An ATMS-based Belief Model for Dialogue Simulation*

J. A. Taylor
Human Communication Research Centre
2 Buccleuch Place
Edinburgh EH8 9LW
United Kingdom
email: jaspert@cogsci.ed.ac.uk

October 1, 1993

1 Introduction

This paper describes a belief model that has been developed for use in a computational model of dialogue. The model combines a basis in logic with restrictions on the propositions and reasoning steps that can be supported, resulting in a system that supports the type of reasoning that people sem to carry out subconsciously during simple, task-oriented dialogues. It is anticipated that this model will be extended to make more complex inferences available, but restrict their use to situations in which the restricted process has led to a breakdown in communication.

1.1 The problems of belief modelling

Understanding peoples' states of mind in terms of beliefs, desires and other attitudes is more than a prerequisite of everyday social interaction; many philosophers [6, 7] argue that such categorisations are the best start we have for the detailed scientific analysis of cognition. One tool for such analysis is the generation of models of cognition based on representations of these attitudes. Such models are to be found in computer programs from many disciplines within cognitive science, including models of the processes of generating and understanding utterances in natural language, as well as human-computer interfaces which adjust their behaviour in accordance with their model of the state of mind of the user.

In a tradition that originated with Liebnitz, the first-order predicate calculus and the analytical procedures associated with it were held to be the idealisation of rationality. However, for the purposes of creating computer systems that embody a belief-modelling capability, using FOPC presents problems due to its all-or-nothing nature. Hence most work on belief modelling in cognitive science (for instance Fagin and Halpern [8], Levesque [14] emphasises the development of varieties of logic which are well suited to problem solving in the real world, in which computational resources are limited, not all relevant knowledge is available and plausible assumptions must be made.

In contrast, the belief model to be described here is derived from an attempt to emulate the restrictions placed on logical inference by the mechanism of human thought. It is therefore not an embodiment of any formal logic, but its behaviour

*The work described here is funded by Joint Councils Initiative UK grant #SPG9200139, running for three years from November 1992.

can be described as making use of a subset of the inferences sanctioned by such a logic.

1.2 Dialogue simulation

We are interested in producing a belief model that can be used in a variety of contexts, including plan-based models of dialogue. There are several of these already in existence, mostly for the simulation of task-oriented dialogue, and they address many linguistic topics, including speech act theory and the generation of appropriate referring expressions (eg. Appelt [2], Reiter [18]).

A model of the simulated agents' state of mind is an essential component of any such system. The model must keep track of agents' acquaintances with concepts in the domain of discourse, as well as beliefs that particular statements about the domain are true or false.

Planning and interpreting utterances requires reasoning about the plans or potential plans of one's conversant, so the belief model must also be able to represent nested beliefs. Houghton's system [12] includes nested beliefs as preconditions for some utterances. Taylor's CYNIC [22] models agents' beliefs and desires in terms of a variant of epistemic logic. Its simulated conversants can reason recursively about each other's reasoning to any depth, although this approach gives rise to problems both of computational complexity and of psychological implausibility for some of its inferences.

These problems do not occur with Carletta's JAM [4], which is capable of making inferences about what inferences other agents will draw, but has implicit limits on the depths of nested reasoning and nested beliefs. The JAM belief model bears little explicit relationship to a formal logic of belief, but the belief model to be described here can be set up to replicate the functionality of the JAM model closely enough to support the same dialogues, and moreover has a clear basis in epistemic logic.

1.3 Reason Maintenance Systems

In typical reasoning problems, not all the required premises are known beyond doubt to the reasoner. Part of the task of reasoning is the making of assumptions, and the withdrawing of assumptions when contradictory facts come to light. Whether a new fact is added through reasoning or as a result of observation, it may cause a group of assumptions to become mutually contradictory. An assumption from the group must then be withdrawn, and this may allow other assumptions, previously withdrawn, to be reinstated. Each change in a reasoner's assumptions can result in major changes to his state of belief, as many facts may follow by implication from something that has been assumed. Gärdenfors [9] has described a model of how an agent's state of knowledge is updated in response to new knowledge in these terms. Various nonmonotonic logics for describing this type of process have been postulated, such as Raymond Reiter's [19] and McCarthy's [16]. Implementations of nonmonotonic reasoning rely on reason maintenance systems (RMSs), which cooperate with the problem-solving part of the system. RMSs accept data concerning dependencies and contradictions between propositions, and generate consistent *environments* — sets of propositions and assumptions that form a state of belief for the reasoner.

The belief model described here, like the attitude model of CYNIC, is implemented as a combination of an implication generator and a reason maintenance system. The interface between the belief model and the dialogue handling system is very simple; the belief model accepts constraints on the status of propositions or concepts, and can be queried about the status of related propositions or concepts. When a proposition is added to the belief model, a node associated with it is created

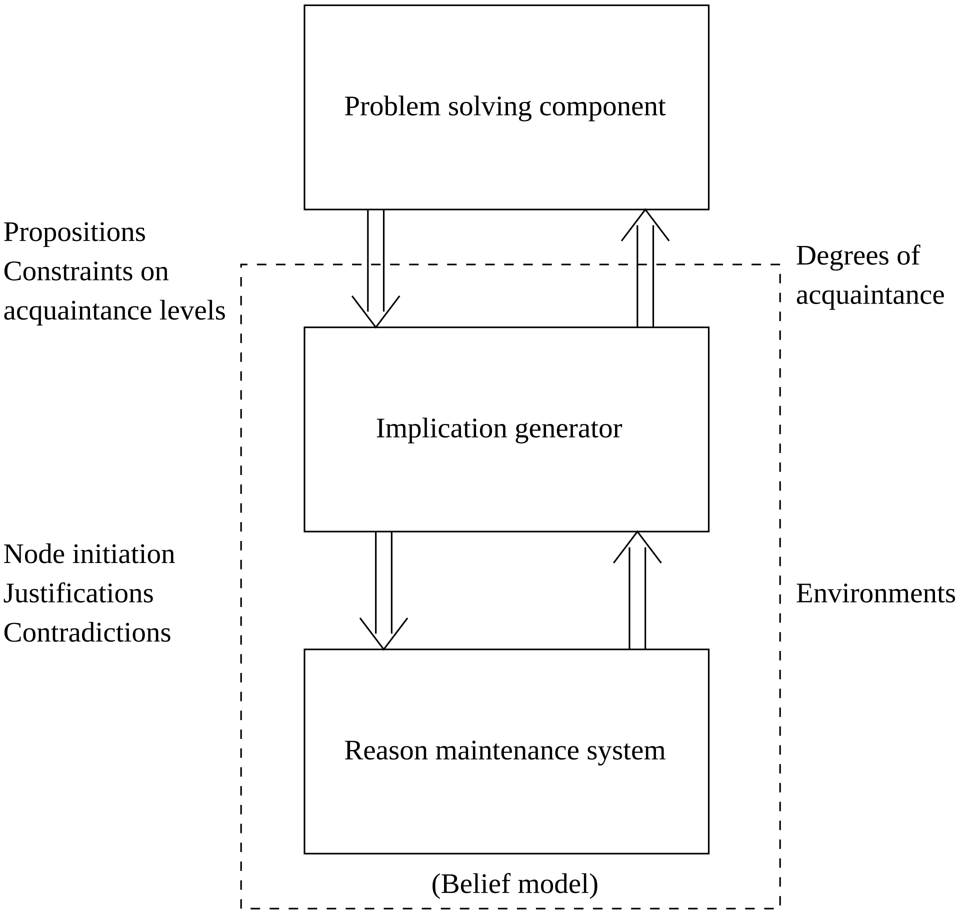

Figure 1: Components of an ATMS-based reasoning system.

