodes that indicate the knowledge states involved in the problem-solving process. A problem-solving session can be perceived as a traversal of a set of states, starting from the *initial state*, using various operators to move to different intermediate states, and finally reaching the *goal state*, a knowledge state that contains the solution to the problem. The approach a problem solver adopts in traversing the problem space is referred to as the *search strategy* or the *process model* of problem solving. A problem-solving process is in essence the formation of the problem space elements in response to demands of the task environment [25,8].

Different problem solvers, varying in their expertise in the task, may possess different knowledge states, adopt different operators, and display different search strategies. A large body of research has reported the differences between the ways novices and experts solve real world problems. These research findings help identify reasons for the better performance of experts. Examples of the domains that have been studied include: program debugging [9], mathematical programming [10], physics problems [11,12], and conceptual data modeling [13].

A number of studies used the *Problem Behavior Graph* (PBG) to study the problem-solving process. This representation describes problem-solving activities in a time sequence, from an initial state (a vague description of needs) to a goal state (a solution that satisfies the needs). This detailed representation of the problem-solving process is derived by first splitting up the searcher's interaction logs and verbal protocols into their *semantic elements*, which consist of the *knowledge elements* and the *operator elements* [8,26]. The *knowledge elements* specify the kinds of knowledge the subject has about the task. The *operator elements* are a finite set of actions that take a state of knowledge as input and produce a new state of knowledge as output. Problem solvers typically use a finite number of operators to change the knowledge states.

Bouwman compared the decision-making processes of experts and novices in a financial analysis task [27]. The verbal protocols obtained in this study were used to construct a PBG to indicate the problem solver's process model of financial analysis. The study revealed that the financial analysis task can be broken down into three stages: examination of existing information, integration of observations and findings, and reasoning. These findings were used to construct computer programs that simulate financial experts' behaviors. Huguenard, Prietula, and Lerch presented a study in which the fragility of expertise in *reactive scheduling* was investigated [28]. A PBG was used to identify the various strategies taken by the experts and novices, and these researchers observed that simple modifications to the task environment were sufficient to degrade the performance of the experts, sometimes to the level of the novices.

One conclusion that can be drawn from research investigating the problem-solving activity is that different problem solvers may possess different knowledge elements and search methods, which affect the formulation of problem, the selection of strategy, and eventually the performance of the task. *Protocol analysis* has been proven to be an effective technique for identifying the "intelligence" of problem solving in a complex, real world task environment.

2.3 *Protocol analysis for document retrieval*

In order to make analysis of online document retrieval meaningful, we represented the online search process in terms of a PBG. In the online retrieval task environment, the essential knowledge elements are indicated by the *terms* used in the search. The operator elements are the operations performed by searchers. As shown in Table 1, we identified two classes of operators. First, operators may be used for *searching*. Eight search options were used by the searchers. They were: author search (AUT), title search (TIL), controlled subject search (SUB), keyword author search (AUTK), keyword title search (TILK), keyword subject search (SUBK), Boolean search (BOL), and number search (NUM). Second, operators may be used for *scanning* citation records. *Brief scanning* (BS) examines the summarized information about a book, including title, author, and call number. *Full scanning* (FS), on the other hand, examines the detailed information about a book including author, title, publisher, notes (abstract), index terms assigned to the book, and physical features of the book.

With these semantic elements, we were able to construct a PBG to describe the progression of online search. Two constructs are used in the PBG: nodes and labeled links. The nodes in the PBG, which have also been referred to as the knowledge states or knowledge elements, represent the characteristics of the task at a given point in time during problem solving. The labeled links represent operators that accomplish transition from one state of

Table 1. Semantic elements for online subject-based search

Semantic Elements	Description
Knowledge Element:	
TERM	Terms used by the searcher.
Operator Element:	
Searching:	*Online search options.*
AUT	Author search.
TIL	Title search.
SUB	Controlled subject search.
AUTK	Keyword author search.
TILK	Keyword title search.
SUBK	Keyword controlled subject search.
BOL	Boolean search.
NUM	Number search (ISBN and call number).
Scanning:	*Examine the citation information.*
BS	Scan the summarized citation record.
FS	Scan the full citation record.

the task to another. Figure 1 shows a search example in which a searcher subject looked for documents about the "analytical hierarchy method of decision making." In the left hand part of the figure, the search process is schematized as a PBG in a time sequence (from the top of the figure to the bottom). The knowledge states the subject acquired are indicated by the darkened round-cornered boxes. The operators that move the process from one state to another are shown by the arrows linking the knowledge states. The search results, which are the outcome of applying certain operators on the knowledge states, are shown in the ovals adjacent to the knowledge states. The verbal protocols associated with each knowledge state and the strategies used during the search are shown on the right hand side of the figure.

In this example, the subject initially used the term "analytical hierarchy method of decision making" (the first darkened box on the upper left hand corner of Fig. 1). After deriving no matches, the subject used a broader term, "analytical," in keyword title search (TILK) instead. But the results derived (>50 matches) were judged irrelevant by the subject. She then changed the search term to "analytical hierarchy method," a term that is narrower than "analytical" but broader than "analytical hierarchy method of decision making." No matches resulted. The above process lasted for three minutes and is referred to as the *TRIAL-AND-ERROR* strategy (the top shaded-edge box in Fig. 1). Details about this strategy will be discussed in the next section.

The subject did not give up at this point, however. After pondering for a few seconds, she decided to handle the problem with a different approach. Instead of using title search or subject search, she used an author search by identifying a known author in the area, "Saaty, T." The subject's knowledge in the domain area (knowing some authors) contributed to the selection of this approach. By using "Saaty, T." in the author search (AUT), she found 10 books written by the author. Two of them were judged relevant. By scanning the detailed information within the books, she obtained two index terms that were assigned to the relevant books: "decision making" and "system analysis" (see the protocols associated with the node "Saaty, T."). The subject used one of the two index terms, "decision making," to perform a subject search (SUB). Other relevant books were derived. This part of the search process indicates a strategy that we referred to as *KNOWN-ITEM-INSTANTIATION* (the second shaded-edge box in Fig. 1). It is in general effective and efficient.

After obtaining some books classified under "decision making," the subject was lured by the information displayed on the screen. The system displayed a list of index terms that are alphabetically close to "decision making," in particular those index terms with subdivisions (e.g., "—bibliography," "—case studies," etc.). These index terms appeared to be relevant to the subject's query. She spent more than five minutes browsing the screen. However, no relevant matches were derived. This strategy is referred to as *SCREEN-*

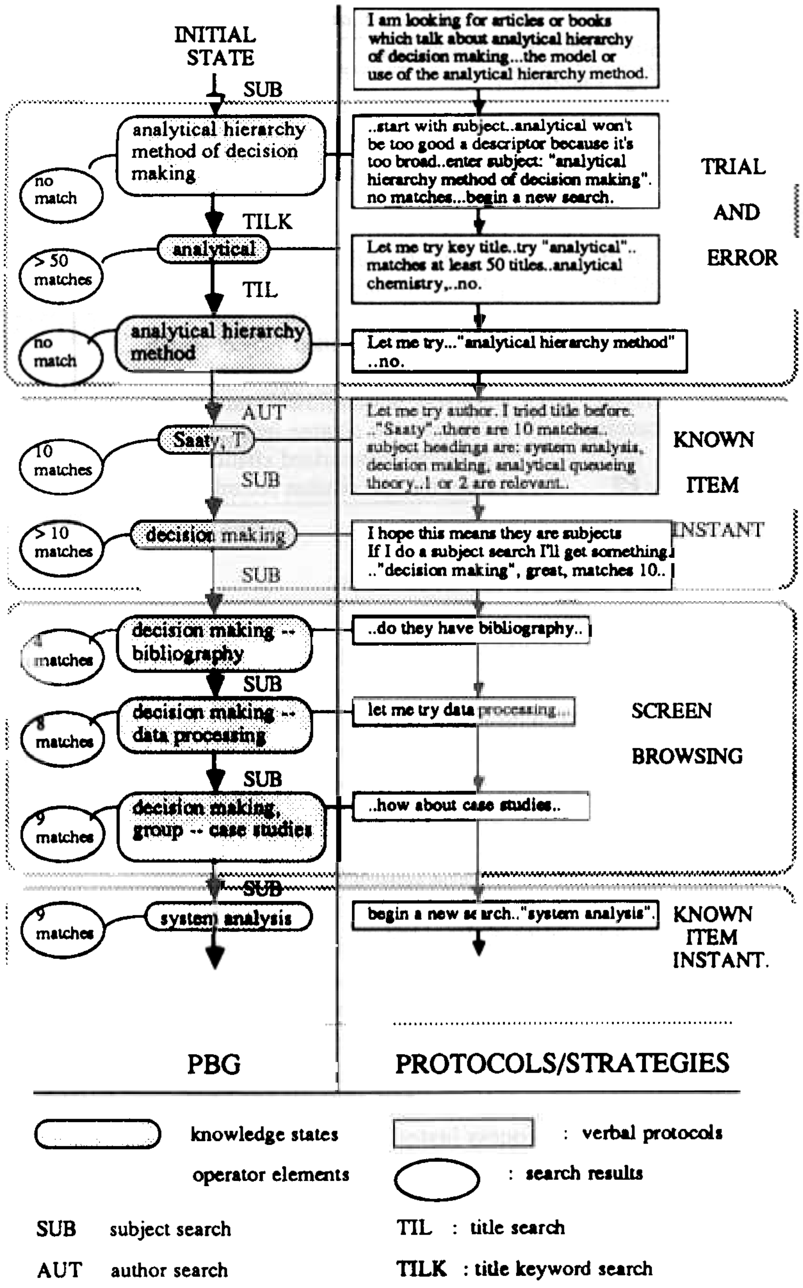

Fig. 1. A PBG example.

BROWSING (the third shaded-edge box in Fig. 1). After this stage, the subject used the other previously derived index term, "system analysis," in a subject search (SUB).

In the remaining part of the interaction (which is not shown in Fig. 1), the subject used another two authors, 'Shaanteu" and "Stohr" in the search (an indication of attempting the KNOWN-ITEM-INSTANTIATION strategy). She also used other terms such as: "pair-wise" and "information economics," in an attempt to ensure the completeness of the search. The whole search lasted for 35 minutes.

