Belief Revision and Dialogue Management in Information Retrieval

Final Report on MRC Grant SPG 8930752
"Testing a theory of belief revision: human computer collaboration for information retrieval"

Brian Logan, Steven Reece, Alison Cawsey,
Julia Galliers and Karen Sparck Jones

Computer Laboratory University of Cambridge,
New Museums Site, Pembroke Street, Cambridge CB2 3QG
(Contact address: ksj@cl.cam.ac.uk)

May 1994

Acknowledgements

This work was supported by the UK Tri-Council Initiative on Cognitive Science and Human-Computer Interaction, under MRC Grant SPG 8930752: we are very grateful for this funding. We are also grateful to Professor N. Belkin of the School of Communication, Information and Library Studies, Rutgers University, for supplying transcript data for user-librarian dialogues recorded by himself, Helen Brooks and Penny Daniels; to Professor Barbara Grosz of Harvard University for discussions; and to Professor R.M. Needham and Dr D.F. Hartley of the Computer Laboratory for the provision of computing and other facilities.

Present addresses: Dr J. R. Galliers and Dr K. Sparck Jones, the Computer Laboratory; Dr A. Cawsey, Department of Computing Science, University of Glasgow; Dr B. Logan, Martin Centre for Architectural and Urban Studies, Department of Architecture, University of Cambridge; S. Reece, Robotics Research Group, Department of Engineering Science, University of Oxford.

Abstract

This report describes research to evaluate a theory of belief revision proposed by Galliers in the context of information-seeking interaction as modelled by Belkin, Brooks and Daniels and illustrated by user-librarian dialogues. The work covered the detailed assessment and development, and computational implementation and testing, of both the belief revision theory and the information retrieval model. Some features of the belief theory presented problems, and the original 'multiple expert' retrieval model had to be drastically modified to support rational dialogue management. But the experimental results showed that the characteristics of literature-seeking interaction could be successfully captured by the belief theory, exploiting important elements of the retrieval model. Thus though the system's knowledge and dialogue performance were very limited, it provides a useful base for further research. The report presents all aspects of the research in detail, with particular emphasis on the implementation of belief and intention revision, and the integration of revision with domain reasoning and dialogue interaction.

Contents

1 Introduction 5
1.1 Background 5
1.2 Project Work and Outcomes 10
1.3 Organisation of the Report 12

2 A Model of Belief and Intention Revision 13
2.1 Belief Revision in AI 13
2.1.1 Uncertain Belief 15
2.1.2 Foundation and Coherence Theories 18
2.2 A Model of Autonomous Belief Revision (ABR) 20
2.2.1 A Logical Framework for ABR 21
2.2.2 Ordering Relations for ABR 22
2.2.3 Increased Coherence 23
2.2.4 Endorsements and Minimal Change 25
2.3 Revising Intentions 27
2.4 Embedding Inference and Planning within the ABR Framework . 29

3 Testing the Model: The Information Retrieval Task 31
3.1 The Information Retrieval Task 32
3.2 The BBD Model 33
3.3 Problems with the BBD Model 36
3.4 Simulating the BBD Model 39
3.5 Distributed Architectures for Information Retrieval 40
3.6 Revising the BBD Model 43

4 The Library Dialogues 44
4.1 Belief Revision in Communication 44
4.2 Speech Acts 46
4.2.1 Speech Acts for Planning 47
4.2.2 Speech Acts for Dialogue 49
4.2.3 Successful Speech Acts 51
4.3 A Typology of Communicative Outcomes 52
4.3.1 Case Analysis 54
4.3.2 Successful Plans 56
4.4 Maintaining Coherence in Dialogue 57