in the RMS. The propositions already in the belief model are checked to see if any directly imply or are directly implied by the new proposition. If so, justifications are added to the RMS connecting the nodes corresponding to the related propositions. The action of adding justifications causes the RMS to update the status of related nodes. Figure 1 is an illustration of this process. The RMS used is an implementation of DeKleer's ATMS [5], a multiple-environment assumption-based RMS, although a similar system could be implemented with other assumption-based RMSs.

2 Background: the BELMOD project

This belief model has been developed as the first step in a project intended to provide a psychologically plausible model of belief. The tradition of Artificial Intelligence offers belief models which are implementations of various epistemic logics, such as those described by Fagin and Halpern [8]. These suffer from two basic problems — they are computationally intractable and psychologically implausible. These characteristics follow from the property of the underlying logical systems that all possible inferences are supported, and that therefore agents are modelled as knowing all the consequences of what they know — they are *logically omniscient.* Some current work in belief modelling restricts belief inference by using more complex logics in the model's specification (e.g., Fagin and Halpern [8], Levesque [14]). This allows agents not to be logically omniscient, and hence to hold contradictory beliefs. However, tailoring the specifications of the logics to describe only plausible belief updates for humans involves making them more and more convoluted and less and less plausible as models for what humans actually do.

Human believers can cope with extremely large amounts of potentially relevant data, but have greater difficulties with particular types of problem that are not reflected in the performance of logic-based computational models. (See, for example, Stenning [21]). An approach to belief modelling is required which takes into account the fact that it is the nature of human memory which restricts the kinds of inferences which humans can make. For one thing, the capacity of working memory is severely limited, and this limits the depth of nested beliefs which can be handled. For another, it is difficult to represent propositions containing variables or disjunctions; "John, Jack and Henry are Canadian" and "Susan doesn't have a dog" are easier propositions to work with than "Either John, Jack or Henry is Canadian" and "Someone other than Susan has a dog", although their logical representations are equally simple syntactically in each case. It has been suggested by Stenning and others that this property is due to the tendency of human memory to build models of fully-defined states of affairs rather than maintain separate representations of different constraints on the current state of affairs [20, 21].

However, these limits on human reasoning ability are to some extent flexible. Generally, a problem that causes someone to pause and think during a conversation (eg. "Does my conversant already know that I know Fred knows which flight I will be taking?") can be solved with some conscious effort, and problems involving variables or disjunctions can be solved with careful thought and perhaps a pencil and paper. As with any type of reasoning problem, practice leads to improvements in performance. These observations suggest a belief model containing two interacting components: firstly, a model of the believed state of affairs consisting only of shallowly-nested propositions in which every individual is uniquely defined, and secondly a system more akin to the logic-based problem solvers or theorem provers of recent artificial intelligence research. The latter will be brought to bear on problems not amenable to solution by the former, and will generate a solution set of sentences possibly containing variables or disjunctions which will be used as templates for a new range of belief sentences to be employed in everyday reasoning.

The starting point for this project is the belief model used in Carletta's dialogue system, JAM [4], which models HCRC Map Task Corpus dialogues [1]. The problem domain comprises a set of concepts (such as map landmarks) and relations between concepts, with which each agent may or may not be acquainted. Agents may also know, or be unsure of, one another's degree of acquaintance with a concept. There is a potentially unlimited variety of beliefs an agent may have relating to a concept, including deeply nested or disjunctive beliefs, but JAM limits the number of ways in which an agent may stand with respect to each concept, thereby assuring the system is able to operate efficiently, as well as having a human-like inability to cope with propositions involving variables or deeply nested beliefs.

The aim of the project is to combine such a belief model with a more conventional, logic-based approach, thereby creating a more useful reasoning system and at the same time shedding light on the relationship between the reasoning process in a strictly limited system and the operation of an unrestricted reasoning system for which formal proofs of correctness and completeness exist. In concrete terms, the goal is to produce simulated dialogues in which a misunderstanding arises, which is then resolved by one or both conversants resorting to the use of a more general reasoning process.

Many obstacles must be surmounted to reach this goal. Firstly, the JAM belief model must be reimplemented in terms of an explicitly defined set of limits on a theoretically unrestricted problem-solving system. This has recently been completed; this paper describes the resulting system. Not only does this put the behaviour of the JAM model on a sound theoretical footing, it also allows the belief model to represent a number of alternative situations concurrently. This enhancement of the model's capabilities is nonetheless compatible with some psychological data, which suggests that experimental difficulty depends to some extent on the total number of alternative world models — see Byrne [3].

These changes have not extended the functionality of the system beyond that of the original JAM, although we can now add maximum depth of nesting of beliefs handled to JAM's set of parameters for characterizing agents. However, the next stage of the project will involve enhancing parts of the system to handle a richer problem domain. Agents in JAM make the implicit assumption that every object in their domain of discourse has a unique name, so there is no possibility of misunderstanding. By allowing ambiguity, and changing the design of agents in order that their unique-name assumptions become explicit, we can create situations in which misunderstandings arise and agents attempt to resolve them. The goal of this phase of the project will be to categorise the sorts of problem that can arise in this context in terms of the depth of nesting of beliefs required for their recognition and for their correct solution.

Finally, the unrestricted reasoning system for the solution of as many potential dialogue problems as possible must be designed. This will make use of the same representation language for beliefs as the restricted model, possibly with additional constructs for variables. This system will use a scheme for control of reasoning deriving from Taylor's planning system CYNIC [22] which uses heuristics to guide reasoning in an unrestricted domain. At the conclusion of a bout of unrestricted reasoning, those beliefs upon which the solution depends will be identified and any variables they contain instantiated to known individuals, producing beliefs which can subsequently be handled by the model of rapid, everyday reasoning.

The overall goal of this project is to draw a connection between the findings of psychologists who tell us which sorts of problems the human mental equipment is better or worse suited to, and those of logicians who provide us with techniques for solving arbitrarily complex problems of a given class. The result should be a system that can cope with dialogue problems with a similar pattern of success to that shown by humans, but which is demonstrably linked to an underlying logic, and can call upon the full problem-solving power of such a logic when it is required.

