16. list. no. 27

Information Processing & Management Vol. 23, No. 5, pp. 395–409, 1987
Printed in Great Britain

0306-4573/87 $3.00 + .00

France will actively
ion researchers and
nce is sponsoring a
d Image Handling"
ns from both sides

vernments and the
t is encouraging to
l.
any country to con-
comparison and an
e old paradigms of
ly copy (as the one
out it at (201)932-

Tefko Saracevic
Editor-in-Chief

d management. Infor-
nd information science
ion policies. Informa-

DISTRIBUTED EXPERT-BASED INFORMATION SYSTEMS: AN INTERDISCIPLINARY APPROACH

Nicholas J. Belkin[1], Christine L. Borgman[2], Helen M. Brooks[3], Tom Bylander[4], W. Bruce Croft[5], Penny Daniels[6], Scott Deerwester[7], Edward A. Fox[8], Peter Ingwersen[9], Roy Rada[10], Karen Sparck Jones[11], Roger Thompson[12], Donald Walker[13]

Affiliations:

1,3) School of Communication, Information and Library Studies, Rutgers University, New Brunswick, NJ 08940

2) Graduate School of Library and Information Science, University of California at Los Angeles, Los Angeles, CA 90024

4) Computer and Information Science Department, Ohio State University, Columbus, OH 43210

5) Department Computer and Information Science, University of Massachusetts, Amherst, MA 01003

6) Admiralty Research Establishment, Ministry of Defense, AXT4, Queens Road, Teddington, Middlesex, England

7) Graduate Library School, 1100 East 59th Street, University of Chicago, Chicago, IL 60637

8) Department of Computer Science, Virginia Polytechnic Institute and State Univeristy, Blacksburg, VA 24061

9) Royal School of Librarianship, 6 Birketignet, DK 2300 Copenhagen, Denmark

10) National Library of Medicine, Bethesda, MD 20894

11) Computer Laboratory, University of Cambridge, Cambridge CB2 3QG, England

12) Hughes Aircraft, Building 618, MS M306, PO Box 3310, Fullerton, CA 92634

13) Bell Communications Research, 435 South Street, Morristown, NJ 07960

(Received 26 May 1987; accepted in final form 17 June 1987)

ABSTRACT. An international workshop on Distributed Expert-Based Information Systems (*DEBIS*) was held at Rutgers University in March 1987. The aims of the workshop were to discuss problems and issues in the design of such systems, and to develop research and implementation strategies for them. The workshop attendees discussed both models and implementations of *DEBIS*. A prototypical implementation operates on one or more workstations and connects an end-user to an information source after invoking multiple expert functions. The design of these functions depends in part on careful study of end-user and search intermediary behavior. Such studies suggest a dozen basic functions which must be incorporated in a *DEBIS*, including ones to model the user, generate search strategies, and manage the interface. The favored methods of implementation use blackboards to simplify communications between functional modules. Two large systems, I^3R and CODER, have been developed which illustrate the complexity but also the feasibility of *DEBIS*.

Keywords. Information systems, expert systems, distributed systems, user models, blackboards, workstations, search intermediaries, distributed artificial intelligence.

INFO. PROC & MAN. 1987 pp. 395-409

0002

PREFACE

Several groups around the world are working on various aspects of Distributed Expert-Based Information Systems (*DEBIS*). Realizing these activities, Rutgers University School of Communication, Information, and Library Studies along with Bell Communications Research have co-sponsored a workshop on *DEBIS*. The workshop had 13 participants and convened for 3 days and nights at Rutgers University in March 1987. The participants included members of the academic community in library, information, and computer science, as well as participants from industry and government. The goals of the workshop attendees were to extend their contribution towards the development of *DEBIS*. The participants were to identify problems and issues in the design of such systems, but more particularly to develop research and implementation strategies for them. Given that each participant had a background and direction that differs somewhat from that of other participants, the first goal of the workshop was to bring these backgrounds and directions together in order to create a broader and deeper framework within which *DEBIS* can continue to be pursued. This involved intellectually exercising the interpretations of the problem and building a framework with which all could be comfortable. From this unified perspective, participants could consider the domains in which collaboration among different groups might be profitable.

INTRODUCTION

In the publishing field there is an explosive growth of machine readable output. Simultaneously, computer hardware is becoming better able to deal with large amounts of textual and graphical material. This is due to the development of compact disks, graphic displays, powerful micro and parallel processors, and networks. Software is also improving along all the fronts that are important for improved information access. User interfaces make it easier for novices to communicate with the machine, as information retrieval systems help hone the users' request into forms that the machine understands. The end-user is now targeted by information providers. This is the result and the cause of societal changes which are stimulating the development of ever better information systems.

The idea of applying artificial intelligence's knowledge-based systems' concepts and techniques to build document retrieval system interfaces is attractive for a variety of reasons.[1] As online search systems tend to rely on specialized access mechanisms--commands, index terms, query forms--it is natural to seek effective, automatic ways of mapping the user's request onto a search query, both because assistance by human intermediaries is costly and because it would be nice to offer the end-user direct access to the search system. However, there is also the important business of establishing the user's real need, so a more significant function of an intelligent interface could be to help the user explicitly formulate a statement of his need.

Other areas of artificial intelligence, notably natural language processing, also have obvious potential for document retrieval (see, e.g.,[2]). But progress here is likely to be very slow, and applying knowledge-based or expert system techniques seems more immediately promising. The processes to which the technology would be applied are obviously important, and expert system methods have been found useful even when applied in very limited ways. They also appear to offer a good upgrade path in that if simple methods give only modest, if helpful, user support, it is possible to subsequently identify additional desirable and attainable improvements.[3]

An information system can provide access to such diverse sources of information as numeric, document, knowledge, and image data bases. For the purposes of this paper, the focus will be on document data bases. A document may be anything which basically comes as text. In other words, books, articles, memos, messages, and more fit the definition of document. The standard paradigm of an information system shows documents and queries as the input to a system which must be able to translate new documents and new queries into some canonical form in order that documents relevant to the query can be returned to the user (see Figure 1).

