Automating the librarian: a fundamental approach using belief revision

Alison Cawsey
Julia Galliers
Steven Reece
Karen Sparck Jones

Computer Laboratory, University of Cambridge
New Museums Site, Pembroke Street, Cambridge CB2 3QG, England
ac, jrg, sr, ksj @ cl.cam.ac.uk

A shorter version of this paper entitled 'Automating the librarian: belief revision as a base for system action and communication with the user' appeared in *The Computer Journal* 35, 1992, 221-232.

Abstract

This paper describes a current research project investigating belief revision in intelligent systems by modelling the librarian in interaction with a literature-seeking user. The work is designed both to test a theory of agent behaviour based on belief revision proposed by Galliers, and to evaluate a model of the librarian developed by Belkin, Brooks and Daniels, through computational implementation. Agent communication is seen as motivated by and motivating belief changes, where belief revision is determined by coherence, combining endorsement, connectivity and conservatism. The librarian is viewed as a distributed expert system with many individual specialised functions operating in particular belief domains. The paper describes our first implementation of the belief revision mechanism and of a very primitive librarian, designed to test the basic viability of our ideas and to allow us to explore different forms of the distributed system architecture.

Keywords
belief theory, information retrieval, distributed system

Acknowledgement
This research is supported by Grant SPG8930752 of the UK Tri- Council Initiative on Cognitive Science and Human-Computer Interaction.

Contents

Automating the librarian: a fundamental approach using belief revision

1 Introduction

The work described in this paper is basic research aimed at a challenging and necessarily very long-term goal, automating the librarian. Searching online bibliographic databases to obtain literature references or documents for end users is an important component of a modern librarian or information officer's work. It requires professional knowledge and skill, so providing conveniently direct access to bibliographic services for end users instead calls for sophisticated interfaces able both to determine the user's need and to express this in a way suited to searching the bibliographic file. In general, that is, it is necessary both to identify the user's topic and to specify this in the indexing or classification language used to describe documents in the file. But even when the search language is the natural language of the file documents' titles, abstracts or texts, professional knowledge and skill is required for effective searching.

Some first steps have already been taken in automating the intermediary by, for example, Pollitt (1986), Vickery et al (1987) and Brajnik et al (1990); but what has been done so far has been very limited, especially in the system's subject scope. More effective systems would call on artificial intelligence (AI) techniques for reasoning on knowledge in interacting with and acting for the user. The project described in this paper is therefore concerned on the one hand with appropriate general mechanisms for agents manipulating beliefs and conducting dialogue, and on the other with deploying these mechanisms within the framework supplied by the literature searching task and by a model of the librarian's characteristic knowledge and actions.

The project is intended to investigate key ideas about what is in principle involved in automating the intermediary. But given the complexity of the real librarian's task, it can only attempt an initial, very simplistic, laboratory system: the foundational character of the work which is needed for eventual proper systems means we cannot envisage realistic prototyping. The project is also in progress, so this paper describes the work's starting point, plan, and what has been done so far, but cannot provide final results or evaluate these. It must further be emphasised that while the research is presented here as directed towards document retrieval, because automating the intermediary requires general capabilities as well as specialised knowledge, and developing these capabilities is a concern of AI, this work is also seeking to contribute to AI as a whole. Thus while from one point of view the aim is to apply AI ideas to information retrieval (IR), from another IR provides a valuable study context for modelling the way any agents adopt or change their beliefs about the world, particularly through engagement in dialogue. At the same time, while the work is concerned with helping the user to obtain literature, it is not primarily concerned with the type of display-oriented support system developed by e.g. McAlpine and Ingwersen (1989), or with 'plain' public access as in Okapi (Walker 1988).

Section 2 of the paper describes the general theory of belief revision underlying the whole; Section 3 the view of the intermediary adopted as a project starting point; and Section 4 the current state of the research and intentions for the future.

2 The theory of belief revision

As noted, the aim is to provide the power needed for an automated intermediary by exploiting a general theory of belief revision as a mechanism motivating both the system's external interaction with the user and its internal problem solving. Intelligent agents are continually revising their beliefs, and this applies to interaction between library users and librarians as much as to other dialogues. Interaction on literature seeking is not driven by fixed goals or manifested in a unidirectional flow of data from one party to the other. The system dialogue fragment of Figure 1, of the kind recorded for actual sessions, clearly shows both parties revising their beliefs about what is wanted. Thus the librarian, having started by assuming that when people ask for books on plants they want books on growing plants, is obliged to revise this belief to accommodate a request for books on other aspects of plants. But equally, the user, having started by saying he wants a book on cacti, revises this belief to accept that books on other sorts of plants may be appropriate. For this illustration we may envisage the eventual output of the mutual belief revision process as a submitted search request of the conventional sort for online services, in the form of a Boolean combination of terms in some controlled indexing language of the kind exemplified by MeSH. Both parties, that is, have collaborated to arrive at the actual search specification aimed at retrieving literature from the file to meet the user's real need.

2.1 Galliers' approach

The particular theory of belief revision the project intends to apply is that proposed by Galliers (1989, 1991, in press). This starts from the position that an intelligent agent is obliged, in a changing world of which any agent has only partial knowledge, to operate autonomously. An agent, that is, cannot rely on predictable states of the world, or on predictable behaviour by other agents within the world, and therefore has to do the best with the knowledge and powers it does have in setting its goals and in planning and acting to achieve these. An agent also seeks to behave rationally by maximising its own outcomes, so in a context of uncertainty this implies adaptation. In particular, a continuously changing environment stimulates changes of mental state in agents, i.e. since all knowledge is actually belief, changes in the environment stimulate the revision of beliefs. This revision depends on the agent's goals, but as the environment changes, the goals can change too. Equally, having or adopting goals, which is a fundamental property of agents, implies action in and reaction to the world motivated by planning, and especially by strategic planning, to effect changes in other agents' mental states.

The nature of communicative behaviour in interactive dialogues between agents follows from the characteristics of and constraints on agent behaviour in general. Inputs from other agents suggest changes to beliefs, and an agent's own outputs are prompted by potential or actual changes relating to the agent's evolving goals. Thus an agent's contribution to a dialogue may be intended to check candidate changes of belief, that is to gather information to choose between competing beliefs, as well as to do what is normally thought of as simply collecting data or seeking to influence others, which are in fact also processes to be viewed as deploying beliefs, i.e. as revising an agent's beliefs in order to attain or determine goals.

Beliefs about other agents are clearly important in the interaction, but not just because they are part of the furniture of the world. As any agent has only limited powers to effect action, it needs cooperation to achieve its goals. This, however, in turn requires that it be cooperative. Thus dialogue is a process of negotiating and mutually accepting beliefs and hence intentions to act. Dialogue is a public manifestation of pervasive, goal-motivated belief revision in each participating agent, operating at every grain level in the characterisation of mental states. For example in the dialogue of Figure 1, the agents have beliefs (separate or shared) about the whole area of discourse, namely plants, about particular matters within this area, namely diseases or allergies, about needs to seek information, and about the means of seeking information.

As these examples imply, moreover, belief revision is also pervasive because it covers belief changes of all kinds, not just simple reversals but modifications of all sorts, including both changes in content, like more specialisation of beliefs, and changes in status, like less commitment to beliefs. Moreover, as beliefs are inferentially related, revision affects belief sets, not just single beliefs but whole webs of related beliefs. Thus a new belief may allow inferences affecting several other beliefs, and may mean there is more or less support for other beliefs. In general, a change to a single belief stimulated by interaction with the world or other agents affects the evidence supporting a whole network of related beliefs; equally, individual beliefs gain their value from the way they figure in a whole network of beliefs.

This very general picture is relatively uncontroversial: it has of course to be fleshed out in substantive enough detail for computational implementation. Thus given some new input from a dialogue, how does an agent decide whether and how to revise its existing beliefs to accommodate the new information? More specifically, how does the agent choose what to reject when there is a conflict of any sort? Conflict resolution is not an occasional requirement having the form of a raw true/false opposition for an individual proposition: it is a normal requirement, having the form of a choice about alternative sets of beliefs representing different ways of responding to any change associated with new information. These responses, moreover, refer not only to beliefs embodying an agent's knowledge, but also to beliefs representing an agent's goals.

A proper theory of belief revision must involve three things: a way of characterising beliefs; a set of criteria for preferring some revisions to others; and a mechanism for applying these criteria to identify the preferred set of beliefs. Specifically, as beliefs form webs of related beliefs, what is needed is a means of handling the way individual beliefs contribute to the structure and solidity of a whole web, and of taking account of the propagation effect of changes at the level of individual beliefs, whether the change modifies an existing belief, adds a new one, or deletes an old one.

2.2 Details of the theory

The theory developed in Galliers (1989, 1991, in press) is essentially pragmatic, and in the spirit of work by Gardenfors (1988), Doyle (in press), and Harman (1986). It focusses on belief revision, characterising beliefs qualitatively rather than quantitatively, and regards all accepted beliefs as certain but variably corrigible, rather than as variably certain. Both of these features of the theory make it rather different from most approaches adopted in AI, where the choice of what revision to make is not normally addressed, and from approaches using quantitative certainty.

In the theory some beliefs are more persistent than others: because they have more information value or explanatory power, the agent is less willing to abandon them. Ground assumptions, that is those beliefs which inferentially justify other beliefs, are particularly important here. These cannot themselves be justified: they are taken as a baseline. But they are *endorsed* with source information (cf Cohen 1985), and some endorsements are stronger than others. For example, beliefs embodying information received at first hand may be more strongly endorsed than those received at second hand. This applies to perceptually received or linguistically conveyed information. The theory has a number of types of endorsement which can be naturally, i.e. heuristically, ordered to provide a base for discriminating among ground assumptions. Agents will be more unwilling to give up more strongly endorsed assumptions. Moreover, though endorsement is not propagated directly to derived beliefs, since it is not obvious how derived endorsement values can be calculated from multiple different input values, it does provide an indirect means of discriminating among derived beliefs.