3 Characteristics of the Map Task domain

The problem domain in which in which the BELMOD belief model will be tested is that of the HCRC Map Task corpus [1]. Briefly, the corpus consists of dialogues in which two participants are each given a map, on one of which is marked a route. The maps consist of a coastline, and some labelled landmarks. The maps purport to represent the same terrain, but differ with respect to some of the landmarks and some of their labels. The purpose of the task is for the participant with the route marked (the instruction giver, or IG) to communicate it to the other (the instruction follower, or IF) as accurately as possible.

Although the apparatus for the task is simple, humans employ a wide variety of strategies in its solution, describing parts of the route with reference to everything on the map including its edges, and using multifarious terms to communicate distance, direction and shape, such as "come down in a J shape". For the purposes of JAM and BELMOD, the domain knowledge available to the agents has had to be reduced to the positions of the objects on their maps and a small group of prepositions. Specifically, we are avoiding any complex spatial reasoning, as will be described in section 3.1. However, the state of knowledge of the other agent is germane to each agent's strategy, and agents therefore must have nested knowledge. The representation of nested knowledge will be described in section 3.3. Furthermore, agents must know how to interpret utterances in the task, in terms of what knowledge the speaker must have had and what he must have lacked. This knowledge, along with that required for the generation of inferences, is mutual to both agents and is represented procedurally, as will be described in section 3.4.

3.1 Concepts and propositions

Concepts and propositions are the two basic types of entity which make up an agent's representation of a map. Carletta's thesis treated concepts as fundamental, and propositions as a type of 'glue' required for the functioning of the database, but here we wish to expound a viewpoint on the nature of concepts that will provide a justification for the way they are handled by the belief model.

In the JAM system, a concept is either an object that appears on a map, i.e., a landmark or a portion of the route, or a description of such an object in terms of other objects (eg. "The palm beach is to the left of the swamp"). Agents can be acquainted with concepts to different degrees, as will be explained in section 3.2. The concepts are linked together into a network, through which their degrees of acquaintance are propagated in min-max fashion: an agent's degree of acquaintance with a description is that of the referent in the description with which he is least well acquainted, and his degree of acquaintance with a map object is that of the description of that object with which he is best acquainted.

The status of a description concept also depends on a propositional, or 'glue' node. This node has a degree of acquaintance, like a concept, but refers to the proposition that the relationship between map objects predicated by the description actually holds. Only two degrees of acquaintance are possible for a proposition — lack of awareness and full knowledge. In addition to acquaintance derived through descriptions, agents are also directly acquainted with map objects that appear on their own maps, and fully know the propositions relating to any descriptions all of whose objects are on their maps.

The upshot of this type of representation is that agents' acquaintances with map objects can be reasoned about without any necessity for representing beliefs about their actual coordinates on the map. One obvious disadvantage of the system is that spatial inferences cannot be modelled, but we do not envisage any necessity to model such inferences even in the final stage of the project. However, since we are pursuing the goal of giving the model a basis in logic, we will state in section 3.2 exactly how the behaviour of concepts in our model relates to the underlying logic.

The new belief model retains the idea of concepts, but only map objects — not descriptions — are referred to as concepts. The degrees of acquaintance are propagated between map objects as described above but without an explicit description layer; for instance if an agent knows the propositions

$$describes(section\_one, between(palm\_beach, swamp)) \tag{1}$$

and

$$describes(section\_one, under(cliffs)) \tag{2}$$

then his degree of acquaintance with $section\_one$ (provided it does not appear on his map) is

$$D(section\_one) = max(min(D(palm\_beach), D(swamp)), D(cliffs)) \tag{3}$$

The removal of the explicit description layer not only clarifies the idea of a concept, it also restricts the types of query with which the database can be addressed, constraining the planning system to work only with map objects and propositions.

3.2 Degrees of acquaintance

If a concept is sufficiently well known to an agent for that agent to be able to use it in a description, then the agent is said to have that concept *vivid*. Vividness is the highest degree of acquaintance. The other degree used in JAM is *aware-only*; this expresses that the agent is aware of the existence of a concept but has no further information about it. Nested attitudes may have an *unknown* degree of acquaintance; this corresponds to an agent's uncertainty about how another agent stands with respect to a concept. JAM also allows an agent to be unaware of a concept, in which case the belief model for the agent contains no mention of the concept.

The idea of vividness is similar to Moore's notion of an executable definition [17]. For example, he asks what we need to know in order to telephone Sue; "Sue's telephone number is 453-2562" will do, "Sue's telephone number is Bob's telephone number" will not, and "Sue's telephone number is the square root of 20544118283844" is not helpful either. It is difficult to elucidate the exact nature of an executable description, but for the purpose of drawing a bit of route on a map, a pair of Cartesian coordinates expressed as numerals would be sufficient. In the following explanation, we will use this as the criterion for vividness, although some more qualitative sort of description of place could fill the same role.

We do not wish to clutter up our underlying logic with the concepts of geometry that would be necessary to allow our agents actually to deduce the coordinates of whatever map features their concepts of were vivid. Furthermore, if we were to do so, the resulting theory would allow agents to create new descriptions of landmarks once their concepts had become vivid due to learning other descriptions, and this behaviour does not accord with what people actually do in the map task. So instead we imagine that an agent knows the coordinates of what is on the map in front of him, and can apply the mathematical operations corresponding to the application of a preposition to an object.[1] Now the concepts that an agent has vivid are exactly those whose coordinates he can work out by the decision procedure of following the descriptions backwards to a set of objects on the agent's map, then applying the mathematical operations corresponding to the descriptions to get the coordinates of the vivid object.

The upshot of this is that we can treat the vividness of a concept like a proposition in epistemic logic, since it is equivalent to the proposition, "The coordinates of this object can be derived". Since vividness is the highest degree of acquaintance, the propagation procedure described at the end of section 3.1 causes a concept to become vivid iff all the concepts in some known description of it are vivid, which is exactly the criterion for propagating the derivability of the object's coordinates. Lesser degrees of acquaintance could be used to represent the derivability of different subsets of the information that can be derived about a feature that is vivid; such degrees could be placed on a lattice according to how the information corresponding to each one subsumes that corresponding to others.

3.3 Nested beliefs

Our belief model must include representations of how agents believe each other to stand in relation to concepts and propositions. Like most other belief models, we will have our agents represent the beliefs of others in the same terms as they represent their own beliefs, i.e., agent A may believe agent B to be only aware of a concept or to have it vivid, or to know or not to know a proposition. But an

[1] For instance, if "The waterfall is under the cliff", then the Y coordinate of the waterfall is 2cm less than that of the cliff and the X coordinate of the waterfall is the same as that of the cliff.

agent may also be unaware of how another agent stands with respect to a piece of knowledge. To represent this state we use a new degree of acquaintance "unknown".

We can apply this state to propositions as well as concepts, and the propagation algorithm described in section 3.1 will handle it correctly if we count it as coming higher up the hierarchy of degrees of acquaintance than only-aware, and lower than vivid. However it is not, strictly speaking, a degree of acquaintance and could not be ordered with respect to any degree of acquaintance representing partial knowledge about the concept itself.

