Modelling Information Retrieval Agents with Belief Revision

Brian Logan, Steven Reece* and Karen Sparck Jones

Computer Laboratory, University of Cambridge
New Museums Site, Pembroke Street, Cambridge CB2 3QG, UK
(sparckjones@cl.cam.ac.uk)

Abstract

This paper describes the development and computational testing of a model of the information intermediary based on an AI theory of belief revision. We describe the theoretical foundations of the work in a general account of the way an agent's beliefs and intentions are formed and modified, and in an analysis of the functional tasks an intermediary has to carry out; we indicate the specific developments required to automate and integrate both aspects of intermediary behaviour, as determinants of interactive dialogue with the user; and report, with illustrations, on tests and findings. The research shows that such approaches can be implemented in an essentially principled manner, though there are many large problems still to be overcome, and our experiments are only the first, extremely simple, trials of the basic strategy for intermediary simulation.

1 Introduction

We have sought to model the information intermediary by combining a general theory of belief revision, due to Galliers, with the functional characterisation of the intermediary proposed by Belkin and his colleagues. We have developed and tested a computational implementation showing that our model supports simple information-seeking dialogue with the adaptive and cooperative properties of real ones. Our experiments validate some, though not all, of the major claims of the belief revision theory; they also suggest that while Belkin at al's functional account of the intermediary's skills is rational, modelling the intermediary as a collection of independent agents is not. Galliers' theory underpins communication between agents, but effective communication between user and librarian requires dialogue control that an unconstrained distributed agent architecture does not provide. This paper describes our development and implementation of the two theories as a single integrated system, and presents illustrative results. These lay foundations for further research, though a great deal more work is needed before such a sophisticated approach to interaction between user and intermediary could be expected to issue in any practical interface. A full account of this work is given in [1].

2 Background

We described the motivation for our work, and both Galliers' and Belkin et al's theories from this point of view, in some detail in [2]. Galliers' theory itself is presented in [3] and the MONSTRAT model work by Belkin, Brooks, Daniels and others in [4], [5], [6], [7] and [8] (hereafter collectively called BBD).

As an illustrative starting point here, our aim is to model the librarian's capacities in such a dialogue as:

```
U:  I am looking for books on Classical architecture.
L:  Are you more interested in Greek or Roman architecture?
U:  No, like the British Museum.
L:  Ah, you mean Classical Revival architecture.
    .....
```

The key features of Galliers' theory are that agents are autonomous but also knowledge and resource limited, so cooperative interaction with other agents, notably by dialogue communication, is required to reach an agent's goals. Formulating and achieving goals, as in literature seeking, is based on belief revision: thus formulating an adequate statement of information need and a corresponding retrieval strategy involves communication, driven by current intentions and plans, aimed at resolving conflicts of belief due to inadequate knowledge and obtaining a solid base for action. Belief revision is determined

[*] Present addresses: Brian Logan, Martin Centre for Architectural and Urban Studies, Department of Architecture, University of Cambridge; Steven Reece, Department of Engineering Science, University of Oxford.

by how ground assumptions are endorsed, derived beliefs are connected, and change is minimised. In the revision process a new belief is evaluated in terms of preferences among the alternative consistent sets of beliefs that adopting or rejecting the candidate belief imply. In general agents are evaluating many competing belief sets in order to establish which one(s) are superior in terms of stronger *endorsement*, greater *connectivity* (proofs), and *minimal change*.

BBD's intermediary/librarian model, based on a detailed study of real interviews, treats the intermediary as a collection of specialised functional experts, agents engaged with subtasks of the overall task of satisfying the user's information need. These experts include 'Problem Description', 'User Model', 'Retrieval Strategy' etc, each operating as an autonomous agent requesting and passing information among themselves and also communicating with the user. BBD noted that in the real dialogues there was no overall protocol, but that different segments, or 'foci', in the dialogues concentrated on a specific expert agent's goals within a flexible and varied flow visiting and revisiting subtasks. They thus viewed the intermediary as a distributed system, and initially advocated an open, blackboard architecture.

It appeared that these two theories could be very effectively combined to obtain the required full intermediary model, since Galliers' belief revision theory seemed to provide the mechanism for driving both the individual functional experts and their joint 'collective' operating as a single agent vis-a-vis the user.

2.1 Required Developments

While Galliers' and BBD's theories appeared to fit well together, applying them together required not only computational implementation, but significant development of both in order to supply missing pieces and ensure effective integration. It was necessary to extend Galliers' account of operations on beliefs to treat intentions and planning, and to provide for inference, in order both to get the inputs required to choose which beliefs and intentions to adopt and to pursue these through action. The computational modelling itself called for a specific mechanism to manage the data and processes of revision.