Figure 2 shows the types of endorsement, and the rank ordering over them. This ordering refers to strength in a general sense: individual endorsement types may themselves embed more specialised notions of strength appropriate to their particular character, for example strong or weak linguistic communication. Thus strong first hand communication provides stronger endorsement than any second hand communication can, but strong second hand communication is stronger than weak first hand communication. The latter in turn provides stronger endorsement than values (for example ethical or social ones), and values in turn rate higher than mere hypotheses. It must be emphasised that 'communication' is used in a very abstract way here to refer to what is deemed meaningful about the world by the receiving agent, so first hand communication is limited to direct perception, and linguistic communication, necessarily depending on another agent and hance having some element of indirection, is second hand. Communication in this paper will normally refer to the linguistic case, relevant to dialogue, so this is second hand in terms of the general theory, but may nevertheless within its own linguistic framework be deemed strong or weak by the receiving agent.

There is, however, more to the general idea of persistence, i.e. resistance to change, than endorsement. It is also necessary to consider the relations between beliefs, i.e. as beliefs are inferentially *connected*, the way in which connectivity reinforces beliefs. The more support a new belief offers to others, the more useful it is. Thus in evaluating alternative revisions of a set of beliefs, i.e. proposed alternative revised sets, as responses to an input, it is necessary to consider how these improve the derivational, and hence explanatory, justification for beliefs as this is embodied in the connectivity among the beliefs in a set.

Endorsement and connectivity together determine the *coherence* of a set of beliefs: one set of beliefs is more coherent than another because of the way its constituent beliefs are connected and its assumptions are endorsed. Belief revision is thus a matter of evaluating alternative belief sets, constituting different responses to new data, to identify the most coherent set (or, possibly, sets). In general, more connectivity and stronger endorsement give more coherence, but connectivity is treated as more important than endorsement. Thus in considering alternative ways of revising beliefs, connectivity is examined first so alternatives with more connectivity are preferred, and endorsements are only investigated when connectivity alone does not unequivocally determine a single preferred set of beliefs.

This way of handling beliefs fits a conservative approach to revision which is intuitively plausible. This is to look at the justification for a belief only if it is challenged, and to abandon it only if the result is a more coherent web of beliefs. Thus one belief set embodying some particular belief of concern will be revised, i.e. will be replaced by another, only if there is good reason to do this. There is no need for an agent to evaluate and seek consistency among its beliefs unless this is required, and it is thus quite possible for an agent to have a mass of miscellaneously endorsed and variably consistent beliefs, in fact implicitly representing alternative internally consistent belief sets. But this generally conservative approach also takes a more specific form as a *conservation* rule which is applied when there are alternative coherent revisions, to select the one(s) making least change to the previous state.

It must be emphasised that these notions presuppose some delimitation of the universe of beliefs, that is some context- driven means of bringing beliefs from the agent's overall stock into the current focus of attention. There may thus be beliefs in the current set which are not accompanied by their supporting ground assumptions, i.e. assumptions are ground assumptions with respect to the current context, and as the context changes, the assumption set may change. However for simplicity we are assuming that an agent's entire stock of beliefs is relevant to the current situation and thus to any ongoing dialogue. It is also the case that further concentration on where revision matters follows naturally from the idea of *core* beliefs. A pragmatic theory of beliefs accepts that some beliefs (relative to a context) will be held as core, for whatever reason; these may be ground assumptions or derived beliefs. Connectivity, endorsement and conservation are then considered primarily as to how they affect support for these core beliefs.

Implementing this theory of belief revision computationally requires a specific mechanism for constructing and evaluating all the belief sets which constitute alternative ways of dealing with some new input. One of these will in fact, because agents are autonomous, be the no-change existing belief set. At any one time there may indeed be several current alternative belief sets in play which the agent has no information for choosing among and for which decision information is required. Thus in the face of new input, the agent may have to consider alternatives for each of these current sets. All this seems very elaborate, but is a natural consequence of the fact that agents had only partial knowledge, that is have alternative hypotheses about the state of the world and corresponding alternative goals and plans. The implementation also requires proper definitions of the evaluation criteria as these apply to and allow comparisons between whole sets of beliefs, that is definitions of connectivity over a set of beliefs, of endorsement for a whole set, and of conservation in a set.

It is in addition necessary to develop the theory to handle goals and any other concepts (e.g. intentions, plans) required to drive agent and system action. While the belief revision theory, viewed in a sufficiently abstract way, subsumes these notions, it is necessary to ensure that they are made specific enough to be effective for agents that have to organise and execute actions.

2.3 Increased Coherence Model implementation

The generic class of mechanisms for creating and modifying belief sets, truth maintenance systems (TMSs), has an intrinsically exigent job to do. We have implemented a particular mechanism, ICM (for Increased Coherence Model), which can operate efficiently as the effort of set manipulation is reduced from the general case both by staged processing and by confining this to sets affecting core beliefs.

Processing has four stages, each addressing one of the factors contributing to the preference ordering on sets of beliefs. One establishes a baseline by identifying all the maximal sets involving core beliefs that are internally consistent and self- justifying relative to the context. The remaining three stages deal successively with connectivity, endorsement, and conservation in relation to these consistent sets. Connectivity is investigated to identify those sets offering the most additional derivational support links (proofs) for core beliefs; endorsement is evaluated to identify the sets with the best overall endorsement for core beliefs; and conservation is used to identify the sets making the least change to the previous state. As connectivity is more important than endorsement, and endorsement than conservation, the ordering is significant, with each stage constituting a filter: the processor for the next stage is only invoked where the previous stage has not selected a single preferred set and further discrimination is required. It could thus happen that revision is determined solely by connectivity considerations, or that endorsement has to be taken into account as well, or that conservation has also to be invoked, perhaps even then without final resolution: this reflects the absolute priorities rather than relative status the theory gives to different types of information about beliefs, within its generally conservative framework.

The specific way in which the three criteria are defined is as follows. Connectivity is considered only in relation to core beliefs. Revisions are scored according to the number of core beliefs for which the revision provides additional justification, i.e. new derivational links. (It does not take into account the number of new links for any single belief.) Thus one revision is preferred to another if it provides additional support for more core beliefs. The endorsement data for a set of beliefs is evaluated using simple heuristics: the first is that one set is preferred to another simply if it has more of the top-ranking kind of endorsement, i.e. more first-hand, strong communications. If this heuristic is not sufficient to select a single set, further heuristics are applied. Thus preference is next given to strongly communicated or specific (as opposed to generic, default) assumptions, regardless of negative, default or value assumptions; and if further sorting out is needed, weak or default assumptions, and values in conjunction with other evidence, are considered (Galliers, in press). Conservation is evaluated simply by considering the intersection of an initial and revised belief set, and preferring the revised set with the largest intersection. (We are experimenting with specific choices here: currently we are intersecting with the previous set of pervasive beliefs - see below.)

The ICM has three components, an incremental and possibly inconsistent database, an inference engine, and a belief maintenance component. It is focussed on positive undermining (Harman 1986), where the principle of positive undermining states that if all the justifications for a belief are disbelieved it does not follow that this belief itself is necessarily disbelieved: beliefs may be disbelieved only if they are themselves in question. However though the ICM is ultimately motivated by cognitive modelling, some details of the way it works, for instance in relation to the database, follow from mechanistic considerations.

The ICM builds and maintains a 'cognitive state', the collection of currently preferred belief sets. As indicated earlier, each belief set is a consistent whole, involving more or less linked beliefs drawn from the system's total stock of beliefs. Beliefs are simply propositions, and particularly when they are considered or manipulated independent of any endorsement information, may be referred to as 'propositions'. Thus the system's total stock of beliefs is essentially a mass of propositions. The system's database however also indicates the derivational links between propositions, and the endorsements on those propositions which are ground assumptions. The beliefs in any of the currently preferred belief sets are distinguished as 'current beliefs' and those beliefs which occur in all the the currently preferred sets are referred to as 'pervasive beliefs'.* As indicated earlier, the beliefs in a set will be a mixture of ground assumptions and derived beliefs. Assumptions not only have explicit endorsements; they may also have justifications in the form of derivation links from other beliefs in the set ultimately reflecting, but not directly embodying, distinct endorsements.

We distinguish two classes of proposition: those which are fed into the system from some external source, which we call 'observations', and those propositions inferred from observations, which we call 'derived propositions'. Observations are all assumptions, but the reverse is not the case: we are especially interested in observations as they represent the system's dialogue inputs. We apply the constraint that each assumption, and thus each observation, must figure in one polarity or another, positive or negative, in each preferred belief set. This requirement was motivated by the desire to fit each observation somehow into the system's collection of possible worlds, as represented by the current belief sets constituting its cognitive state. While it might be assumed that observations, if they are raw perceptions, are in some sense always positive, we allow them more informally and generally to be either positive or negative: a received utterance, for instance, may be negative. An observation will therefore always be taken as embedding a positive or negative operator, so though its opposite form is always formally a negative, its meaning may be positive. We refer for convenience to complementary 'pairs' of assumptions or observations, and to an assumption or observation and its pair. Observations will normally have stronger than minimal endorsements; their pairs, like the constructed pairs for other assumptions, are naturally only hypotheses. Finally, as the sources and occasions of observations are important in relation to their endorsements as assumptions, we index observations: we currently only do this only in a very simple way by source, but we intend to index by time as well. The propositional content of two different observations may therefore be the same, as it will correspondingly be for their pairs. We indeed more generally treat assumptions as distinct, even though their endorsements as well as propositional contents are the same, to allow for a better and richer use of index information in relation to endorsement than our current simplistic provision of input data does.