3. PROCESS MODELS OF DOCUMENT-BASED RETRIEVAL: SEARCH STRATEGIES

We constructed one PBG for each interaction and then categorized each PBG according to the characteristics underlying the process. Five distinctive process models of online information retrieval (search strategies) were identified. We describe them below.

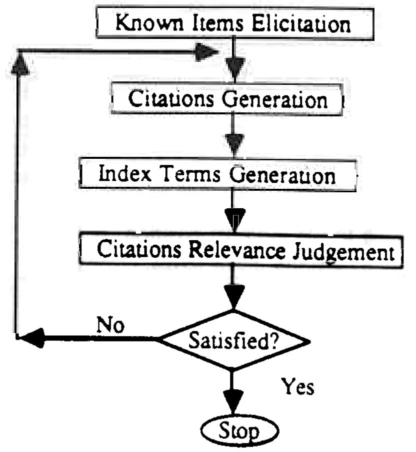

Fig. 2. KNOWN-ITEM-INSTANTIATION strategy.

3.1 *A taxonomy of search strategies*

The search strategies derived from our two empirical studies were: the *KNOWN-ITEM-INSTANTIATION* strategy, the *SEARCH-OPTION-HEURISTICS* strategy, the *THESAURUS-BROWSING* strategy, the *SCREEN-BROWSING* strategy, and the *TRIAL-AND-ERROR* strategy. We present these search strategies using flowcharts, with boxes indicating the stages involved during problem solving, arrows indicating the flow of problem solving, and diamonds indicating the major decision points.

3.1.1 KNOWN-ITEM-INSTANTIATION *strategy.* The *KNOWN-ITEM-INSTANTIATION* strategy was formalized because of the observation that some searchers used the known item search options first (author, title, and call number) in locating some initial relevant documents (*Known Items Elicitation* and *Citations Generation* in Fig. 2). After those relevant documents were retrieved, searchers could obtain index terms that describe the contents of these documents (*Index Terms Generation* in Fig. 2). This process of instantiating the information in known citations is iterative in nature. As shown in Fig. 2, if the searchers were not completely satisfied with the search results, they could use the newly-derived index terms to perform a subject search (represented by the arrow pointing back to *Citations Generation* in Fig. 2).

We can think of this strategy as an online variant of the *pearl-growing* strategy identified by Bourne [20]. The example we described in the previous section illustrates this strategy. The searcher used the index terms derived ("decision making" and "system analysis") from a known item search ("Saaty, T.") to perform a subject search. The problem of matching the searcher-supplied terms with the index terms (the *term matching* problem, described earlier) was avoided because the searcher could use the index terms directly).

This strategy, however, may not be applicable for all searchers. Searchers must have knowledge about some authors, titles, or call numbers in order to retrieve at least one book in the subject area. The strategy is suitable for searchers who have some subject area familiarity.

3.1.2 SEARCH-OPTION-HEURISTICS *strategy.* Searchers used different online search options to perform subject-based searches.* In particular, controlled subject search† (SUB), title search (TIL), keyword subject search (SUBK), and keyword title search (TILK) were the four most frequently used search options. Each option appeared to be suitable for different situations. In our studies, the reference librarians and the searchers with good system knowledge used this strategy frequently. Based on the searchers' stated heuristics and our analysis of the characteristics of these options, we derived a set of heuristics that indicate appropriate situations for applying each search option.

(1) *Heuristic for controlled subject search:* Controlled subject search is appropriate if the index terms have already been identified, possibly via other strategies (such as the

*Search for unknown documents in some subject areas.
†Online search using index terms assigned by the library.

Table 2. Heuristics for different search options

Search Options	Precondition	Effect
Controlled Subject Search (SUB)	Index terms are known.	Derive other relevant index terms.
	Search term has one or more words.	Match alphabetically close index terms.
	Term is likely to appear in the leftmost position of some index terms.	
Keyword Controlled Subject Search (SUBK)	For one word search term.	Match all index terms with the term in some position.
	Term is likely to appear in some position of the index terms.	
Title Search (TIL)	Search term has one or more words.	Match alphabetically close titles.
	Term is likely to appear in the leftmost position of some book titles.	
Keyword Title Search (TILK)	For one word search term.	Match all citations with the term in some position of the titles.
	Term is likely to appear in some position of the titles.	

KNOWN-ITEM-INSTANTIATION strategy). If the index terms are not known, this option can act as a *string matcher*, generating a list of alphabetically close index terms. This option is appropriate for search terms consisting of more than one word* and when the search terms are likely to appear in the leftmost position of some index terms. For example, by using "international corporation" in SUB, a searcher can get index terms such as "international corporation—United States," "international corporation"—Japan," etc.

(2) *Heuristic for keyword subject search:* Keyword subject search takes only single-word input from the searchers. This word, however, can appear in any position of the index term. It is therefore a more exhaustive search option than the controlled subject search, which only matches index terms from the leftmost position. For example, by using "planning" in SUBK, the system can match index terms such as "hierarchical planning," "system planning," etc.

(3) *Heuristic for title search:* The title of a book usually reflects its content. If a searcher uses words likely to appear in the title of a book, title search performs a function similar to controlled subject search. Title search is appropriate for multi-word queries where search terms are likely to appear in the leftmost position of book titles. For example, by using "corporate finance" in TIL, searchers can find books with titles like "corporate finance, an introduction," "corporate finance for MBAs," etc.

(4) *Heuristic for keyword title search:* Similar to title search, keyword title search is appropriate when the search term is likely to appear in the book title. However, only one word can be used in keyword title search. Since the keyword title search option finds all books that have the specified term in some position of the title, the set of matched citations is often larger than that matched by a regular title search. It is appropriate for searchers who need an exhaustive search. For example, by using "deregulation" in TILK, searchers can find all books that have the term "deregulation" in the title, such as: "airline deregulation," "deregulation of the banking industry," etc.

The *SEARCH-OPTION-HEURISTICS* strategy makes use of the applicability of various online search options. The four search options discussed above, along with the heuristics for applying them, are summarized in Table 2. In order to improve search performance, it is important to evaluate the characteristics of the searcher's problems and to choose the

*A keyword search options available on the retrieval system we studied, Bobcat, can only take one word. The keyword search options in Bobcat are appropriate for single-word search terms.

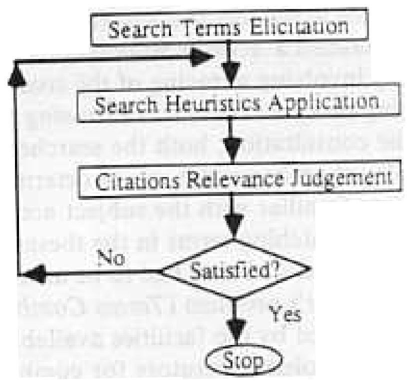

Fig. 3. SEARCH-OPTION-HEURISTICS strategy.

appropriate search options. This process is indicated by the box labeled *Search Heuristics Application* in Fig. 3.

For example, one subject used "airline industry" in a controlled subject search in a fruitless attempt to search for materials about airline industry deregulation (the system generated a lot of materials about the airline industry, but nothing on deregulation). The search would have been more successful if the searcher had performed a title search with "airline deregulation" or a keyword title search with "deregulation." Terms like "airline deregulation" or "deregulation" are likely to appear in the titles of books devoted to this topic. Similarly to the KNOWN-ITEM-INSTANTIATION strategy, the SEARCH-OPTION-HEURISTICS strategy is also iterative in nature. Searchers can use different search terms and search options repeatedly until they are either satisfied or decide to give up.

3.1.3 THESAURUS-BROWSING *strategy*. Many reference librarians adopted the *THESAURUS-BROWSING* strategy in assisting searchers. They alleviated the *term matching* problem by consulting the thesaurus. Once the index terms were identified, the indeterminism involved in the search was greatly reduced.

In order to retrieve documents that would address the specific needs of the searcher, librarians solicited search terms from the searcher (the box labeled *Search Terms Elicitation* in Fig. 4). These searcher-supplied terms then had to be sharpened and translated into

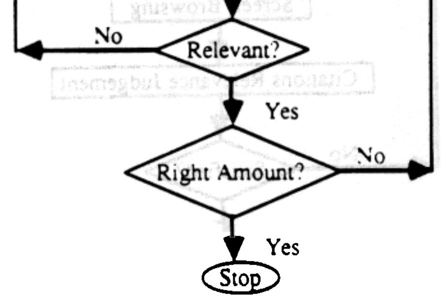

Fig. 4. THESAURUS-BROWSING strategy.

index terms. The chance of terms in the searcher's query matching index terms is generally low. The librarians therefore initiated a *terms translation* process which often included consulting the thesaurus (possibly involving a tracing of the cross referencing structure of the thesaurus) and a brainstorming process (*Thesaurus Browsing* and *Index Terms Generation* in Fig. 4). In this stage of the consultation, both the searchers' and the librarians' familiarity with the subject area played an important role in determining the appropriate terms. If a searcher or a librarian was familiar with the subject area, more terms could be proposed, increasing the chance of matching terms in the thesaurus.

After the index terms were generated they had to be arranged in a way that expressed the semantic content of the searcher's problem (*Terms Combination* in Fig. 4). The combination of terms is generally limited by the facilities available on the system being used. Many online databases provide Boolean operators for combining terms. Combining the terms results in a listing of documents that match the structured query. If the resulting set contains too many documents, the query must be tightened; this can be done by substituting ANDs for ORs in the query and/or regenerating more specific terms by consulting the thesaurus. Similarly, if the resulting set is too small, the query can be loosened by substituting ORs for ANDs or, as before, regenerating broader terms from the thesaurus (see the arrows leaving from the two decision points: "*Relevant?*" and "*Right Amount?*").

A consultation terminated when a reasonable number of relevant documents had been found. This strategy is similar to the *building-block* strategy proposed by Bourne [20]. Moreover, this strategy captures the dynamic (sometimes iterative) nature of consultation in a flowchart (see the loops in Fig. 4). Using this strategy, a human information specialist is able to generate index terms systematically. A detailed discussion of the role of the information specialists in reducing search indeterminism appears in [29].