A single piece of knowledge in an agent's model is thus characterised by:

- A list of agent identifiers, indicating the nesting path of the belief. This is empty in the case of the agent's own beliefs about the domain.
- The name of the concept or proposition.
- The degree of acquaintance/knowledge.

We shall write these states as ordered triples, for instance `{[A], swamp-0, vivid}` refers to the modelled agent's belief that agent `A` has the concept `swamp-0` vivid.

Such a representation is ambiguous in the case of beliefs at the second or deeper level of nesting where the degree of acquaintance is unknown. If an agent has belief `{[Doris, Fred], cliffs-0, unknown}` this may mean the agent believes agent `Doris` does not know how agent `Fred` stands with respect to `cliffs-0`, or that the modelled agent does not know how agent `Doris` believes agent `Fred` to stand with respect to `cliffs-0`. States of these types were conflated in JAM's model, and making the distinction between them does not appear necessary for the generation of plans to produce definite states of knowledge, so they are not distinguished in the current belief model's responses to queries. In the fully-expressive model to be developed in the next stage of the BELMOD project, however, it will be necessary to make a distinction between these sorts of states. The means by which we plan to do this will be described in section 5.1. 28 In theory, a representation such as this can be used for beliefs at any degree of nesting. For the model of rapid belief updating described here, a limit of one or two levels of nesting is set, depending on the complexity of the strategies (eg. plan recognition) being employed by each agent. When the database is queried by the planning component for beliefs at a deeper level of nesting than those which are explicitly represented, it calculates the response on the basis that beliefs at the deepest explicitly represented nesting are mutually known. An exception is made for agents' nested beliefs about their own beliefs, in which case the state that would be returned for the shallower belief on the part of the same agent is returned.

3.4 Effects of utterances on belief states

In the JAM belief model, the effects of producing or hearing any utterance were described by a transition table. One of the first tasks in the BELMOD project was to analyse the information in this transition table and design a principled set of rules for utterances that could be applied by speakers and addressees to produce the same updates to their states of belief, at least to the depth to which JAM handled nested beliefs.

This has turned out to be fairly simple; agents need only consider the conditions of making an utterance. If one of the conditions is knowing a proposition, that proposition becomes mutually known between the conversants (at least to the depth explicitly modelled in the database). If a condition is having a concept vivid, it becomes mutually known that the speaker has the concept vivid, and if a condition is that the speaker have some nested belief, it becomes mutually known that the speaker has that belief.

Example	Utterance type	Content type	Conditions
"The ...is..."	inform	proposition	{[], <proposition>, vivid}
"OK"	OK	proposition	{[], <subject>, vivid}
"I do not understand"	dont-understand	proposition	{[], <subject>, only-aware}
"Do you have the..."	yes-no	concept	{[hearer], <concept>, not-known}
"Yes"	yes	concept	{[], <concept>, vivid}
"No"	no	concept	{[], <concept>, only-aware}
"Where is the..."	wh-ask	concept	{[], <concept>, only-aware} {[hearer], <concept>, not-only-aware}
"I am done..."	close	concept	{[], <concept>, vivid}

Table 1: Conditions for utterances

Conditions are expressed as restrictions on the degree of acquaintance of a concept. For instance, one of the conditions for asking a wh-question is that the speaker's (S) belief about the addressee's (A) state with respect to the subject is at least unknown, i.e., not only-aware. When the question is asked, A's beliefs about the subject at nesting levels $[S, A], [S, A, S, A], \ldots$ are upgraded to unknown if currently only-aware, and the same happens for S's beliefs about the subject at nesting levels $[A], [A, S, A], \ldots$ (except that the first of these must already not be aware-only or the utterance could not have been produced). The process is cut off at the preset limit of explicit nesting.

Table 1 gives the conditions for all the utterances that can be modelled by the system. The following points must be noted about this table:

- These representations of utterances are not currently used for planning or plan recognition.
- The conditions are given in the form of triples with the last element being the *constraint* that the speaker's belief must satisfy if the utterance is to be produced. There are six of these; `only-aware`, `not-known` and `vivid` indicate that the speaker must have the concept at the corresponding degree of acquaintance, and `not-only-aware`, `known` and `not-vivid` indicate that the speaker must not have the concept at the corresponding degree.
- Replies to wh-questions are counted as informs.

4 Implementation of the model on the ATMS

As described in section 1.3, the belief model must record the beliefs that are passed to it, generate any necessary implications that apply between them, and propagate belief state changes along the paths of these implications. The task of generating implications is of linear complexity, thanks to the following features:

- Propositions are never implied by any other database items.
- All implications involving a concept can be found by searching for propositions naming that concept.
- Implications only occur within one level of belief nesting; we are not concerned with agents' beliefs about their own beliefs, and there is no direct connection between different agents' beliefs.

The task of propagating belief states according to these implications is handled by the RMS — in this case, an implementation of DeKleer's ATMS [5]. As well as being a theorem prover for a multi-valued logic as described by Ginsberg [10], the ATMS is an efficient system for carrying out large volumes of propagations. Like the state transition tables, the belief propagation rules used in JAM were not produced specifically in accordance with some logical basis, but using the ATMS as will be described produces only minor differences in the overall behaviour of the belief model.

4.1 Operation of the ATMS

An ATMS node is a triple consisting of a *datum*, a *label* and a list of justifications in which it is the consequence.

- The datum merely identifies the node, and in this system it is the belief corresponding to the node. In labels and justifications, nodes are denoted by numbers indicating their positions in the list of all nodes.
- The label is a set of *environments.* An environment is a set of assumptions. Each environment uniquely defines a truth assignment to all formulae represented by nodes in the network. The environments in any node's label are all the minimal sets of assumptions that must be made in order for that node's formula to be true. Labels are updated each time a new justification is added to the ATMS.
- Each justification is a list of *antecedents*, other nodes whose combined truth implies the truth of the node with that justification. These are added by the problem solver as described above.

A premise node has a justification with an empty list of antecedents. The label of a premise node has just one environment, the empty set. Thus its formula is true even if there are no tenable assumptions.

The label of an assumption node has just one environment, consisting of the assumption itself. An assumption node also has only one justification, indicating that it follows from itself.

The labels of other nodes depend on their justifications. Whenever a new justification is added to a node, its label is recomputed by combining the labels from the nodes in the justification. This is a complex process, and the interested reader is advised to refer to DeKleer's description [5]. When a node's label has changed, the labels of all nodes whose justifications include this node must be similarly updated.[2] The result is that assumptions in labels propagate along chains of justifications between nodes.

The ATMS has one special node representing contradiction. This can be given justifications like any other; for instance any pair of formulae which would lead an agent to know something that they have denied knowing form a contradiction. The environments of the contradiction node are called *nogoods.* They are the minimal sets of assumptions which cannot all be valid. When recomputing any node's label, environments which are supersets of others in the same node's label are thrown away. For nodes other than the contradiction node, environments which are supersets of nogood environments are also thrown away.