Belief revision is invoked by the stimulus of some external observation. If the proposition in question has not been encountered before, it is automatically negated and the two complementary propositions are passed to the database. Limited * We do not call them 'common beliefs' as this expression has a well-established different meaning. Pervasive beliefs are often referred to just as beliefs, since they can be viewed as the potential beliefs that are actually believed, but we do not want to make this distinction here, and retain the wider meaning of 'belief'. (i.e. not logically omniscient) inferences are drawn from each member of the pair using the inference engine, and the two propositions and all their respective derived propositions are passed to the belief revision component.

The system reasons both about whether to accept a observation and about how to accept it, where these are both interpreted as applying to the observation and its pair. Thus not accepting the observation implies accepting its pair. The system considers both members of the pair, in an attempt to find the one which coheres most strongly with its current belief sets, though it may not be able to retain these sets, even if modified, and may also not be able to choose between the two members of the pair, as there may be competing, equally coherent sets for both members of the pair. Belief revision as we are broadly interpreting it, or belief maintenance, informally covers just adding beliefs to an existing body of beliefs, or just taking them away, or both. But working explicitly with complementary pairs means that while we may add a belief to a set, we cannot just remove one, as doing this requires we add its pair instead. We thus have two belief maintenance processes: belief *addition* and belief *modification*

In belief addition, the observation or its pair are in turn added to each belief set in the current cognitive state, to produce augmented belief sets. If all of these sets are consistent, the new cognitive state is built as the subset of these augmented sets which are preferred according to the connectivity, endorsement and conservation criteria. As indicated earlier, these three criteria are applied in order, and the ordering is significant, so they operate as filters: connectivity is more important than endorsement because endorsement is only considered if there are competing best-connected sets, and endorsement is more important than conservatism because conservatism is only considered if there are competing best- endorsed sets. This ordering is intuitively motivated, but also has practical advantages since it means that belief sets are not in general independently constructed and held, only to be discarded much later. Each criterion is applied comparatively to candidate sets, and as for each criterion only the one or equal best candidate sets are retained and there is no requirement to order all the candidates, the procedure for determining the final preferred belief set(s) can be carried through its successive stages in a relatively simple way.

Revision is limited in belief addition to determining preferences within the boundaries of the current state: there is no reference to any other beliefs in the stock held in the system's database. This procedure can be justified on the grounds that if it is possible to accommodate new information within the system's current focus of interest, this is an efficiency saving and is therefore an obviously sensible strategy. Belief modification applies as soon as any of the augmented sets is found to be inconsistent, showing that the information given by a new observation cannot readily be accommodated: as it is evident that a more radical reconsideration of the system's beliefs is needed, the addition process and the limitation to the current cognitive state are abandoned and the work of determining the most rational belief sets restarts from the whole database. However as treating all beliefs in the database on all fours in this process would be too taxing, we start from a more limited subset defined in terms of assumptions as follows.

We take all of the ground assumptions in the database, and enlarge this as a set of asumptions by adding to it all those pervasive beliefs in the last cognitive state. These beliefs have some claim to superior status just because they are pervasive, and also serve to give weight to the information represented by derivations in which they figure. They are adopted as assumptions, with a slightly higher level of endorsement than as hypotheses, called 'pervasive', designed to enforce minimal change (but their pairs are just hypotheses). Using the previous pervasive beliefs in this way can thus be viewed not just as a crude economy device, but as giving a theoretically desirable bias to the prior cognitive state, even though the attempt to preserve this by limiting revision just to adding beliefs has failed. The processor now considers the input observation in relation to the new assumption set, and constructs all the maximal consistent belief sets it can for its current core beliefs from the database, applying its connectivity, endorsement and conservation metrics to select the new preferred set(s) constituting its new cognitive state. (The consistent sets are constructed and compared for connectivity on the fly, so there is in general no need to hold all candidate sets at once.)

All this processing is carried through using an ATMS-type conflict resolution algorithm (DeKleer 1986), which is appropriate to the way we regard foundations for beliefs as important. An ATMS mechanism distinguishes 'assumption-type' propositions, which are not justified by other beliefs and appear in a belief set solely on merit, from 'node-type' propositions, which can only figure in a belief set if they are justified by other beliefs in the set. Our implementation limits the evaluation of assumptions to the subset of beliefs we have defined as assumptions. Node-type propositions are labelled with 'environments', where a node's environment is the minimal set of assumptions required to support the node. The ATMS also maintains a set of 'nogoods', or minimal inconsistent environments: any belief set which is a superset of a nogood is automatically inconsistent too. Inconsistencies take the form of a complementary pair of propositions, where either or both may be assumptions or derived nodes. They are tackled by removing one of the pair plus propositions which immediately support it, and in turn propositions supporting these propositions, back to ground assumptions: deleting these then achieves consistency. The process removes all the derivational chains which support the offending proposition, so if there are alternative justifications for a proposition both are removed; however if a proposition is jointly supported by others, only one of these needs to be removed.

If an assumption also happens to have derivational support, and the assumption is deleted, its support chain must also be removed. It is also necessary, when assumptions are deleted, to work forward from them removing derived propositions which are no longer supported. But we limit this process so that derived propositions are only removed if they are not themselves assumptions (or pervasive beliefs adopted as assumptions): those which are assumptions are retained unless they are independently attacked. Overall therefore, assumptions persist as long as they are not challenged through an inconsistency, and propositions derived from these are only removed if they are not also viable assumptions: thus assumptions may be deprived of support but can continue, just because they are assumptions which have not been challenged.

From a mechanical point of view, responding to observed inconsistencies is essentially a vast checking operations tracking through chains of justificatory links. The end result is a new candidate set for evaluation for connectivity and so forth. The foundational character of the whole is embodied in the fact that as inconsistencies suggest faulty foundation beliefs, these must be identified and rejected, along with all their dependent consequences. The way nodes are labelled with their environments makes it easy to carry out the checking since the assumptions underpinning derived nodes, and thus implicitly further derived nodes along justification chains, can be easily identified. The labelling also makes it easy to compute comparative connectivity when competing sets are evaluated, as well as to consider competing endorsement status. Overall, the way we have implemented the ATMS means that we can experiment with different belief revision theories which assign different weights to different types of information. For example propositions can be given assumption status at run time, which allows us to explore the effects of interpreting the principle of positive undermining in a specific way, say by treating all beliefs as self-justifying assumptions unless they are explicitly subverted.

2.4 An example

The way the whole works can be illustrated with a highly simplified example, presented in more detail in Galliers (in press), and summarised in Figure 3.

This example is about a car repair situation. The agent, J, has a whole mass of beliefs about cars and garages, both general and particular, which form a context, with some ground assumptions and core beliefs, for a particular car repair episode. For the purposes of the example, the initial state, State 0, is taken to be that J's car had collapsed yet again and been taken to J's usual garage, and J is now going to collect it. J has several initial belief sets including groups of alternatives relating to core beliefs about payment, i.e. about whether she will or will not have to pay. For example in one group there is a set to the effect that the garage is respectable and will have mended the fault, so as in this situation there is usually a bill to pay, J will have a bill to pay. In the other group there is a set to the effect that as the garage failed to mend the fault earlier and they are a respectable garage, they will feel guilty, and as in such situations people often do not charge, J will not have a bill to pay. The preferred two alternatives are both in the to pay group: they are of comparable weight as far as connectivity and endorsement are concerned, but differ on whether the fault is intractable or there are multiple faults.

When J gets to the garage, the assistant says the car is ready to go and as far as he knows there is nothing to pay - but he's not certain about this. J's next belief state, State 1, is that there is somewhat better endorsement of the not to pay ground assumption and therefore of the derived core belief, since the ground assumption is not now endorsed only as a default, but rather as a communication, albeit only as a second hand (because linguistically rather than perceptually conveyed) and also rather a weak one. But when connectivity and endorsement for all possible alternative sets is taken into account, while allowing there is more support than before for the not to pay option, this is outweighed by conservation, since the pay option implies less change overall. There are in fact five alternative sets embedding the pay core belief, representing different possibilities in relation to this by considering either multiple faults or one fault which is hard to find.

The need to resolve the uncertainty naturally leads J to seek more information from the garage proprietor, who confirms there is nothing to pay. Thus as far as J's new belief state, State 2, is concerned, the definite way in which the proprietor says that there is nothing to pay now provides sufficiently strong endorsement for the nothing to pay ground assumption and therefore for the core belief that J has nothing to pay that this option is now unequivocally preferred. There are however still alternative belief sets relating to this core belief, to do with whether the garage feels guilty or there is in fact nothing wrong with the car: the subsequent dialogue could naturally stem from J's desire to get more information to choose among these.

This is an extremely selective account of what is in fact much more complex, with a much richer set of contextually and derivationally related beliefs. The example is designed to illustrate the basic principles of the theory, but it must again be emphasised that in reality it is the cumulative effect of many small contributing factors which determines the outcome belief state, with one or more alternative sets of beliefs in play, for any agent at any particular time.

3 The model of the librarian

While the theory of belief revision provides a general base for motivated action including dialogue, it is also necessary, in seeking to automate the intermediary, to consider the task-specific goals and knowledge the intermediary has: what particular characteristics does a librarian have that need to be modelled by the system as the agent interacting with the information-seeking user?