3.1.4 SCREEN-BROWSING *strategy.* Most online catalogs generate a list of index terms alphabetically adjacent to the user's search terms (*Alphabetical Index Terms Generation* in Fig. 5). Browsing the list of alphabetically close index terms displayed on the screen has become one principal strategy of performing online subject-based searches (*Screen Browsing* in Fig. 5). Sometimes this strategy may be useful because of the morphological characteristics of words. However, in most cases this strategy is likely to be unproductive. This is because terms that are semantically relevant may not be syntactically similar and vice versa. Searchers may spend a lot of time browsing the screen fruitlessly (see the loop in Fig. 5).

For example, a subject who had a query, "bargaining problem in game theory and incentive design," used the *SCREEN-BROWSING* strategy extensively. The subject derived some initial matches by using "game theory." He then spent the next 20 minutes or so

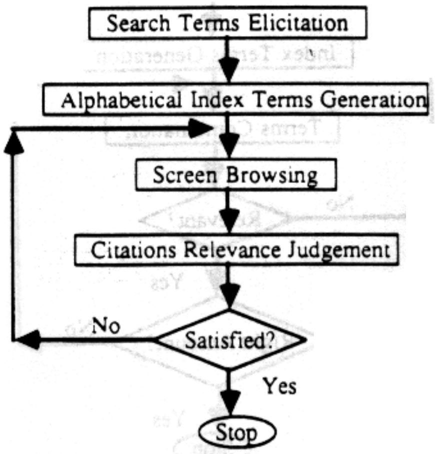

Fig. 5. SCREEN-BROWSING strategy.

browsing the index terms and citations displayed on the screen. No relevant information was derived from this process. The example we described in the previous section also used the SCREEN-BROWSING strategy (indicated by the third shaded-edge box in Fig. 1).

3.1.5 TRIAL-AND-ERROR *strategy*. The least effective search strategy is the *TRIAL-AND-ERROR* strategy. Under this strategy, searchers use whatever terms they have in their minds. The retrieval system merely performs a string matching function (*String Matching* in Fig. 6). If the system matches some relevant index terms, the searcher can employ them to generate citations; otherwise, the searcher will try other terms. Searchers do not use any clues provided on the screen or consult the thesaurus. The search process is essentially one of trial and error. It was often used by searchers who had little knowledge about the system's functionality and the classification scheme.

For example, a subject who was looking for marketing-related information about the soft drink industry used this strategy extensively. She repeatedly used the terms "beverage," "soft drink," and "Coke" in subject search (SUB) in a fruitless attempt to obtain relevant information. The example in Fig. 1 also involved the use of this strategy. The subject used "analytical hierarchy method of decision making," "analytical," and "analytical hierarchy method" in three consecutive steps before she decided to give up on this strategy and used the KNOWN-ITEM-INSTANTIATION strategy.

3.2 *Analysis of the search strategies*

It is alarming that the least effective and efficient strategies, namely, the SCREEN-BROWSING and the TRIAL-AND-ERROR strategies, were used overwhelmingly by the subject searchers. The majority of the reference librarians and the sophisticated searchers, on the other hand, adopted the KNOWN-ITEM-INSTANTIATION, the SEARCH-OPTION-HEURISTICS, and the THESAURUS-BROWSING strategies.

We have derived some interesting findings from the analysis of these search strategies and describe them below.

(1) *Searchers' knowledge affects their selection of search strategies.* Our studies revealed that most searcher subjects used more than one search strategy. On an average, between two and three strategies were used by each searcher subject. The SCREEN-BROWSING and TRIAL-AND-ERROR strategies were used most extensively (over 80% of the subjects used these two strategies). We posit that, in order to apply different search strategies searchers need to possess different knowledge. In prior research by Chen and Dhar [30], we reported three knowledge elements that are important for online search: subject area knowledge, classification scheme knowledge, and system knowledge. We postulate that the precondition of applying the KNOWN-ITEM-INSTANTIATION strategy is a high level of familiarity with the subject area (the searchers need to know the subject area well enough to identify some book titles, authors, etc.). The precondition of using the SEARCH-OPTION-HEURISTICS strategy is familiarity with the system's functionality

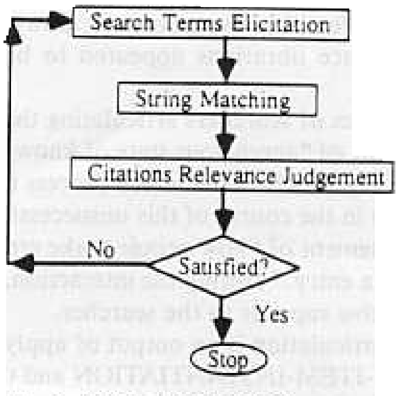

Fig. 6. TRIAL-AND-ERROR strategy

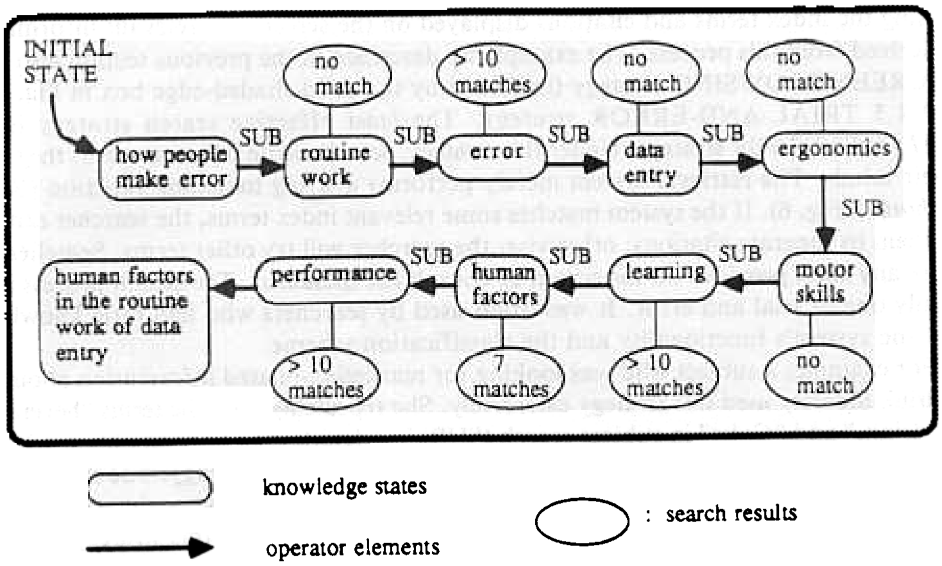

Fig. 7. An example of query articulation.

(system knowledge). Lastly, in order to use the THESAURUS-BROWSING strategy, searchers need to have good classification scheme knowledge (knowledge about the cross-referencing structure, the indexing principle, etc.). We refer to the KNOWN-ITEM-INSTANTIATION strategy, the SEARCH-OPTION-HEURISTICS strategy, and the THESAURUS-BROWSING strategy as *strong methods* of problem solving [31]. Knowledge about the domain, the classification scheme, and the system leads to effective search in the problem space. By exploiting regularities in the task environment, such methods produce behavior that is more specialized and suitable for solving the problem at hand.

On the other hand, the SCREEN-BROWSING and the TRIAL-AND-ERROR strategies are considered *weak methods* of problem solving. They are somewhat similar to other general problem-solving methods such as: means-ends analysis, generate-and-test, goal reduction, and hill climbing [32,33], which require no special task-related knowledge. (The TRIAL-AND-ERROR strategy is similar to the generate-and-test method.) The behaviors these methods produce are simpler, and often less efficient and effective.

(2) *Query articulation is the result of applying search strategies.* In addition to revealing five online search strategies, the searchers' PBGs also disclosed the process of articulating queries. Searchers tended to approach a search by specifying broader terms first. There might be several reasons for this. One hypothesis is that searchers often did not have "queries," but what Belkin calls an "anomalous state of knowledge" [34]. Searchers often expected to refine this anomalous state into a query, *through* an interactive process. The organization of a catalog or a system, however, does not always facilitate this type of query refinement. In contrast, reference librarians appeared to be particularly adept at this function.

We observed ample examples of searchers articulating their queries in our studies. A good example of transition from an "anomalous state of knowledge" to a precise statement of the problem is exhibited in Fig. 7, where the search process is represented in a PBG. The intermediate states cropped up in the course of this unsuccessful interaction. The searcher progressed from the vague statement of "how people make errors" to that of "human factors in the routine work of data entry." During the interaction, the system's responses and messages provided no interactive support to the searcher.

We postulate that query articulation is an output of applying search strategies. Some strategies such as the KNOWN-ITEM-INSTANTIATION and the THESAURUS-BROWSING strategies, appeared to be effective in helping searchers articulate queries.

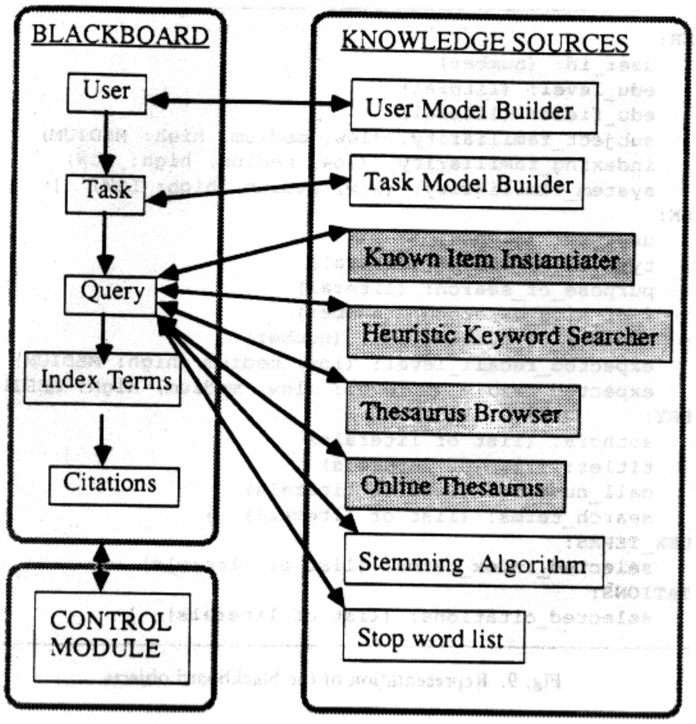

Fig. 8. The blackboard architecture of Metacat.