It was necessary to supplement BBD's account of the intermediary, on the other hand, with some means of sustaining and organising dialogue with more power than the BBD 'Response Generator' and 'Explanation' modules. Though Belkin and his colleagues (cf [4]) report experiments with the internal architecture of the intermediary, they did not address the question of dialogue management. However it is clearly necessary to decide when to ask the user for information, and maintaining effective interaction with the user implies that dialogue content has to be ordered and grouped: system outputs cannot simply be triggered, as separate items, by independent internal prompts. Moreover, as discussed in [2], BBD's model needed development to integrate dialogue management and overall control and to substantiate the metafunctions 'Plan' and 'Match' BBD do not analyse. Though Ingwersen in [9] has proposed a much more elaborate model, essentially subsuming BBD's, he has not tackled these issues either.

3 Implementation of Belief Revision

In Galliers' theory ground assumptions (relative to a context) are endorsed as communicated, given ('innate'), or hypothetical. Belief communication may be first-hand or second-hand, positive or negative, so e.g. a strongly communicated belief from a highly reliable source may be endorsed *2c-pos*. The options for given are either specific (*spec*) or default (*def*) generalisation. Any beliefs not otherwise endorsed are hypothetical (*hypoth*). Thus beliefs (which may themselves be propositions or their negations) can be ordered by the difficulty of disbelieving them, from those endorsed *1c-pos* down to *hypoth*. Beliefs are related to other beliefs, forming internally consistent sets, so one set may be more strongly endorsed than another. Sets may also be more or less strongly connected, or internally dense: thus individual beliefs, especially *core beliefs* independently defined as important in some context, may have more or less derivational (proof) support in different belief sets. So if an agent is entertaining a candidate belief, adopting or rejecting it (i.e. adopting its negation) involves examining all the different maximally consistent belief sets containing either it or its negation, and preferring those that are better endorsed and more connected (*mc*). In addition, since the whole approach is a conservative one based on the principle that agents are reluctant to change their beliefs without good reason, if other things are equal sets that make minimal change over the previous state are preferred. These types of reason for preferring belief sets are considered in order, first by connectivity, then endorsement, then minimal change: however any may determine the final outcome. This will normally be a set of equally preferred belief sets requiring further resolution through the acquisition of new information.

Intentions depend on beliefs, and revising beliefs entails revising intentions. We have therefore incorporated intentions into an extended account of cognitive *attitudes*: cognitive states are combinations of propositional attitudes, with derivational links between beliefs, intentions, and predicted future states. The general process of belief revision applied to all attitudes then handles intention revision either directly in the face of conflict with communicated intentions or indirectly as a result of conflicts between motivating beliefs. Connectivity applies to intentions in a straightforward way. Commitment, the key feature of intention, is captured in part by the strength of endorsements on supporting beliefs, and in part by characterising intentions directly according to the utility of their goal states and the effort required to reach these. Thus goal states are either *desire-pos* or *desire-neg* and either *effort-pos* or *effort-neg* (on simple heuristic definitions). The difference between beliefs and intentions is that the strengths of beliefs depend on their sources, the commitment to intentions on their (expected) outcomes. Then in choosing between competing sets of attitudes involving both beliefs and intentions, these are rated first by the belief preference criteria and then the intention ones.[1]

It is further necessary to provide for inference and planning. This is straightforward. Normal inference mechanisms (e.g. modus ponens) add derived beliefs or intentions to existing attitude sets, resulting in either conflict or additional support for existing attitudes and therefore provoking revision. Planning is also handled by inference, applying intention generation rules along with planning operators to existing attitudes. There is thus a basic *agent action cycle*: incorporating new information in existing attitude sets, firing deductive inference rules over existing attitudes to obtain new ones, performing revision to obtain preferred sets, and executing preferred intended actions. This is the way an agent responds to communicated input in a dialogue, modifying its attitudes and itself taking communicative action.

3.1 Computational Apparatus

The computational implementation requires an apparatus for constructing and evaluating all the attitude sets constituting responses to input. Our approach, the 'Increased Coherence Model' (ICM) is based on de Kleer's ATMS [10, 11]: for a full account see [1]. An agent has a message interception unit (MIU) for storing incoming and outgoing messages and a cognitive unit (CU) reading from and writing to the MIU board. The CU has a database, an attitude revision (i.e. ATMS-based) component, an inference engine, and a planner: the engine contains axiomatic rules operating on the information in the database which consists of derived attitudes and their supports; the revision component does belief and intention revision given the available justifications, taking the attitudes as ATMS assumptions; the planner constructs and assesses plans, working on intentions derived from beliefs and resulting in actions. As an agent's cognitive state covers believed, intended, or uncertain attitudes, and the agent's commitments to its attitudes, the engine and planner use this information to guide their choice of inferences and plans, and the ATMS to maintain consistent sets of attitudes.

Notationally, we refer to an agent's attitude to a proposition or state of affairs, the time of holding the attitude and the endorsement of or commitment to it (for convenience we may refer to individual attitudes as possible attitudes). For convenience we refer to commitments to beliefs, i.e. to the strength with which we hold them, as well as to commitments to intentions: commitments to beliefs are heuristically computed from their endorsements and may be strong, weak or uncertain. Then if an agent's belief state is its set of preferred belief sets, a given proposition can figure in these in various ways reflecting the agent's degree of commitment to it. As noted, intentions are based on beliefs but are explicitly marked, and there are intention sets analogous to belief sets. Belief and intention are linked by rules to the effect that an agent cannot intend a state it believes true. Overall, revision results in consistent sets of interdependent beliefs and intentions characterised by the agent's commitments to these attitudes. We say an agent believes a belief or intends an intention if these attitudes occur in every preferred set, i.e. are *pervasive*; attitudes in some sets only are uncertain.