The project is taking work by Belkin, Brooks, and Daniels (Belkin, Seeger and Wersig 1983, Belkin Hennings and Seeger 1984, Brooks, Daniels and Belkin 1985, Brooks 1986, Daniels 1987) - hereafter referred to as BBD - as a starting point. This work was based on real library dialogues, but it must be emphasised that everything has to be ruthlessly simplified for our project. This applies whether the real library situation is one where the literature is to hand and the usual means of access is via a conventional catalogue, or where references to literature are obtained via an online search service. The BBD model is a completely general one, intended indeed to apply to all types of information-seeking situation and not just library or literature search service ones. It is also intended to cover the range of enquiries stretching all the way from quite definite requests for known items to very indefinite, barely formulated needs for unknown items. But as the earlier examples suggested, the typical situation is the topic or subject search for unknown items, of the kind associated with online search services. Other research so far by eg Pollitt (1986), Vickery et al (1987) and Brajnik et al (1990), has also been concerned, in different ways, with this situation. The example of Figure 1 assumed a subject-based search of an online book catalogue, rather than the more common subject search of journal literature, but the generic situation is the same. Searching in these contexts is of course usually iterative: our initial simplification for experimental purposes is to treat the point at which the first actual search formulation is submitted to the online system as a stopping point; but this does not affect the general form of the agent-user interaction, and iteration can be incorporated later, as it is clearly essential for a realistic and effective system. The situation being modelled will be referred to for convenience as the library situation, regardless of whether there is an actual library with literature to hand, and of whether books or papers are in question.

The essential point about the situation being modelled is that the user has a need for information, and knows what the context motivating this need is, but that he cannot by definition fully characterise the information needed because he has not yet read the documents which supply this information. The user does not have technical knowledge of the access routes to the literature either, i.e. of the indexing vocabulary, classification scheme or whatever, or of the library or information service holdings and coverage. The librarian, on the other hand, does not, indeed cannot, know the user's individual need, or the user's personal motivating context. But the librarian does have technical access and holdings knowledge, and typically also has generic subject area knowledge, and user population knowledge. Thus as the earlier dialogue showed, the two parties to the library interaction have mutually complementary starting knowledge, but the process of putting these to work on one another is not just a transfer operation: it is a constructive one, since it is necessary to formulate the user's need sufficiently fully and explicitly for it to serve as a basis for a search specification which is intended to be an effective means, descriptively and selectively, of obtaining relevant literature, given the particular properties of the available document collection or file.

3.1 Belkin, Brooks and Daniels' approach

After considering all these factors, BBD have suggested that an appropriate way of modelling the librarian is as a set of subtask processors, or functional experts, each with their own specific resources and each satisfying their own data-gathering goals, but in doing this collectively contributing the data required to achieve the overall system goals, namely to enable the user to satisfy his information need. In general, this may be done either directly, or indirectly by providing pointers to documents. But in some cases it may prove impossible to help the user: thus the outcome for the system is more correctly characterised as satisfying the goal of doing the best for the user, as mutually agreed. For the simple experimental case being studied by the project, however, this is taken as agreement on a first pass search specification.

The justification for the model BBD propose is that very distinctive bodies of knowledge and processes are required for the various tasks contributing to the overall goal of satisfying users' information needs. Thus librarians deploy quite specific knowledge about indexing languages and techniques, for example, and have particular knowledge about individual document collections, even if they also back up this specialised knowledge with a more general "ordinary" knowledge base. At the same time, forming an effective or adequate search specification calls not only on the topic description itself but on information about the type of user, the type of literature wanted and so forth. Individual processors may also seek data satisfying a variety of subgoals, for example for the user both general educational experience and level of familiarity with the particular area in question.

The complete set of processors BBD propose is quite large. It includes both what may be thought of from the global task point of view as central processors and support processors. The complete set, embodying some compromise between BBD's various publications, and with some renaming for present convenience is shown in Figure 4, along with very simple illustrations of the kinds of state they might be in at about (though not necessarily precisely simultaneously) the end of the dialogue fragment of Figure 1. These illustrations are simply indicative, however, and are not intended to make any claims about the proper way of representing processor results. The central processors are those bearing directly on the user's information need. They include the Problem Description expert, intended to capture the user's topic and its broader conceptual context or subject area, deemed in the example to be conflated as the notion represented by 'cactus cause disease'; the Problem State expert, showing the status of the user's progress with his subject and topic, in this case just starting finding out; the Problem Mode expert, characterising the manner of information gathering taken as appropriate for the user to supply his need, in this case reading (as opposed to, say, talking to someone); the User Model expert, giving the relevant properties of the user, e.g. householder (not horticulturalist); and the Retrieval Strategy expert which produces the means of access to the description or document file, in this case taken as a Boolean request in a controlled indexing language.

The supporting subprocesses cover Dialogue Mode for the form of interaction between the user and the librarian, for instance continuing talking about the user's topic etc as opposed to looking at actual documents; Explanation Provision, concerned with the kind of information the librarian gives the user about what is going on, in this case we may suppose that a rather broad search specification has been formed because the library holds little material on plants; Input Analysis, designed to interpret the user's natural language input, e.g. "No, on diseases they cause"; Response Generation, for planning and organising the form and content of system responses to the user, e.g. checking whether material on non-cacti would be appropriate; and Output Synthesis, for producing natural language output, e.g. "Other house plants ...".

BBD's claim for the range and nature of the knowledge sources contributing to the librarian's task performance as a whole is based on a detailed and careful analysis of human examples, including protocols taken from dialogues between library users and online search service intermediaries. The analysis also shows that the functional processors may be quite complex, with subprocessors with subgoals to be satisfied in support of a processor's overall goals. BBD thus argue that the natural model for the librarian is as a distributed expert system with multiple agents having their own individual tasks, but cooperating by supplying data any other experts may use by posting messages on a common blackboard. From this point of view indeed, the user is just another agent, albeit one mediated by the Input Analysis processor.

The motivation for adopting this data-driven model is that the detailed study of human user-librarian interaction shows how very free and flexible dialogue structure is in terms of how far individual goals are pursued at any point, and in what order, when they are revisited, and so forth, and also in terms of the way any individual item of data is obtained. The dialogues show exchanges delimited by conversational boundary markers and shifts of discourse topic, with each exchange or focus, concentrating on one task or another. Overall the dialogue may show a gradual tendency to move from concern with the User Model, through the Problem Description to the Retrieval Strategy, but there is great variation in the detailed pattern reflecting the way in which the needs of different subtask processors are addressed. At the same time the analysis of the dialogues shows that at some times a piece of data required to satisfy some goal comes directly from the user, at other times may be derived indirectly from data primarily relating to another goal. For example information about the user's expertise relating to the User Model or about the user's Problem State may be supplied by the user, or it may be inferred from the type of literature requested, itself a concern of a Problem Description expert; for example a request for an introductory textbook suggests the user may be a beginning student and/or someone just beginning work in the relevant area. The general presumption is that as the individual processor's data needs are satisfied, whether via responses from the user to system data requests or contingently via other processors, the system's collective needs are also satisfied.

Interestingly, Chen and Dhar (1987) proposed, evidently completely independently of BBD, a similar but rather simpler model for the intelligent assistant, also based on a study of actual interactions between users and librarians. They found that the observed interactions followed a two-phase pattern, with the first establishing 'handles' selecting indexes or databases for the second phase of specific topic searching (though there might be iteration over as well as within phases). Chen and Dhar found user and librarian collaborated even in the first phase, and saw this phase as important for an envisaged (but apparently not actual) implementation of the intelligent assistant, though its relative contribution to delivering the user with suitable goods is not in fact clear.

3.2 Control problems

As each processing agent in BBD's model has its own area of knowledge which it deploys in the context of communications from other agents including the user, it is easy to see that BBD's approach can be couched in terms of belief revision at the level of the individual processors and hence that of the system as a whole. But there are significant difficulties with it which need resolution before any computational implementation, however simple, can be attempted. There are of course questions about the message language used for internal *communication*, and about the way individual processors interact with the blackboard. But the serious issue is overall control, and also control of the external *dialogue* with the user. The way BBD appear to see control operating is essentially responsively, applying 'syntactic' criteria relating, for example, to message or sender status, rather than 'semantic' criteria relating to message content, to determine which messages require responses from the user and when the response should be sought. Thus the notion in Belkin, Seeger and Wersig (1983) seems to be that output is triggered when there is enough pressure from the data state (indicating hypotheses to be tested or information to be sought) on the blackboard.

The problem with this is that it does not provide sufficiently for sensible dialogue control. BBD invoke the Hearsay-II architecture as a model without considering whether their task is sufficiently like the one for which HEARSAY-II was designed. The overall distributed data-driven model is attractive in allowing for the heterogeneity of the resources and processes involved and for the arbitrariness of the data, in terms of both the nature and the timing of items of information. But effective dialogue cannot be conducted simply by picking off the individual most pressing request for data. The interaction between librarian and user required to determine information needs and candidate ways of meeting these cannot be carried out as a series of independent system questions to the user. The system needs to be able to make a more informed evaluation of the state of the blackboard and to have a more controlled organisation of dialogue as a means of data gathering.

This is necessary both for efficiency and for effectiveness, as rational dialogue chunking is essential not just for comprehensible interaction with the user, but because it reflects a motivated consideration of what information needs to be got from the user which can only be based on a review of the various current blackboard messages, their relations and, perhaps, implications. Thus the controller itself has to take account not only of the fact that information is sought by processors P, Q, ... etc: it has to be able to study what information is needed, in order to decide whether and how the user should be approached. This implies a much more substantial capability in the overall controller, and in the dialogue conductor embedded in or dependent on it, than appears to envisaged by BBD's combination of a reactive syntactically-driven global controller and the specific Response Generation processor. However if there has to a powerful controller with judgemental and planning capabilities as a manager of the dialogue between other processors and the user, what happens to the original aggressively distributed model?