5 Implementing Belief and Intention Revision **60**
- 5.1 Agent Architecture . . . 60
- 5.2 Representing the Cognitive State . . . 62
  - 5.2.1 Belief-Type Attitudes . . . 63
  - 5.2.2 Intention-Type Attitudes . . . 66
  - 5.2.3 Database . . . 67
- 5.3 Deriving the Belief State . . . 68
  - 5.3.1 Computing the Preferred Sets . . . 70
  - 5.3.2 The Agent's Pervasive Beliefs . . . 73
  - 5.3.3 Commitment to Pervasive Beliefs . . . 75
- 5.4 Deriving the Intention State . . . 78
  - 5.4.1 Computing the Preferred Sets . . . 79
  - 5.4.2 The Agent's Pervasive Intentions . . . 80
  - 5.4.3 Commitment to Pervasive Intentions . . . 81
- 5.5 Inference . . . 83
  - 5.5.1 The Rule Binding Algorithm . . . 85
  - 5.5.2 Defeasible Inferences . . . 88
  - 5.5.3 The Inference Algorithm . . . 91
- 5.6 Planning . . . 92
  - 5.6.1 Action Schemata . . . 93
  - 5.6.2 Planning Rules . . . 94
  - 5.6.3 Minimal Plans . . . 95
  - 5.6.4 Alternative Plan Branching Points . . . 96
- 5.7 The Agent Action Cycle . . . 98
  - 5.7.1 A Blocks World Example . . . 100
  - 5.7.2 Revising the Plan . . . 103

6 Implementing the BBD Architecture **105**
- 6.1 The Functional Experts . . . 105
  - 6.1.1 The Problem Description . . . 106
  - 6.1.2 Retrieval Strategy . . . 110
- 6.2 System Architecture . . . 112
  - 6.2.1 Task Decomposition . . . 112
- 6.3 Information Retrieval Rules . . . 117
  - 6.3.1 Problem Description Rules . . . 118
  - 6.3.2 Domain Rules . . . 120

7 Implementing Dialogue **121**
- 7.1 Action Schemata . . . 122
  - 7.1.1 The Tell Schema . . . 122
  - 7.1.2 The Adopt Schema . . . 123
  - 7.1.3 The Infer Schema . . . 124
- 7.2 Dialogue Rules . . . 124
  - 7.2.1 Ascription Rules . . . 125
  - 7.2.2 Adoption Rules . . . 126
  - 7.2.3 Prediction Rules . . . 126
  - 7.2.4 Desire Rules . . . 133

7.2.5 Planning Rules 134
7.3 Agent Action Cycle Revisited 135
7.4 An Example 136
7.5 Stability of Intentions in Dialogue 139

8 Testing 140
8.1 Belief Revision in the Library Dialogues 141
8.2 Simulation 144
8.3 Results of the Tests 171

9 Problems and Future Work 172
9.1 The Problem of Computational Complexity 172
9.1.1 Definite Beliefs 176
9.1.2 Focus Set and Focussing Algorithm 177
9.1.3 Possible Solutions to the Complexity Problem 180
9.2 The Problem of Increased Coherence 181
9.3 The Problem of Prediction 184
9.3.1 Predicted Intentions 186
9.4 The Problem of Communicated Commitment 187
9.5 The Problem of Dialogue Management 188

10 Conclusions 190
10.1 Project Overview 190
10.1.1 Work Done 191
10.1.2 System Design 192
10.2 Major Issues and Future Research 193
10.2.1 Belief Revision Issues 193
10.2.2 Future Work in Belief Revision 194
10.2.3 Information Retrieval Issues 195
10.2.4 Future work in Information Retrieval 196

A An Example Transcript of a User Librarian Dialogue 204

B A Planning Demonstration 210
B.1 Introduction 210
B.2 Simulation 210

C A Multi-Agent Implementation of the BBD Model 219
C.1 Introduction 219
C.2 Research Issues in Multi-Agent Architectures 220
C.2.1 Communication between Agents 220
C.2.2 Resolving Conflicts 221
C.3 A Multi-Agent Implementation of the BBD Model 222
C.4 Conclusions 224

Chapter 1

Introduction

This report describes work carried out under grant number SPG 8930752: "Testing a theory of belief revision: human-computer collaboration for information retrieval", funded by the Tri-Council Initiative on Cognitive Science and Human-Computer Interaction.