4. A COGNITIVE PROCESS-BASED DESIGN

In this section we present a cognitive-process based design for document retrieval systems. This design was grounded on the findings we derived from our empirical studies. Our system adopted the *blackboard architecture.* This architecture has been applied successfully in the design of various knowledge-based systems [35,36,37].

The first *blackboard-based system* was the HEARSAY-II speech understanding system [35] that evolved between 1971 and 1976. Subsequently, many systems that have similar organization have been built. Typically a blackboard structure consists of three components [38,39]. First, the knowledge needed to assist the system's users is partitioned into different *knowledge sources.* Each knowledge source acts as a small expert. Second, the data and the partial results involved in the problem-solving process are kept in a global database, the *blackboard.* Lastly, a *control module* monitors the sequence of executions of the system.

In our system, Metacat, five types of data are posted on the blackboard (as shown on the left of Fig. 8). All of them are represented through the use of *frames.* Each frame consists of between one and seven attributes (slots). The five data types are organized as a hierarchy in the blackboard to depict their closeness to the searcher's information need. Data that are more relevant to the searcher's information need are posted closer to the bottom of the hierarchy. At the highest level, a *user model* is built, which captures the long-term characteristics of the searcher, such as the searcher's educational background, the searcher's educational level, etc. At the second level, a *task model* is constructed to represent the searcher's task-related information, such as the type of material, the recency of the publications, and the number of relevant citations the searcher desires. At the third level, the system creates a model of the actual *query,* which includes all the query-related inputs supplied by the searcher. These include: titles of books, authors' names, call numbers, and search terms. At the fourth level, *index terms* are generated by the system to represent the searcher's query. At the bottom level, *citations* that are derived from the index terms are posted on the blackboard.

In Fig. 9, we list these five data types on the blackboard with their associated attrib-

```
{USER:
      user_id: (number)
      edu_level: (literal)
      edu_field: (literal)
      subject_familiarity: (low, medium, high: MEDIUM)
      indexing_familiarity: (low, medium, high: LOW)
      system_familiarity: (low, medium, high: LOW)  }
{TASK:
      user_id: (number)
      type_of_material: (literal)
      purpose_of_search: (literal)
      #_of_hits_expected: (number)
      recency_of_publication: (number)
      expected_recall_level: (low, medium, high: MEDIUM)
      expected_precision_level: (low, medium, high: MEDIUM)
{QUERY:
      authors: (list of literals)
      titles: (list of literals)
      call_numbers: (list of literals)
      search_terms: (list of literals)  }
{INDEX_TERMS:
      selected_index_terms: (list of literals)  }
{CITATIONS:
      selected_citations: (list of literals)  }
```

Fig. 9. Representation of the blackboard objects.

utes. The domains and default values (indicated by the upper case terms in the attributes in Fig. 9) of these five data types are also shown.

Our system contains several knowledge sources for assisting online search. The *user model builder* and the *task model builder* (shown on the right side of Fig. 8) apply a few empirically-derived heuristics in modeling the searcher. The *stemming algorithm* generates legal suffixes for the searcher's terms. The *stop word list* eliminates non-semantic bearing words from the searcher's input. For details about these components, readers are referred to [40]. In this article, we discuss the other four knowledge sources: *Online Thesaurus, Known Item Instantiater, Heuristic Keyword Searcher*, and *Thesaurus Browser* (the shaded boxes on the right side of Fig. 8).

4.1 *Online thesaurus*

Our system includes an *online thesaurus* corresponding to a portion of the LCSH Handbook. It consists of terms and cross references. This online thesaurus is represented as a semantic network where nodes are the terms (the index terms and the non-index terms) and links are the relationships between terms. We schematized this network in Fig. 10. A frame-based representation for this network is shown in Fig. 11. The slots of each term frame in Fig. 11 indicate the cross reference relationship including NT (narrower term), BT (broader term), RT (related term), USE (synonymous official term), and UF (synonymous unofficial term). The other two slots in the term frame, number of matched citations and titles of matched citations, are considered *if-needed facets* (values are computed and obtained when they are needed).

We derived our semantic network-based online thesaurus by extracting a portion of the computer readable form of the LCSH Handbook (with the assistance of OCLC*). Our online thesaurus consists of nearly 3,500 terms (both index terms and non-index terms) in the areas of mathematics and computer science (the areas we chose for the system evaluation). Each term has between a couple of and a few hundred relevant terms associated by means of the cross referencing structure of the thesaurus. This online thesaurus represents the domain knowledge of the system. The objects on the blackboard and the online thesaurus were represented in FLAVORS, a representation language embedded in Franz Lisp that

*Online Computer Library Center, Inc., Dublin, Ohio.

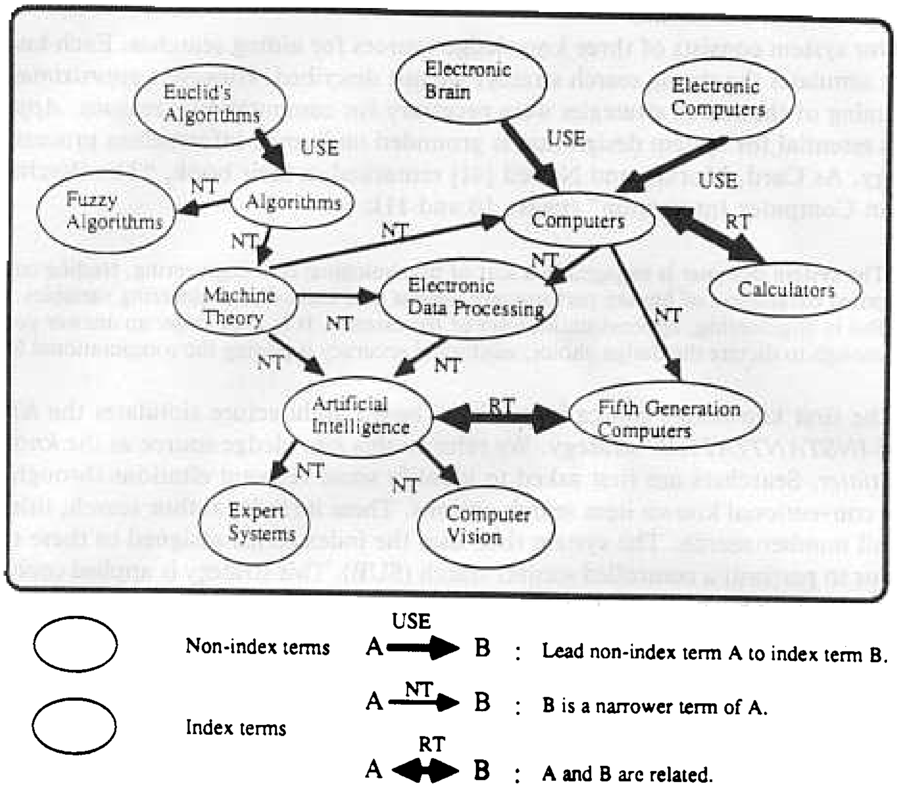

Fig. 10. A portion of an LCSH semantic network.

incorporates many of the features of object-oriented programming. Since the complete LCSH is in an online format, the process of generating a semantic network representation is easy. With the proper heuristics, the online thesaurus can be extremely useful in helping searchers articulate their queries.

```
Term Object Frame:
  {TERM: (name of the term)
        Type of term: (* for unofficial term; nil for official term)
        NT: (list of narrower terms)
        BT: (list of broader terms)
        RT: (list of related terms)
        USE: (list of synonymous official terms)
        UF: (list of synonymous unofficial terms)
        Number of citations: (if-needed: integer)
        Titles of citations: (if-needed: list of matched titles)
  }

Example of Term Instance:
  {TERM: Computers
        Type of term: nil
        NT: (Electronic-Data-Processing Fifth-Generation-Computers)
        BT: (Machine-Theory)
        RT: (Calculators)
        USE: nil
        UF: (Electronic-Brain Electronic-Computers)
        Number of citations: (if-needed: 56)
        Titles of citations: (if-needed: "Introduction to Computers"
                              "Computers" ...)
  }
```

Fig. 11. Frame-based representation for LCSH.

4.2 *Known item instantiater*

Our system consists of three knowledge sources for aiding searches. Each knowledge source simulates the strong search strategy we just described. However, *approximation* and fine-tuning of the search strategies were necessary for computational reasons. *Approximation* is essential for system design that is grounded on human information processing psychology. As Card, Moran, and Newell [41] remarked in their book, "The Psychology of Human Computer Interaction" (pages 10 and 11):

> The system designer is engaged in a sort of psychological civil engineering, trading computed parameters of human performance against cost and other engineering variables... But in engineering, approximations are of the essence. It is vital to get an answer good enough to dictate the design choice; additional accuracy is gilding the computational lily.

The first knowledge source in our blackboard architecture simulates the *KNOWN-ITEM-INSTANTIATION* strategy. We refer to this knowledge source as the *known item instantiater.* Searchers are first asked to identify some relevant citations through the use of the conventional known item search options. These include: author search, title search, and call number search. The system then uses the index terms assigned to these retrieved citations to perform a controlled subject search (SUB). This strategy is applied continuously in a "chain reaction" way. When a new citation is derived, the system can obtain a few new index terms. These new terms may lead to other new citations, which in turn may suggest other new index terms. By following this process continuously, we can identify a set of relevant index terms and citations.

In Fig. 12 we illustrate this chain reaction of known item instantiation graphically. For example, by identifying the author of BOOK1 ("Salton," as shown in Fig. 12), the searcher can find an initial relevant citation (BOOK1). Book1 is classified under SUB1 (follow the *index* link in Fig. 12). Following the links associated with SUB1, the system identifies BOOK2 and BOOK3. These two books lead to two new index terms, SUB2 and SUB4. These two index terms link to BOOK4, BOOK5, and BOOK6, which in turn lead to SUB3 and BOOK7. This instantiation process continues until no links can be instantiated further.

