Technical Report

UCAM-CL-TR-131
ISSN 1476-2986

Number 131

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Architecture problems in the construction of expert systems for document retrieval

Karen Spärck Jones

December 1986

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

https://www.cl.cam.ac.uk/

ISSN 1476-2986

ARCHITECTURE PROBLEMS

IN THE

CONSTRUCTION OF EXPERT SYSTEMS

FOR

DOCUMENT RETRIEVAL

Karen Sparck Jones

Computer Laboratory, University of Cambridge
New Museums Site, Pembroke Street, Cambridge CB2 3QG, England

December 1986

Abstract

The idea of an expert system front end offering the user effective direct access to a document retrieval system is an attractive one. The paper considers two specific approaches to the construction of such an expert interface, Belkin and Brooks and their colleagues' treatment of the functions of such a front end based on the analysis of human intermediaries, and Pollitt's experimental implementation of a query formulator for searching Cancerline. The distributed expert system model proposed by Belkin and Brooks is a plausible one, and Pollitt's system can be regarded as a first step towards it. But there are major problems about this type of architecture, and the paper argues in particular that in seeking to develop more powerful front ends of the kind envisaged there is one important issue, the nature of the language used for communication between the contributing experts, that requires more attention than it has hitherto received.

This paper was prepared for a Seminar on Artificial Intelligence and Information Retrieval, organised by the Royal School of Librarianship in Copenhagen in December 1986, and will appear in a volume based on the Seminar.

Architecture problems in ¡he construction of expert systems for document retrieval

Karen Sparck Jones

Introduction

The idea of applying AI's expert system concepts and techniques to build document retrieval system interfaces has been canvassed for some time, and is attractive for a variety of reasons. As online search systems tend to rely on specialised 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 human intermediaries are 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 formulate his need and express it as a request input to the technical mapping process. (Of course these two processes should not actually be treated as separate, since the mapping possibilities available may constrain the formulation; but both formulation and translation are involved in the approach to the search system.)

Other areas of AI, notably natural language processing, also have obvious potential for document retrieval. But progress here is likely to be very slow, and applying 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 appled 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 identify desirable and attainable improvements.

The purpose of this paper is to explore what is involved in having an expert system as an intelligent interface to a document retrieval system, using two recent investigations as a focus. This analysis suggests that there is an important problem to be addressed in building intelligent interfaces of the kind exemplified to which not enough attention has been paid. It is by no means the only significant issue arising in this context, but it is both important in itself and illustrative of the difficulties to be overcome before really serious intelligent interfaces for document retrieval systems can be built.

For the purposes of the discussion the assumption is that we are concerned with access to document retrieval systems of the kind hosted by Dialog, where we have very large collections, summary document descriptions, and small relevance sets, with all the fundamental challenge this implies for indexing documents and requests to achieve both accuracy and discrimination in description. I shall also assume for the moment that we have controlled indexing languages directly intended to overcome the variability of natural language as an indication of document content and indirectly, through this, to promote proper index description.

It is also necessary to bear in mind, in considering intelligent interfaces to document retrieval systems, two important points. One is that we are concerned with access and, more materially, indirect access to the information the user wants: he wants the information in the documents, but the system only gives him the documents. The other is that the documents are linguistic objects and that the way the information they contain is linguistically expressed can itself be as important as the information, i.e. the two may be indivisible. The aim of a document retrieval system is to give the user the texts where he can read about X. The fact that we have an indirect interface may affect its performance as an expert system, and the fact that we are dealing with natural language objects may constrain the interface's internal operations (see Sparck Jones 1983).

Illustrative approaches

The two projects I shall analyse in more detail are both directed towards automating the operations of the trained intermediary in extracting the user's need, expressing it in the indexing language, and forming a query geared to the collection file being searched.

Brooks

Brooks' work (Brooks 1986) has been carried out within the broader framework of the analysis of information systems as wholes proposed by Belkin and his colleagues (Belkin et al 1983, 1984, 1985; Brooks et al 1985; Daniels et al 1985). Belkin and his colleagues analyse information systems, in the very general sense of information provision mechanisms, in terms of a set of functional components designed to treat the user's problem. More specifically they propose a distributed problem solving model with individual experts for particular functions. These work independently to achieve their own goals, but the system is cooperative in the sense that the goals achieved by the functional experts all contribute to the overall system's higher-level goal of providing the user with an appropriate response via a suitable search formulation. The functional experts also cooperate in a more specific way by communicating information (or hypotheses) via a common message space, or blackboard. The ten system functions Belkin and his colleagues have identified are shown in Figure 1. The precise details of these functional specifications, and also the merit of the set as a whole, are not in question here. Though it is possible to argue about the proposals it is clear that something like these functions have to be carried out by information provision mechanisms. The set of functions also reflects an approach to information systems from the perspective of the user: a broader view could suggest other functions, for example a matching function and, in the most general case, a document indexing function, for example. However there is no doubt that the functions listed are material ones, and that the Problem Description function in particular, is central to the whole concept of an information retrieval system.

But the important point here is that the functions of Figure 1 are all functions served by a human intermediary when viewed as part of an information system, so we would want to provide for these functions in seeking to automate the intermediary. Of course in a specific context some functions may be heavily constrained, for example the type of retrieval strategy allowed, or range of response possibilities offered.

Belkin, Brooks and Daniels' (Brooks et al 1985; Daniels et al 1985; Brooks 1986) detailed analysis of the interaction between user and intermediary in interviews recorded at the University of London further identifies subfunctions of the higher-level functions listed in Figure 1, aimed at achieving subgoals serving those of the higher-level functions. The subfunctions identified for the User Model, Problem Description and Retrieval Strategy functions are shown in Figure 2. Brooks' investigation of the interviews, concentrating on the Problem Description and Retrieval Strategy functions, shows how the interaction between the user and the intermediary addresses these functional goals. Individual utterances, i.e. utterances by one participant, on a single topic, are directed towards some functional or subfunctional goal, but information may be supplied which is relevant to other goals, as shown in Figures 3 and 4. Sequences of utterances directed to a common goal form a natural group, or dialogue focus, which may also be bounded by dialogue framing expressions. As goals may be addressed or re-addressed in very variable orders, the structure of an interview can be very complex, as appears in the kind of focus transition map that can be constructed for it, illustrated in Figure 5. (These figures do not illustrate the same interview.)