Thus if the model is redesigned for a dominating controller with subordinated subprocessors, it is not clear how far these can operate autonomously in parallel and in a data-driven way. But even if they can, it is not obvious how control and dialogue management as a whole are to be achieved, given three critical features of the task situation being modelled. These are first, the weakness of the notion of satisfaction for subprocessors, especially key processors like the Problem Description one. Data gathering cannot be driven, as it can for many other tasks, by a check-list approach, certainly not at the level of offering a range of specific choices, but even of generic ones. With a menu system the relevant variables (slots) would be given, and perhaps even the potential values (possible slot fillers). Limited implementations of the automated intermediary like Pollitt's are able to operate effectively with known slots and filler possibilities, and this may be feasible for e.g. simple versions of the User Model. But it is not possible in general, for example, to capture topic information by a menu approach because the range of possibilities is too large, unless the menu is more notional than real, with generic slots like 'Concept1', 'Concept2', and so forth, and the notion of satisfaction applied is minimal, e.g. three concepts is by definition enough. BBD's presumption is that obtaining a proper or adequate topic description is a serious matter, and this implies a sophisticated approach to determining whether a given topic characterisation is adequate, which can only be based on a number of criteria which are individually weak. (This is setting aside the fact that it is hard to establish what the set of criteria is or how they work together, and also the fact that the criteria may be very hard to apply.) It is also difficult to get mileage out of a notion of obligatory data. For example it is not necessary to have any individual information in the User Model at all (and the default user characterisation may be very simple indeed).

The weakness of the satisfaction criteria applies everywhere, but is especially awkward as far as the crucial Problem Description processor is concerned: what is the right, or a good, problem description? It is clearly naive to suppose that effective dialogue can be conducted simply by the system applying a 'tell me more' strategy, but when satisfaction is weak it is difficult to determine what a system's output should be. It will certainly require the informed self-evaluation capability mentioned earlier. The satisfaction problem of course also applies at the level of the system as a whole: what, in the likely absence of clear indications from individual processors, determines whether the entire 'information need problem' has been satisfied? It is not evident that relying on the user to declare this, especially without constructive system suggestion, is efficient or effective.

The second major problem to be resolved for control and dialogue management is the open data sourcing, that is the fact that useful or desired pieces of information can come from other processors or from the user. For example, the Retrieval Strategy processor may be able to obtain data for a search specification from the Problem Description or User Model or Problem State modules, or from the user via the Input Analysis module. This makes it difficult to determine whether an attempt to obtain information should be forced by embarking on dialogue with the user or should be awaited from any source (including volunteering by the user).

The third problem is the separateness of the user. At the fine grain information level, there is no predictability in the user, however cooperative the user may be both in relation to the task as a whole and in relation to the local dialogue context. This is not so much because individual user responses to system questions or statements may not fit tightly, but because the user is a genuinely independent agent (in the way the other processors are not) who may choose to take his own initiative in the way the dialogue is conducted. This implies a need for great flexibility in the system's controller, and in turn, as for the previous problem, that it has a far from trivial capacity to continually re-evaluate its data state and action possibilities.

Quite apart from the possible need for relatively powerful global control in the interests of dialogue management, it is not obvious that there is no need for control of the system's internal communications in general, i.e. for more comprehensive blackboard management than that required for dialogue purposes. Is it reasonable to assume that effective overall system behaviour will emerge from the aggregated operations of the individual agents able to control only their own activities according to their own criteria, whatever and however many messages there are on the board?

Thus with BBD's model of a distributed system for their characteristic task type, the issues are whether internal communication can only be in the open, blackboard style; how much control is needed to regulate internal activity and to manage external dialogue; and how these two control processes are related if, as is possible given their rather distinct functions, this involves two distinct system components, a global system controller and a specific dialogue manager.

3.3 Architecture refinement

Belkin, Hennings and Seeger (1984) (BHS) began to address some of the questions just raised in simulation experiments designed to study different architectures for the automated intermediary. In these they compared blackboard and actor versions of the distributed model, i.e. architectures where internal communication is via a blackboard with architectures where internal communication is direct between an agent and other specified agents, and they compared uncontrolled and controlled communication regimes, i.e. regimes with no and with some monitoring, prioritising etc of message flows. BHS concluded that their experiments showed that a blackboard architecture is appropriate, and specifically that it is superior to an actor one. But they also concluded that it needs a positive control regime: a simplistic free-for-all model is too weak.

BHS divided their blackboard into areas, one for each expert: each expert had a list of other experts whose boards could be read, i.e. whose messages were acceptable, but not a list of other experts who could read its own messages. In the uncontrolled regime for the blackboard messages were freely posted and collected, and interaction with the user was simply via the Response Generation expert's reaction to individual board messages. In the controlled regime there was a Blackboard Analyst (BA) whose main role was to filter messages relevant to the user for the Response Generation expert, applying its knowledge of the state of the system and capacities of the individual experts, and naturally also relying, given the lack of explicit addressee labels, on an ability to understand and evaluate messages, to do this.

Unfortunately, though the experiments were quite carefully conducted, the fact that human agents were involved meant that the simulations were not specified at the level of detail required for machine implementation, and crucial questions about the powers of the BA and the relationship between BA and Response Generation were therefore finessed: as BHS note, the experts' judgements and behaviour were 'improperly' well informed. BHS nevertheless found that there were problems with the blackboard architecture, even when controlled, stemming from the need to identify message versions, to cope with poor quality messages, and to allow for both formal and substantive feedback. They also note that the satisfaction criterion, namely the user's calling a halt, was too simple.

BHS's conclusions about the relative merits of the different architectures are open to the criticism that there was not enough rigour in the comparison. But their detailed analyses bring out, as BBD's of human dialogues did, the complex dependencies among the experts' activities: any one action done by an expert might be stimulated by inputs from several others, and might in turn stimulate actions by several others. There was also, as with the human dialogues, a gross flow of activity through the set of experts over a whole session, but there was still a great deal of varied interaction between experts, and individual experts could remain active throughout a session.

As noted, there are many problems with these simulation experiments in the lack of detail about the capabilities of the BA, though as BHS observe, to do its monitoring and decision-taking job properly it clearly needs a message interpretation ability and extensive knowledge of the system's resources; and there are problems about the relationship between the BA and the Response Generation expert: this affects both decisions about which of the messages that Response Generation could in principle consider should actually be passed to it, and about the detailed organisation of the dialogue with the user. For example, is there meant to be some strategic/tactical division of responsibility for dialogue management? BHS found that while messages were originally intended only to convey hypotheses, more varied types, including requests for information, emerged in the simulations, and this clearly bears on the conduct of dialogue with the user.

3.4 The CODER System

Fox and France (1987)'s CODER system design is an explicitly computational attempt to tackle the problems of blackboard architectures for information systems. CODER is a multi-function information system shell, intended primarily to support a wide range of experiments. As it is multi-function, e.g. is for indexing as well as retrieval, it allows for different clusters of experts, each with their own blackboard, for the major task areas. These can communicate and share resources; however for present purposes it is the structure of any one of the clusters which is relevant. Thus for, say, the retrieval task area, Fox and France allow for a set of distinct experts like BBD's, though they see individual experts as typically quite limited in scope, implying either more at one level or a hierarchical decomposition. The examples they give, e.g. morphology expert, clustering expert, are more definite and limited in their system function than BBD's. The experts communicate with the blackboard via operations like 'post', 'view', and 'retract', and their specification includes that of the message content predicates they can read/write. The experts have their own internal knowledge sources but can also call on shared external sources. Following established practice, the group blackboard is divided into sub-boards, one for posting questions and answers, one for the set of consistent hypotheses forming the best overall group task hypothesis, and others for specific subject areas. All the experts have access to the first two, but to others only as appropriate for the the individual expert needs.

But the important point about the CODER design in the present context is that the group board has a powerful controller, namely a strategist/planner, with a whole range of directive functions of the kind mentioned earlier as required, and implying a message interpretation capability. The strategist keeps models of the experts and monitors and schedules their activity, and maintains blackboard consistency and selects best hypotheses. It subsumes both a generic TMS component to maintain consistency and an application-specific rule set relating to task conditions and events, as well as a mechanism for identifying answer specialists for questions and a dispatcher for allocating pending jobs to experts, using commands like 'attempt hyp', 'attend to area'. The CODER strategist is thus much more powerful than the controller of BHS's simulation, and in fact has the capabilities needed to deal with the control issues BHS identified in evaluating their simulation results. Interaction with the user is, however, seen as the responsibility of a separate user interface manager, linked to the specialists but not the scheduler, which seems to suggest a much more limited view of interaction with the user than BBD's, and one which is more in accord with current operational system designs.

The CODER state described in Fox et al (1988) suggests that while the principle of the distributed expert architecture has been retained, the implementation has been simplified in key respects. Thus the user interface is system driven and menu based, and problem mode, state and description have been combined as a single expert which has become the dominant module since its rule base determines most of the system state changes. As processing includes actual searching there is a major feedback loop here, as also through lexical browsing, but otherwise there is a strongly linear flow with, apart from the problem expert's major contribution, a significant role for user modelling at the beginning and search formulation and execution at the end. Other modules, like input analysis and explanation, play a part throughout. The strategist, on the other hand, appears now to have an essentially middle management role, keeping things running.

Subsequent accounts of CODER (Fox et al 1988, Fox et al 1991), while they show that considerable effort has been put into other aspects of the system, do not provide any fuller detail about the architecture or about its conditions and performance in actual use. However it is evident that, as most of the system's capabilities have naturally been initially based on current technologies for query construction and searching, much of what is supplied is simpler than BBD's desiderata and more in line with Vickery et al's (1987) system. Thus the fact that CODER has a report generation, rather than response generation, expert seems to signal its actual level of sophistication. But it is in consequence difficult to see CODER as a real demonstration of BBD's distributed architecture.