[2] The implementation of the ATMS used here provides each node with a list of the other nodes in whose justifications it appears, to avoid searching through all the justifications.

4.2 Modelling nested beliefs with the ATMS

There is no direct connection between the states of mind of different agents — they only affect each other through utterances that have taken place. So if we are modelling the beliefs of more than one agent at once, we use a separate instance of the entire ATMS for each agent. Furthermore, an agent's acquaintance with any proposition is independent of his beliefs about the states of mind of other agents with regard to that proposition. Therefore, a separate node in the ATMS is created for each nested belief about a proposition or concept. However, only one node is needed to represent the state of belief for each path of nesting, as the different degrees of acquaintance represented by JAM at each level are encoded by the different sets of assumptions that may be present in the node's environment.

The datum of each node in this system is composed of three parts, a *nesting*, an *object* and an *acquaintance.* These correspond to the three components of a piece of information in the system, described in section 3.3. Their roles are as follows:

- The nesting is a list of agents' names. If it is the empty list, the node corresponds to the modelled agent's direct acquaintance to some feature of the world. If it contains one name, the node corresponds to the modelled agent's belief about how the named agent stands with regard to some feature of the world. If it contains more than one name, it represents a multiply nested belief. For instance, the nesting [Fred Doris] indicates a node corresponding to the modelled agent's belief about Fred's belief about Doris's acquaintance with some aspect of the world.

- The object identifies the feature of the world to which the attitude relates. In the world of the map task, this may be a landmark, a route section, or a proposition relating to a description. In contrast to JAM's database, the descriptions themselves do not have nodes here. This is because a description corresponds to the implication that its subject is vivid if its referents and proposition are vivid, so it is represented by a *justification* within the ATMS.

- The acquaintance distinguishes the node corresponding to the belief itself from assumption nodes which are added to represent uncertainty about the belief. In the former case, the acquaintance is an empty list; when querying the ATMS to get the status of this belief, it is the node's label rather than its datum that is checked. In the latter case, this field identifies the type of assumption we are making about this attitude; "vivid" is the only one currently in use.

In order to model the three degrees of acquaintance which are used in JAM, only one type of assumption is needed. The JAM levels are only-aware, unknown and vivid (unknown is never the case for propositions or concepts in the domain, only for other agents' states of mind). In this system, a belief is only-aware if the label of its node contains no environments, unknown if it contains environments which contain "vivid" assumptions, and vivid if it includes the empty environment. This may not seem clear, but can be visualised by supposing that the node in the unknown state needs assumptions if it is to be believed vivid, whereas a node in the vivid state needs none, and a node in the aware-only state cannot be thought vivid whatever assumptions are allowed.

4.3 Setting and changing degrees of acquaintance

The initial state of affairs in a Map Task dialogue simulation contains beliefs at all three possible degrees of acquaintance. Beliefs that are only-aware, such as a route follower's acquaintance with the route sections, are given nodes with no justifications and hence empty labels. Beliefs that are vivid, such as agents' acquaintances with objects on their own maps, are given premise nodes. Beliefs that are initially unknown, such as an agent's belief as to how his partner stands with respect to landmarks that appear on the first agent's map but not necessarily his own, are given initial assumptions from which they are justified. Thus the label of the node for such a belief initially contains a single environment consisting of a single assumption.

As mentioned above, updates to the belief model are specified in the form of constraints. When an update is made, the current state of the belief is evaluated and then the minimum change is made to bring it into line with the constraint. If a belief is vivid, this cannot be changed. If it is aware-only, it can be made unknown by adding a justification for it consisting of a new assumption, or vivid by adding one with an empty antecedent list. An unknown belief can be set to vivid the same way, as the ATMS removes environments which are supersets of others, so the new empty environment will cause environments containing assumptions to be removed. If an unknown belief is to be made only-aware, it could simply be marked as a contradiction in the ATMS, but this would prevent that belief from becoming anything but only-aware in the future, so instead each group of assumptions constituting one of its environments is contradicted directly, causing the environments to be removed without preventing new ones appearing later.

It can be seen that the propagation process of the ATMS will cause the status of related beliefs to change according to the description in section 3.1, with the exception of the propagation backwards of a change from unknown to aware-only. This change comes about through the removal of the appropriate environments by the ATMS contradiction mechanism.

4.4 A Map Task example

For an illustration of how this system updates the value of nodes, picture a map task situation in which the instruction giver, let us say Doris, wants to use a particular landmark, say, the "palm beach", in her description of the route to the follower, whom we shall call Fred.[3] We will describe the nodes that are created in Doris' belief model which represent Fred's acquaintances with the concepts involved. The palm beach has not yet been mentioned in the conversation, so in Doris's belief model, Fred's degree of acquaintance with it is unknown. The node representing Fred's state of mind about the palm beach has a justification from its own "vivid" assumption. The basis of the assumption is that it is likely from Doris' point of view that the palm beach appears on Fred's map, but she does not know for certain that it does. In this example we allow the route giver to refer to landmarks which she assumes the route follower to have without asking questions first, though we could set it up so that any state of the partner's was questioned unless its node's label included the empty environment, i.e., was definitely vivid. Figure 2 illustrates the nodes and their states at this point. The ellipses represent nodes in the database; the small rectangles attached to them represent individual justifications for the node, the antecedent nodes being linked to them by arrows.

Doris refers to the palm beach with an utterance of type *inform*, whose English equivalent would be something such as "The first section goes to the left of the palm beach". The effect of making this utterance on Doris' own belief state is evaluated by looking up the condition in table 1 and applying the knowledge conditions described in section 3.4. The result is that she now believes Fred to know

[3]Here we are not intending to refer to the communicative postures assigned to the agents with these names in Carletta's JAM examples.