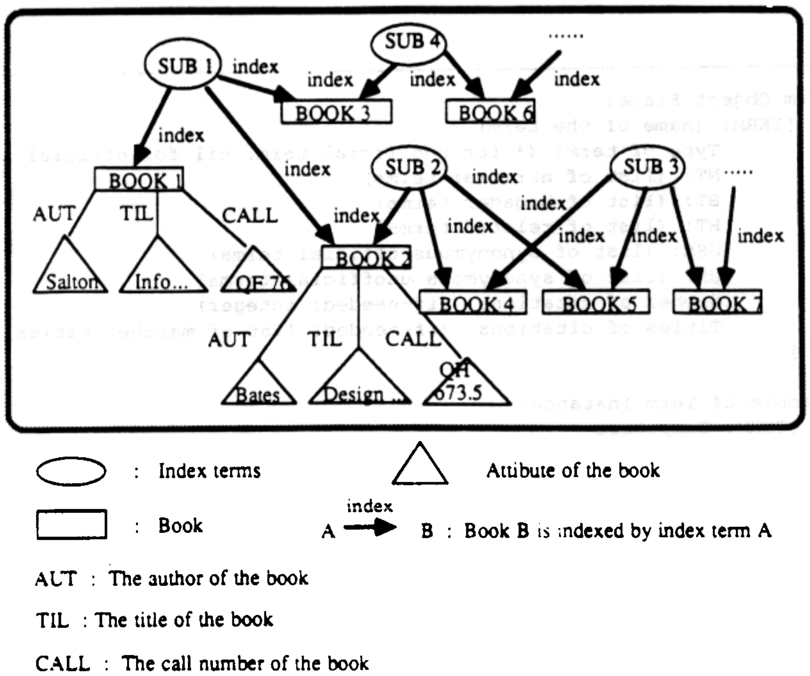

Fig. 12. A graphic representation of known item instantiation.

This module also terminates after it instantiates two levels (or three index terms, e.g., from SUB1 to SUB2, and from SUB2 to SUB3). This stopping condition is for computational reasons and can be relaxed. The computation involved in this process is relatively "inexpensive" (compared to the next two strategies we will discuss). By using the existing search options wisely, a search process can become productive and efficient.

4.3 *Heuristic keyword searcher*

The *heuristic keyword searcher* is another knowledge source that utilizes the existing search options in a new way. It is grounded on the *SEARCH-OPTION-HEURISTICS* strategy. While the known item instantiater exploits the known item search options, the heuristic keyword searcher utilizes the non-known item search options, in particular the controlled subject search (SUB) and the title search (TIL).

This heuristic keyword searcher consists of ten different search functions, which are based on the SEARCH-OPTION-HEURISTICS strategy and engineering considerations (suffixing). Each search function has a different level of *credibility* indicating its likelihood to generate index terms that are (semantically) close to the searcher's search terms. When the system invokes this knowledge source, it first tries the most credible search function. If it fails, then the system proceeds to the next most credible search function. This process continues until either the system finds some index terms by using a certain search function or until the system exhausts all ten search functions. These ten functions are listed in descending order of credibility in Fig. 13. We describe each of them here.

1. SUB with the original search term: Use the original search term in controlled subject search (SUB). This is most credible because if there is a match, the index term would completely represent the search term.
2. SUB with the suffixed search term: If there is no match from the first function, this function applies a stemming algorithm to add suffixes to each word in the search term. For example, if there is no match with "online computer," then the system will try, "online computers," "online computing," etc., automatically.
3. Thesaurus lookup with the original search term: If the search term does not match with any index terms in the thesaurus (using the two functions described above), the system expands its search space by including the non-index terms the system can recognize (via the thesaurus). If the system matches a non-index term in the thesaurus, it can obtain the synonymous index term by following the USE link of the non-index term (see Fig. 10). This function, which is nearly as reliable as the previous two functions, is only a simple one-level thesaurus lookup. No extensive activation of the links in the online thesaurus is required.
4. Thesaurus lookup with the suffixed search term: As in the case of the second function, if there is no match with the original term, our system uses the search term's suffixed forms in thesaurus lookup.
5. SUBK with the original search term: If our system finds no match by using controlled subject search and thesaurus lookup (the previous four functions), it proceeds to perform a keyword subject search (SUBK in Bobcat) with the original search term. Since the keyword search options can find partial matches for the

```
- SUB with the original search term
  - SUB with the suffixed search term
    - Thesaurus lookup with the original search term
      - Thesaurus lookup with the suffixed search term
        - SUBK with the original search term
          - SUBK with the suffixed search term
            - SUBK with the suffixed words derived from the search term
              - TILK with the original search term
                - TILK with the suffixed search term
                  - TILK with the suffixed words derived from the search term
```

Fig. 13. Search functions within the heuristic keyword searcher.

search term, the search space for the system is further expanded (however, the matched terms derived will be less credible than those generated from the previous functions).

6. SUBK with the suffixed search term: As in the case of the second function, our system expands its search space by using the search term's suffixed forms in keyword subject search (e.g., "online computers" and "online computing" from "online computer").
7. SUBK with the suffixed words derived from the search term: The system uses the suffixed *words* that are derived from the search term in SUBK (a search term often consists of more than one word). For example, if there is no match with "online computers" and "online computing" (terms generated from the previous function), then the system will try "online," "computer," "computers," and "computing" for the search. These words (not the original term) are used in SUBK, respectively. By decomposing the search term into suffixed search words, we expand the system's search space significantly.
8. TILK with the original search term: As reported in our empirical findings, title keyword search can assist searchers in performing subject-based search. For online catalog records, which typically have little content information, title is a good source from which to glean the contents of the book. It is, however, less credible than searches using the index term directly (i.e., SUB, thesaurus lookup, and SUBK).
9. TILK with the suffixed search term: Again, we can expand the search space by using the suffixed search term in keyword title search.
10. TILK with the suffixed words derived from the search term: An even larger search space can be obtained by using the individual suffixed *words* derived from the search term.

The rationale behind the heuristic keyword searcher is to expand the search space incrementally until some hits can be found. As indicated in an online catalog design principle proposed by Bates [42], it is crucial for the searcher to get into the system in the first place (i.e., match/find something from the system)—what she described as the (hit the) "side-of-the-barn" principle. This principle suggests an information retrieval system design that provides a "big target" for the searcher to hit. That is, a retrieval system should provide as many entry points as possible. Our heuristic keyword searcher is grounded on this principle. Moreover, instead of merely presenting a big target to the searcher, our system suggests an incrementally expanded target. This *search-space-expansion principle* not only helps identify good index terms but also is computationally efficient.

4.4 *Thesaurus browser*

The last knowledge source of our system is the *thesaurus browser*. This knowledge source simulates the *THESAURUS-BROWSING* strategy described earlier. During a search, our system first invokes the known item instantiater and the heuristic keyword searcher. These two knowledge sources, which exploit the capabilities of the existing search options, generate index terms and citations for the searcher's query. If the searcher is not satisfied with the results derived from these two knowledge sources, the system will then activate the thesaurus browser.

The index terms generated from the above two knowledge sources can match with some nodes (terms) in our online thesaurus. These nodes are taken as the *source nodes* by our thesaurus browser. There are links associated with these nodes. Our system applies a *heuristic spreading activation* process (similar processes have been described in [43] and [5]) on the semantic network-based thesaurus to generate relevant terms. We developed a few heuristics to guide this activation process. They are presented as follows:

1. *The Specific Terms First Heuristic:* Based on the analysis of the LCSH structure, we observed that nodes (terms) that have fewer neighbors in the semantic network are generally more specific (in content) than nodes (terms) that have more neighbors. Since searchers have a tendency to state their information needs more broadly

than they should, our system applies a heuristic that first expands the nodes with fewer neighbors (the more specific terms).

2. *The Specific Links First Heuristic:* Links associated with the index terms are of three types: NT, RT, and BT (the reverse of NT). Our system adopts a heuristic for expanding the links in the order of: NT, RT, and BT. That is, our system will search the NT links first before it searches the RT links, and before it searches the BT links. This heuristic guides our system in activating the more specific links first (which will eventually lead to more specific terms).
3. *The Shorter Distance First Heuristic:* This heuristic is related to the distance between a node and the source nodes. During the activation process, our system will expand the nodes that are closer to the source nodes (shorter distance) earlier than those nodes further away from the source nodes (longer distance). The rationale is that terms that are more remote from the source nodes are less relevant to the source terms than terms closer to the source nodes. Therefore, they should be expanded only after the more relevant terms (closer nodes) are expanded.
4. *The Two-Level Expansion Heuristic:* The number of links between two nodes (terms) in a semantic network indicates the semantic proximity of the terms. In order to find only terms that are closely relevant to the source nodes, our system expands the source nodes by two levels. That is, we activate only nodes that are two links away from the source nodes. (Because each node in the network may have between a few dozen and a few hundred links, two-level expansion still requires a lot of computation.) This two-level expansion heuristic was derived from the query articulation characteristic of searchers (they rarely broadened or narrowed the specificity of their search terms for more than two levels) and an analysis of the cross referencing structure of LCSH. This heuristic ensures that the system only finds terms that are semantically relevant to the source nodes (terms). (A spreading activation algorithm without the level of expansion heuristic will generate many irrelevant terms and is computationally inefficient.)

The four heuristics described, which consider the specificity of the nodes, the specificity of the links, the distance between nodes, and the expansion level, are used to direct our system's *spreading activation* effort. We represent the problem as a *search* task, where the goal is to identify the relevant terms associated with the searcher's search terms.

We assign *costs* to each expanded path on the semantic network based on the nodes visited, the types of links traversed, and the number of links in the path. Cost is a metric we used for indicating the "semantic distance" (relevance) of terms. It is similar to the "Distance" metric used by Rada *et al.* [44], which represents the conceptual distance between concepts. However, Rada *et al.* used only the path length to determine this conceptual distance. Our algorithm considered a number of factors including the specificity of the nodes visited (by counting its neighbors), the types of links (NT, BT, RT, and USE), and the path length. A path that contains more relevant terms will have lower cost than a path that contains fewer relevant terms.

We use a branch-and-bound algorithm to guide the search. This algorithm computes and orders costs for each partial path and expands the least-cost path. It allows our system to locate the optimal (least-cost) path (i.e., identify the most relevant terms in the online thesaurus).