The objective of the project was to test Galliers' computational theory of belief revision in the context of the interaction between a library user and a librarian. In doing so it combined a general theory of cooperative problem-solving with a specific task about which a good deal is already known from other studies. It is hoped that the project will both establish the theory as better than those hitherto proffered, and lay a foundation for an eventual real automated library interface. This chapter summarises the problems and ideas taken as starting points for the work. The main body of the report shows how we have developed, computationally implemented, and tested these ideas.

1.1 Background

The general framework of speech act theory appears well suited to the design of natural language interfaces to task systems, where it is necessary to recognise intentions and their effects on belief states. But the particular theories that have been developed so far (see, for example, Cohen, Morgan and Pollack (1990)) have relied on some simplistic assumptions, notably that the participants' utterances are sincere and reliable and that their beliefs are persistent, and therefore especially in relation to tasks, that the participants are mutually benevolently disposed, and are engaged primarily in conveying and adopting previously formed plans which do not conflict with existing beliefs. Galliers' theory starts from the position that there is necessarily much less clarity and certainty in communication, and that neither participant is a vessel waiting to be filled, so engaging in dialogue involves a much more continuous and pervasive process of belief assessment and revision supporting mutually accommodating joint plan determination and execution.

The need for the proposed lack of imposed 'helpfulness' and associated assumptions about agents as reliable and informed and hence 'knowing what they are talking about', is because multi-agent environments are 'open environments'

(Hewitt 1986). No agent can know everything about its environment. No agent can *know* another's belief states. Such a state of affairs would not even be desirable as there would be unnecessary bottlenecks of information processing (Hewitt 1986, Gasser et al. 1989, Galliers 1990). Hence the use above of words such as 'presumed' and 'predicted' in phrases referring to others' mental states. This lack of complete information, together with the dynamic nature of both the physical and multi-agent world, is the background within which belief revision is viewed as fundamental to rational interaction. It is also the background to collaborative dialogue as a series of negotiated or mutually accepted revisions of belief.

Galliers' theory thus treats dialogue, which is normally task oriented, as strategically driven cooperative problem solving, motivated by belief revision. Each participant's actions stem from the revision of their own beliefs about the subject of the discourse, and about the other participant's views, which is stimulated by the other participant's actions; and each participant's actions are also intended to revise the other's beliefs about the subject and the actor. Each participant seeks to maximise the dialogue outcome from their own point of view by achieving a desired mental state in the other. This has to be done by negotiation, to mutual satisfaction. Thus even in the 'baseline' case where the specific intended task outcome is known to the dominant participant, interactive dialogue implies changes of belief in the participants. Belief revision is then, more importantly, much more central to the many cases where the desired outcome not merely is not, but cannot be, specified in detail in advance, so the dialogue defines the outcome.

This revision, determining and determined by each dialogue utterance, is an autonomous process in each participant; but this autonomy is mutually recognised, so while utterances cannot have guaranteed effects, dialogue as a whole is jointly controlled. Modifying the participant's web of beliefs is not confined to simple changes treated as local phenomena, like true/false switches or data additions. The theory presented in Galliers (1989) offers an account of belief revision as a principled process grounded in explanatory and justificatory assumptions. These assumptions underlying the reasons for beliefs are more critical than the reasons themselves because the explanations and justifications they embody have the coherence needed to support consistent beliefs. The individual's response to possible changes of belief thus ultimately depends on the nature and strength of the underpinning assumptions, and in seeking to modify another's beliefs the individual addresses what he sees as the assumptions underlying the other's beliefs.

Assumptions are differentiated in status by being endorsed (Cohen 1985) in various ways, according to their sources; and means are provided, by a set of general heuristics, for evaluating endorsed assumptions individually and in combination, in order to determine whether dependent beliefs should be revised. The combined assumptions grounding the explanatory and justificatory coherence relations for each belief are compared according to their endorsements: more firmly grounded beliefs are harder to revise. This is a non-numeric approach to strength of belief, where the more persistent belief is retained in the context of specific challenge.