Brooks' investigation makes it clear that the activity of the (intermediary's notional) functional experts, both individually and collectively, is very complicated: interpreting the user's need so as to get the best system response is very demanding. Her analysis also shows the variety and range of knowledge required to carry out the functional tasks effectively. The Problem Description function, for example, requires, or at least can benefit from, all the knowledge sources listed in Figure 6. Further, though some sources may be relevant to more than one individual expert, others may be specific to one expert.

The distributed expert system model proposed by Belkin and his colleagues is a very general one. The claim is that the functional model is supported by an examination of various forms of information provision mechanism, but both more data and more validation, through an autonomous functional analysis of the task, of the task characterisations suggested by the transcripts, are required. Investigations like Brooks' are very much a preliminary clearing of the ground for the design of a comprehensive intelligent interface, with no detailed specification of the mechanisms involved. Brooks' analysis, though very careful, is a human analysis of a sophisticated human activity, and, quite apart from the challenges involved in automating the intermediary's primary activities, like determining the problem description, there are also those of, for example, automating the interpretation of input utterances to determine what their topics are to consider.

I shall return to the implications of Brooks' work for any attempt to build a computational interface after considering an actual implementation of an expert front end, Pollitt's recent experimental system (Pollitt 1986b). This is, not surprisingly, much more limited in its capacities than even a fairly uncommitted human intermediary, but it is useful as an illustration of what can be done now.

Pollitt

Pollitt focuses on the specific task of helping the end user achieve the artificial index language expression of his need. The aim is to allow the user to approach the search system directly, without having to know the technical details of the index language. This has always seemed an attractive idea, particularly if it can be supported by real-time feedback and iteration. The same motivation underlies various forms of front end proposed for non-bibliographic databases.

Pollit's system, CANSEARCH, is very limited in its scope. It was designed to provide access, through Cancerline, to cancer therapy literature indexed using MeSH. MeSH is a highly controlled, i.e. very artificial, indexing language, so one cannot expect the end user to know how to apply it. The object of the system is therefore to derive a valid MeSH search query, in terms and structure (e.g. with respect to modifying terms), for the user's concepts. 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, as shown in Figure 7. 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 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 (or subaspect) of a query, in a simple and straightforward way passing messages or finally actual search terms for eventual assembly into the search query. The system's architecture and blackboard set are shown in Figure 8, and simplified, illustrative rules of the various context types in Figure 9.

Pollitt's system was tested, both from the informal practical point of view as acceptable to users and in a controlled formal evaluation designed to compare search queries generated via CANSEARCH with those prepared by human intermediaries. The testing constraints meant that simple natural language need statements were taken as starting points for query preparation, either by a user operating CANSEARCH or by an intermediary applying his or her professional skills. Each worked alone, without interacting with the individual originator of the need statement, so the experimental CANSEARCH user is properly described as a pseudo user. It is important to emphasise that CANSEARCH does not actually interpret natural language input text, as some non-bibliographic database query systems do: the starting natural language text was simply used, for the purposes of controlled experimental comparison, as a common starting point by the CANSEARCH operator and intemediary alike. What the experiment was testing, therefore, was

CANSEARCH's effectiveness in leading the (pseudo) user from a specified need to a search query, i.e. in getting him to the MeSH query of Figure 10 from the expressed need. (The test results demonstrated that, within the test framework, CANSEARCH was competitive with human intermediaries.)

But even setting aside the constraining test conditions, 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 clarificatory dialogue except in the particular form of iteration if the user has made inconsistent menu selections. The system is essentially tackling only one of Brooks' functions, Retrieval Strategy (and in the test was specifically confined to this), though the system indirectly offers the user, through its menu displays, some support in establishing the Problem Description. But this is only within the context supplied by the frames, primarily the individual frames but also the hierarchy as a whole; the user can otherwise only abandon an unprofitable path and restart the whole query formulation process, or modify the query after a collection search. The system is also confined to a very small number of knowledge sources and these are, as it were, flavoured ones. The main source is the indexing language which gives the set of available concept labels and their hierarchical structure, i.e. MeSH. 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, modifiers, etc.

However 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, i.e. that they deal with types, sites, forms of therapy, etc. These sources are not, however, explicit and distinct, but are implicit, embedded in the frames and rules.

We can indeed identify other sources underlying Pollitt's system, for example we can view MeSH as embodying some knowledge of the subject literature as much as of the subject. But it is important to recognise the constraints imposed on these sources by the fact that the knowledge is expressed through the medium of MeSH, i.e. through a limited artificial language. It is true that the fact that MeSH encapsulates a range of relevant knowledge, or facilitates its incorporation in the system (for example in taking account of the typical elements of a therapy request), provided Pollitt with a good deal of leverage in setting his system up: MeSH was a means as well as an end in the construction of the rules and frames. The use of MeSH is of course primarily justified as the necessary channel of communication with the back end, since Pollitt deliberately excluded the possibility of searching in anything else. But it is possible that it is prematurely binding on the user, especially if the system is intended to have a real role in determining the user's problem description rather then just 'translating' it, in the way CANSEARCH is set up.

Implications of Pollitt and Brooks

Suppose now, however, that we consider what Brooks and Pollitt taken together suggest, on the grounds that if Brooks lays out requirements to be met by an automated intermediary, and proposes a general system design, Pollitt's CANSEARCH shows that some automation is possible and indeed that a specific implementation of the type of system proposed by Brooks is feasible. Thus is is possible to view CANSEARCH as a miniature distributed expert system, with the sets of context rules as functional experts (and perhaps frames and their operationally associated rules as subfunction experts), cooperating both globally by contributing to the overall task of constructing a search query and more immediately via their blackboard messages. As already indicated, CANSEARCH is very limited, so the natural question is how do we move from such intermediary systems to the more powerful ones Belkin and his colleagues are seeking.