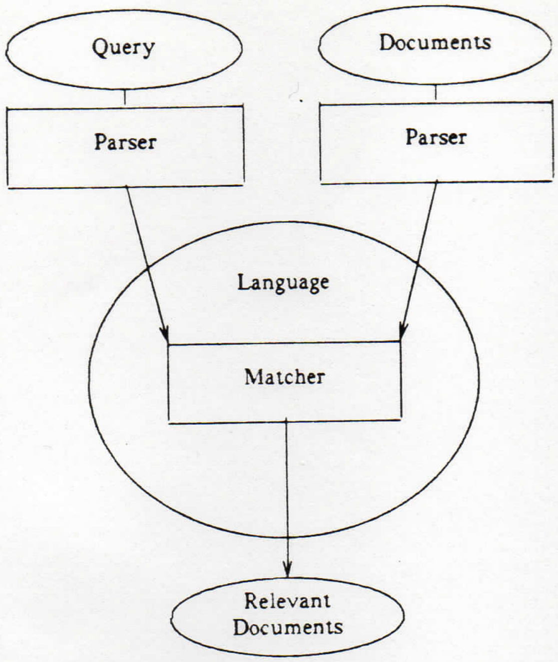

Figure 1: Flowchart of information retrieval system emphasizing the role of language for representing documents and queries.

As one imagines a distributed version of such a system, many possibilities come to mind. The documents could be distributed across different systems. There could be many searchers who come one after the other to the system. Or all the users could simultaneously be interacting through many workstations with many document bases.

The process of making these systems more intelligent or expert involves placing a variety of sources of knowledge or expertise into the system that can help the searcher gain access to information. This leads to a type of distributed system where a single workstation interacting with a single user who wants a document from a single document base has distributed expertise within itself. The distribution here is among the kinds of expertise that the information system must have in order to operate so as to best satisfy the user. In some cases the functional modules embodying that expertise can be distributed among several workstations or larger computers.

MODELS

In the traditional Information Retrieval (IR) system a searcher goes to a search intermediary and the two engage in a dialogue about the searcher's interest. The search intermediary uses his knowledge about the IR system (with its data collections) and the searcher to formulate requests directly to the IR system. The search intermediary has formulated a model of the user and taken advantage of his existing model of the IR system. To improve the direct accessibility of the IR system to the end user it would be important to incorporate in the computer both the user model and IR system model. The understanding of how these and other components of the successful intelligent IR system come together requires yet a higher-level model.

Search intermediary and user models

In one of the ongoing *DEBIS* projects in Europe an expert search intermediary is being interviewed to help determine what of his knowledge should and could be incorporated into the computer. The particular project is funded by ESPRIT, being performed in Denmark, and is called KIWI-KIRA. The KIWI-KIRA project aims to handle several attached databases of different types. KIWI-KIRA is expected to be able to successfully communicate with systems in which information is stored in relational database format or inverted file format and where the basic information may be factual, numeric, or textual. The KIWI-KIRA computer experts are obtaining knowledge from a search intermediary in several ways. First the search intermediary is asked to describe on paper the types of knowledge he uses and the functions he performs. Then the computer experts meet with the search intermediary in day-long sessions to probe the significance of what the search intermediary has written. During these sessions the KIWI-KIRA designers grasp rules for questioning users and generating search statements. The designers also discuss the validity of these rules with the expert search intermediary.

Information retrieval is a good testbed for theories of human-computer interaction because it is general--covers a diverse population, range of needs, and resources.[4] Ideally, one should be able to develop some general models of user behavior in the retrieval process. Such general models have proved elusive, however, due to the large number of variables to be managed. Fidel and Soergel[5] identified an impressive range of variables--several hundred--that had been studied to characterize information retrieval behavior, as it relates to bibliographic retrieval systems and search intermediary-users. They classified these searching variables into nine categories for further research: the end-user requesting the search, the search intermediary, the search system (features that may cause user problems), the search process (the way in which the search is performed, including errors made), search setting, search request, and database.

Any individual study has been restricted to control only a few of these variables, attributing the effects of the rest to random variance. Most user behavior studies of information retrieval systems have used expert subjects (typically search intermediaries) and experimental or commercial bibliographic retrieval systems. The body of knowledge about user retrieval behavior has been expanded through a recent series of studies on online catalogs. Online catalogs tend to be studied in libraries, with a focus on the end user (library patron) as subject. Online catalog studies have not built upon the lessons learned from studies of intermediary-searched retrieval systems, perhaps on the grounds that they involved different situations altogether.

Upon closer inspection, it appears that user behavior on bibliographic retrieval systems and online catalogs may have a lot in common, especially as the systems themselves become more similar.[6] Users of both kinds of systems have difficulty with the mechanical and conceptual aspects of searching. Search intermediaries interact with information systems far more often than do most end users, and the differences in behavior might be due to effects of training, experience, and frequency of use. Differences in performance may also be due to differences in knowledge about information structures. Librarians are trained intensively in data structures and vocabulary control; their education is focused on the mechanics of the system. Non-librarian searchers must all at once learn the nature of the database content and its organization as well as the mechanics of the system, putting them at a severe disadvantage.

Similarly the "user friendly" front ends to information systems that are assumed to simplify searching attempt to be all things to all people. They have not been designed with a specific target audience in mind. Human factors and computing research has shown the importance of iterative design with a group of subjects drawn from the intended population of users. A recent attempt to design a front-end tailored to one narrow user population (energy researchers as end-users of the Energy Database on DOE/RECON) has shown that it is possible to tailor an interface to a group of users. Given such tailoring, that user group can accomplish an assisted search with relatively little training.[7]

General system

The overall function of an information system must depend on a cooperative effort between the user and the IR mechanism--that is between the user and an intermediary mechanism (either human or machine) which in turn accesses and manipulates some knowledge resources. The functions of such an intermediary mechanism in its interaction with users have been elaborated in the MONSTRAT model.[8, 9] The MONSTRAT model is meant to apply to a variety of environments, ranging from online bibliographic retrieval systems, to referral systems, to advisory situations. Most of the work with MONSTRAT has involved investigating what a human intermediary does and extrapolating from there to what the functions of an automated intermediary mechanism should be.