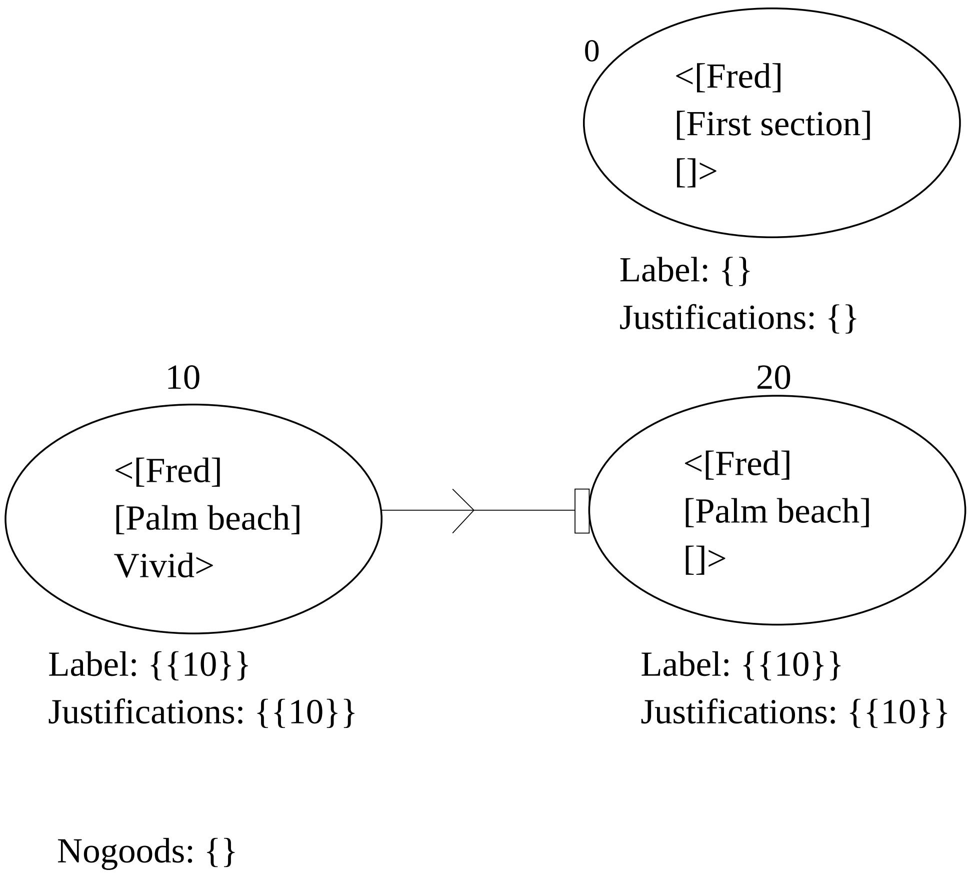

Figure 2: Assumption and consequent node for Fred's acquaintance with the palm beach.

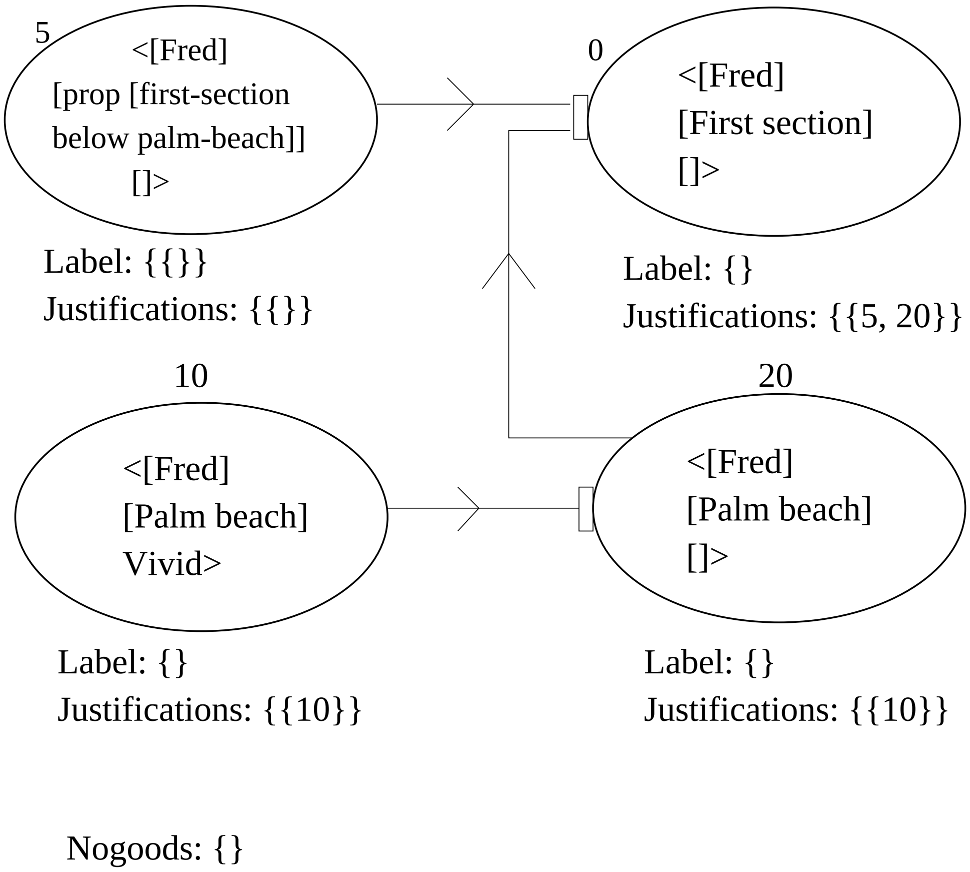

Figure 3: State of the same nodes after Fred indicates failure to understand description.

the proposition corresponding to the description (plus deeper nested beliefs if these are being explicitly modelled). She also adds the justification corresponding to the description, from Fred's acquaintance with the palm beach and the proposition for the description, to his previously only-aware acquaintance with the first section. As a result of this, the assumption that the palm beach is vivid for Fred is propagated up to the first section node, whose status for Fred in Doris' model becomes unknown.

However, the reply is an utterance of type *dont-understand*: "I do not understand". The effect of this on Doris's mental state is to set her belief about Fred's acquaintance with section one to only-aware. Therefore any group of assumptions that allowed Doris to believe that Fred had it vivid form a contradiction. In this case there is only one group of one assumption, i.e., that Fred has the palm beach on his map. It is added to the contradiction list, and all environments containing it are removed from the system. This includes the environment of the node corresponding to Doris' belief about Fred's state of awareness about the palm beach. The node now has no environments, reflecting its only-aware status. Figure 3 illustrates the nodes and their states at this point.

Suppose Doris tries to repair her plan by giving a description of palm beach in terms of two other landmarks, let us say the waterfall and the swamp. Uttering the description, "The palm beach is between the waterfall and the swamp" will result in knowledge of the proposition becoming mutual down to the depth limit, plus a new justification being added to Doris's database, as follows: Fred's acquaintance