For instance as a very simple example, consider the user – librarian dialogue:

U1 I want a book on house plants.

L1 You want something on growing them?

U2 I seem to be allergic to a plant in our house.

L2 Oh I see, something on common house plants and allergies?

The librarian's L1 is based on the assumption given by, i.e. 'told', in U1, plus a general assumption linking house plants with growing them. But the new told assumption from U2 generates a conflicting inference for the librarian, that the user wants something on house plants and allergens. The conflict is resolved to generate L2 because told assumptions are better grounded than generalities. The user has himself produced U2, not just to supply more information, but to counter the assumption he infers underlies L1 and to provoke revision of the librarian's beliefs, by relying on the greater weight dialogue participants give to told assumptions. In the exchange the theory is accounting for utterances in relation to belief contexts which are continuously tracked and evaluated.

In other speech act theories proposed so far (see, for example, Cohen et al. (1990)), the effect of U2 would be a mutual belief shared by user and librarian that the user believes he wants to know about house plants and allergies. Transferring this belief to the librarian, so she also believes it of the user, depends on its fitting in, or being consistent with, all of the librarian's other beliefs. But as L1 shows this is not in fact the case: the librarian's belief about the user is quite different. A system which allowed non-monotonic reasoning could cope with this: the new evidence in U2 would provoke a replacement for the earlier belief and the librarian's whole belief set would be maintained to suit. But the system would always assume, too strongly, the incoming evidence was true: thus if the user then suggested, as U3, that a health book on allergies in general might be appropriate, this would automatically stimulate another round of modification to the librarian's belief set. Galliers' theory is non- monotonic, unlike the earlier ones, but it also allows reflection as a meta-reasoning process applied to incoming evidence and existing beliefs, prior to revision. This means that it would still be possible for the librarian to persist in believing that a book on house plants and allergies would be best for the user.

Galliers' approach to belief revision, building on aspects of the work of Harman (1986) and Cohen (1985), is more broadly based and realistic than those hitherto used in AI. With a small but still discriminating range of possible endorsements, revision has a wider interpretation than simple contradiction or variation in (well-defined) specificity; and more importantly does not rely, as Dempster/Shafer for example does, on numeric measures of certainty and on mechanisms for manipulating these, without tackling the key issue of the sources for the numbers. In the approach advanced here, in contrast, beliefs are qualitatively founded .

Literature seeking in its many forms is a common activity sharing with other types of information gathering the basic property that someone wants to get to know something they do not already know, in the context of what they think they do already know. But more particularly, it is natural for the literature seeker, not knowing exactly what to look for or precisely where, to ask for help from the librarian.

The specific type of library interaction we have worked with is that between the user seeking document references from an on-line service like DIALOG, and the skilled intermediary, familiar with the available databases, forms of indexing, and types of retrieval strategy, who normally conducts the actual search. ‘Automating the librarian’ is thus automating the intermediary as a front-end system with an external natural language interface to the user and an internal connection with the backend service.

The interaction between user and intermediary is normally via a cooperative problem-solving dialogue of the sort envisaged in Galliers (1989). The user’s literature need, even if presented as well-defined, is usually inadequately formulated and expressed, fundamentally because the user is seeking information to remedy ignorance, but also because he is having to make his need explicit in a new context, and because this has to be characterised in a manner suited to searching, for instance by being translated into a set of index terms. The primary function of the dialogue is to identify the user’s need, so it can be transformed into a search request for submission to the back end. This identification is a mutual effort by the user and the intermediary because, though the intermediary is generally experienced in searching, she cannot know what the user knows about his own situation: the user determines and contributes this information, reacting to the intermediary as she develops her interpretation in the light of her professional knowledge. Both parties thus modify their respective beliefs to arrive at their mutually agreed outcome request.