Internally, commitments are handled as a form of endorsement: the various types of commitments are included in the endorsement orderings for beliefs and intentions respectively, and can therefore be used in preference evaluation to determine, as a consequence of revision, what the agent's new commitments to its attitudes are. Individual attitudes can be multiply endorsed, and may also be *definite*, i.e. ascribed to another agent and treated as premises. An agent's database thus consists of all the agent's attitudes, their endorsements (including commitments), and their justifications, i.e. inference supports. In processing via the ATMS, preferences between sets in terms of their attitude endorsements are computed by the procedure described in [1]: as endorsements are not propagated through to *derived* attitudes, the effect of the algorithm is to test whether attitudes should be retained or adopted by seeing if they remain viable when their weakest ground assumption is removed. Evaluating attitude sets for connectivity (for core attitudes) is easily done via the ATMS: this records minimal proofs via justification chains and these can be checked to see whether one set includes all the proofs in another set plus additional ones, and so should be preferred. Finally, the ATMS can be used to evaluate for minimal change by checking to see which sets preserve most of their previous pervasive beliefs.

Inference rules are applied to obtain new justifications for attitudes. These are general rules, with endorsements, that are not part of the database. However when they are instantiated via the rule binding algorithm they become specific attitudes providing justifications in the database and therefore subject to defeasible inference. Given the many possible inferences that can be drawn relating to new attitude,

[1] Note that previous states may emerge as still preferred.

e.g. a communicated belief, we constrain inference by concentrating on the current task, as defined by *recency* and modelled by a stack: thus inference is applied to attitude sets from the top of the stack. The inference algorithm (described in [1]) exploits *relevance* relations between attitudes, where commitment to one attitude depends on that to another.

As mentioned earlier, for action (as in dialogue) to satisfy intentions, planning is needed. This is done in STRIPS style, implemented via rules applied through the inference engine and exploiting the belief revision mechanism to choose preferred plans. The rules include desire rules determining the leading intention and utility of a plan, and planning rules to decompose intentions: the planning rules in turn operate on *action schemata* embodying primitive agent actions, which may be 'internal' (e.g. perform an inference) or 'external' (e.g. send a message to the MIU for communication to another agent).

Computationally therefore, the agent's cognitive architecture is operationalised with the ATMS, working on the database, supporting the belief revision mechanism, which plays a central role: it supplies attitude information to, or receives it from, the inference engine, and it supports planning both via inference in plan formation and through plan assessment. Agent processing in an action cycle is provoked by MIU messages stimulating possible beliefs.

For example, in the dialogue fragment presented in the previous section, it turns out U is mistaken about the meaning of the term 'Classical architecture'. This involves two changes of belief: L revises her original assumption about the period U is interested in; and U revises his belief about the meaning of the term 'Classical'. U's initial belief sets contain the pervasive belief

(*bel U* (*problem-desc classical-architecture*) *strong*)

i.e. U is strongly committed to the belief that *classical-architecture* is a suitable problem descriptor, and the intention

(*p-int U* (*bel L* (*problem-desc classical-architecture*)) *desire-pos*)

i.e. U intends to share his problem description with L. U generates a plan to achieve this intention, which results in his first utterance. U's utterance causes L to revise her beliefs to include the fact that the topic of U's query is 'Classical architecture'. Immediately after U's utterance, L believes

(*p-bel L* (*bel U* (*problem-desc classical-architecture*)) *2c-pos*)

and, following inference and belief revision, L adopts U's belief as her own:

(*bel L* (*problem-desc classical-architecture*) *strong*).

L does this because she has no reason to believe that 'Classical architecture' is not a good description of U's problem (e.g. U has not previously stated he is writing an essay on fish farming). L constructs a plan from this new belief to determine whether U is more interested in Greek or in Roman architecture (perhaps because L infers that the problem description 'Classical architecture' is too general). However U's response to L's question leads L to realise that there has been a misunderstanding and that the appropriate problem description is 'Classical Revival architecture', not 'Classical architecture'. L therefore abandons her plan to determine if U is more interested in Greek or in Roman architecture.

4 Implementing the BBD Model

As mentioned earlier, implementing the BBD model required a decision about the specific architecture to be used, provision of a dialogue management capability, and a supply of actual IR task knowledge (data and rules). The version of the BBD model we have used for reference is that given in [7]

The architecture design problems of distributed systems are well known, particularly the challenges presented by a full-blown open-access blackboard approach with multiple agents interacting under an uncontrolled, 'evolutionary' regime. As noted in [12], this view of the intermediary, as originally suggested by BBD, presents great difficulties in the face of unpredictable inputs from arbitrary agent sources, only very general prior definitions of task and task satisfaction, and the need for communicatively effective (i.e. coherent) dialogue with the user. In these circumstances viable system operation calls for a controller with so much knowledge and power that it actually subsumes the individual expert agents.