Croft and Thompson (1987)'s prototype implementation of an intelligent intermediary in their I3R system has much in common with Fox's. The I3R system is a data-driven blackboard one with a powerful scheduler operating on strongly preplanned lines. The various experts, User Model Builder, Request Model Builder, Domain Knowledge Expert, Browsing Expert, Search Controller, and Explainer collaborate to build a user model and a request model, communicating with the user via an Interface Manager. The Scheduler implements its default or alternative exception plans for satisfying the user as the conditions for its various experts' rules are satisfied and transitions can be made from one agent's activities to another (of course allowing for iterations).

Much of the retrieval interest of the system is in the sophisticated use of statistically-motivated information and of terminological inference, and also in the types of display and details of the user interface. From the architecture point of view, in the context of our project concerns, I3R is relatively straightforward: the restricted view of the form and manner of need and search specification, as embodied in the scheduler's plans for deploying the system's contributing experts and in the firmly system-driven interaction with the user, makes for well-organised control. Thus requests for information from the user are always explicit and are systematically preferred, and his answers are constrained enough to be of direct utility.

The architecture of both of these systems is thus less distributed in practice than in principle, and is much like that used in Brajnik et al (1990)'s IR-NLI II. This prototype intelligent intermediary essentially combines a sophisticated version of Vickery et al (1987) as a rule-based expert for handling search formulation and reformulation with an ambitious user modelling component. The retrieval subsystem exploits knowledge about search strategies and tactics of a professionally established kind with domain terminological knowledge, and is designed to develop an adequate characterisation of the user's need and appropriate search specification: this may involve iteration using retrieved output. The user modelling component, starting from stereotypes dealing with user experience, background and retrieval history etc, constructs and maintains a current model. Both subsystems may thus involve inference. However IR-NLI's operation is essentially system driven through a well-defined, possibly iterative, sequence of steps from initial request capture to final search specification, with communication with the user modulated by the user-modelling component (and not yet in free natural language). The system design makes control relatively straightforward, and the prototype implementation gains by being able to rely heavily on a quite restricted application domain and user community.

The proposal for a distributed model of the librarian thus has to be evaluated at three levels: whether distributed processing is in itself right; whether processors should interact in blackboard or in actor style; and whether interaction should be essentially uncontrolled or controlled. For all three, there are no independent arguments one way or the oither: the particular choice of system organisation depends on the characteristic properties of the librarian's task.

The discussion in this section might perhaps suggest that a distributed architecture is not the one for an automated librarian. But there are merits, for a complex, technically based and skilled task, in the notion of distinct special-purpose knowledge sources. It is thus fair to start from BBD's basic position that a distributed architecture is appropriate for the information-seeking case. However it is not merely desirable in principle, but necessary in practice for computational implementation, to tackle the question of the nature of inter-agent communication and of control, both overall and for dialogue management. There is the further matter, from this point of view, of seeing how Galliers' theory of belief revision works out within the distributed framework. One of the two major objects of our project is to develop, and validate (if simply) by computational implementation, this theory. It fits very well with the notion of distributed processing, as it is a theory of interacting agents. But it is necessary to show that it can be implemented so that both each agent's individual behaviour, *and* the agents' collective behaviour which defines the librarian, are sensible.

4 Project strategy and progress

There are clearly many tricky problems to investigate for our project, associated with the system architecture and the management of dialogue with the user, and with the performance of the belief revision mechanism as the modus vivendi of any system component and hence of the system as a whole. In the longer term, of course, there is the nature of the task and domain knowledge needed for effective functional experts.

4.1 Initial system design and implementation

The approach we have adopted to begin the project is to build a very simple version of the librarian, meeting only the most basic requirements for conducting a simple interaction with a simple user. This is primarily to obtain a working implementation of the essential belief revision and communication apparatus instantiated for the library case, using BBD's functional model with distinct experts. We are currently testing and evaluating alternative architectures and control regimes within the context of this simple initial system.

The initial Mark I system (see Figure 5) has only a few of the functions of the full BBD model, namely the key Problem Description and Retrieval Strategy ones, the User Model function, and an 'Interactor' function, which is concerned with managing the interaction with the user. This latter function is not strictly one of those enumerated by BBD, but conflates some of the simpler functions of the Input Analysis, Response Generation and Output Synthesis experts. (The appropriate relations between these are not obvious, and BBD's treatment is not convincing.) As the Interactor is the exclusive channel of communication with the user, the user may be deemed from the system's point of view to be the Interactor, in a more general and abstract model the actual user is yet another agent within a larger 'system' embracing the actual computational one.

In our first version system the knowledge contained within each expert consists of a small number of inference rules and some simple data structures, in fact motivated by an actual dialogue about literature on Greek-Turkish relations BBD recorded and analysed (Brooks 1986 pp 284 ff). Thus in Retrieval Strategy, for example, we have data structures representing the attributes of the different literature databases, and inference rules linking desired document attributes with the most appropriate database (much like Chen and Dhar's handles). The document attributes include subject area, for instance history, to which a whole database might be devoted; document type, for instance journal; and document restriction, for instance date specifiability meaning, as would be natural for historical materials, that there is a specific field in each document description indicating the time period to which the document refers (e.g. '19th Century'). The inference rules may then be :

```
'if subject-area = history ===> DATABASE 6',
'if document-restriction = date specified ===> DATABASE 6',
'if document-type = journal ===> DATABASE 5 or 6'.
```

Illustrative concepts for the other experts are shown in Figure 5. Thus Problem Description has 'user-topic = greece', for which an inference rule would derive 'subject-area = geography'. Inference rules can applied either forwards, to obtain further conclusions based on new data, or backwards, to try and satisfy some goal (such as determining whether a particular hypothesised database is appropriate). The form of the knowledge and the inference strategies are the same in each expert module. This is an acknowledged simplification as BBD have suggested qualitatively different ways of representing the knowledge and reasoning appropriate to the different modules.

In order to test BBDs proposal for a distributed architecture we have implemented the different expert modules as parallel distributed processes operating on separate machines. We are currently investigating two contrasting types of communication structure and associated control regime within the distributed architecture, each in its most basic or 'naked' form, as well as different hybrids. Thus we are comparing basic *blackboard* and *actor* architectures, and some combinations of these.

In the blackboard architecture, as originally proposed by BBD, all the modules communicate via a common data structure, the blackboard, which holds messages reflecting different aspects of the current global problem state. Message are sender-labelled, but not receiver-labelled, and agents have no prior knowledge of the capabilities of other agents, though they make inferences about them from their messages. We assume that any agent can in principle use any message, and also that any agent can in principle supply any message. The results delivered by any agent's internal operations are in principle communicable, though an agent's evaluation of its task state may mean they are not or not immediately communicated (see below). The messages communicated are of different types, requesting or delivering information, either voluntarily or in response to requests; when hypotheses are communicated which conflict with other blackboard hypotheses, they may be accompanied by their reasons. Figure 5a illustrates a hypothesis state for the model as it might hold at some point in a dialogue about Greek-Turkish relations.

From the control point of view, the blackboard model is implemented in the simplest possible way. There is no global controller, evaluating the system's state and manipulating the contents of the blackboard (as there is in Fox's CODER). The blackboard simply holds all posted messages which are read by all agents, and processing terminates, i.e. the system's overall goal is deemed satisfied, when all the agents (and hence also the user) have no pending tasks. As messages posted to the blackboard are read immediately by all the agents, when all the agents have no messages or other tasks requiring attention, this implies there are none outstanding on the board either. As a termination condition this is clearly very simple, in fact too simple, but is adequate as a beginning. The whole operation of the system is essentially data driven, though each agent has of course its own internal control mechanism.

In the actor architecture, expert modules communicate directly with each other, each module having knowledge of the sorts of knowledge that will be relevant to particular other modules. For example, the User Model module might know that the Retrieval Strategy module is interested in information about the user's status (e.g., student, lecturer). When the User Model comes to new conclusions about the user's status it may communicate this information directly to the Retrieval Strategy. If, say, this information conflicts with Retrieval Strategy's existing beliefs it can negotiate directly with User Model. Subject to the constraints of its specific knowledge of the other communicating agents, each expert operates in the same general way as in the blackboard model, and in fact in our initial version each expert may communicate specifically with any other expert, as needed. Control for the actor model again has the simplest form, in this case meaning that there is no attempt to restrict the flow of communication between modules: each agent has to decide itself what to do with whatever arrives. The system's processing is again terminated when all the agents' have no outstanding tasks. Figure 5b illustrates the actor model for the same dialogue situation as is shown for the blackboard model in Figure 5a.

In discussing the behaviour of an individual module in more detail, we will assume agents are communicating with one another directly, as in the actor model. However the points made generally apply to the blackboard model as well.

Communication between agents is based on a simple common formal language. We are currently not addressing natural language per se, so we communicate with the user 'behind' the Interactor exactly as with any other agent, in the formal language. A *message* consists of a type of speech action, the sender name and receiver name, a proposition or package of propositions, and optionally a 'strength', which can be weak or strong. For example, the message 'tellref(UM, RS, user-status(student), strong)' informs the Retrieval Strategy module that the User Model module strongly believes the user is a student, while the message 'askwhy(RS, UM, user-status(student))' is a request from the Retrieval Strategy module for further justification for this belief. The speech acts thus embody the simple message types we allow, currently two forms of information request, 'askref' for data and 'askwhy' for explanation, and two forms of information delivery, 'tellref' for hypotheses and 'tellwhy' for explanations ('tellref' is not necessarily an answer to a request). It must be emphasised that strength here refers to the status of a belief for its communicator (i.e. utterer), not for its receiver. It is a simple, ad hoc version of a belief's status as this is associated, for the communicator, with the difficulty of disbelieving it: a strong belief is harder for the communicator to disbelieve, because of its revision consequences for him, than a weak one. Of course strength for the communicator does not necessarily imply strong endorsement, as defined earlier, by the receiver.