MONSTRAT model. The MONSTRAT model specifies ten functions which an IR mechanism needs to perform in order to achieve its goal of helping the user with his problem.[10, 11] These functions were derived from analyses of information seeking interactions. In the general information seeking interaction, the IR system needs to have (see Table 1 for a brief listing of the ten functions and their acronyms):

* an understanding of the state of the user in the problem solving process (PS),
* an idea about what kind of response or system capability is appropriate for this user and problem (PM),
* a model of the user himself, including goals, intentions and experience (UM)
* a description of the problem the user is facing and the user's knowledge about it (PD), and
* a hypothesis about what sort of dialogue mode is appropriate for this user and problem (DM).

MONSTRAT Components	
Name of Function	Description
Problem state (PS)	Determine position of user in problem treatment process, e.g., formulating problem
Problem mode (PM)	Determine appropriate mechanism capability, e.g., reference retrieval
User model (UM)	Generate description of user type, goals, beliefs, e.g., graduate student, thesis
Problem description (PD)	Generate description of problem type, topic, structure, environment
Dialogue mode (DM)	Determine appropriate dialogue type for situation, e.g., natural language, menu
Retrieval strategy (RS)	Choose and apply appropriate retrieval strategies to knowledge resource, e.g., best match
Response generator (RG)	Determine propositional structure of response to user appropriate to situation
Input analyst (IA)	Convert input from user into structures usable by functional experts
Output generator (OG)	Convert propositional response to form appropriate to user and situation
Explanation (EX)	Describe mechanism operation, capabilities, etc., to user as appropriate

Table 1: Functions and their descriptions for the MONSTRAT model.

This information will be gained through interaction with the user, which will require analysis of the user's part of the dialogue (by the IA) so that it can be used by the other functions. The results can then be used to specify what aspects of the knowledge resource or database might be relevant to the user at this time (RS). From this potentially relevant 'world' a response particular to the specific situation can be generated (OG). Finally, it may be necessary to explain the IR system's operation and competence to the user (EX). These functions are necessary for solving subproblems of the overall IR problem. Routines solving these subproblems thus constitute the 'expert' components of a distributed expert model of an IR system.

The MONSTRAT model was intended to be a *general* model, applicable to different types of information systems. The model itself was empirically tested by means of a simulation, using humans to perform the functions.[11] The results of the simulation supported the MONSTRAT model. A group at City University of London then did a number of studies exploring the validity of the MONSTRAT model in the online search service environment. The method for doing this was through functional analysis of transcripts of human-human presearch interviews between human online search intermediaries and online service users, in an academic environment. This empirical research supported the MONSTRAT model and enabled three of the functions, UM, PD, and RS, to be specified in more detail.[12]

Further specification. The general MONSTRAT model is being elaborated. The ideal of the front-end as a distributed system of autonomous experts cooperatively achieving a global system goal was perhaps achieved because of hidden assumptions in the experiments that have been done, but for the model to be more accurate these hidden assumptions should be less hidden. One approach to further specifying a workable model is to refine its intended domain of applicability. There may be too much diversity among users, systems, and search questions for a general model to be workable. A starting point for specifying such constraints would be to divide the information retrieval world into three parts supporting different types of questions: referral (to bibliographic or other sources external to the system), source (fact-based or full-text based, where the answer, either textual or numeric, exists internal to the system), and advice-giving, where the purpose is to counsel the user based on cases in the database. A different set of functions would be required for each of these system classes. Users could also be divided into classes by degree of information retrieval knowledge, by subject area, by depth of subject knowledge, by degree of computing knowledge, or by other parameters. It may not be possible to create a model that is orthogonal to the classification on types of questions. Rather, a tailored classification of users may be a subset of each class of question.

In addition to refining the general *DEBIS* model by applying it to certain information sources and user classes there are ways to further specify the MONSTRAT model by indicating more about the control and blackboard properties of the model. It is hard to quarrel with the idea of functional expertise as the driving force of the front end. The specific idea of a distributed expert system with a set of individual experts each specialized to seek some particular goal but collectively, in achieving these, contributing to the system's overall goal, is a plausible one. But there are problems about these ideas. Distributed expert systems are thoroughly complicated.[3] Thus it is necessary to be quite clear about what having such a cooperative system implies.

The first and major problem with distributed systems is control: what is the mechanism for determining the flow of control? The assumption in the document retrieval case is that the system is not wholly data driven. Rather it is a mixed-initiative system. Imitating the intermediary, the system itself frequently will have the dominant role, working towards the overall goal of satisfying the user. An evaluation methodology needs to be developed, possibly by taking the Problem Description function as the key function and arranging that if the Problem Description function is satisfied, by whatever means it has of determining its own goal satisfaction, and if the Retrieval Strategy function has consumed all of the germane information output by the Problem Description, this is sufficient. But it is not easy to interpret this more concrete-looking suggestion precisely, because a strategy of simply asking the user whether he has said his all is clearly inadequate: the problem is really whether the Problem Description function is capable of establishing, given its putatively sophisticated model-building capacities, that it has taken its model of the user's problem as far as it can. The control difficulty comes from the fact that the system is not modeling an external phenomenon with a well-defined, or definable, structure. The nature of the interaction between user and intermediary can only be characterized in a schematic fashion covering very many, quite different, lower-level possibilities. Imposing some specific organization on processing is liable to lead to interaction which is too arbitrary for the user. These control problems have no general solution: the solution has to depend on the particular design of the system.[3]

IMPLEMENTATIONS

While substantial work by psychologists, librarians, and information scientists remains to be done in the study of what humans do when they somehow obtain information, some computer scientists are building prototypes that test various assumptions about what would constitute an adequate *DEBIS*. In this Implementations Section, one generic module which has been implemented in a variety of expert systems and that depends on hierarchic classification is described. Then a simple but practical system alled CANSEARCH, that serves as an interface between health practitioners and cancer literature, is described. Two large systems, called I^{3}R and CODER, contain multiple expert modules, communicate through a blackboard, and access document databases. I^{3}R and CODER are used in this paper to illustrate how the implementations of complex schemes can produce working systems and allow insights into principles of operation of human-machine systems.