We chose this task as a study context because the work of Belkin et al., Brooks and Daniels (hereafter BBD) (Belkin, Seeger and Wersig 1983, Brooks 1986, Daniels 1987) work provides a detailed analysis of the intermediary on which to build. This analysis is based on an attractive general theory of information-seeking interaction, and is itself a substantial contribution to our understanding of the librarian’s particular expert activity. BBD model the intermediary as a distributed expert system with some ten individual functional components addressing subtasks which together supply the information needed for the request. The User Model component, for instance, gathers information about such matters as the user’s educational level, the Problem Mode module information about the status of the background work stimulating the search, and the Problem Description component the characterisation of the need, while the Retrieval Strategy component builds the actual search request. These subsystems apply their specific knowledge, for example the Retrieval Strategy of the index language, but cooperate both locally by supplying information to one another (e.g. being a first year student in the User Model may imply wanting review articles in the Problem Description), and globally by making their respective contributions to the search request (e.g. wanting review articles, if confirmed for the Problem Description, could lead to a Retrieval Strategy searching on appropriate review journals).

The individual modules may be viewed generically as rule-based systems, but may vary considerably according to their specific tasks. For example Daniels envisages the User Model as instantiating hierarchically-organised frames covering, for instance, occupational status, and work goals motivating the literature search. These are supported, especially where the options are not obvious and limited, by rules: for example in relation to the user's experience of on-line searching, rules would be used to discover what databases the user is aware of, and how much he has used them, from which his level of experience can be inferred. The Retrieval Strategy component on the other hand, treating Pollitt (1986)'s system for cancer literature searching as an illustration, is primarily a set of rules for converting the user's choice of medical topics, expressed in his usual language, into appropriate search terms in the particular artificial indexing language used by the search service, and for grouping them according to their categories (e.g. drug name, cancer type) into a Boolean search formula (e.g. several drug terms are linked by disjunction, but conjoined with a cancer term).

The system as a whole interacts with the user to obtain information serving the different functions. For instance the example dialogue given earlier addresses the Problem Description function. If the user agreed with L2 he would revise his beliefs to replace those underlying U1; if he disagreed, remembering a particular plant, and offered U3 "Actually, it's a cactus", this would stimulate revision of the librarian's beliefs. However even if the user agreed with L2, the librarian, considering the Retrieval Strategy, might find there was no obvious index term corresponding to "house plant" but an assortment of candidates including 'Pot Plant', and so would have to interact further with the user on the Problem Description.

BBD's detailed studies of actual interviews show the dialogue as a sequence of foci, each represented by a subsequence of utterances concerned with a particular functional task. Focus choices are made by either party, and within the framework of a general shift towards the particular characterisation of the search request representing the overall task goal, there may be very varied patterns of focus movement addressing and readdressing individual tasks. These patterns clearly display a joint effort to establish mutual beliefs in subareas and in pursuing the consequences for others, subject to the general constraints of discourse management like maintaining cohesion, signalling shifts, etc, and of polite behaviour. Belief revision as a generic activity therefore takes the specific form, in task-driven dialogues like this, of 'finding-and-convincing' cycles. Each utterance, or contribution, represents a minimal find-and-convince cycle; the individual foci, and the discourse as a whole, larger embedding cycles.

The information retrieval task is an ideal testbed for evaluating Galliers' theory. The dialogue is task driven, but requires much more constructive interaction than TDUS (Grosz 1978) did, for instance. BBD's interviews provide reference data; and technical practice is well established and can be exploited to supply specific knowledge, for example about retrieval strategies. BBD's model, moreover, provides a useful analysis on which to build. At the same time, trying to apply their model is of value in its own right in laying the foundation for an automated intermediary, as the model needs detailed development, both as a whole and in specific areas, notably control. Though some very limited expert retrieval systems have been built (Pollitt 1986, Vickery et al. 1987), attempting to implement the model as such is new work.