For the initial paths (paths with source nodes only), the system assigns a cost computed as the source nodes' (the initial search terms') total number of neighboring terms. After computation, the cost for the new path is determined as follows:

```
(cost of the path accumulated before expanding to the new node)
  (relative weight of the link) *
    (# of neighboring terms of the new node to visit)
```

For the *weights* of the links, we use the following assignment (relative weights):

```
USE   NT   RT   BT = 3 : 9       1
```

These *relative weights* represent the relative frequency of the links (USE, NT, RT, and BT) used by the searchers in our empirical study of the searcher/system interactions. For example, the searchers were three times (NT : USE = 9 : 3) more likely to follow the NT links than the USE links in query refinement (because they had a tendency to state terms more broadly than necessary). When expanding a path via a USE link, an *actual weight* of 1 is assigned to the link. Since the USE link connects two synonymous terms, the weight of 1 for the USE link will ensure that the new path, which is created by expanding to a synonymous term, has the same cost as the path before expansion (i.e., cost of the new path = cost of the old path * 1). Treating USE as the basis, we assigned the *actual weights* of 3/9, 3/5, and 3/1 to NT, RT, and BT, respectively. For example, the cost for path, {"MATHEMATICS" NT "ALGEBRA"} (i.e., a path expanded from "MATHEMATICS" to "ALGEBRA" via a NT link), is equal to 75 * 3/9 * 14 (i.e., 350); where the cost accumulated before the expansion is 75 (the total number of neighbors of "MATHEMATICS"), the weight of the link is 3/9 (the weight of the NT link), and the cost of the new node, "ALGEBRA", is 14 (ALGEBRA's total number of neighbors). Via this ranking scheme, our system maintains a list of ranked paths. We present our branch-and-bound algorithm in Fig. 14 (the algorithm is based on [32]).

This algorithm is able to alleviate the computational explosion problem that frequently occurs in the semantic network spreading activation process and identifies mostly semantically relevant terms for the searchers.

The output from our branch-and-bound algorithm is multiple connected components. (The algorithm we used to determine the connected components is based on a simple connected component algorithm in [45].) Each component consists of the source nodes and the paths that link these nodes. We referred to these connected components as *concept groups*. Paths and nodes that are connected are considered to be in the same *concept group* (there is at least one path from one node to another node in the concept group). A concept group consists of terms that have strong relationship (or *association*, the term used in Cognitive Psychology [33]). The nodes (terms) on the paths within the same concept group address a similar underlying concept.

This concept grouping process had been frequently used by human information specialists (reference librarians often attempt to identify the various topics patrons try to address during a consultation session). New terms (terms different from the source terms) found in each concept group will become good candidate terms for the searchers' queries. The thesaurus browser ranks the concept groups in order (based on the number of source terms involved in each concept group) and suggests new terms to the searchers. This thesaurus browsing process is useful in assisting searchers to articulate their queries.

```
BRANCH-AND-BOUND SEARCH:
1. Form a queue of paths. Let the initial queue consist of the ''source
   nodes'' only. Sort the queue by a cost calculation
   that is based on the number of the source node's neighbors, with
   the least cost paths in the front of the queue.
2. Until the queue is empty or the goal has been achieved, determine
   if the first path in the queue reaches the goal state.
   GOAL STATE: All source nodes are connected by some paths in the queue.
   2.1. If the first path reaches the goal state, do nothing.
   2.2. If the first path does not reach the goal state,
        2.2.1. Remove the first path from the queue.
        2.2.2. Form new paths from the removed path by extending one step from
               the last node of the removed path.
        2.2.3. If a path has less than the maximal expansion level (2),
               add the new path to the queue.
               If a path has reached the maximal expansion level or the last
               node in the new path has no neighboring nodes to expand to,
               remove the path from the queue.
        2.2.4. Sort the queue by the sum of the cost accumulated so far and
               the cost remaining, with the least cost paths in the front
               of the queue.
3. If the goal state has been achieved, return one group with paths
   that link all source nodes; otherwise, return multiple groups, each group
   consisting of the paths that link the source nodes within the group.
   The source nodes within each group are connected while the source
   nodes in different groups are not linked.
```

Fig. 14. A branch-and-bound algorithm for the thesaurus browser.

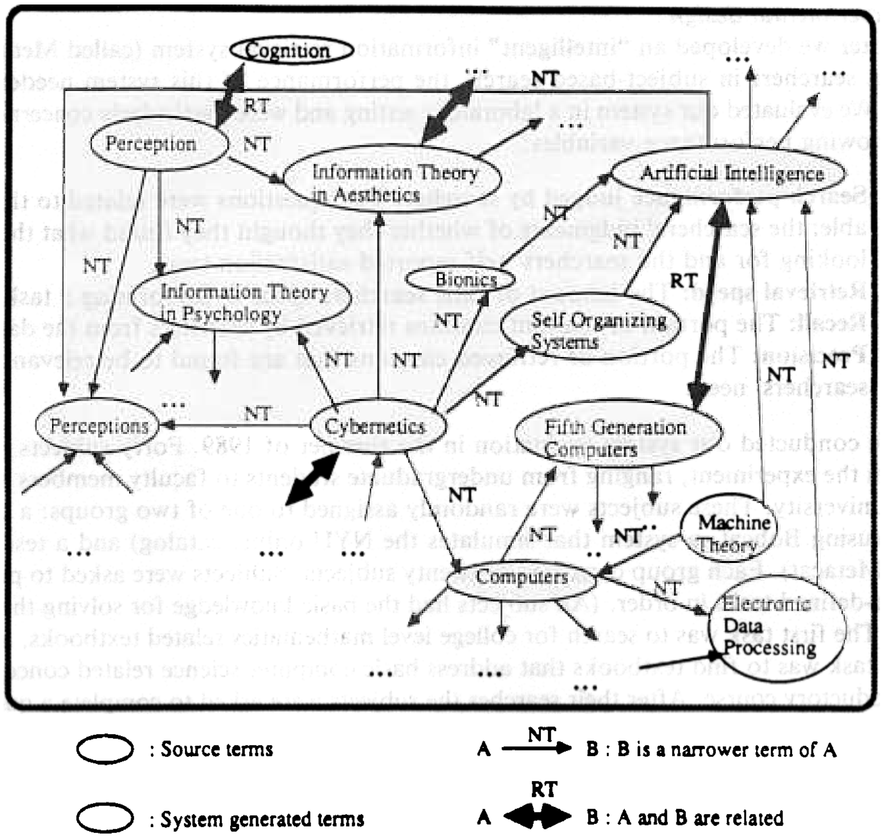

Fig. 15. A graphic representation of thesaurus browsing.

In summary, the thesaurus browsing process generates relevant terms by activating the links in the semantic network-based online thesaurus. This process has been made possible by means of a heuristic-based branch-and-bound algorithm. The thesaurus browser, along with the known item instantiater and the heuristic keyword searcher, are applied iteratively in our system.

In Fig. 15 we present a sample session of thesaurus browsing. We list the source terms (nodes in shaded ovals) and the relevant paths and terms generated from the browsing process (nodes in unshaded ovals). The initial terms consist of: "Cognition," "Artificial Intelligence," and "Cybernetics." The thesaurus browser enables our system to suggest to the searcher other relevant terms in ranked order (e.g., "Perception," "Bionics," "Information Theory in Psychology," etc., in Fig. 15). This example involves only one concept group (all terms are linked together). The whole browsing process explores 345 paths in about 15 seconds.

Our prototype "intelligent" document-based retrieval system, which is called Metacat, was developed in Franz Lisp, Opus 42, and runs on the SUN. Lisp was mainly used to represent the search approaches we identified. FLAVORS, an object-oriented language (which is embedded in Franz Lisp), was used to represent the database records, the online thesaurus, and the data on our blackboard-based system.

5. SYSTEM EVALUATION

This section presents the results of our system evaluation. In the first subsection we describe our experimental design. We summarize the hypotheses for system evaluation in the second subsection. In Subsections 5.3, 5.4, 5.5, and 5.6, we report and discuss the results of the system evaluation. In the last subsection, we present conclusions.

5.1 *Experimental design*

After we developed an "intelligent" information retrieval system (called Metacat) to support searchers in subject-based search, the performance of this system needed to be tested. We evaluated our system in a laboratory setting and were particularly concerned with the following performance variables:

1. **Search performance judged by searcher:** Two questions were related to this variable: the searchers' judgments of whether they thought they found what they were looking for and the searchers' self-reported satisfaction level.
2. **Retrieval speed:** The amount of time searchers spent in performing a task.
3. **Recall:** The portion of relevant citations retrieved by searchers from the database.
4. **Precision:** The portion of retrieved citations that are found to be relevant to the searchers' need.

We conducted our system evaluation in the summer of 1989. Forty subjects participated in the experiment, ranging from undergraduate students to faculty members of New York University. These subjects were randomly assigned to one of two groups: a control group (using Bobcat, a system that simulates the NYU online catalog) and a test group (using Metacat). Each group consisted of twenty subjects. Subjects were asked to perform two pre-defined tasks in order. (All subjects had the basic knowledge for solving these two tasks.) The first task was to search for college level mathematics related textbooks, and the second task was to find textbooks that address basic computer science related concepts for an introductory course. After their searches the subjects were asked to complete a questionnaire eliciting their evaluation of the system, information about the retrieved citations, and their comments about the search and the system. The first task was used for training purposes. This was necessary because most subjects had used Bobcat before, but never used Metacat. We asked three experts (faculty members) to select a list of relevant books from the database for each task. (Each expert selected a list of relevant books by browsing all the 200 books in the database. The books selected by more than one expert were included in the final list of relevant citations.) We obtained a list of 31 relevant books for the mathematics task and the same number of books for the computer science task. Based on the information derived from the subjects and experts, we were able to measure the variables in which we were interested. We report the results concerning both tasks in the following subsections.

5.2 *Hypotheses*

We proposed four hypotheses for system evaluation:

```
Hypothesis 1: There is no difference between the search performance
              of Bobcat and Metacat as judged by searchers using
              the two systems.
Hypothesis 2: There is no difference between the amount of time
              Bobcat searchers spend in performing searches and the
              amount of time Metacat searchers spend.
Hypothesis 3: There is no difference in recall between searches
              on Bobcat and searches on Metacat.
Hypothesis 4: There is no difference in precision between
              searches on Bobcat and searches on Metacat.
```

These hypotheses have also been adopted in other studies that evaluated the performance of online information retrieval systems [1,2,3,4,5,6,7].