Communication between the modules, and thus also between the system and the user, is both motivated by and causes belief revision, according to the theory developed by Galliers as outlined in Section 1. As we saw there, this theory of belief revision determines how a new communication influences the hearer's beliefs. The communicated proposition may or may not be taken on by the hearer, and if taken on may influence a whole web of related beliefs, whether the new communication is directly sent as in the actor model, or read off the blackboard as soon as posted. Depending on whether the proposition is taken on or not, and on how other beliefs are changed, different messages may be sent or posted in response. These are constrained by a set of Interaction Rules (IR) which, in conjunction with the belief revision process, dictate how an agent may respond to particular communications and communicate belief changes following the operations of belief revision. For example, if an agent is informed of a proposition P but does not take on that belief, then depending on a number of factors such as past communications it has received and its view of the communicating agent's beliefs the receiver may respond, directly or indirectly via the blackboard, by trying to convince the sender of not-P or by asking the sender why they believe P.

When one agent in the system is attempting to convince another, whether an individual agent in the actor case or presumed other(s) in the blackboard case, of the truth of some proposition, the agent uses simple models of the other to strategically plan a message which it believes will cause the desired belief change. Of course these models of other agents may be inaccurate, so the desired change may not occur. Because of this there may be a complex negotiation between agents as they argue for or against the truth of the proposition in question.

This general picture is complicated by the fact that any agent in the system may have several things to work on at once. They may try to determine the truth of some proposition, and may be able to draw new conclusions from new beliefs, to respond directly to incoming messages or to communicate new beliefs to interested agents. An agent in the system has to prioritize these tasks, for which it uses a set of Task Prioritization Rules (TPR). For example, the initial system deals with new messages before old, and puts at low priority drawing new conclusions from weakly held evidence.

As far as the individual agent is concerned therefore, it has three components, the IR component, the TPR component, and the BR component, embodying the belief revision mechanism (BRM) and sets of beliefs constituting its belief state (BS). In general terms, though the precise relationships between the components have still to be determined, the TPR component manages the task agenda for tasks derived either from incoming messages or from changes in the BS, while the IR component manages the precise form of input-output message linking and output message expression. In the current simple model, information requests just access the BS, while information deliveries stimulate the BRM.

4.2 An illustration

The way our simple models work can be illustrated by what happens when we emulate the fragment of the Greek-Turkish relations dialogue mentioned earlier which is shown in Figure 6. In this example the user wants to get hold of documents on Greek- Turkish relations, post-1974. The intermediary suggests the history database, presumably because it is the only one that allows the user to ask for documents with post-1974 content. However, the user rejects this suggestion, presumably believing that the history database would not have recent material, and not understanding the need for choosing a database which allows documents to be selected based on an explicit date restriction. We show below how we can model the crucial features of this dialogue using our belief revision apparatus. The example is simplified for the purposes of illustration, and we assume an actor architecture. It should be emphasised that in this discussion, as we are imagining we are communicating with the real user, we need to separate the Interactor from the user and to refer explicitly to ’User’ as a communicating agent in messages.

We represent the system’s initial beliefs, and its initial beliefs about the user’s beliefs as follows, where beliefs are either assumptions which are endorsed using the endorsement types of Figure 2 or are derived from the indicated assumptions:

```
System’s beliefs:
      1. doc-content(post-1974) ===>doc-restriction(date) : 2cs.
   2. doc-restriction(date) ===>database(history) : default.
   3. doc-content(post-1974) : 2cs.
   4. doc-restriction(date) : [from 1,3].
   5. database(history) : [from 1,2,3].
System’s beliefs about user’s beliefs
   6. believes(User,doc-content(post-1974)) : 2cs.
```

Belief 3 is in the Problem Description module, and the other system beliefs, 1,2,4 and 5, are in the Retrieval Stategy module. The system’s beliefs about the user’s beliefs, here belief 6, are held in the Interactor module, as they are needed in order to 1communicate effectively with the user. Belief 3 is also in Retrieval Strategy, to which it has been communicated.

Initially the system, and specifically Retrieval Strategy, has the task of getting the user to believe that the history database may be appropriate. Using the ICM mechanism Retrieval Strategy predicts (based on its model of the user’s beliefs, transmitted from the Interactor) that simply informing the user (via Interactor) of this should cause the desired belief change. This message is communicated weakly as it is not strongly held by the system, via the Interactor:

```
RS: tellref(RS, I, database(history), weak)
I: tellref(I, User, database(history), weak)
```

The user responds by strongly rejecting the suggestion:

```
User: tellref(User, I, not database(history), strong)
```

The Interactor now updates its model of the user's beliefs with the information that the user does not believe that the history database is appropriate. This belief is then passed on to Retrieval Strategy with a strong commitment. However Retrieval Strategy modifies this endorsement to a weak one because it does not believe the Interactor is knowledgeable about database selection. Retrieval Strategy also recognises that the new communicated user belief conflicts with its existing belief (that the history database is appropriate), and considers revising its beliefs. However, the result of the belief revision process is that the Retrieval Strategy module holds on to its existing beliefs. But as it recognises that this conflicts with the beliefs of the Interactor (and hence the user), it responds by trying to convince the Interactor that the history database is appropriate. It does this by providing support in terms of the following information:

```
RS: tellwhy(RS, I, (database(history) because
       [ doc-content(post-1974) ===>doc-restriction(date) strong &
doc-restriction(date) ===> database(history) weak &
doc-content(post-1974) strong ] )
```

(ie, "The reason for choosing the history database is that wanting post-1974 material is a kind of date restriction, and the history database is suitable if there is a date restriction") The Interactor takes on this justification and passes it to the user, who then decides (for whatever reasons he has) to weakly accept the system's suggestion:

```
User: tellref(User, I, database(history), weak)
```

4.3 Evaluation of the simple models

The example dialogue illustrates the basic functioning of our initial distributed system in one of its modes. But much more work is needed before we can assess the overall approach. The initial system works only for a small range of restricted problems, and is also unacceptably slow. The models we have implemented are very simple indeed, though they are still a helpful base for investigating crucial aspects of our approach to information communication and belief revision in a distributed system. Currently control is entirely in the hands of the agents, who each manage their own operations. But an agent may find it difficult to determine whether the input it receives from other agents, especially in response to requests, is 'adequate' and is all it is going to get. With more agents with more knowledge engaged in a more serious literature-seeking interaction, more work will be required and more messages can be expected, suggesting a need for more global control on the operations of the system as a whole. Conducting rational and comprehensible dialogue with the user also suggests a need for more control on the interaction between system and user. Currently the system's behaviour is unfocussed, with messages poorly related to the overall task or to the prior pattern of communication, whether internal or external. The system currently has no serious notion of satisfaction either for any individual agents or for the system as a whole. These are important issues that have to be addressed, and there are also embedded ones to do with specific aspects of our general approach that need tackling. One of the most pressing of these is developing a proper view of how goals are related to beliefs within the framework of the theory of belief revision: in the present simple models the way goals as represented by tasks are established is somewhat arbitrarily determined by the details of messages and belief revisions. A more coherent and motivated account of the way goals emerge from or are associated with beliefs is required. Finally, it is clearly necessary to give the experts more knowledge to deploy, and to increase the number of agents, for instance by adding a Problem State module.

5 Conclusion

As described, the motivation for our project is to combine a general theory of agent behaviour based on belief revision with a specific theory about the librarian as a collective agent. The two fit naturally together, and our aim is to use each to throw light on the validity of the other, and to take the first steps towards towards automating a more powerful intelligent librarian or information intermediary than the other techniques so far tested allow. The architecture and control investigations and measurements with which we are currently engaged should give us the more solid base we need to construct a next version of the model implementation with a little less trivial, if not very extensive, relevant task knowledge.

References

Belkin, N.J., Hennings, R.D. and Seeger, T. 'Simulation of a distributed expert-based information provision mechanism', *Information Technology* 3, 1984, 122-141.

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

Brajnik, G., Guida, G. and Tasso, C. 'User modelling in expert man-machine interfaces: a case study in intelligent information retrieval', *IEEE Transactions on Systems, Man, and Cybernetics* 20, 1990, 166-185.

Brooks, H.M. *An intelligent interface for document retrieval systems: developing the problem description and retrieval strategy components*, PhD Thesis, City University, London, 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 *Informatics 8: advances in intelligent retrieval*, London: Aslib, 1985.

Chen, H. and Dhar, V. "Reducing indeterminism in consultation: a cognitive model of user/librarian interaction", *AAAI-87, Proceedings of the Sixth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, 1987, 285-289.

Cohen, P.R. *Heuristic reasoning about uncertainty*, Boston: Pitman, 1985.

Croft, W.B. and Thompson, R.H. 'I3R: a new approach to the design of document retrieval systems', *Journal of the American Society for Information Science* 38, 1987, 389-404.

Daniels, P.J. *Developing the user modelling function of an intelligent interface for document retrieval systems*, PhD Thesis, City University, London, 1987.

De Kleer, J. 'An assumption-based truth maintenance system', *Artificial Intelligence* 28, 1986, 127-162.

Doyle, J. 'Rational belief revision' in *Belief revision* (Ed Gardenfors), Cambridge: Cambridge University Press (in press).