The comparative experiments reported in [5] simulating various architectures indeed showed that some degree of control was necessary, though they concluded that a blackboard architecture was still preferable to an actor one. The various implementations of expert intermediaries done by [13], [14] and [15] have all been of a quite strongly controlled kind, with experts distinctly subordinate to a controller rather than working as a collective of equals.

We carried out our own comparative simulations with alternative blackboard and actor architectures [16]; these showed somewhat better performance for actors, but again emphasised the need for control. But more importantly, when we came to implement dialogue management in detail, which neither BBD nor any of the other systems mentioned had significantly addressed, we found it too difficult, at least for our 'Mark 1' system, to work with a set of functional expert agents. We therefore not merely abandoned the original blackboard proposal, but the idea of a multi-*agent* implementation for the intermediary. We adopted a much more conventional approach with a single control module and the functional expert agents replaced by distinct specialised rule sets. The BBD model had some ten functional experts, which could be grouped as central and support processors, the former including e.g. Problem Description and Retrieval Strategy, the latter e.g. Input Analysis; but the nature and relations of the support processors are not well-developed in BBD's account. We wanted to concentrate on belief revision, and not to engage with language processing per se. Our Mark 1 architecture therefore had a controlling Dialogue module managing both interaction with the user and the other specialised retrieval task modules, which was thus the locus of the intermediary agent's belief revision. The system does not do any actual natural language processing: the (notional) input/output interface receives and sends communications in a simple propositional language with speech act operators, which we assume current NLP interpretation and generation could handle; we also assume that communication at the strictly linguistic level is transparent, i.e. that librarian and user agents correctly 'hear' and 'parse' messages they receive.

4.1 Dialogue Modelling

We have had to develop and implement a specific model of dialogue pragmatics suited both to the way agents are motivated by belief and to the nature of information-seeking dialogues as illustrated by BBD, and hence designed both to achieve dialogue goals and to maintain proper dialogue flow. We have drawn, as obviously appropriate, on the theory of speech acts, where communication depends both on an agent's own beliefs about the world and on its beliefs about its interlocutor(s), and is strategically motivated to reach end states that the dialogue participants have cooperated to define and attain. Within this framework each individual dialogue contribution is motivated by a desire to change participants' cognitive states, i.e. beliefs and intentions; but it is also, to ensure effective communication, constrained by the need to maintain dialogue coherence which relies on the use of recognised structures for interaction, like question-answer, as well as topic continuity.

The primitive speech acts we use are conventional in their general style, but have been defined to reflect our need to capture the actual conditions of dialogue participation (where little can be assumed) rather than advance planning (as in [17]). We have three speech acts, *tell*, *ask* and *answer*, with specific preconditions and effects expressed primarily in terms of the agent's beliefs about its own and its interlocutor's states. For simplicity, the propositional language used to communicate is that also used for the internal representation of attitudes; however, reflecting the real case, it does not follow that the internal 'meanings' of linguistic terms are the same for all participants: e.g. what the predicate *'classical'* denotes in the architectural domain. Communication includes an indication of the speaker's commitment to the attitude being communicated, which may be *strong* or *weak*.[2]

In our implementation successful communication is guaranteed at the basic level, i.e. where the hearer recognises the speaker wishes the hearer to have a belief (though the speaker may not themselves hold it). However at the higher level speech acts may be successful or not for various reasons. An act succeeds if the hearer not merely recognises but adopts the speaker's intention, i.e. if the speaker's intention for the hearer is satisfied, and fails otherwise. However these outcomes have to be specifically characterised in belief revision terms. Thus success implies that all of preconditions, communication, and effects must be achieved. Failure may occur e.g. when the hearer does not believe what the speaker supposes the hearer believes; it may be from speaker's or hearer's point of view, and may be 'obvious' or involve misunderstanding. By extension, a communicative plan will be successful if each speech act step succeeds.

To maintain dialogue coherence (or *d-coherence* to distinguish it from coherence of belief), we have found it sufficient to work with simple conventions in the spirit of dialogue games [18], which both participants know. Thus if each takes turns, there are legal responses to each of the three possible outcomes for each speech act. The responses are of *continuation* type or *repair* type, with the former depending on context and the latter on perceived failure. Thus we have a transition table, so e.g. a *tell* may be continued by a *tell* or *ask* or repaired by a *tell* or *ask*, while *ask* is continued only by *answer* and repaired only by *tell*. This table ensures d-coherence (at least over relatively short segments), and also naturally supplies segmentation analogous to the 'foci' of BBD's dialogue analysis, in a simple model of task-oriented dialogue structure. Specifically, continuation must lead to segmentation, which may or may not be on the same subject, while repair may lead to a new segment, but must be on the same subject; thus segmentation occurs when agents disagree or the dialogue subject changes.