5.3 *Hypothesis 1: Search performance*

```
Hypothesis 1: There is no difference between the search performance
              of Bobcat and Metacat as judged by searchers using
              the two systems.
```

The search performance as judged by the searchers was measured by two questions. The first question was: In your first (second) computer search, did you find (1) nothing you

Table 3. Subject's evaluation of information found: Bobcat vs. Metacat

Infor. Found	Task 1		Task 2	
	Mean	Std. Dev.	Mean	Std. Dev.
Bobcat	2.350	0.745	2.550	0.945
Metacat	2.400	0.821	2.250	0.639
Sig. Level	P = 0.840		P = 0.250	

Table 4. Subject's reported satisfaction: Bobcat vs. Metacat

Satis.	Task 1		Task 2	
	Mean	Std. Dev.	Mean	Std. Dev.
Bobcat	3.150	0.813	2.800	0.894
Metacat	2.700	0.865	3.300	0.571
Sig. Level	P = 0.098		P = 0.043	

were looking for, (2) some of what you were looking for, (3) all you were looking for, or (4) more than you were looking for. Scores of 1, 2, 3, or 4, respectively, were assigned to each choice. The second question was: Considering what you were looking for, was your first (second) search (1) very unsatisfactory, (2) somewhat unsatisfactory, (3) somewhat satisfactory, or (4) very satisfactory. Again, a score of 1, 2, 3, or 4 was assigned to each selection. We used the Student's t-test to compare the means of the responses for these two questions between the searches on Bobcat and the searches on Metacat. The statistics are presented in Tables 3 and 4.

There was no statistically significant difference between the searchers' evaluation of the information found by Bobcat and that by Metacat (see Table 3, significance levels are $p = 0.840$ and $p = 0.250$, respectively). For the searchers' reported satisfaction, there was a discrepancy in the results concerning the two tasks (see Table 4). On task 1, the Bobcat group's average reported satisfaction level (3.150) was higher than the Metacat group's (2.700). On task 2, to the contrary, Metacat group's average reported satisfaction level (3.300) was significantly higher than the Bobcat group's (2.800). We postulate that for subjects using Metacat their first search was a learning experience. Errors occurred during the first search. But after performing the first task, most Metacat subjects were able to use the system without difficulty.

5.4 *Hypothesis 2: Retrieval speed*

```
Hypothesis 2: There is no difference between the amount of time
              Bobcat searchers spend in performing searches and the
              amount of time Metacat searchers spend.
```

In our experiment, we recorded the amount of time each subject spent in performing the tasks. Again, we used the t-test to compare the amount of time the Bobcat subjects spent in search and the amount of time the Metacat subjects spent in search. The statistics are shown in Table 5. The unit of measure in Table 5 is seconds.

Table 5. Time spent in search: Bobcat vs. Metacat

Time (in seconds)	Task 1		Task 2	
	Mean	Std. Dev.	Mean	Std. Dev.
Bobcat	818	360	810	405
Metacat	1397	534	935	360
Sig. Level	P = 0.0003		P = 0.310	

Table 6. Recall: Bobcat vs. Metacat

Recall	Task 1		Task 2	
	Mean	Std. Dev.	Mean	Std. Dev.
Bobcat	0.156	0.106	0.098	0.070
Metacat	0.237	0.114	0.174	0.139
Sig. Level	P = 0.026		P = 0.038	

On average, the subjects in the Bobcat group spent less than 14 minutes in performing task 1. The subjects in Metacat group, however, spent almost 24 minutes on task 1. The difference was significant ($p \leq 1\%$). On task 2, however, there was no statistically significant difference between the average amount of time the Bobcat subjects spent (13 minutes and 30 seconds) and the average amount of time the Metacat subjects spent (15 minutes and 35 seconds). Bobcat subjects used about the same amount of time for both tasks, but there was a dramatic reduction in the amount of time the Metacat searchers spent for the second task.

We have two postulates concerning the causes of these results. First, there appeared to be a learning curve for the Metacat subjects. During the course of performing the first task, Metacat subjects spent a lot of time reading the screen instructions, studying the search options, and attempting to understand how the system worked. After the searchers performed the first task, the second task became easy. Second, many subjects wasted time demolishing their initial perceptions of the system before they performed their actual searches. We observed that many subjects had an initial perception of our system (Metacat) as a system like Bobcat. Sometimes, it was only after 10 minutes of interaction that the searchers realized that Metacat functioned differently from Bobcat. Once the subjects became familiar with the system, they proceeded with their searches without problems.

5.5 *Hypothesis 3: Recall*

```
Hypothesis 3: There is no difference in recall between searches
              on Bobcat and searches on Metacat.
```

The operational definition that we used for recall is as follows:

```
Recall = the ratio of the number of relevant retrieved citations to the
         number of total relevant citations.
```

Recall and precision are the most important variables for measuring information retrieval systems' performance [7]. In this study, Bobcat subjects were asked to write down their selections of citations. Metacat, on the other hand, recorded its searcher's selections during the search (part of the system's functionality). The statistics for recall are summarized in Table 6.

On both tasks, Metacat subjects out-performed Bobcat subjects ($p \leq 5\%$). The average recall values of the searches on Metacat were 23.7% on task 1 and 17.4% on task 2. These recall values were 52% and 78% better than the recall values of the searches on Bobcat for task 1 and task 2, respectively. The Metacat's recall values are comparable to the recall value of 20% on STAIRS* [3,46,47] and 19% on MEDLARS†.

5.6 *Hypothesis 4: Precision*

```
Hypothesis 4: There is no difference in precision between
              searches on Bobcat and searches on Metacat.
```

*STAIRS is a full-text retrieval system developed by IBM.

†MEDLARS was considered a study conducted by Lancaster in the late 1960s. MEDLARS was based on manual indexing by subject experts using a controlled indexing language described in the MeSH (Medical Subject Heading) thesaurus. After a manual indexing operation and a manual query formulation, the retrieval operations were performed.

Table 7. Precision: Bobcat vs. Metacat

Precision	Task 1		Task 2	
	Mean	Std. Dev.	Mean	Std. Dev.
Bobcat	0.653	0.328	0.549	0.319
Metacat	0.812	0.227	0.555	0.162
Sig. Level	P = 0.083		P = 0.94	

The operational definition that we use for precision is as follows:

```
Precision = the ratio of the number of the relevant retrieved citations
            to the number of the total retrieved citations.
```

The statistics for precision are summarized in Table 7. On task 1, Metacat subjects outperformed Bobcat subjects in search precision ($p \leq 10\%$). The average precision value of the Metacat subjects performing task 1 was 81.2%, which was comparable to the precision value of 75% on STAIRS and 80% on MEDLARS. It was also 25% better than the average precision value on Bobcat (65.3%). The average precision value of the Metacat subjects performing task 2 was 55.5%, which was worse than the precision level of both STAIRS and MEDLARS. However, there was no significant difference in precision between the searches on Bobcat and the searches on Metacat for task 2.

5.7 *Summary*

We summarize the results of our system evaluation pertaining to task 2 (see Table 8) as follows (task 1 was used as a training task):

1. The Metacat users had higher satisfaction level than the Bobcat users. Most Metacat subjects were happy with the results generated from their searches and the processes involved in deriving these results. Many Metacat users were impressed by the system's capability in performing automatic thesaurus browsing and suggesting ranked relevant terms and citations.

2. There was no significant difference between the amount of time Bobcat users spent in performing searches and the amount of time Metacat users spent. While users of Bobcat spent time making their decisions during the search, our system spent time on the actual computation. The heuristic keyword searcher and the thesaurus browser, in particular, require a lot of computation. For a query that consists of several search terms (users can supply as many terms as they like), it takes the heuristic keyword searcher longer search time. Similarly, thesaurus browsing may take longer when more initial terms are present (more spreading activation is needed).

3. The average recall values of the searches on Metacat were significantly higher than the average recall values of the searches on Bobcat (78% better in recall). Our system was able to suggest more relevant citations to the users than the traditional system. Because our database is limited in its size, it was difficult for the user subjects to identify some known citations. Therefore, utilization of the known item instantiater rarely took place. Nevertheless, the heuristic keyword searcher and the thesaurus browser were very useful in generating relevant terms and citations. Because of the *search-space-expansion principle*, the heuristics keyword searcher could always identify some relevant matches for a query. This alleviated the problem of "getting into the system." The thesaurus browser, on the other

Table 8. Results pertaining to Task 2

	Bobcat	**Metacat**	Level of Sig.
Satisfaction	2.800	3.300	0.043
Search time	810 (secs)	935	0.310
Recall	0.098	0.174	0.038
Precision	0.549	0.555	0.940

hand, exploits the semantic knowledge of the thesaurus. It helped articulate the users' queries.

4. There was no significant difference in precision between the two groups. We postulate that precision is related to the users' subject area knowledge. Because of the random assignment of the subjects to each group, it was not surprising to see no difference in precision.

6. DISCUSSION AND FUTURE DIRECTION

In our research, we applied the cognitive modeling paradigm to the study of the document-based retrieval activity. Representations of the search strategies and eventually a computational model of the search strategies were developed. This computational model was incorporated into the design of an "intelligent" document-based retrieval system. An experiment was also conducted to determine the "intelligence" of the system.

We are investigating the potential of applying our results to other types of text-based information retrieval systems. These systems typically have a controlled vocabulary for classifying information, a thesaurus or table to organize the terms and the relationships between terms, and a computer system to perform the search. We believe the structure and design of our system, which contains an online thesaurus and various heuristics-based search strategies, the employment of a semantics-based query refinement module, and the use of the task-oriented blackboard architecture, can be applied to these types of systems.

Several new research projects are in progress. These projects illustrate our overall attempt to create more "intelligent" and useful information management and retrieval systems. We describe them below.

Various knowledge base components have been incorporated into the design of other knowledge-based information retrieval systems [36,37,48]. Most of these knowledge bases (including our LCSH thesaurus) were either based on existing thesauri or created via extensive knowledge acquisition (from human experts). We, on the other hand, are investigating an automated approach to the creation of a knowledge base.