Generic task modules

Many problems are characterized by a search for the category or categories within a predefined classification hierarchy that apply to a given situation. Often these tasks can be solved by using the generic method of hierarchical classification.[13] The mechanism of hierarchical classification is establish-refine. Simply put, the establish-refine strategy is: if a category is confirmed or determined to be highly relevant (the establish part), then consider its subcategories in the classification hierarchy (the refine part). If a category is rejected, then its subcategories are also rejected. For systematic search, establish-refine starts at the root category of the hierarchy.

Classification hierarchies cannot simply be copied from textbooks. Even if a textbook supplies a hierarchy, it is dangerous to take it literally.[14] For example, a classification of the causes of jaundice might be given, which would include hepatitis, suggesting that hepatitis should be a subcategory under jaundice. However, ruling out jaundice does not rule out hepatitis, i.e., hepatitis does not necessarily cause jaundice.

The categories in the classification hierarchy become hypotheses for an abduction problem. An abduction is a non-deductive inference that follows a pattern something like this:[15] If D is a collection of data or observations and H is a likely hypothesis that accounts for D and no other hypothesis accounts for D as well as H does, then conclude that H is the best hypothesis. The abduction problem is solved through an intelligent search of the classification hierarchy and repeated testing of the extent to which the hypotheses match the data.

The theory of generic tasks[16] proposes a number of simple information-processing architectures, in which each generic task provides a mechanism for performing a restricted type of function and a representation for decomposing the problem into conceptual parts. Given a collection of simple information-processing architectures (such as generic tasks) and a function to perform, there are two issues that need to be resolved. First, what generic task best matches the desired function, and how can the subfunctions of the generic task be specified. The second issue is whether the knowledge is available in the right form for the generic task. For the hierarchical classification example, a category's plausibility needs to be reliably estimated without considering the plausibility of many other categories. If the interactions are too complex, then a deeper model may be needed. It is hoped that generic tasks can be used in future construction of DEBIS.

Elementary system

CANSEARCH takes advantage of an existing hierarchy, called MeSH, for classification of information in its course of leading users to the proper formulation of their query. CANSEARCH[17, 18] was designed to provide access, through Cancerline, to cancer therapy literature indexed using MeSH. CANSEARCH implements some of the principles of the hierarchic-classification generic task[16] but also addresses several of the problems unique to information retrieval systems.

MeSH is an artificial, hierarchic classification language, and one cannot expect the end user to know how to apply it. The object of CANSEARCH is therefore to help construct an appropriate MeSH search query, with proper terms and structure. The underlying system structure is a rule base designed to lead the user, via touch screen menus, through a hierarchy of frames covering the aspects of a therapy topic specification. Thus the main areas of the hierarchy deal respectively with cancer itself and its sites, types and therapies. The rules are grouped into types of context, for selecting frames, for checking and for processing term selections, and for constructing the actual search query. In operation those rules for selecting frames and processing term choices are associated with specific frames, so a frame can be thought of as having its own rule packet. The rules communicate via blackboards, one for each aspect of a query, in a simple and straightforward way passing messages or finally actual search terms for eventual assembly into the search query.

CANSEARCH is essentially implementing only one, or at most two, of the intermediary's functions, and in a very limited style: there is, for example, no clarifying dialogue except in the particular form of iteration, if the user has made inconsistent menu selections. The main source of knowledge for the system is the indexing language MeSH which gives the set of available concept labels and their hierarchical structure. There is a closely related source, which specifies the forms index descriptions, and more specifically search queries, can take with respect to the links between and roles of terms, i.e., Boolean operators. There is also expressed in MeSH some knowledge of what cancer is, and further, in the rule set, knowledge, again mediated by MeSH, about what the constituent notions of cancer therapy requests are.

MeSH also embodies some knowledge of the subject literature as much as of the subject. The fact that MeSH encapsulates a range of relevant knowledge, or facilitates its incorporation in the system, provided the developer of CANSEARCH with a good deal of leverage. MeSH was a means as well as an end in the construction of the rules and frames. The use of MeSH is, of course, partly justified as the channel of communication with the information source.

There is no doubt that CANSEARCH could be substantially improved, for example in its coverage of request types and MeSH, and doing this might well provide systems that were practically very useful. But it could be that in the long run gains could only be made by starting from a more radical architecture. CANSEARCH is both limited in its scope and constrained in its specific task: it is essentially concerned with translating a certain sort of input into a certain sort of output. It thus has no real control problem: there is a natural path, globally through the rule sets and locally through the frame hierarchy, and there are essentially structured boards with simple messages because the messages are targeted and have a context for their interpretation. It is also the case that at the lower level of the individual rules, the system has an appropriate granularity. But it might not be easy to extend the system's scope, for example to allow natural language rather than MeSH queries.

Intelligent interface

I^3R (Intelligent Interface for Information Retrieval) is a system designed to help overcome the difficulties of using text retrieval systems. As an interface system, it is responsive to a wide variety of users, who have varying levels of ability in computer use and varying levels of knowledge about the topic being investigated.[19, 20] The overall structure of the system is based on a blackboard architecture, a collection of independent cooperating experts which communicate indirectly using a shared global data structure.