The foregoing provide the basic dialogue apparatus. Dialogue is driven by the action schemata an agent has for planning, in this case for planning dialogue communication, supported by its desire and planning rules and also by specific dialogue rules. The desire rules, triggered by attitude conflict, provoke planning to resolve the conflict, say by one agent giving the other a justification for its belief. We have three schemata: `tell`, `adopt` and `infer`. These are complex objects characterising agents, attitude types, preconditions, effects, constraints etc. The `tell` schema is used for communication about attitudes; the `adopt` schema allow an agent to adopt an attitude held by another agent; and the `infer` schema drives

[2] "Speaker" and "hearer" are conventional dialogue terms: there is of course no actual speech processing in our system.

inference from a rule. To support these schemata there are further rules for ascription, so e.g. the hearer assumes the speaker believes the dialogue preconditions hold; for attitude adoption, so e.g. if the speaker is strongly committed to a belief, this is reason for the hearer to adopt it; and for prediction, so e.g. the speaker knows that if it communicates a strong commitment to an attitude, this is reason for the hearer to adopt it. Specifically, for instance, if the speaker communicates commitment *strong*, the prediction is that the hearer will endorse the communicated belief *2c-pos* or communicated intention *desire-pos*.

Prediction is complex, and to implement it adequately we have introduced finer levels of endorsement distinguishing an agent's predictions about its own (*auto*) attitudes from its predictions about another agent (*alter*)'s attitudes. These endorsements allow justifications like *auto* taking *alter's* strong endorsement of a belief as ground for assuming that *alter* will continue to hold the belief, so *auto* can plan on this basis. These *auto* and *alter* endorsement types are ordered, like the others, so prediction can be extended over combinations of current and predicted attitudes in the calculation of preferences between attitude sets. Supported by desire rules to identify leading intentions, and planning rules for intention decomposition, the rules just described support an enriched version of the agent action cycle allowing for prediction and action based on it, as well as just choice among attitudes.

All of these rule types are essential for planned action, and hence for the formulation and expression of communications as one kind of action. However this apparatus not only generates speech acts. Our approach as a whole also ensures stability of behaviour under dialogue, partly through minimal change in belief revision, partly through the desire to minimise effort, and partly through a general predisposition to avoid conflict.

For instance, in the earlier example dialogue fragment U's first utterance, a *tell* act, leads to a continuation response, an *ask* act by L on the same topic. U's first utterance successfully communicates U's intention and and appears successful in changing L's cognitive state. However U and L attach different meanings to the term 'Classical architecture'. Thus in contrast, while L's first utterance is successful in communicating L's intention, it does not result in the desired communicative outcome (an answer to L's question). The attitude conflict that is now evident leads instead to a repair response and a new segment. (Note that L predicted that the performance of the *ask* act would be successful in achieving the goal of determining whether U was more interested in Greek or in Roman architecture, otherwise L would not have asked her question.)

4.2 The IR Functional Component

As mentioned, the task experts are not fully independent agents in our actual implementation, but rather data and rule sets. In our experiments so far, moreover, they have been extremely simple. It is clear from studies like [7] that a Problem Description agent in particular, if intended for a practical implementation, would have to have ramified knowledge of a kind that is very hard to capture; and this is also implied by the amount of resource required even for a domain-limited system like [19]'s expert. However our primary aim has been to evaluate the mechanism of belief revision within the IR task context, and thus for our Mark 1 implementation we provided only enough task knowledge to see whether the system could simulate the *kind* of cooperative interaction, with a mutual exchange of information and development of a search specification, that BBD's transcripts exhibit. But even so, as the computational complexity of the mechanism, which has to work on many attitude sets, makes computation very effortful, we could only provide very simple task resources. Thus as described further below, we have only replicated human behaviour in an extremely limited way, and whether we have demonstrated either the adequacy of the model or the long-term practical viability of the approach is therefore a matter for discussion.

Taking [7] as our guide, as mentioned earlier, we have explored an architecture with all of BBD's five central experts: Problem State (PS), Problem Mode (PM), User Model (UM), Problem Description (PD) and Retrieval Strategy (RS). The crucial problem is in defining each module's goals in a way that makes satisfaction specifiable and attainable. We have treated the first three of these modules in a straightforward way in terms of limited attributes e.g. early or middle for PS, document type for PM, and novice or expert for UM, which can be checked by simple system questions if not supplied, and which are not mandatory for searching. PD and RS, especially the former, are more challenging even for a very limited intermediary but realistically-intended task modelling, and we were particularly concerned that both individual module knowledge bases and the 'translation' relationships between them should not be completely obvious.