1.2 Project Work and Outcomes

As stated above, the objective of the project was to test Galliers' computational theory of belief revision in the context of the interaction between a library user and a librarian. To this end, we have built a series of successively more capable (though necessarily still very simplified) versions of an 'automated librarian' in order to study Galliers' theory in a systematic way.[1] The final version of this system, described in detail below, consists of a knowledge base with task knowledge and inference rules, with an associated ATMS providing the mechanism for maintaining the ramifications of proposed changes of belief and also intention, operating under the overall control of a planner. Though BBD envisaged an architecture with totally independent specialised functional modules, we have adopted a cognitively more plausible approach with a single knowledge base which nevertheless allows for de facto clusters of functionally motivated knowledge. The great value of BBD's model is that it clearly exhibits the varied subtasks, with their distinctive specialist knowledge, involved in the intermediary's activity. As the project progressed it became clear that the original goals were extremely ambitious and it has proved necessary to revise these in some instances. At the same time, considerable development of the original theories has been necessary.

We have not built, nor did we ever intend to build, a complete implementation of the BBD model. Not only is capturing the professional expertise of a librarian currently well beyond the state of the art, the detailed operation of the underpinning theory of belief revision is itself the subject of the research. We have therefore limited the intermediary system to the few most critical functions, namely Problem State, Problem Mode, User Model, Problem Description, and Retrieval Strategy. However we believe this is sufficient to investigate the effects of functional needs, and their relationships, on the manipulation of belief and conduct of dialogue.

Nor have we attempted to implement an actual language processor, choosing instead to work with content representations in propositional form, of a well-established kind. This is because we wanted to focus on belief revision and the structural organisation of dialogue, rather than on the finer-grained, especially pragmatically- relevant, features of its linguistic expression, which require language processing research in their own right. We did not believe, either, that it would be useful to engage in any routine language processing, or sensible to devote specific effort to extending processing along lines already being studied elsewhere, for instance to deal with propositional attitudes. We have therefore simulated input and output processing, assuming the kind of framework being developed by SRI Cambridge (Alshawi 1992), in order to concentrate on the propositional content of speech acts associated with individual utterances, and on their manifest structural roles in the dialogue. Thus at the discourse level

---

[1] Note that this report describes the *final* version of the system, and in many cases the description given below conflicts with that in earlier project papers.

we have assumed an ability to manage discourse organisation cues, for example topic shift or framing expressions.

The focus of the research has therefore been first, on the way belief revision works and second, on the closely related problem of control. This was not defined by BBD for the retrieval task and had to be developed, both for the internal system operations and external dialogue, in a way which meshed properly with the working of the belief revision mechanism. This emphasis is reflected in the type of evaluation we have been able to perform. We have not in this research, because everything has had to be ruthlessly simplified compared with library reality, engaged in any direct operational evaluation. We have however, attempted to simulate some portions (or analogues varying their subject matter) of the interviews collected by BBD.

This work can been seen as basic research aimed at a challenging and necessarily very long-term goal, automating the librarian. While the research is presented here as directed towards document retrieval, it also seeks to contribute to artificial intelligence as a whole. Thus while from one point of view the aim is to apply artificial intelligence ideas to information retrieval, from another information retrieval provides a valuable study context for modelling the way any agents adopt or change their beliefs about the world, particularly through engagement in dialogue.

The work done under the project has demonstrated that Galliers' theory in its most important aspects and extended to handle intentions as well as beliefs can be implemented, and that it can constitute an effective underpinning for cooperative task-directed dialogue. In particular we have shown that exploiting qualitative endorsements for belief revision leads to dialogue interchanges with the same properties as real ones. We have not however been able to demonstrate the value of one component of the theory, namely that emphasising connectivity between beliefs; and we cannot emphasise too strongly that our test dialogues are very modest indeed compared with the real thing. Moreover while we have implemented the entire apparatus, and carried out many individual tests, the intrinsic complexity of the system means performance is slow. At the same time from the retrieval point of view, while the project has shown that modelling the information intermediary as an agent applying belief revision to specialised functional knowledge is appropriate, the problems of implementing the full BBD distributed model while maintaining rational dialogue control meant that we had to adopt a more straightforward model, and suggest that the original BBD model is fundamentally flawed.

We nevertheless believe that overall, the system described below represents a considerable advance in combining a general theory of cooperative problem solving with a specific task about which a good deal is known from other studies. It both helps to establish Galliers' theory as one with advantages over previous ones and begins to lay a foundation for an eventual interactive library interface.