It is hard to quarrel with the idea of functional expertise as the driving force of the front end; and the specific idea of a distributed expert system with a set of individual experts each specialised to seek some particular goal but collectively, in achieving these, contributing to the system's overall goal, is a plausible one. Again, there is a good case for assuming, in the context of the set of functions identified, that pieces of information may be relevant to more than one goal, so communication between experts is required, and that this communication could be effectively achieved by open messages posted on a common blackboard.

But there are known problems about these ideas. Distributed expert systems turn out to be thoroughly complicated (see, for example, the synopses in Smith 1985). Thus it is necessary to be quite clear about what having such a cooperative system implies, and the specific question of interest is whether there are properties of the document retrieval task which make solutions to these problems unusually difficult, or alternatively relatively easy, to achieve.

Control

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. Imitating the intermediary, the system itself must have the dominant role, working towards the overall goal of satisfying the user by attacking one contributing goal, i.e. one function, and then another, but at the same time adapting to user initiatives.

For this some means of determining that goals have been satisfied is needed. The assumption must be that some sort of global state evaluation is called for. This has to be more than simply polling the functions to see whether they are all individually satisfied because, quite apart from the fact that it is not at all obvious how any particular function can determine how its goals are satisfied, it is unrealistic to require that every function is satisfied. At the same time, because the user is a free, but by definition inadequate, agent, there is no natural evaluation base in a given body of user input: i.e. it is not clear that the idea of system evaluation by coverage of the input, following the Hearsay model (Erman et al 1980), applies in this case. On the other hand, though the interface is essentially system driven, i.e. it is not external data driven, it is quite evident that providing a global evaluator designed to determine whether the system as a whole greater than the sum of its parts knows enough, or should seek more information from the user, is not a trivial business.

It is in any case possible that the evaluation should be more focused, most plausibly 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 and also an ability to call on other functions for information, that it has taken its model of the user's problem as far as it can.

It is also necessary, to support control, to have some means of ensuring cooperation. Cooperation applies at two levels. The more important one is that represented by the notion that all the component function goals are subgoals of the overall system function of satisfying the user, and therefore that satisfying all the individual function goals will ensure that the global goal is also satisfied. Cooperation also figures in the subsidiary form represented by the transmission of messages between functions, both in the action itself and in the supposition that these messages are helpful. However while it may be that in the document retrieval case, unlike the robot one for example, there is no reason to suppose that one function, in satisfying its goal, will specifically clobber another, it is not obvious that there is no need for positive action to stimulate and modulate functional activity, i.e. that it is sufficient if the system is quite passive, and internally data driven through messages spontaneously generated by individual functions. Functions should be able to request as well as just accept inputs from other functions.

This is not primarily a problem for the individual functional experts if it is assumed, not unreasonably, that they can carry out their internal functions autonomously, i.e. without competing for processing resources: responding to an input request is much like responding to input information, though in both cases internal scheduling for responses may be required; it is rather a problem for the system as a whole. Thus it is not necessarily the case that satisfying one function's goal contributes effectively to the system's overall goal, for example that it is useful to go on elaborating the user model regardless of how rudimentary a problem description can be achieved: indeed it may be positively unhelpful to have the User Model function harassing Problem Description to get more information. The system has therefore to apply some criteria for cooperative behaviour, taking into account the status and state of the various functions, to its internal operations. Achieving balance here is distinct from maintaining a balance in the interaction between the different functions and the user; but there is also a problem in the requirement to balance the needs of the various functions with the restriction imposed by the linear interaction with the single user, and indeed with that of ensuring acceptability to the user through coherent dialogue.

Both evaluation and cooperation, therefore, imply some mechanism for determining, in a system with interdependent components of the kind envisaged, what to do next.

CANSEARCH has a clear control structure determined primarily by the rule types and in a more detailed way by the frame hierarchy, since the rules are ordered both by context and within a context in a way which is locally open but globally closed. Brooks' analyses of the focus transitions in interviews (cf Figure 5) show a complex flow (as indeed there is a similar complexity within the sequence of utterances making up a focus), but what determines it is far from clear. Daniels et al's (1985) analyses of interviews suggest that it is possible to detect patterns of logical and temporal ordering from which it might be feasible to derive a characterisation of the front end task from which a control flow would follow, though this would have to allow for many alternative paths in an interaction and for a good deal of initiative by the user even if the balance of direction was with the system. But it would appear, at any rate, that the nature of the task would imply a more constrained sequence of operations that those allowed for in a general dialogue model like Reichman's (Reichman 1985), for example.

Even so, Daniels et al's analyses show very clearly what the real control problem is. It is possible to detect a common gross structure in the interviews, namely a gradual shift of attention from User Model and Problem Description to Retrieval Strategy and Response Generator, but this is a very weak structural characterisation, not so much because it only reflects a tendency as because the characterisation is a very high-level one which subsumes much lower-level variation.

The control difficulty in the document retrieval case comes from the fact that the system is not modelling an external phenomenon with a well-defined, or definable, structure. The 'problem structure', to use Belkin and his colleagues' term, i.e. the nature of the interaction between user and intermediary, can only be characterised in a schematic fashion covering very many, quite different, lower-level possibilities (i.e. possible structures, not possible individual instantiations: there are these as well, of course). There are so many potential combinations of user properties, and of document retrieval system properties, representing a retrieval situation, and also so many possible processes for identifying a situation. There are, for example, all the many search requests that could be addressed to a search system and how these could be established. There are thus no very material constraints that can be exploited to impose any particular organisation on the system's construction and application of its models. We are not in a position where, because we have a well-defined phenomenon to model, we can get a correspondingly well-defined system structure which provides a strong context for the actions and interactions of the system's components, perhaps implying that these actions, or more importantly interactions, may themselves be quite simple.