For test purposes we chose architecture as our subject domain. Problem descriptions have several components: *topic*, *subject area*, *document type* and *document level*. Retrieval strategies have *term*, *database* and *document type* components. These components essentially correspond to BBD subfunctions as given in [7]. Document type and level do not exactly match 'slit', but they are clearly natural notions for retrieval, and having several components met our requirement to model satisfaction. We assume, partly for clarity in distinguishing various aspects of literature searching and partly as reflecting a common real situation, that searching uses a controlled indexing language (thesaurus). Thus while for topic components the PD module works with a *description* of the user's need in a simple *descriptor* meaning language, the RS module works to provide a *request* with Boolean operators linking *terms.* Then to guide the search development task we have definitions of *valid*, *minimal* and *good* descriptions/requests. Thus for our experiments we have naively defined a minimal description as one with a topic component with at least three descriptors, collectively neither too general nor too specific; and a good description as one with such a topic component and also document type and document level components (search area is inferred from topic). Requests are similarly defined, so a minimal request has a term component (with at least three terms), a good request has term, database, and document level components. Descriptors are organised in hierarchies, tagged with broad *subject* labels, and are supplied with *synonym* alternatives (reflecting natural language variation of concept reference). We assume that user and librarian have overlapping, but not identical, descriptor vocabularies. The essential business of forming a sound problem description is thus of choosing a useful number of descriptors from hierarchies with the appropriate subject orientation, at the right hierarchical levels.

As noted, though the PD descriptors may look like index terms, we wanted to allow both for the common separation of user language from search terms and for the task of formulating search strategies. We therefore require a mapping of description onto request which also satisfies our RS goals for requests. (For our simple experiments, however, we have stopped at the point of actual search and do not revise requests after retrieval feedback.)

Like the descriptors, our terms are organised in hierarchies, with subject labels and with descriptors as *entry words.* But we do not have a simple-minded one-one mapping between elements or structures. There are three possible descriptor-term mappings: direct; or synonym to term; or 'shift' to less or more general term. These data resources supply one part of the task apparatus: the other is provided by rules for obtaining descriptions and requests meeting the goal criteria, and especially the RS goal. The architecture thus assumes processing driven from the RS goal, with top-down goal resolution done heuristically (by abduction). The mechanisms for PS, PM and UM are quite simple, essentially asking basic questions. The system then has to derive an appropriate request from the PD description which may, e.g., mean getting a modified description from the user. The system has to work over both description and request attempting to satisfy the PD and RS goal criteria in the mapping transformation from description to request.

All of this seems quite straightforward from the retrieval point of view. It is however necessary to relate all these operations specifically to those of the belief-revision apparatus: they have to be expressed in terms of attitudes, embedded in planning etc. In particular, the task goals have to be stated not just as desires, but in such a way that they force the development of a satisfactory request. This is done by the need to achieve a balance between effort cost for the librarian and relevant retrieved benefit for the user.

Thus in our illustrative dialogue fragment, L's question about the type of architecture U wants is motivated by a desire to improve the problem description, since U's initial topic component, the single descriptor 'Classical architecture', does not constitute even a minimal problem description and would result in a poor search request. To achieve L's goal of a good request (to serve U's goal for information), L must devise a plan to modify U's cognitive state, but one which also does not consume too many computational resources.

We simulated the specific task model for PD using OPS5 and showed it was acceptable. However when we attempted the actual implementation we found further simplification beyond that described above was needed and we therefore reduced the task modules to three: UM, PD and RS, with UM and RS very rudimentary.

5 Computational Testing

The main problem computationally is the cost of manipulating all the very many attitude sets that may have to be considered: quite modest propositional proposals very naturally and plausibly generate large numbers of alternative responses for assessment. For instance with a simple three-turn interchange and small knowledge bases and rule sets, thousands of attitude sets may be produced. Our initial runs were extremely time consuming, but with 'optimisations' of various sorts (see below) we reduced the time required to a reasonable level, i.e. from days to minutes.[3]

Our test methodology was to see whether we could replicate example dialogue phenomena, i.e. forms of information exchange from both topic content and pragmatic points of view. Thus we ran the system in 'duplication' mode, i.e. simulating two agents, one the librarian and one the user. We could not of course,

[3] Note that we can't reduce the number of belief sets by eliminating those sets which are least preferred. Not only would this make major revisions in the agent's beliefs impossible (the agent having discarded the necessary 'improbable' belief sets), it causes problems when an intended state is achieved or when the world changes even in predictable ways. Presumably the agent is reasonably sure a state it intends does not currently hold, otherwise it would not have been trying to achieve it. However this means that the intended state will always be least preferred and therefore will be discarded.

with ‘free’, ‘intelligent’ agents expect that our system behaviour would simply copy that of the source examples, as it were word for word: we were looking for similar behaviour. We also lacked the knowledge sources to replicate the topics of dialogue examples taken from the BBD transcripts. What we did was identify specific types of phenomenon which could be convincingly described as e.g. showing a failure of a default assumption, and then, by providing an appropriate start state in out architecture domain, run the agents to see what happened. Thus given a simple architecture domain version of a transcript dialogue fragment, F, we sought a system-produced analogue F’. Altogether we ran four such simulations, covering a range of phenomena including ‘failed inform’, the case where one agent’s knowledge is incomplete and the agent knows it is; ‘misunderstanding’, where an agent lacks knowledge but does not realise it; and ‘failed prediction’, i.e. by an agent about the effects of an utterance. The simulations utilised a total of 11 domain-specific rules and 33 dialogue rules.