The I^3R system can be compared to the Hearsay-II system. Hearsay-II is a speech understanding system that synthesizes the partial interpretations of several diverse knowledge sources into a coherent understanding of a spoken sentence.[21] Knowledge sources communicate by reading and writing on a blackboard. The blackboard has several distinct levels which hold different representations of the problem space. Typical blackboard levels for speech understanding are sound segments, syllables, words, and phrases. The knowledge sources are pattern-action productions: if the information on the blackboard matches the pattern of a knowledge source then its action can be executed. At any time, many knowledge sources are likely to have patterns that match the contents of the blackboard. The scheduler decides which knowledge source is to be executed next, choosing the knowledge source which has the highest priority.

Architecture. The I^{3}R system differs substantially from the kind of blackboard system exemplified by Hearsay-II.[21] Hearsay-II solves a problem that is easily cast into a hierarchy, so the blackboard could be structured to match. Also, since Hearsay-II solves a bounded recognition problem, the concept of a "hypothesis", a possible interpretation of data, makes sense. However, the problem with which I^{3}R deals has no overall hierarchy. Instead, it has a number of different aggregations of information that are pertinent to different aspects of the problem. The structure of the I^{3}R blackboard reflects this; it is composed of a number of different models and communication areas.

For I^{3}R to be adaptable, it must be able to assess the user's abilities so it can adjust the interface to match them.[22] This requires a user model builder. As each user may have his own view of the subject area being searched, it would be valuable to capture this information and remember it from session to session in a domain knowledge expert. The system also contains modules for controlling search, browsing, and managing the interface.

As with most blackboard systems, the activity of the I^{3}R experts is controlled by an agenda mechanism. Typical agenda management involves ordering posted actions and then selecting the one on the top. In I^{3}R the scheduler controls the agenda by determining what experts can post actions and in what order those actions are executed. The basis for the scheduler's decisions is derived from the analysis of user/intermediary dialogues performed by Belkin and others.[23] The dialogue structure is reflected directly in the structure of the scheduler. This structure is essentially a goal tree with extra transitions (see Figure 2).

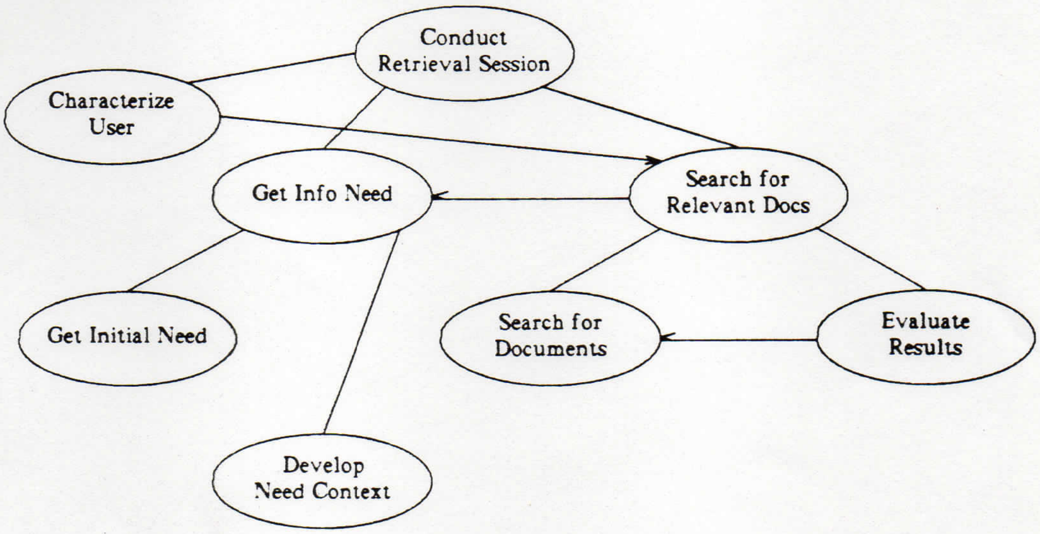

Figure 2: I^{3}R Scheduler Structure.

A basic session would involve passing through the goals as a preorder traversal of the tree. The horizontal transitions are taken when the scheduler decides that the session is not proceeding normally. What transition to take is determined by the rules associated with each goal. Expectations about the number of relevant documents desired and the number of searches to use are derived from the User Model Builder applied to the user. For example, say the user was familiar with the system, was an expert in the subject area of the search topic, and wanted a precision-oriented search, then the User Model Builder would set the expectations of the system to be: five documents required and two searches allowed. If, after two searches, five documents were not found, the scheduler would take the transition from "Search for Relevant Documents" to "Get Information

Need" and initiate further dialogue with the user.

Implementation of the experts. In typical blackboard systems the experts can be implemented in any manner as long as they conform to the interface defined by the blackboard structure. In I^3R the expert control knowledge is represented as rules. The rules provide a uniform control knowledge representation but not a uniform data representation. In I^3R the data is structured in ways most suitable for each expert's function (see Table 2).

I^3R Data Structures		
Model	Conceptual Structure	Implementation
Request Model	Probabilistic Request	Hash Table of Structures
Domain Knowledge Model	Semantic Net	Hash Table of Structures
User Model	Stereotypes and Expectations	Association List
Document Representation	Network of Documents and Terms	Relational Database
Browsing Model	Semantic Net Fused with Network of Docs & Terms	Hash Table of Structures

Table 2: Table of data structures accessed by expert rules in I^3R.

For instance, the Domain Knowledge Model is conceptually viewed as a semantic net and is implemented as a hash table of structures. The Document Representation is modeled as a network of documents and terms but implemented in a relational database. Given the hardware limitations of the current I^3R, the relational database is housed on a second machine. The communication to this second machine is the major bottleneck in the timely execution of I^3R. The interface manager, while not the subject of research in I^3R, has nevertheless required as much computer code as the entire rest of the system. Most of this code is devoted to handling windows, menus, and graphics. This experience of substantial effort required for the user interface has been shared by other research groups.[24]

The control rules are 4-tuples consisting of:

<*expert name*> <*rule #*> <*conditions*> <*actions*>.