1.3 Organisation of the Report

The report is organised into three main parts: theory, implementation, and testing and evaluation. The earlier background chapters largely reproduce material from previous project publications, included here for completeness and reference, in some cases also with significant modification. The later, more substantial chapters on the detailed project research are new.

In chapter 2 we outline Galliers' theory of belief revision as it was when we started the project, and also extend it to cover intentions and planning and inference, while in chapter 3 we describe the information retrieval task and present the BBD model for the intermediary. Chapter 4 introduces our treatment of speech acts and dialogue communication, where we relate relatively conventional ideas about dialogue to our own belief revision situation. The next three chapters provide the detailed account of our implementation of belief and intention revision in chapter 5; of our modified BBD architecture in chapter 6; and of dialogue management in chapter 7. The tests we have done are summarised in chapter 8, with four extended examples. Chapter 9 discusses the problems we have encountered with the various aspects of belief and intention revision and their interaction with dialogue communication. In conclusion in chapter 10 we assess the project as a whole.

Chapter 2

A Model of Belief and Intention Revision

In this chapter we describe a model of autonomous belief and intention revision (ABR) based on Galliers' work (Galliers 1989, Galliers 1991, Galliers 1992) that we have used for the project. This discriminates between possible alternative belief sets in the context of change and determines preferred revisions on the basis of the relative persistence of competing cognitive states. The project work has been the first serious test of Galliers' approach. It has needed some development, but we have found the theory appropriate and our studies therefore provide support for it as a theory of belief revision as well as as a base for modelling our information retrieval task.

We start by indicating the general issues of belief revision and types of approach to handling it. We then outline the theory of ABR and communication and discuss the problem of multiple alternative revisions. We present a logical framework for describing alternative theories of belief revision as various different ordering relations. A new ordering relation *mc* based on maximising coherence with other beliefs is then proposed as particularly suited to the requirements of ABR. On recognition of a speaker's intention via an utterance, the hearer applies this general principled basis of maximal coherence, not only to determine how to accommodate the new evidence, but also whether to accommodate it at all. In addition to this logical basis relying on connectivity and conservatism, the theory exploits the status of ground assumptions or endorsements for beliefs, giving a heuristic four-tiered ordering method which represents a blend between coherence and foundational theories of belief revision. In the last part of the chapter we consider the treatment of intentions as well as beliefs. The enlarged account of propositional attitudes which results provides a natural framework for inference and planning operations.

2.1 Belief Revision in AI

Much of the work in AI has been concerned with the design of automated systems which can plan and execute actions. These actions should be appropriate to the goals of the system, and its context or environment. In this sense they are rational behaviours, and the system a rational 'agent'. Being an autonomous as well as a rational agent, means having the ability to reason about relations and behaviour appropriate to self and the world, where the world includes other agents, who similarly reason in order to act autonomously and rationally. Primary in this reasoning are representations, beliefs or cognitive states generated through perception and inference, and related to desires and action according to the rules of rationality encoded into the system. But these cognitive states are inevitably constantly changing. The world is dynamic. Expansion and contraction of a belief set occurs as new data is perceived or inferred, and old data is lost over time or in the light of new evidence. Often expansion and contraction occur together. This is belief revision as described by Gärdenfors Gärdenfors (1988), namely changing one's cognitive state.

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

Belief revision in AI is associated with non monotonic reasoning; reasoning with inferences potentially withdrawable at some later stage. Doyle specifies two aspects of non monotonicity. Firstly, temporal non monotonicity in which attitudes are lost and gained over time, and secondly logical non monotonicity, in which unsound inferences are made as the joint product of sound reasoning, incomplete information and a 'will to believe' (Doyle 1988). An example of the latter is default reasoning.

Reason maintenance systems (RMS's) are AI's mechanisms for belief revision. They maintain consistent sets of beliefs in the light of new evidence. de Kleer's ATMS (de Kleer 1986a) maintains a number of consistent sets of beliefs appropriate to different assumptions or contexts, whereas the RMS's of Doyle (1979) and McAllester (1980) maintain just one.