Similarity coefficients can be obtained between pairs of distinct descriptors based on co-occurrences in the term assignments of the documents of the collection. This similarity coefficient indicates the "semantic distance" (or the degree of relevance) between two terms. Two algorithms, one based on normalized cosine computations (see [49]) and one based on the probability of terms co-occurrences (an algorithm developed by the authors), were adopted by us [50]. We were able to generate relevance weights (in terms of probability, between 0 and 1) between any two relevant terms in an international computing text-based database (the Arizona Analyst Information System, AAIS [51]). The database contains over 36,000 pieces of text and 35,000 unique terms. The thesaurus we created consists of more than 20,000 terms and their weighted relationships with other descriptors (on an average, each descriptor has eight "relevant" neighbors). This knowledge base is stored in INGRES, the underlying DBMS of the AAIS. For more detail, readers are referred to [50].

The AAIS thesaurus contains terms in the areas of international computing. These terms are complementary to the computing terms used in the LCSH and the ACM Computing Review Classification System (ACM CRCS)*. We have stored the LCSH computing related terms (about 5,000 terms) and the ACM CRCS (1,100 terms) in INGRES. Currently, we are exploring the approaches of merging the LCSH and ACM CRCS (existing thesauri from other sources) with our AAIS thesaurus (created automatically). This project is similar to the research of the National Library of Medicine that augmented the various existing thesauri: Current Medical Information and Terminology (CMIT), Systematized Nomenclature of medicine (SNOMED), and Medical Subject Headings (MeSH) [52]. Issues of merging man-made thesauri with the algorithm-generated thesaurus will be explored (e.g., inconsistency detection, conflict resolution, etc.). We are also in the process

*The AAIS and LCSH thesauri are network-based, while the ACM CRCS is hierarchical.

of developing a heuristic-based spreading activation algorithm for our knowledge base (for information indexing and retrieval).

Our research, which originated from the Information Science discipline and which was grounded on the Cognitive Psychology research methodology, addresses problems common to information retrieval systems. We believe our research has contributed to the understanding of the document-based information retrieval process and to the design of "intelligent" and useful document-based retrieval systems.

Acknowledgement—Many thanks for comments on earlier versions of the paper are offered to Edward Fox, Kevin Lynch, Al Croker, Margi Olson, Bill Sasso, and anonymous referees.

REFERENCES

1. Saracevic, T. Relevance: A review of and a framework for the thinking on the notion in information science. Journal of the American Society for Information Science, 26(6): 321-342; 1975.
2. Buckland, M.K. Relatedness, relevance and responsiveness in retrieval systems. Information Processing and Management, 19(3): 237-241; 1983.
3. Salton, G. Another look at automatic text-retrieval systems. Communications of the ACM, 29(7): 648-656; 1986.
4. Shoval, P. Comparison of decision support strategies in expert consultation systems. International Journal of Man-Machine Studies, 24(5): 475-487; 1986.
5. Cohen, P.R.; Kjeldsen, R. Information retrieval by constrained spreading activation in semantic networks. Information Processing and Management, 23(4): 255-268; 1987.
6. Rich, E. Users are individuals: Individualizing user models. International Journal of Man-Machine Studies, 18(3): 199-214; 1983.
7. Eastman, C.M. An approach to the evaluation of catalog selection systems. Information Processing and Management, 24(1): 23-30; 1988.
8. Newell, A.; Simon, H.A. Human Problem Solving. Englewood Cliffs, NJ: Prentice-Hall; 1972.
9. Gugerty, L.; Olson, G.M. Debugging by skilled and novice programmers. In CHI'86 Conference Proceedings, Human Factors in Computing Systems, pp. 171-175, Boston, MA, April, 1986.
10. Orlikowski, W.; Dhar, V. Imposing structure on linear programming: An empirical analysis of expert and novice models. In Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), pp. 91-95, Philadelphia, PA, August, 1986.
11. Larkin, J. Problem representation in physics in mental models. In Mental Models, Gentner, D. and Stevens, A.L., editor. Hillsdale, NJ: Lawrence Erlbaum Associates; 1983.
12. Chi, M.T.; Feltovich, P.J.; Glaser, R. Categorization and representation of physics problems by experts and novices. Cognitive Science, 5: 121-152; 1981.
13. Batra, D.; Davis, J.G. A study of conceptual data modeling in database design: Similarities and differences between expert and novice designers. In Proceedings of the 10th International Conference on Information Systems (ICIS-89), pp. 91-100, Boston, MA, December, 1989.
14. Ramaprasad, A. Cognitive process as a basis for MIS and DSS design. Management Science, 33(2): 139-148; 1987.
15. Bourne, C.P. On line system: History, technology and economics. Journal of the American Society for Information Science, 31(3): 155-160; 1980.
16. Swanson, E.B.; Culnan, M.J. Document-based systems for management planning and control: A classification, survey, and assessment. The Management Information Systems Quarterly, 2(6): 31-46; 1978.
17. Fenichel, C.H. The process of searching online bibliographic database: A review of research. Library Research, 2(2): 107-127; 1980.
18. Tagliacozzo, R.; Kochen, M. Information-seeking behavior of catalog users. Information Storage and Retrieval, 6(5): 363-381; 1970.
19. Lancaster, F.W. Information Retrieval Systems. New York: John Wiley & Sons; 1979.
20. Markey, K.; Atherton, P. Online Training and Practice Manual for ERIC Data Base Searchers. Syracuse, NY: ERIC Clearinghouse on Information Resources, ERIC: ED 160109; 1978.
21. Palmer, R.C. Search strategy development. In Library Science Text Series in Online Reference and Information Retrieval, pp. 74-94; 1987.
22. Bates, M.J. Information search tactics. Journal of the American Society for Information Science, 30(4): 205-214; 1979.
23. Smith, P.J.; Shute, D.; Galdes, D. In search of knowledge-based search tactics. In Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval, pp. 3-10, Cambridge, MA, June, 1989.
24. Ericsson, K.A.; Simon, H.A. Protocol analysis: Verbal report as data. Cambridge, MA: The MIT Press; 1984.
25. Newell, A. Reasoning, problem solving and decision processes: The problem space as a fundamental category. In Attention and Performance VIII. Nickerson, R., Editor. Hillsdale, NJ: Lawrence Erlbaum Associates; 1980.
26. Waterman, D.A.; Newell, A. Protocol analysis as a task for artificial intelligence. Artificial Intelligence, 2(3/4): 285-318; 1971.
27. Bouwman, J.M. Human diagnostic reasoning by computers: An illustration from financial analysis. Management Science, 29(6): 653-672; 1983.
28. Huguenard, B.; Prietula, M.J.; Lerch, F.J. Performance ≠ behavior: A study in the fragility of expertise. In

Proceedings of the 10th International Conference on Information Systems (ICIS-89), pp. 101-117, Boston, MA, December, 1989.
29. Chen, H.; Dhar, V. Reducing indeterminism in consultation: A cognitive model of user/librarian interaction. In Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87), pp. 285-289, Seattle, Washington, July, 1987.
30. Chen, H.; Dhar, V. User misconceptions of online information retrieval systems. International Journal of Man-Machine Studies, 32(6): 673-692; 1990.
31. Newell, A. Heuristic programming: Ill-structured problems. In Progress in Operations Research, Volume 3. Aronofsky, J., Editor. New York, NY: Wiley; 1969.
32. Winston, P.H. Artificial Intelligence, second edition. Reading, MA: Addison-Wesley Publishing Company; 1984.
33. Anderson, J.R. Cognitive Psychology and Its Implications, 2nd Ed. New York: W.H. Freeman and Company; 1985.
34. Belkin, N. J.; Oddy, R.N.; Brooks, H.M. Ask for information retrieval: Part I. background and theory. Journal of Documentation, 38(2): 61-71; 1982.
35. Erman, D.L.; Hayes-Roth, F.; Lesser, V.R.; Raj Reddy, D. The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. ACM Computing Survey, 12: 213-253; 1980.
36. Croft, W.B.; Thompson, R.H. I^3R: A new approach to the design of document retrieval systems. Journal of the American Society for Information Science, 38(6): 389-404; 1987.
37. Fox, E.A. Development of the CODER system: A testbed for artificial intelligence methods in information retrieval. Information Processing and Management, 23(4): 341-366; 1987.
38. Nii, H.P. Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. AI Magazine, 7(2): 38-53; 1986.
39. Nii, H.P. Blackboard systems: Blackboard application systems, blackboard systems from a knowledge engineering perspective. AI Magazine, 7(3): 82-106; 1986.
40. Chen, H. An Artificial Intelligence Approach to the Design of Online Information Retrieval Systems. Information Systems Department, New York University, Unpublished Ph.D. Thesis; 1989.
41. Card, S.K.; Moran, T.P.; Newell, A. The Psychology of Human Computer Interaction, Hillsdale, NJ: Lawrence Erlbaum Associates; 1983.
42. Bates, M.J. Subject access in online catalog: A design model. Journal of the American Society of Information Science, 37(6): 357-376; 1986.
43. Shoval, P. Principles, procedures and rules in an expert system for information retrieval. Information Processing and Management, 21(6): 475-487; 1985.
44. Rada, R.; Mili, H.; Bicknell, E.; Blettner, M. Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics, 19(1): 17-30; 1989.
45. Horowitz, E.; Sahni, S. Fundamentals of Data Structures. Woodland Hills, CA: Computer Science Press, Inc.; 1976.
46. Blair, D.C.; Maron, M.E. An evaluation of retrieval effectiveness for a full-text document-retrieval system. Communications of the ACM, 28(3): 289-299; 1985.
47. IBM World Trade Corp. Storage and Information Retrieval System (STAIR)—General Information Manual. IBM Company, 2nd Ed., Stuttgart; 1972.
48. Chiaramella, Y.; Defude, B. A prototype of an intelligent system for information retrieval: IOTA. Information Processing and Management, 23(4): 285-303; 1987.
49. Salton, G. Automatic Text Processing. Reading, MA: Addison-Wesley Publishing Company; 1989.
50. Lynch, K.J.; Chen, H. Acquiring knowledge from historical data. University of Arizona, Department of MIS Working Paper; 1991.
51. Lynch, K.J.; Snyder, J.M.; McHenry, W.K.; Goodman, S.E.; Hoopes, L.M. Requirements for next-generation collaborative research systems. In Proceedings of the 23rd Annual HICSS Conference, Decision Support and Knowledge Based Systems Track, pp. 72-83, Kailua-Kona, HI, January, 1990.
52. Rada, R.; Martin, B.K. Augmenting thesauri for information systems. ACM Transactions on Office Information Systems, 5(4): 378-392; 1987.