Adopting some particular problem structure, i.e. imposing some specific organisation on processing, is therefore liable to lead to interaction which is too arbitrary for the user or, if it is not unacceptable to the user, to interaction which is too restrictive and hence less effective in extracting the information that is really required. At the same time it is not feasible, just because the essential starting point is that the user does not know exactly what he wants or how to describe what he is looking for, to let the user drive the interaction and have the system operate in totally responsive mode. This would not be modelling the human intermediary.

We are therefore left with some such picture of the system's control mechanism as the following. The primary work of the system is done by the functional components: these are engaged on substantial tasks and are thus necessarily complex. But because there is no external detailed process specification to reflect, the structure of the system containing these components is itself relatively simple, though its actual behaviour in any individual case will not ordinarily be simple. Control is passed from one functional component to another depending on both external and internal data and on their needs (for example to develop their models) subject to such overall constraints, of a fundamentally preferential rather than absolute kind, as that the system works on the problem description before the retrieval strategy, and that it should not harass the user, for example by switching frenziedly from one topic to another. This implies some method of scheduling an agenda of tasks depending on an evaluation of the various functions' claims and of the consequences of the constraints.

This sounds reasonable, and is perhaps a little less vague as a system characterisation than that the expert intermediary should be a cooperative, distributed system, but it is still only suggestive. The real work is in getting the task evaluation and scheduling processes specified. Essentially the intermediary system has to operate with a nasty mix of data and function driving, combining opportunism and determination. But this requirement is in fact not obviously specific to the document retrieval case. Building a retrieval interface is a challenging expert system application, but it is just one member of a class of applications associated with complex systems combining data and function driving. The intractable design problems which arise in these cases, and the complicated solutions which follow, are well illustrated by Corkill and Lesser (1983), for example.

These control issues have no general solution: the solution has to depend on the particular purpose and resources of the expert system. However it is easy to underestimate the problem in the present case because of the way the exercise of control is hidden in the human intermediary. Pollitt's strategy, with the flow of control largely built in, is effective for his limited system, where it is also feasible to constrain the interaction with the user, but it is not obvious that it is extensible to an interface with a larger range of functions and more flexible interaction.

Control is the dominant problem in designing the expert intermediary, but there are other non-trivial ones.

Blackboards

The second, related problem is that of the nature of the blackboard: is this just, as Belkin and Brooks' model appears to imply, a single open board where messages are posted by any functional expert for any other expert to read? Of course messages from one sender might for convenience be placed in a particular area, for anyone to read, but there is no obvious general case for posting to a specific reader area, since individual messages may well be relevant to several experts: indeed it is probably sensible to assume that the information outputs of any function may be of interest to any other function. In Pollitt's case there are separate boards associated with the various aspects and subaspects of requests, presupposing some selectivity in readers or writers. But this sort of mechanism, which is like Hearsay's (Erman et al 1980), is too restricted for the general case, and Pollitt's boards may be viewed as having a more important role as the internal means of communication for the subsystem represented by a single expert (i.e. context set of rules).

For the system as a whole it seems there can only be a simple board. As there is no well-defined problem structure there can be no derived well-structured board. Essentially having to have an open board is a consequence of the weak problem structure the system is stuck with. We cannot have a board with a structure reflecting that of the problem which would allow the individual experts to make strong inferences from the nature or location of the messages, i.e. from the context in which messages occur. Because there are so many possible pieces of information that can be supplied, there can be no solidly-based expectations and hence none of the context a structured board can provide to support message interpretation. Messages have to be taken as they stand and used in whatever way individual functions allow. Each function has many potential models, and so there are many possible ways of relating these to form a global model. The balance of effort and information in the system is therefore with the functional experts and not with the board, so the board is just for passing messages and not for constraining the global model, as it has no starting guidelines for doing this.

But this then raises questions about the status of messages. There are some questions which stem from the logic of persistent messages for multiple readers, but these present mechanistic problems that can be dealt with by date-stamping, reading to pending in-trays, and so forth. But there are substantive questions about the status of messages as well. The tacit assumption underlying the cooperative idea is that there is a global aggregation of more, and more detailed, information; or if this is too simple, the assumption must be that senders will clearly signal that messages are replacements, or that readers will recognise them as such. But given the variety and detail both of the information supplied by the user-intermediary interaction and that exploited by the intermediary, it is evident that implicit change and modification as well as explicit correction or simple addition will be involved, and that the individual experts will have to have sophisticated mechanisms for evaluating inputs for their actual and potential utility. There are particular problems about recognising that an information item, though not relevant now, may be relevant in the future. The functions will have to rely, but not in too narrow a way, on their internal models to determine the utility of messages, with support from a whole subordinate apparatus for distinguishing old, new, repeat and reinforcing messages.

Messages

The third, closely related problem is the nature and content of the messages themselves: what are they like?

The assumption has to be that the messages are quite rich, and it is easy to see why: if messages are simple they must have a context for interpretation, but as there are no constraints to supply a context and add meaning to a message, the messages themselves must convey all the information. Indeed the functional experts will have to seek to maximise the information passed through messages themselves to provide the best opportunity for other experts to assess what they may value. This implies that we have complex messages intended to give potential users as much information as possible to work from. The only context that can be supplied with incoming messages is that provided by their sources, and this model information itself has to be explicitly expressed as part of the message. But all this raises the question of how rich messages can be encoded and decoded, and further emphasises the need for power in the experts themselves.

This is not to imply, for either of these board and message issues, that the document retrieval case is different from that of other actual or potential expert system applications. In the expert intermediary case we have a situation where everything is very unconstrained with respect to the problem structure as a whole, so it is natural to leave large amounts of initiative to the experts, and the difficulty is then getting them to work efficiently together. But it is clear that other expert system applications, for example in command and control, must pose similar problems. The point is rather that building a distributed expert system for a hard task is a tough problem, and that the kind of characterisation provided by Belkin and his colleagues is only a small step in the direction of the computational system we seek.