Fox, E.A. 'The development of the CODER system: a testbed for artificial intelligence methods in information retrieval', *Information Processing and Management* 23, 1987, 341-366.

Fox, E.A. and France, R.K. 'Architecture of an expert system for composite document analysis, representation, and retrieval', *International Journal of Approximate Reasoning* 1, 1987, 151-175.

Fox, E.A., Weaver, M.T., Chen, Q.-F. and France, R.K. 'Implementing a distributed expert-based information retrieval system', *Proceedings of RIAO 88 Conference on User-Oriented, Content-Based Text and Image Handling*, (MIT, Cambridge MA), 1988, 708-726.

Fox, E.A., Koushik, M.P., Chen, Q.F. and France, R.K. "Integrated access to a large medical literature database", TR 91-15, Department of Computer Science, Virginia Polytechnic, 1991.

Galliers, J.R. *A theoretical framework for computer models of cooperative dialogue, acknowledging multi-agent conflict*, PhD Thesis, Open University; Technical Report 172, Computer Laboratory, University of Cambridge, 1989.

Galliers, J. R. 'Cooperative interaction as strategic belief revision', in *Cooperating knowledge based systems* 1990 (Ed Deen), Berlin: Springer, 1991.

Galliers, J.R. 'Autonomous belief revision and communication', in *Belief revision* (Ed Gardenfors), Cambridge: Cambridge University Press, in press.

Gardenfors, P. *Knowledge in flux: modelling the dynamics of epistemic states*, Cambridge MA: MIT Press, 1988.

Harman, G. *Change in view: principles in reasoning*, Cambridge MA: MIT Press, 1986.

McAlpine, G. and Ingwersen, P. 'Integrated information retrieval in a knowledge worker system', *Proceedings of the Twelfth Annual International ACMSIGIR Conference on Research and Development in Information Retrieval*, Association for Computing Machinery, 1989, 48-57.

Pollitt, A.S. '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.

Vickery, A., Brooks, H., Robinson, B. and Vickery, B. 'A reference and referral system using expert system techniques', *Journal of Documentation* 43, 1987, 1-23.

Walker, S. 'Improving subject access painlessly: recent work on the Okapi online catalogue projects', in *Document retrieval systems* (Ed Willett), London: Taylor Graham, 1988.

U: I want a book on cacti.

L: On growing them?

U: No, on the diseases they cause.

L: Other house plants as well?

U: Maybe.

......

====> HOUSE PLANT ^ HUMAN DISEASE

Figure 1 : Fragment of dialogue between a user and a librarian and outcome search specification

endorsement types :

communication :	1cs = first hand communication, strong
	1cw = first hand communication, weak
	2cs = second hand communication, strong
	2cw = second hand communication, weak
	1c refers to perception
	2c refers to language
kind :	sp = specific
	df = default (generic)
value :	vs = value, strong
	vw = value, weak
hypothesis :	h

ordering on types :

1cs > 2cs = sp > 1cw > 2cw = df > vs > vw > h

Figure 2 : Types of assumption endorsement and their heuristic ordering

```
state 0 : car bust again, at garage, being fetched :


several belief sets :
A) group for 'J to pay' core belief e.g.
   ....                 )
   garage respectable   )
   fault mended         ) ---> J to pay (C)
   bill to pay (df)     )
   ....                 )

B) group for 'J not to pay' core belief e.g.
   ....                 )
   failed mend earlier  )
   garage feel guilty   ) ---> J not to pay (C)
   not bill to pay (h)  )
   ....                 )

2 preferred sets, both A group, equally plausible, differing in
  connectivity, endorsement ;
  content distinction : multiple faults / fault hard to find


J           :  "How's my car?"
assistant   :  "Its OK, you can take it away. I don't think there's
                 anything to pay."


state 1 :

revising gives for B group e.g.
   ....                   )
   not bill to pay (2cw)  ) ---> J not pay (C)
   ....                   )

BUT conservation prefers revisions in A group e.g.
   ....                   )
   bill to pay (df)       ) ---> J to pay (C)
   ....                   )

5 alternative sets revising A equally plausible ;
  content distinction : multiple faults / fault hard to find
```

```
J          :  "He says there's isn't anything to pay."
proprietor :  "No, there's nothing to pay."
```

state 2 :

```
revising gives for B group e.g.
   ....                  )
   not bill to pay (2cs)  ) ---> J not pay (C)
   ....                  )
```

revision in B group with stronger assumption now preferable to revision in A
(3 alternative sets for B :
content distinction : garage feel guilty / nothing wrong with car)

Figure 3 : Belief revised by car repair example illustration

C = core belief

a) central processors :

Problem Description
cactus cause disease, ...

Problem State
starting finding out, ...

Problem Mode
reading, ...

User Model
householder, ...

Retrieval Strategy
CACTUS v SUCCULENT

b) support processors :

Dialogue Mode
talking

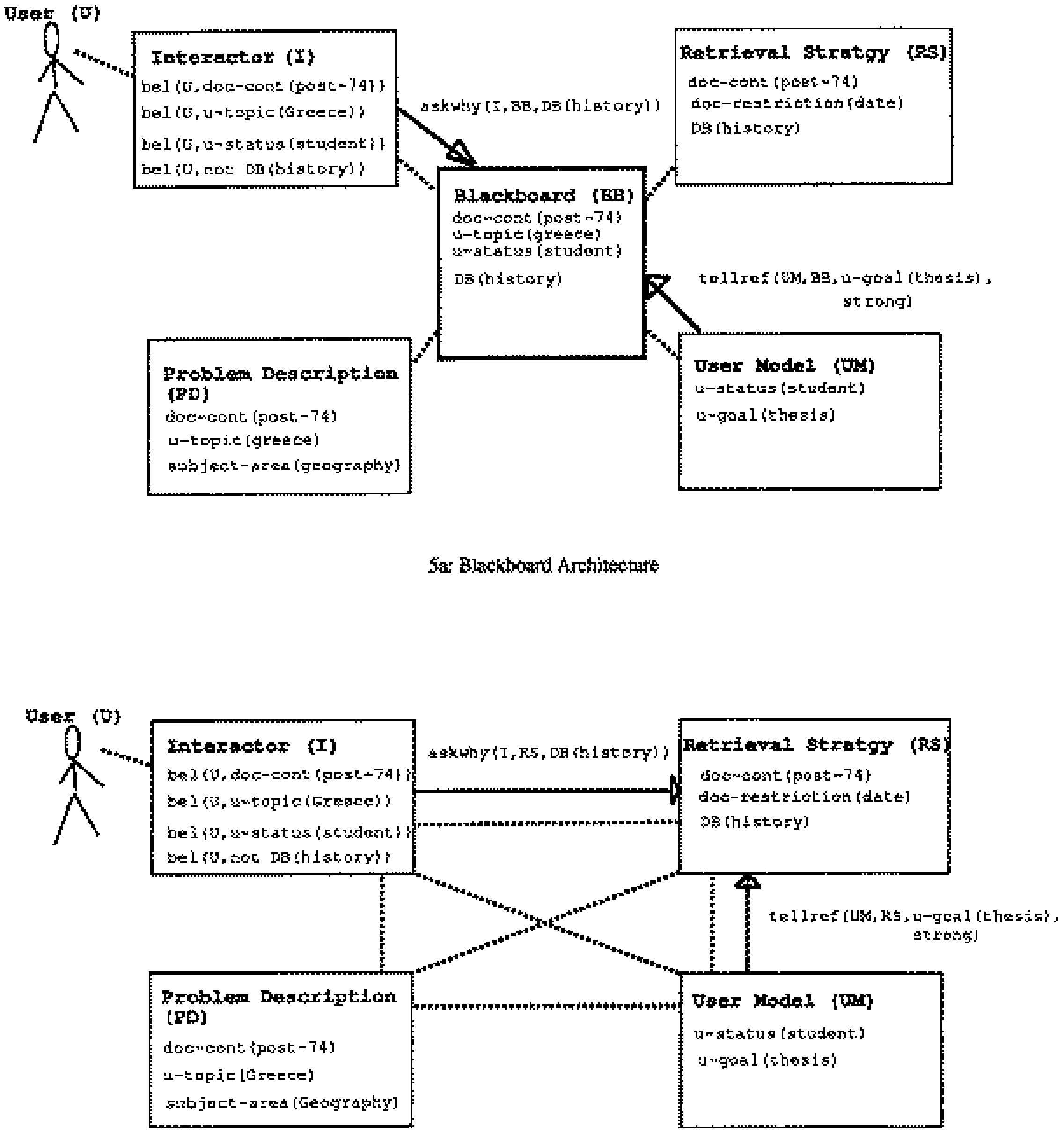
Explanation Provision
little on plants

Input Analysis
"No, on diseases ..."

Response Generation
non-cacti?

Output Synthesis
"Other house plants ..."

Figure 4 : Librarian model processors with illustrative information states for the dialogue example of Figure 1



5a: Blackboard Architecture

5b: Actor Architecture

Figure 5: Illustrating blackboard and actor architectures for the Mark 1 system, working on 'Greek–Turkish relations' dialogue

.....

Intermediary: Um, the only other possibility is Historical Abstracts but it it

User: No.

Intermediary: it is fairly, they CAN include some recent material...

User: Well. Maybe. Maybe OK maybe.

Intermediary: We'll think about it we'll see we'll put a query by that one. Mm.

User: OK OK alright.

Intermediary: It's the only database which has really, obviously because it deals with history tried to, cope with this time limitation.

.....

Figure 6 : Fragment of a recorded dialogue between a librarian
(an intermediary for searching using an online service)
and a user seeking literature on Greek-Turkish relations
(from Brooks 1986, p 293)