The <*expert name*> and <*rule #*> simply indicate where the rule belongs. The conditions and actions are the meat of the rule. Each condition is a 4-tuple itself consisting of

<*blackboard place*> <*action name*> <*predicate name*> <*arguments*>.

The interpretation of <*conditions*> is that if an <*action name*> is at a <*blackboard place*>, then check the place with the given predicate and arguments. Each <*actions*> from a control rule is a 3-tuple of <*blackboard place*> <*action name*> <*arguments*>. This 3-tuple means perform the specified action on the given place using the arguments. The <*blackboard place*> and <*action name*> provide the mechanism for notifying experts when something has happened that is of interest.

A blackboard architecture suits I^3R because:

1) multiple search strategies are required,
2) there is no analytic way of determining which choice of strategies is best, and
3) a complex interaction is required for query formulation and evaluation.

Although the architecture of I^3R resembles that recommended by MONSTRAT in many ways, it should be remembered that it is based on an entirely different model of information retrieval. The basis of this model is that retrieval is a process of plausible inference, where information about relevance is gathered from a variety of sources. The emphasis on the formulation of a detailed request model is a natural consequence of this approach.[25]

General testbed approach

The Composite Document Expert Retrieval System (CODER) has been designed to serve as a testbed for determining how useful various artificial intelligence techniques are in improving the effectiveness, efficiency, and usability of information storage and retrieval systems.[26, 27] It is being developed in a modular fashion so that new approaches can be easily compared with more traditional methods. While a skeletal version of CODER is now operational, the emphasis is on constructing a rich environment for experimentation rather than demonstrating functionality of a limited prototype. A comprehensive lexicon, a powerful knowledge representation language, a set of communication predicates, and a general-purpose blackboard are all parts of the CODER support environment. CODER is able to both automatically parse documents into the canonical form of the information retrieval system and to handle queries from users (see Figure 3); in this way, CODER is more robust than I^3R which is designed exclusively to interact with people who are requesting information.

CODER was initiated in 1984 with the idea of using logic programming for analyzing and retrieving messages from a collection of AIList Digests. The AIList Digest is an electronic newsletter that contains a wide variety of announcements and short papers about artificial intelligence. Early in the CODER effort key online reference works were obtained. The *Handbook of Artificial Intelligence* was analyzed to obtain people's names, key words and phrases, and a subject hierarchy. The Collin's dictionary was transformed into a Prolog fact base. This was usable directly since a Prolog system, called MU-Prolog, with the ability to handle very large databases on secondary storage was selected. Knowledge is crucial to the task at hand,[28] and more attention should be given to the automatic preparation of thesauri and other aids from existing machine readable works.

CODER is made of external knowledge bases, managers, experts, and blackboard/strategist complexes. Each such unit operates as a process on some computer, and may serve one or more users. There may be several copies of a unit operating on the same computer or on several computers. Typically, a new user interface manager and a blackboard/strategist complex are assigned to each system user at logon time, but the lexicon manager only operates on the computer that has the very large database stored online. Thus if there are many computers, CODER can operate in such a way that each expert is always trying to process any data on the blackboard that is of a type it can handle. Rules in the domain task scheduler portion of the blackboard/strategist are responsible for much of this control. CODER supports both centralized and decentralized modes of control, or any combination of the two. By isolating control issues into one module, CODER facilitates experimentation with control variations.

It is still unknown how best to manage a retrieval session, when many different experts each suggest different actions. Yet, a hierarchy of plans, goals, and rules that force discourse to reflect the current consensus can help smooth the human-computer interaction. Goals should include:

* keep the user engaged without having to wait long periods of time and
* search incrementally rather than in batches unless the user likes to scan.

CODER has regular blackboard areas, a pending hypothesis area, and a question/answer area. The latter is only for questions that can be answered without much work. The former is for normal blackboard communication. However, since each posting to the blackboard has an associated confidence value assigned by whatever agency made that posting, it can be left to the blackboard manager to fill in entries in the pending hypothesis area. Specifically, whenever there is a consistent set of hypotheses with sufficiently high confidence, that set of hypotheses is moved to the pending hypothesis area. Thus, new, confident postings, or old postings whose confidence values have increased as a result of new confirming information, both end in the pending hypothesis area. This is important since all experts can see that area, while there is a predefined mapping of experts and blackboards for the normal areas.