But there is a further problem which is very sharply raised by the kind of system structure proposed for the expert intermediary. This is that of the system's internal communication language: what do the functional experts communicate with one another in?

This is an issue whose importance appears not to have been perceived by Belkin and his colleagues, or indeed by those generally engaged with expert systems.

The experts' communication language

Pollitt essentially uses a very simple, ad hoc language for messages, e.g. 'site to specify'. It is a mistake to think of these messages, from the system's point of view, as written in English; they are written in English for the benefit of the rule writer; for the system they are just arbitrary strings. In fact Pollitt's language is strictly a hybrid, as some messages consist of MeSH terms. but Pollitt does not consider the nature of the communication language explicitly, as his system is so limited this is not a material issue.

Brooks does not address the question explicitly either. In the interviews she analysed the communication is all in the head of the intermediary; her representation of the information passed between experts is an informal natural language one (cf Figure 4), and she does not discuss the form of message representation that would be required in the automated case. However Belkin has assumed that information would be communicated in some propositional form.

But when we consider what is involved in an interface with the full functionality postulated by Belkin and his colleages, it is manifest that a very powerful communication language is required, and one that is much more powerful than a straightforward propositional or even predicate logic. It is also clear that the need for any expert to communicate with any other implies that we have to have a general purpose language (there is no point in having polyglot experts). We are not dealing with a Hearsay-type situation where there are subboards of interests to subcommunities of experts, for which different specialised languages may be appropriate.

Consider, moreover, what happens when we take the idea of a set of experts further than it is taken in Belkin and Brooks' work, but in a direction in which it is quite consistent, and proper, to take it.

If we start with the idea of automating the human intermediary this naturally leads to the notion that the communication medium should be a 'language of thought', like the manifestly sufficiently effective one inside the intermediary's head. However there is no reason to suppose that if we wish to obtain the best user interface for a document system, we will get it by confining ourselves to the capacities of the single intermediary. Any single intermediary (however good) is inevitably limited with respect to, for example, the indexing language (since intermediaries are not responsible for the design of the language they use), or the publishing conventions of the journals covered by the collection (since this is a matter for their editors), or with respect to the subject area: thus Cancerline intermediaries are not doctors. It is therefore natural to conclude that one would get a better system if the cooperating experts were each the best possible, which might be achieved by automating the skills of different human experts.

What, then, does this imply for the language used to communicate between the functions? We might argue that logically the situation is just the same as that of having a human polymath, so all we are seeking is a (necessarily explicit version of a) language of thought, which might be, for example, some form of logic, or at any rate some language with an adequately defined semantics.

One problem with this is what happens if we regard the end user himself as one of the member experts contributing to the working of the whole mechanism: he is after all in a sufficient sense an expert on himself. It is difficult then to see how an effective or convenient communication language for him would be anything other than his own natural language. However even if we exclude the user, we have to allow for the need for a very powerful (rich and flexible) language to express the kind of messages analyses like Brooks' suggest would need to be passed between experts e.g. to convey that the user is looking at community education in three African countries because they have long tradition of community education which was introduced in colonial times and was actually called community education. Now exactly as one might argue that two robots would not need to communicate in

English but in their meaning representation language, so the member experts in the document retrieval case need not communicate in natural language. But it is difficult to believe that a powerful enough language would not have to have many if not all of the distinctive properties of a natural language and so, even if it was not any actual language, would be like a natural language in significant respects. One would then have to make a very good case for not in practice using some actual language.

A possible additional reason for this would be that in a document retrieval system we are essentially concerned with natural language objects. These are primarily the document texts we are seeking, but may also if we generalise, as we should, to allow the use of natural language as an indexing language, include natural language objects in the form of index descriptions (keywords, phrases, titles, etc). It is not necessary to suppose that because the objects the experts talk about are natural language objects we should talk about them in natural language; the requirement for a natural language-like communication language then stems only from the requirement for effective communication. However it is also necessary to consider the suggestion that it may be effectively impossible (especially if we envisage an interface supporting genuinely interactive searching) to maintain an object- / meta- language distinction, i.e. that the language about and the language of form a seamless web. In this case the communication language would have to be the natural language of the documents. (Dealing with a multilingual collection is not an issue here.)

The conclusion that the system's internal language has to be, or to have the distinctive properties of, natural language is a strong one. It implies, amongst other things, that each expert has to have a serious language coder and decoder. However the difficulty of providing automatic natural language interpreters and generators also suggests that it is essential, from a practical point of view, to consider what much more modest but sufficiently expressive communication languages should be like, and equally to consider their design as an integral part of the characterisation of the expert system as a whole. As noted, Pollitt works with a very informal assumption that, for his limited need, simple 'code' messages indicating the query component slots to be filled, plus index terms, are adequate. Brooks' discussion of representation formalisms for the models to be built by the experts has implications for this question, though she does not draw them out. Thus in considering whether the semantic network formalism she envisages for the Problem Description's model is adequate, she seeks to show that it can provide the information needed by the Retrïeval Strategy function. She claims that representations she constructed for her example interviews show that the Problem Description's form of model can supply the required information but, as in Pollitt's case, the key type of information, namely the concepts to be transformed into index terms, is relatively straightforward, though establishing the correct specific transformation may not be easy. The question not discussed is exactly what form the information would be transferred in, i.e. whether it would be in the network formalism itself, which would then be the communication language: there seems to be a tacit assumption that it would, though Brooks also recognises that different experts may need different internal formalisms, which undermines this assumption, and there is also the question of what sort of resources would be needed to embed pieces of network for the communication of additional information, for example about their reliability.

The nature of communication languages is an issue which has been thoroughly addressed in the general context of distributed computing systems, and these languages may, when they are programming languages making remote procedure calls for instance, be very sophisticated. The issue in the document retrieval case is the sort of language we require for messages between the functional experts of an interface, even if we recognise that we are only taking the first steps towards natural language as the communication language.

Conclusion

It is clear that building the all-singing, all-dancing expert intermediary is a major enterprise. Pollitt has demonstrated, on the other hand, that a practical and potentially useful system, if only of a modest kind, can be built. The natural next question is how to proceed to get something better.