5.1 An Extended Example

So, for instance, for the exemplar fragment for a ‘failed inform’

```
U: I’m looking for books on Wren.
L: Who is Wren?
U: He designed St Paul’s Cathedral.
```

we obtained the two-agent system dialogue in our ‘message’ language

U : (*tell U L* (*bel U* (*problem-desc wren*)*strong*))

L : (*tell L U* (*int L* (*exists !x* (*bel L* (*class wren !x*))*strong*)))

U : (*tell U L* (*bel U* (*class wren designed-st-pauls*)*strong*))

(where we omit books for simplicity, abbreviate *user* and *libr* to *U* and *L*, and use *!x* as existential quantifier): this may be glossed in English as

```
U: my problem descriptor is wren
L: I want a problem descriptor class for wren
U: the problem descriptor class for wren is designed st pauls
```

The way this works, heavily abbreviated and simplified, is as follows. U is initialised with beliefs about *wren* including the belief that *wren* is a good problem descriptor, and with the possible intention to share this descriptor with L:

(*p-int U* (*bel L* (*problem-desc wren*))*desire-pos*)

From this intention U constructs a plan to communicate with L, issuing in U’s first message above. The plan has 5 steps, covering both telling the message content and doing so strongly enough to ensure L adopts U’s problem description. At the point of ‘utterance’ U has 10 belief-type attitudes and 67 intention-type ones, in 4 candidate sets of beliefs and 12 of intentions. L infers U’s strong commitment to *wren* and hence cooperatively adopts it as U’s problem descriptor. But as L cannot find *wren* in a descriptor hierarchy, as needed to develop a sound problem description, L instantiates a rule to ask U for *wren’s* class. L’s plan for finding out about *wren* from U has 14 steps, including e.g. one from the conjunction of L’s belief that she does not know what class U believes *wren* to be and L’s intention to `adopt` (i.e. to come to believe) what class U believes *wren* to be, to the intention that L should come to know what U’s class for *wren* is. This is represented as (step 3):

(*not* (*bel L* (*exists !x* (*bel L* (*bel U* (*class wren !x*)))))) &
(*p-int L* (*exists !x* (*action* (*adopt U L* (*bel L* (*class wren !x*)))))*effort-pos* ⇒
(*p-int L* (*exists !x* (*bel L* (*bel U* (*class wren !x*)))))

Another, later step (step 13) from L’s prediction that U will `tell` L what class U believes *wren* to be to L’s prediction that she will know what class U believes *wren* to be has the form

(*f-p-bel L* (*exists !x* (*action* (*tell U L* (*bel U* (*class wren !x*)))) ⇒
(*f-p-bel L* (*exists !x* (*bel L* (*bel U* (*class wren !x*)))))

f-p-bel denotes a future possible belief, i.e. a prediction. L’s plan is based on the belief that U has a class for *wren*, which is justified by a rule exploiting the fact that U produced this descriptor. L’s output utterance strongly conveys L’s intention to discover U’s class information, and U therefore adopts this as an intention to satisfy which, as U already has *wren’s* class, leads to U’s reply.[4] At this point, after a 4-step plan, U has 179 propositions under consideration with 4 candidate belief sets and no less than 1152 candidate intention sets. The plans involve many rule applications, e.g. for step 13 above the rule

[4] When an intention is achieved, the intended state switches from disbelieved to believed, and the intention becomes distintended. This change is propagated to higher-level states and intentions that depend on the intended state. For example if, immediately prior to L’s utterance, U were to add “... and his design for St Paul’s”, L’s goal is achieved and her plan is therefore aborted.

(*f-p-bel ?A* (*exists ?X* (*action ?ACT*))) &
(*action-schema ?PRECOND ?ACT* (*?EFFECT*)*?CONSEQ ?EFFORT*) ⇒
(*f-p-bel ?A* (*exists ?X ?EFFECT*))

i.e. if an agent *?A* predicts that an action *?ACT* will be performed, then it believes that the effects *?EFFECT* of that action will be true in the future, is instantiated, and the whole seems extremely top-heavy for such a minimal exchange between two agents. However it correctly demonstrates that when such reasoning is properly decomposed, as it has to be, many non-trivial steps are needed to make even the most obvious transitions from one attitude to another.

We had two problems with the computational testing. The first was that we could not identify natural core beliefs, required for applying the connectivity (*mc*) preference criterion, and so could not evaluate this important part of the basic theory properly. The second was that the system could only be radically speeded up by not preserving previously planned, but unsupported, intentions, which also weakens our claims for the theory. Subject to these qualifications, however, we found that our experiments did, on an intuitive assessment (the only one practicable), deliver cooperative, problem-solving interaction of the kind required for the task.