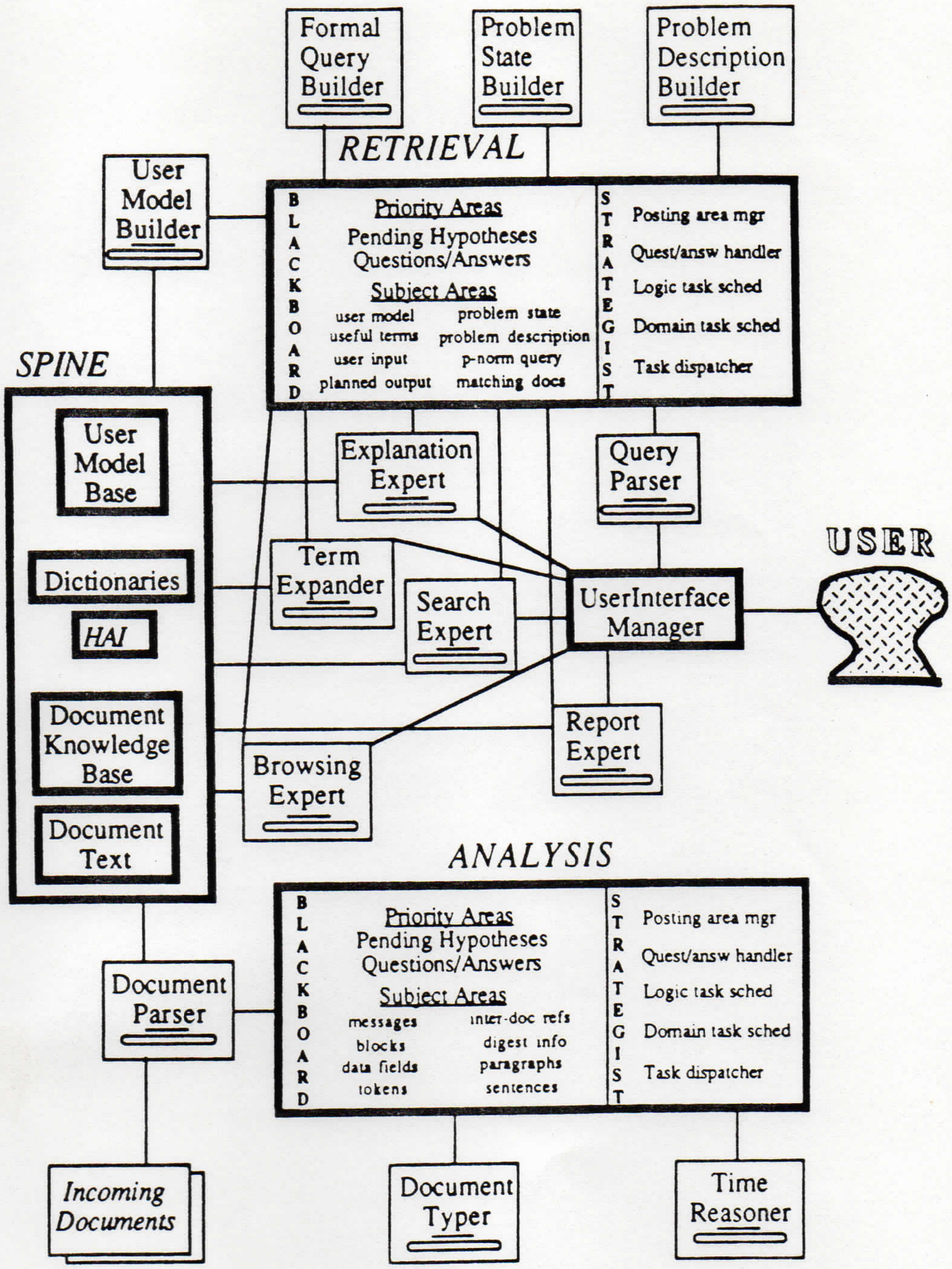

Figure 3: Overview of CODER System.

0013

CODER is being extended in several directions, and there are many unresolved issues. How can the system know when there are gaps in the collection or when the user query has been incorrectly translated? Ideas on evaluation are needed, and ways to measure system efficiency are desired. Another unresolved issue relates to just how much effect one gets from having a knowledge representation scheme for documents that is more comprehensive than vectors. Similarly, one needs to know if the behavior of a human intermediary is the appropriate activity to study regarding building intelligent computer intermediaries; that is, can the human-human-computer interaction be usefully mapped to the human-computer interaction. The design of CODER is intended to facilitate the kinds of comparative studies that will help answer these questions.

CONCLUSION

The role of intelligent interfaces to information systems is growing, and multiple disciplines are contributing to the development of distributed expert-based information systems. One of the emphases of this paper is on architectures for expert information retrieval (IR) systems. Given the importance of IR applications, the functionality to be implemented is crucial. It must be well-specified and motivated. This motivation can come from cognitive research or research on theoretical IR models. Blackboard architectures appear to be an appropriate engineering solution to the implementation of the desired functionality. The architecture is, however, a secondary issue with regard to IR. Future works must concentrate on development of functionality and IR models. In terms of architectures, the evaluation is partly in terms of successful implementations and their relative capabilities.

Some of the areas particularly ripe for future work include user models, multi-agent planning, the analysis of text structure and content, and the use of dictionaries and other resources as search aids. The new implementations of user models do not take as much advantage as they could of the knowledge gained from studies already made of how people interact with information systems.[29] As information systems become more completely integrated into the office place, the need for multi-agent planning becomes paramount. In the office place, several agents may have several activities to perform on several objects simultaneously.[30] One of the backbones of the expert-based approach to information interfaces is, of course, knowledge about the content of the documents being retrieved.[31] Although natural language processing techniques are not likely to provide the basis for a complete understanding of texts for some time, it is possible to exploit resources at a lexical level that identify the information they contain more effectively. In particular, the use of machine-readable dictionaries, almanacs, and encyclopedias, coupled with an analysis of the patterns of occurrence of words and phrases in documents, looks particularly promising.[32]

The *DEBIS* Workshop summarized in this paper focused on developing an integrated approach to serving the information needs of end users. Informed by the behavior of search intermediaries and experimental studies of users seeking information, a functional approach to information access was elaborated. Architectural concerns of system control and representation methods for communication were explored. Design issues of problem decomposition were considered. Approaches to building expert systems using commercial tools were contrasted with Prolog and Lisp approaches with internal or external support for databases. The prototype systems demonstrated the feasibility of the distributed expert approach. Given the increasing availability of machine readable information, the hardware and software tools which support access to information, the models proposed for user-adaptable retrieval, and the existence of systems implementing some of the desired functionality, the need for continued cooperative efforts was stressed.

References

1. Karen Sparck Jones, "Intelligent Retrieval," *Proceedings Intelligent Information Retrieval*, pp. 136-142, Aslib, London, March 1983.

2. Karen Sparck Jones and J I Tait, "Automatic Search Term Variant Generation," *Journal of Documentation*, *40*, pp. 50-66, 1984.

3. Karen Sparck Jones, "Architecture Problems in the Construction of Expert Systems for Document Retrieval," *AI-IR Seminar*, Copehagen, 1987. in press (also presented at Workshop on Distributed Expert-Based Information Systems, Rutgers University, New Jersey, March 1987).

4. Christine L Borgmann, *Information System's Functionality: A User-Driven Perspective*, Rutgers University, New Jersey, March 1987. Presented at Workshop on Distributed Expert-Based Information Systems

5. Raya Fidel and Dagobert Soergel, "Factors Affecting Online Bibliographic Retrieval—A Conceptual Framework for Research," *Journal of American Society of Information Science*, *34*, *3*, pp. 163-180, 1983.