Unfortunately it is not obvious whether a bottom-up strategy, proceeding incrementally from CANSEARCH, would be productive in the long run. 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 targetted 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 would not be easy to extend the system's scope, for example to include a problem description function, or to relax its constraints, for instance by allowing the formulation of natural language rather than MeSH queries.

It is equally clear that there can be no starting, top-down, from the Belkin and Brooks model in its current informal form. In an attempt to realise a comprehensive retrieval system interface. the strategy which naturally suggests itself, therefore, is an attempt to have a system motivated by the fully distributed model, with a number of functions, but with each function of a fairly elementary kind. Choosing the right set of functions and fixing on a sensible simplicity in each are therefore the next research questions.

It is also possible that a useful way of reducing the complexity discussed in this paper would be to propose a less thoroughly distributed model of the user-system interaction itself, in the sense of placing less emphasis on the various independent models constructed by the different experts, and rather more on the way they are integrated into a single model. Brooks' picture is essentially of a collection of separate models used in some way by the system as a whole but not effectively synthesised as a single entity; it could be that there is mileage to be got from synthesising a more comprehensive model, so the system's blackboard was the place where this was explicitly constructed.

Acnowledgements

This paper was written during the tenure of a GEC Research Fellowship. I am grateful to Tony Morgan and Helen Brooks for discussion and comment.

References

Belkin, N.J., Seeger, T. and Wersig, G. 'Distributed expert problem treatment as a model for information system analysis and design', Journal of Information Science 5, 1983, 153-167.

Belkin, N.J. and Vickery, A. Interaction in information Systems, Library and Information Research Report 35, The British Library, 1985.

Belkin, N.J. and Windel, G. 'Using MONSTRAT for the analysis of information interaction', in Representation and Exchange of Knowledge as a Basis for Information Processes (Proceedings of IRFIS 5, 1983)(ed Dietschmann), Amsterdam: North-Holland, 1984.

Brooks, H.M. An Intelligent Interface to Document Retrieval Systems: Developing the Problem Description and Retrieval Strategy Components, Ph D Thesis, City University, 1986.

Brooks, H.M., Daniels, P.J. and Belkin, N.J. 'Problem descriptions and user models: developing an intelligent interface for document retrieval systems', in Advances in Intelligent Retrieval: Informatics 8 (Proceedings of Informatics 8, 1985), London: Aslib.

Corkill, D.D. and Lesser, V.R. 'The use of meta-level control for coordination in a distributed problem solving network', Proceedings of the Eighth International Joint Conference on Artificial Intelligence, 1983, Los Altos: William Kaufmann, 1983, 748-756.

Daniels, P.J., Brooks, H.M. and Belkin, N.J. 'Using problem structures for driving human-computer dialogues', in RIAO 85 (Actes of the Conference: Recherche d'Informations Assistee par Ordinateur, 1985), Grenoble: IMAG, 1985.

Erman, L.D. et al 'The Hearsay-II speech understanding system: integrating knowledge to resolve uncertainty', ACM Computing Surveys 12, 1980, 213-253.

Pollitt, A.S. (1986a) 'A rule-based system as an intermediary for searching cancer therapy literature on Medline', in Intelligent Information Systems: Progress and Prospects (ed Davis), London: Aslib, 1986.

Pollitt, A.S. (1986b) An Expert Systems Approach to Document Retrieval, Ph D Thesis, Huddersfield Polytechnic, 1986.

Reichman, R. Getting Computers to Talk Like You and Me, Cambridge MA: MIT Press, 1985.

Smith, R.G. 'Report on the 1984 Distributed Artificial Intelligence Workshop', The AI Magazine 6(3), 1985, 234-243.

Sparck Jones, K. 'Intelligent retrieval', in Intelligent Information Retrieval: Informatics 7 (Proceedings of Informatics 7) (ed Jones), London: Aslib, 1983.

Problem State (PS)
e.g. being formulated, fully specified

Problem Mode (PM)
e.g. get references, talk with people

User Model (UM) : status; goals ...
e.g. is student; complete thesis

Problem Description (PD) : topic; context ...
e.g. forestry; writing survey

Dialogue Mode (DM)
e.g. speech, menu

Retrieval Strategy (RS)
e.g. Boolean, generalise

Response Generator (RG)
e.g. list titles, give number of documents matching

Explanation (EX)
e.g. indicate search expertise, motivate index terms

Input Analysis (IA)
e.g. parse text, note menu selection

Output Generation (OG)
e.g. produce text, display new menu

Figure 1

MONSTRAT model information system functions
(summary synthesised from various sources)

UM - User Model

IRS	experience of retrieval
USER	status
BACK	background
KNOW	knowledge of field

PD - Problem Description

TOPIC	search topic
RES	content of research
SUBJ	subject background
DOCS	form of documents
SLIT	domain literature

RS - Retrieval Strategy

TERMS	term selection
QUERY	formulation
STRAT	search strategy
DB	database

Figure 2

Subfunctions of three system functions
(summarised from Brooks 1986)

```
utterance 1   I  What's the problem?
          2   U  I'm just beginning a research project
          3   I  mm
          4   U  I'm a student at LSE
          5   I  yeah
         (4)  U  in the geography department doing a thesis on forestry
  ...
```

Figure 3a : simplified example of interview between intermediary and user

focus	speaker	utterance	subgoal
1	I	1	RES
	U	2	PDIM
	I	3	-
	U	4	USER
			USER
			UGOAL, RES

Figure 3b : subfunction goals addressed by utterances in the interview focus

Figure 3

Interview analysis
(simplified from Brooks 1986)