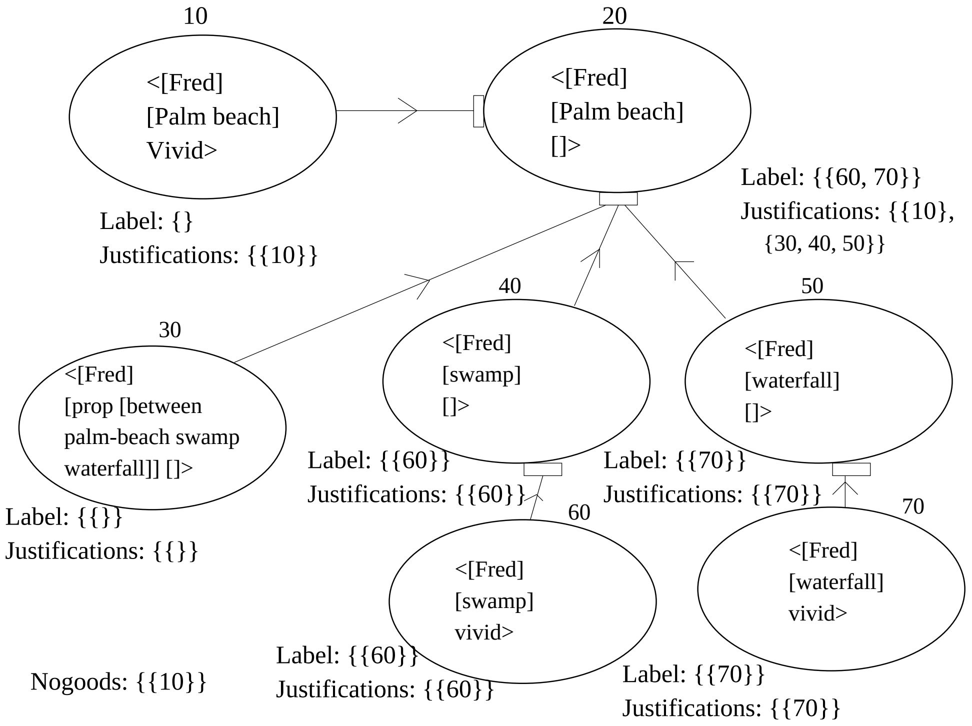

Figure 4: Part of Doris' belief network after she tells Fred the second description.

with the two new landmarks, plus his knowledge of the propositional content of the description, imply a similar degree of acquaintance with the palm beach. His acquaintance with each of the the two new landmarks is justified by a "vivid" assumption if it has not yet been mentioned. The ATMS will propagate the two "vivid" assumptions through the justification to form a new label for the node corresponding to Fred's knowledge of the palm beach. It is once again regarded as being in an unknown state at this point, because assumptions must be made for it to be thought vivid. Figure 4 illustrates the nodes and their states at this point (the nodes relating to the first section have been omitted from this figure for clarity).

Now Fred may once again reply "I do not understand", indicating that he is aware-only of one of the two new landmarks. In this case, the pair of assumptions form a new contradiction, and Fred's acquaintance with the palm beach is once again aware-only, although his acquaintance with each of the new landmarks is still unknown since either of the two assumptions is tenable in the absence of the other one. Alternatively, Fred may reply "OK" indicating that he has the subject of the description vivid. In this case, a new justification from an empty list is added for the node corresponding to his acquaintance with the described landmark (the palm beach), so it becomes a premise. Its empty environment will propagate up to the node for Fred's acquaintance with the first section, resulting in that becoming vivid too. At this point, if beliefs at the second level of nesting are being explicitly modelled, the agents may need to exchange a closing dialogue to establish mutual vivid acquaintance with the first section down to the required level.

5 Future directions

So far, we have implemented the belief model as described above, forming the 'fast' component of BELMOD's hybrid belief model. The next step in the BELMOD project is to extend the system to cope with domains in which references can be ambiguous, i.e., in which one agent can have two landmarks which might be the referent of a given term. The intention behind this change is to allow dialogues to be simulated in which agents proceed for some time before realizing that there is a mismatch, thus adding to the complexity of the reasoning that must be carried out to recover fully from the failure.

This will require another class of proposition to be used, that belief that an object referred to by the partner is the same as one with a similar name on the agent's own map. A new class of assumptions will be introduced to justify such propositions where there is only one apparent referent for some mention of a landmark. Beliefs that depend only on these assumptions will be treated as vivid, but will be defeasible by the removal of that assumption. The goal of this phase of the project will be to categorise the sorts of problem that can arise in this context in terms of the depth of nesting of beliefs required for their recognition and for their correct solution.

After that, development of the unrestricted belief model can commence. Simply extending the limits on nesting from the 'fast' component will not result in a viable system; it is expected that a set of heuristics for guiding reasoning will be required. When reasoning about misunderstandings that originated a number of turns previously, agents may have to compare the times at which different assumptions were added to the network. The need for this will be briefly outlined in section 5.1. It is also expected that the identities of assumptions, rather than just the presence or absence of assumptions of a particular type, will become important, particularly if the planner will have to be made capable of using utterances to achieve belief changes that follow indirectly from them, as CYNIC's planner does [22]. A means of doing this using counterfactual assumptions in the belief model is described in section 5.2. On a different note, it may be possible to make the 'fast' component more humanlike without weakening its appeal to logic, by replacing the standard ATMS with Kelleher's LazyRMS [13]. This will be discussed in section 5.3.

5.1 New reasoning about past events

The belief model as it currently exists only holds information about agents' current states of belief. This is adequate for the example dialogues which can curently be modelled, in which any failure to understand a description correctly causes an immediate repair or replan dialogue. Its limitations can be seen in the way contradictions are handled, which was described in section 4.3. When a description fails, the assumptions of vividness for the referents, rather than the nodes which are queried for their status, are marked as contradictions. This is because what is actually being contradicted is the *initial belief that these were on the addressee's map* rather than the addressee's acquaintance with them, which may change later.

When repairs remote in time from the original dialogue are carried out, one agent may have to reason about what another agent thought at several different points in time. In most cases, the addition and removal of the identity assumptions mentioned above will ensure that at any time only landmarks whose descriptions make use of references currently believed to be valid are marked as vivid. However, if as the result of a clarification dialogue the intended referent of an expression changes, and then there is a problem involving a description containing this expression, it will be necessary to distinguish between beliefs relating to the expression before and after the change was made. This could be done with the addition of another assumption class, for use when nested beliefs become vivid as the result of "OK" utterances. If the subject of the description being replied to were justified with an assumption containing a sequence number instead of as a premise, its status could still be returned as vivid by the belief model until a repair invalidating a description is made.

No exotic temporal logic will be needed here. It would appear to be safe to use a straightforward situation calculus as advocated for planning systems by McCarthy [15] because nothing changes in the domain except as the result of utterances, which are discrete and nonoverlapping. This would be an extension of the current status, where only two times are reasoned about — the current time and the initial state — and hence in keeping with the objectives of BELMOD.

5.2 Counterfactual assumptions and utterance planning

The belief model envisioned for the BELMOD project plays no part in the generation and recognition of plans — it merely keeps track of what is currently believed as a result of utterances that have taken place. The planning and plan recognition system make use of a separate model of the effects of utterances and the propagation of beliefs, which is encoded in the form of a list of plan operators. Some of these correspond to utterances, others to belief propagation steps. The belief model is only consulted to determine if a particular operator is applicable, or if a plan has succeeded.

As part of the aim of the project is to produce a model of the reasoning behind trouble-free dialogue which is consistent with human reasoning styles, it makes sense to try to use the same rules for utterance effects and belief propagation throughout the model. This could be done fairly simply; the plan operators could be generated in more or less their current form in much the same way as new beliefs are generated from the list of conditions for utterances, to express changes at each level of nested belief down to the current limit.

This would still leave JAM's control system for plan generation in place, which uses a plan tree data structure separate from the belief model. Plan recognition builds a similar tree structure to model the planning agent. However, both these structures can be subsumed by the belief model's ATMS network through the use of *counterfactual assumptions.* This method of planning and plan recognition was pioneered in CYNIC [22] but could easily be used in conjunction with a restricted belief model such as that described here.