6. Christine L Borgman, "Why Are Online Catalogs Hard to Use: Lessons Learned from Information Retrieval Studies," *Journal of American Society of Information Science*, *37*, *6*, 1986.

7. Donald Case, Christine L Borgman, and Charles T Meadow, "End-User Information-Seeking in the Energy Field: Implications for End-User Access to DOE RECON Databases," *Information Processing and Management*, *22*, *4*, pp. 299-308, 1986.

8. Helen M Brooks, *The Functions of an Information System: The MONSTRAT Model*, Rutgers University, New Jersey, March 1987. Presented at Workshop on Distributed Expert-Based Information Systems

9. G Wersig and R D Hennings, "The Intellectual Architecture of Information Systems: A Broad Range Research Agenda," in *Representation and Exchange of Knowledge as a Basis of Information Processes*, ed. H J Dietschmann, pp. 7-30, Elsevier Science Publishers, Amsterdam, Netherlands, 1984.

10. N J Belkin, T Seeger, and G Wersig, "Distributed Expert Problem Solving as a Model for Information System Analysis and Design," *Journal of Information Science*, *5*, pp. 153-167, 1983.

11. N J Belkin, R D Hennings, and T Seeger, "Simulation of a Distributed Expert-Based Information Provision Mechanism," *Information Technology*, *3*, *3*, pp. 122-141, 1984.

12. H M Brooks, P J Daniels, and N J Belkin, "Problem Descriptions and User Models: Developing an Intelligent Interface for Document Retrieval Systems," *Informatics 8: Advances in Intelligent Retrieval*, pp. 191-214, Aslib, London, 1985.

13. T Bylander and S Mittal, "CSRL: A Language for Classificatory Problem Solving and Uncertainty Handling," *AI Magazine*, *7*, *2*, pp. 66-77, 1986.

14. Tom Bylander, *Architectures for Distribution of Knowledge Based on Functional and Conceptual Decomposition*, Rutgers University, New Jersey, March 1987. Presented at Workshop on Distributed Expert-Based Information Systems

15. Fernando Gomez and B Chandrasekaran, "Knowledge Organization and Distribution for Medical Diagnosis," in *Readings in Medical Artificial Intelligence*, ed. Edward Shortliffe, pp. 320-338, Addison-Wesley, Reading, Massachusetts, 1984.

16. B Chandrasekaran, "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert System Design," *IEEE Expert 1*, *3*, pp. 23-30, 1986.

17. Steven Pollitt, "CANSEARCH: an Expert Systems Approach to Document Retrieval," *Information Processing & Management*, *23*, *2*, pp. 119-138, 1987.

18. A S Pollitt, "A Rule-Based System as an Intermediary for Searching Cancer Therapy Literature on MEDLINE," in *Intelligent Information Systems: Progress and Prospects*, ed. Roy Davis, pp. 82-126, 1986.

19. Roger H Thompson and W Bruce Croft, "An Expert System for Document Retreival," *Proceedings of the Expert Systems in Government Symposium*, pp. 448-456, IEEE Computer Society Press, Washington, D.C., 1985.

20. Roger H Thompson, *An Implementation Overview of I3R*, Rutgers University, New Jersey, March 1987. Presented at Workshop on Distributed Expert-Based Information Systems

21. L D Erman, "The Hearsay-II Speech Understanding System: Integrating Knowledge to Resolve Uncertainty," *Association Computing Machinery Computing Surveys, 12*, pp. 213-253, 1980.

22. W Bruce Croft, "The Role of Context and Adaptation in User Interfaces," *International Journal of Man-Machine Studies, 24*, pp. 283-292, 1986.

23. P J Daniels, H M Brooks, and N J Belkin, "Using Problem Structures for Driving Human-Computer Dialogues," *Actes of the Conference: Recherche d'Informations Assistee par Ordinateur*, IMAG, Grenoble, France, 1985.

24. Reid Smith and James Baker, "The Dipmeter Advisor System: A Case Study in Commercial Expert System Development," *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pp. 122-129, August 1983.

25. W Bruce Croft and R T Thompson, "I3R: A New Approach to the Design of Document Retrieval Systems," *Journal of American Society of Information Science*, 1987. to appear

26. Edward A Fox, "Development of the CODER System: A Testbed for Artificial Intelligence Methods in Information Retrieval," *Information Processing and Management, 22, 4*, 1987. to appear

27. Edward A Fox and Robert K France, "Architecture of an Expert System for Composite Document Analysis, Representation, and Retrieval," *International Journal of Approximate Reasoning, 1, 2*, April 1987.

28. Robert K France and Edward A Fox, "Knowledge Structures for Information Retrieval: Representation in the CODER Project," *Proceedings IEEE Expert Systems in Government Conference*, pp. 135-141, McLean, Virgina, 1986.

29. Christine L Borgman, "The User's Mental Model of an Information Retrieval System: an Experiment on a Prototype Online Catalog," *International Journal of Man-Machine Studies, 24*, pp. 47-64, 1986.

30. Sidney Harris and Harvey Brightman, "Design Implications of a Task-Driven Approach to Unstructured Cognitive Tasks in Office Work," *ACM Transactions on Office Information Systems, 3, 3*, pp. 292-306, 1985.

31. Roy Rada, "Knowledge-Sparse and Knowledge-Rich Learning in Information Retrieval," *Information Processing and Management*, pp. 195-210, 1987.

32. Donald E Walker, "Knowledge Resource Tools for Analyzing Large Text Files," in *Machine Translation: Theoretical and Methodological Issues*, ed. Sergei Nirenburg, pp. 247-261, Cambridge University Press, 1987. also published as "Knowledge Resource Tools for Information Access," in FGS: Future Generations Computer Systems, 2:3, pp. 161-171, 1986.