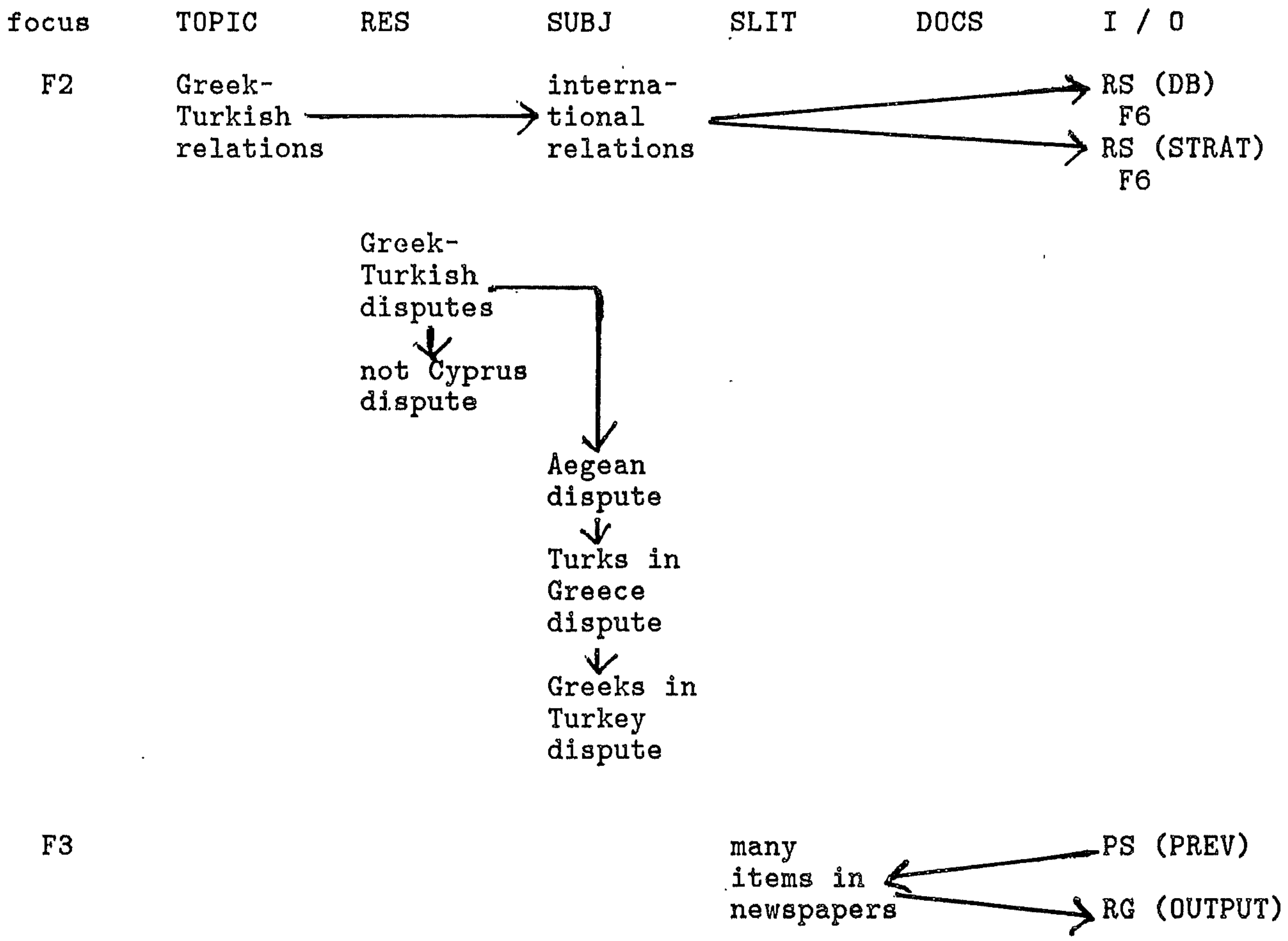

Figure 4

Interview items contributing to Problem Description subfunctions
(simplified from Brooks 1986)

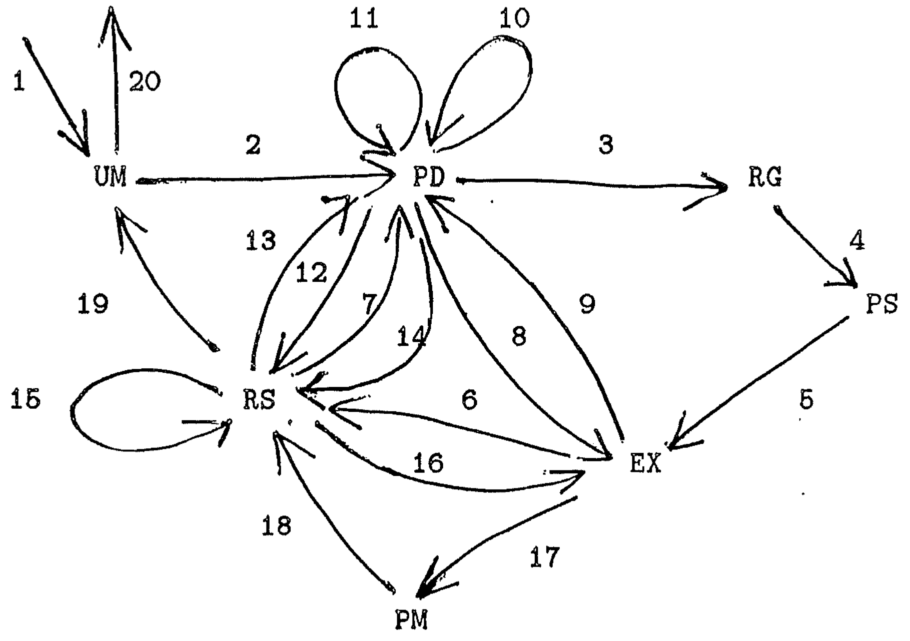

Figure 5

Map of interview focus transitions between functions
(taken from Brooks 1986)

external sources

- verbal input
- presearch form
- manuals

internal sources

- structure of subject areas
  e.g. forestry occurs in places

- structure of world knowledge
  e.g. management is linked to economics

- particular domain

- user types

- subject literature

Figure 6

Knowledge sources used by Problem Description function
(summarised from Brooks 1986)