6 Assessment

Though computationally expensive, the tests were very limited in relation to the extent of task knowledge and scope of dialogue found in the real IR case, as illustrated by e.g. [20] or [21], or as compared with the computational dialogue modelling envisaged for the TRAINS project (see [22]). The problems that emerged from our modelling study as a whole were with computational complexity; connectivity; prediction both of world states and intentions; communicating commitment; and focusing in dialogue, so e.g. a topic shift is recognised. Reducing complexity needs some way of limiting revision by e.g. taking some attitudes as fixed; connectivity needs refinement to allow for quality as well as quantity of proof; predictability needs a theory of common sense reasoning (a tall order); communicating commitment requires another intensional layer in the system; and focusing needs at least something like recency. These are all tough problems, with complexity and dialogue focusing as most pressing. Further, to get a working implementation even for trivial dialogue examples, we had to introduce many rather specific endorsement types. This conflicts both with the claim for a general approach and seems unsatisfactory in itself. It is possible that the correct response is to operate on two levels, with a few universal types subsuming a set of more particular, task and domain specific endorsements. This requires further study.

But we nevertheless believe that, limited though our experiments have been in many ways, they do demonstrate that the sophisticated approach to attitude revision proposed by Galliers does lead to appropriate communicative interaction between agents in an information-seeking situation like the library one, and as such provides a deeper account of the intelligent intermediary than any hitherto proposed or implemented. This may appear paradoxical given the rich, cognitively-motivated account of intermediary functions that [9] gives. Our general approach is sympathetic to Ingwersen's call for open interfaces that support the user; and from one point of view our implementation falls within Ingwersen's model, since the explicit functions implemented are those Ingwersen and BBD share. We have also implemented, though with elementary substance, other implicit functions. But we have done this in ways that cut across Ingwersen's function block structure. Moreover the specific requirement, for computation, to model operational *processing* in detail has meant that we have not only begun to fill in key blanks in Ingwersen's and the BBD models: we have also made them more genuinely dynamic.

Acknowledgement

This work was supported by the UK Tri-Council Initiative on Cognitive Science and Human-Computer interaction, under Grant SPG 8930752: we are grateful for this funding. Dr Alison Cawsey, now of the Department of Computer Science, University of Glasgow, worked on this project in its first years, and Dr Julia Galliers, of the Computer Laboratory, has been associated with it throughout.

References

1. Logan, B. et al. "Belief revision and dialogue management in information retrieval". Technical Report, Computer Laboratory, University of Cambridge, 1994
2. Cawsey, A. et al. "Automating the librarian: belief revision as a base for system action and communication with the user". *The Computer Journal* 1992; 35:221-232

3. Galliers, J. R. “Autonomous belief revision and communication”. In: Gärdenfors P. (ed) *Belief revision.* Cambridge University Press, Cambridge, 1992, pp 220–246

4. Belkin, N. J., Seeger, T. and Wersig, G. “Distributed expert problem treatment as a model for information systems analysis and design”. *Journal of Information Science* 1983; 5:153–167

5. Belkin, N. J., Hennings, R. D. and Seeger, T. “Simulation of a distributed expert-based information provision mechanism”. *Information Technology* 1984; 3:122–141

6. 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.* Aslib, London, 1985

7. Brooks, H. M. *An intelligent interface for document retrieval systems: developing the problem description and retrieval strategy components.* PhD Thesis, City University, London, 1986

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

9. Ingwersen, P. *Intermediary functions in information retrieval interaction.* PhD Thesis, Copenhagen Business School, 1991.

10. De Kleer, J. “An assumption-based truth maintenance system”. *Artificial Intelligence* 1986; 28:127–162

11. De Kleer, J. “Extending the ATMS”. *Artificial Intelligence* 1986; 28:163–196

12. Cawsey, A. et al. “Automating the librarian: a fundamental approach using belief revision”, Technical Report 243, Computer Laboratory, University of Cambridge, 1992

13. 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* 1990; 20:166–185

14. Fox, E. A., Weaver, M. T., Chen, Q.-F. and France, R. K. “Implementing a distributed expert-based information retrieval system”. In: *Proceedings of RIAO 88 Conference on User-Oriented, Content-Based Text and Image Handling.* MIT Press, Cambridge MA, 1988, pp 708–726

15. 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* 1987; 38:389–404

16. Cawsey, A. et al. “A comparison of architectures for multi-agent communication’. In: *ECAI-92, 10th European Conference on Artificial Intelligence*, 1992, pp 249-251

17. Allen, J. *Natural Language Understanding.* Benjamin/Cummings Publishing Company, Menlo Park CA, 1987

18. Kowtko, J.C., Isard, S.D. and Doherty, G.M. “Conversational games within dialogue”. Research Paper HCRC/RP-31, Human Communication Research Centre, University of Edinburgh, 1992

19. Vickery, A. et al. “A reference and referral system using expert system techniques”. *Journal of Documentation.* 1987; 43:1–23

20. Bates, M.J. “Where should the person stop and the information search start?”. *Information Processing and Management* 1990; 26:575-591

21. Shute, S.J. and Smith, P.J. “Knowledge-based search tactics”. *Information Processing and Management* 1993; 29:29-45

22. Traum, D.R. “The discourse reasoner in TRAINS-90”. TRAINS Technical Note 91-5, Department of Computer Science, University of Rochester, 1991