We would need to introduce a class of propositions corresponding to the occurrence of utterances, and a class of assumptions which we shall call "done" assumptions.[4] When planning for a goal, an agent generates all the antecedents for that goal, including propositions for utterances that bring it about, and adds them to the belief model with the justifications that cause the goal to follow from them. Each antecedent node, if not already a premise, is justified with a "done" assumption. The goal node will have one or more environments consisting of "done" assumptions, but is not actually believed, i.e., these are counterfactual. The control system picks an environment in which few things have to be "done", adopts one of them (not an utterance) as a subgoal and applies the same procedure. When the original goal has an environment in which all the "done" assumptions justify utterance propositions, the goal can be achieved by making those utterances. As each is uttered, its proposition is made a premise, so finally all "done" assumptions are removed from the goal node environment that was chosen.

This technique allows agents to plan for their goal to become definitely vivid or merely unknown — in the latter case the plan may not succeed but can be repaired with minimal wastage of reasoning effort. An analogous procedure exists for plan recognition, involving reasoning from causes to possible intended effects.

[4] These are equivalent to CYNIC's "doable" assumptions, named differently because in CYNIC they only result in action when combined with an explicit want for the operation.

5.3 The Lazy ATMS: more restrictions on reasoning

One criticism that can be levelled at the BELMOD project in its current form is that while the beliefs that are represented are restricted to those that humans can deal with easily, the ATMS will propagate a change in belief state immediately to any other belief it affects. This is very different to the way human believers have been shown experimentally to react to new information; for a discussion of propagation in human belief updating see Harman [11]. Essentially, people tend not to change their beliefs unless they have actual cause to question them, in which case the justifications on which they depend may be reexamined.

There is a similarity between Harman's model of human belief updating and the operation of Kelleher's LazyRMS [13]. This was developed in an attempt to reduce the amount of processing required in reason maintenance. It carries out the same job as DeKleer's ATMS, but only updates a node's label when it is explicitly queried by the problem solver. All that happens when a new justification is added is that nodes which may be affected by it are flagged "out-of-date". When an out-of-date node is queried, a search is made backwards along its justification links until nodes are reached that are not out of date, then the labels of these are propagated forwards in the standard way.

This ATMS offers a limited improvement in performance over the standard ATMS for the belief model described here, as many belief schemata are queried to find instances of beliefs that would be useful in a plan. Some "human-like" reasoning could be simulated by allowing some plan operators to be used after only examining the current labels of their condition beliefs rather than requesting full updates; only when failures occurred would updating be done. This is still some way from modelling the observed behaviour of human believers, who according to Harman are lazy to some extent about belief updating even when they do have direct cause to reconsider a consequence.

6 Conclusion

The overall goal of this project is to draw a connection between the findings of psychologists who tell us which sorts of problems the human mental equipment is better or worse suited to, and those of logicians who provide us with techniques for solving arbitrarily complex problems of a given class. The result should be a system that can cope with dialogue problems with a similar pattern of performance to that shown by humans, but which is demonstrably linked to an underlying logic, and applies a subset of the inferences sanctioned by that logic whether modelling everyday, simple belief updating or the more laborious reasoning associated with conscious problem-solving processes in humans.

A humanlike belief model makes use of many abbreviations, conflations and arbitrary limits on complexity. It has been found that representing concepts and agents' degrees of acquaintance with them is a useful shorthand for a lot of propositional information. It has also been found that if a nested belief is "unknown", the level of nesting at which knowledge fails is unimportant from the point of view of planning.

The ATMS provides a natural representation for implications, as well as an efficient mechanism for belief state propagation. It forms part of a nonmonotonic reasoning system in epistemic logic, which can be constrained to work with only the type of expressions which humans find easy to reason with. However, the formats described here for assumptions and nested beliefs can be extended to include more complex beliefs, counterfactual reasoning and reasoning about situations. At the same time, the ATMS's operation can be altered to provide a better model of human belief updating.

References

[1] Anne H. Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. The HCRC map task corpus. *Language and Speech*, 34(4):351–366, 1991.

[2] D. E. Appelt. *Planning English sentences.* Cambridge University Press, 1985.

[3] Ruth M. J. Byrne. The model theory of deduction. In Andrew Rutherford Yvonne Rogers and Peter A. Bibby, editors, *Models in the Mind.* Academic Press, 1991.

[4] Jean C. Carletta. *Risk-taking and Recovery in Task-Oriented Dialogue.* PhD thesis, Edinburgh University Department of Artificial Intelligence, 1992.

[5] Johan de Kleer. An assumption-based truth maintenance system. *Artificial Intelligence*, 28(2):127–162, 1986.

[6] Daniel C. Dennett. *The intentional stance.* MIT Press, Cambridge, MA, 1987.

[7] Richard Double. The case against the case against belief. *Artificial Intelligence*, 13(1):81–132, 1980.

[8] Ronald Fagin and Joseph Y. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39–76, 1988.

[9] Peter Gärdenfors. *Knowledge in Flux.* MIT Press, Cambridge, MA, 1988.

[10] Matther L. Ginsberg. Multivalued logics: A uniform approach to reasoning in artificial intelligence. *Computational Intelligence*, 1988.

[11] Gilbert Harman. *Change in View: Principles of Reasoning.* MIT Press, Cambridge MA, 1986.

[12] George Houghton. *The Production of Language in Dialogue: A Computational Model.* PhD thesis, University of Sussex, 1986.

[13] Kelleher and van der Gaag. The lazyrms - avoiding work in the atms. *Computational Intelligence*, (3), 1993.

[14] H. J. Levesque. A logic of implicit and explicit belief. In *Proceedings of AAAI-84*, 1984.

[15] John McCarthy. Programs with common sense. In Marvin Minsky, editor, *Semantic Information Processing*, pages 403–418. MIT Press, 1968.

[16] John McCarthy. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1):27–39, 1980.

[17] Robert C. Moore. A formal theory of knowledge and action. In Jerry R. Hobbs and Robert C. Moore, editors, *Formal theories of the commonsense world*, pages 319–358. Ablex, Norwood, New Jersey, 1985.

[18] Ehud Reiter. Generating descriptions that exploit a user's domain knowledge. In Robert Dale, Chris Mellish, and Michael Zock, editors, *Current Research in Natural Language Generation*, pages 257–286. Academic Press, London, 1990.

[19] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1):81–132, 1980.

[20] Keith Stenning. Distinguishing conceptual and empirical issues about mental models. In Y. Rogers, A. Rutherford, and P. Bibby, editors, *Models in the Mind*. Academic Press, 1991.

[21] Keith Stenning and J. Levy. Knowledge-rich solutions to the 'binding problem': some human computational mechanisms. *Knowledge Based Systems*, 1(3):143–152, 1988.

[22] Jasper Taylor. *A Multi-Agent Planner for Modelling Dialogue*. PhD thesis, School of Cognitive and Computing Sciences, University of Sussex, forthcoming.