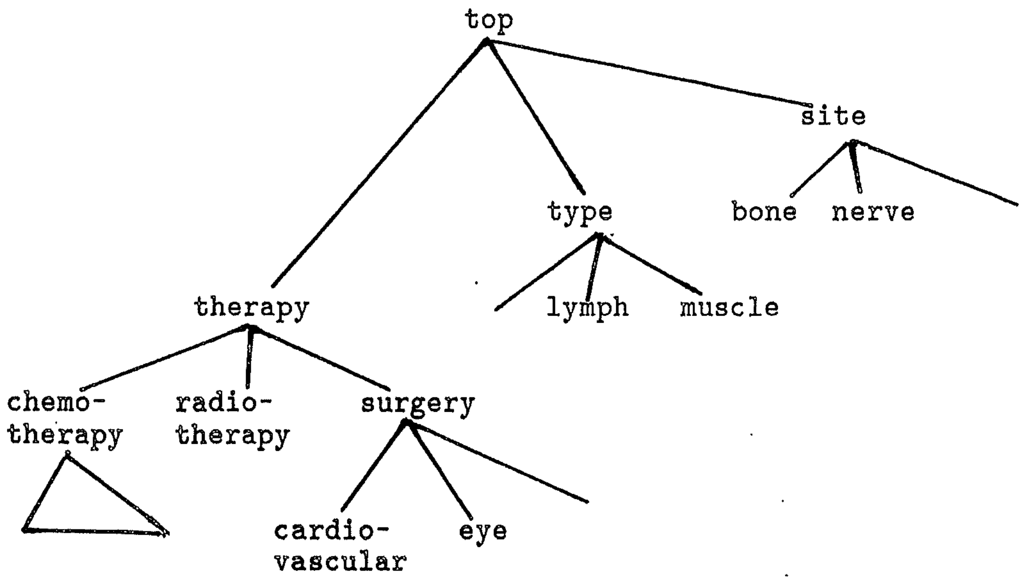

Figure 7

Frame hierarchy
(simplified from Pollitt 1986b)

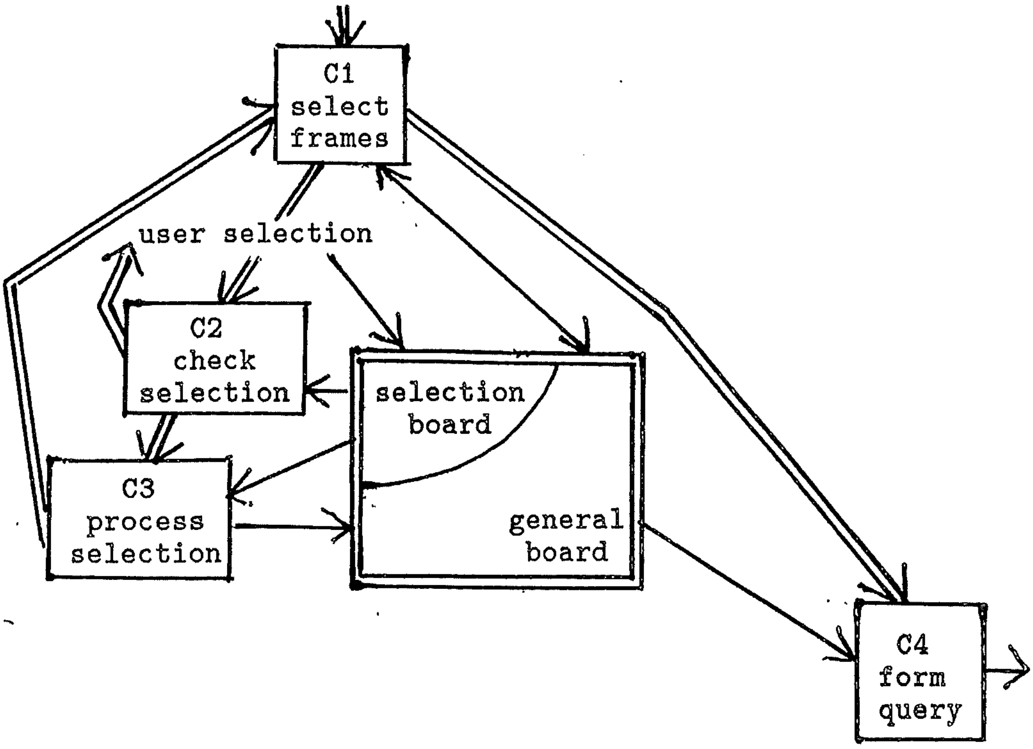

Figure 8a : system architecture

```
control
site
primary site
secondary site
type
therapy
therapy subboard
drug therapy
radiotherapy term
surgery therapy
drug subboard
surgery subboard
patient
miscellaneous
statement
```

Figure 8b : blackboards

Figure 8

CANSEARCH structure
(simplified from Pollitt 1986b)

```
Context 1 : frame selection rules

   IF 'site to specify' on site board
   THEN erase 'site to specify'
        display frame 15 <choice of primary, secondary>
        get user selections


Context 2 : selection checking rules

   IF 'all cancers' is selected
    AND 'cancer at a particular site' is selected
   THEN display 'select all cancers or particular cancers'
        return for reselection


Context 3 : handling user selections

   IF 'cancer at a site' is selected
   THEN deselect 'cancer at a site'
        write 'site to specify' on site board


Context 4 : forming the MeSH query

   IF 'ear' is selected
   THEN deselect 'ear'
        write 'EAR NEOPLASMS' on primary site board
```

Figure 9

Examples of rules
(simplified from Pollitt 1986b)

'5 FU in the treatment of breast cancer'

```
"SUBS APPLY DT
1: BREAST NEOPLASMS
"SUBS CANCEL
"SUBS APPLY TU
2: FLUOROUACIL
"SUBS CANCEL
1 AND 2
3 AND HUMAN
```

Figure 10

Input need statement and output MeSH query
(taken from Pollitt 1986b)