New evidence may be accommodated into a belief set in alternative ways, and all of these maintain consistency. This is known as the 'multiple extensions' problem. For example, given

(a) $P \vee Q$
(b) $R \supset Q$
(c) $P \vee R$

and new evidence

$$\neg P \wedge \neg Q$$

incorporating the new evidence results in two logically equivalent extensions. These are (b) and the new evidence, or (c) and the new evidence, because

(a) is inconsistent with the new evidence, and *either* (b) *or* (c) are consistent with it, but not both (Rescher 1964) Alternatively again, the new evidence can be rejected if it is not assumed as 'truth' in which case the third possible extension is (a), (b) and (c). This latter alternative is more likely to arise as a result of communication as long as there are no assumptions regarding the communicator's omniscience or sincerity.

The only way of determining a preferred option from these kinds of possibilities is to incorporate some factor in addition to consistency. This factor should be the basis for ordering or prioritising the various alternative combinations of belief. The following section deals with various aspects, problems and solutions to this issue of preference in belief revision.

2.1.1 Uncertain Belief

In general, AI approaches to non-monotonic reasoning do not consider beliefs to vary in strength. All beliefs are equal for the purposes of inference and decision. Strength of belief is an accepted notion within inductive logic, however. It can involve acceptance theories comprising sets of confirmation functions and acceptance rules. Alternatively, Jeffrey's theory of partial belief (Jeffrey 1983) assigns degrees to beliefs as subjective probabilities computed using Bayes' theorem from a set of evidence hypotheses. Some AI approaches similarly assign numbers as probabilities to every belief. For example, certainty factors in expert systems, and Dempster/Shafer theory (Shafer 1976). In these cases, individual beliefs are differentiated in a manner which provides a ranking or order. The values assigned to new beliefs inferred from old or as evidence is gained or lost, reflects the combinations of values from their multiple sources.

Some AI approaches maintain beliefs as equal but differentiate the rules which generate those beliefs. This is a kind of preemptive approach whereby beliefs that would be inferred on the basis of less preferred rules are not inferred in the first place. Examples are systems employing prioritised competing default rules, such as in HAEL (Hierarchic AutoEpistemic Logic) (Konolige 1988). In this, the belief set is divided into a hierarchy of evidence spaces. Sentences in lower spaces are considered stronger evidence in being more specific, than those higher up. Inferences drawn from rules situated lower in the hierarchy override potential inferences higher up. An individual bat for example, can be inferred to fly even though the following two default rules contradict each other:

1. Normally mammals do not fly
2. Bats are mammals which do fly

The latter default rule relies on the more specific information, and is thus placed lower in the hierarchy. The bat as a mammal that cannot fly is not less preferred; it is never inferred in the first place. The issues involved in structuring priorities into the belief set in this way are discussed below.

In the example of HAEL above, priorities are structured into the belief system. The priorities are therefore reasoned with, but not something to be reasoned *about*. Alternative ordering schemes such as Gärdenfors' ordering of sentences in a belief set according to their ‘epistemic entrenchment’ (Gärdenfors 1988), Nebel’s ‘epistemic relevance’ (Nebel 1989, Nebel 1992), Doyle’s system of rational revision (Doyle 1991) and the *mc* relation described below (Galliers and Reichgelt 1990) provide a qualitative basis for assessment, as opposed to a fixed measurement or structure. Cohen (1985) deals explicitly with this issue; he discusses the importance of being able to reason about uncertainty.

The primary limitations of fixed structural ordering are inaccessibility and inflexibility. Doyle and Wellman (1989) refer to Konolige’s specification of the hierarchy in HAEL as ‘dictatorial’ in its inflexibility. It violates the modularity principle, critical to successful construction of complex structures such as commonsense knowledge bases. Modularity offers general rules of combination applied as the need arises, as opposed to employing a ‘sovereign authority’ whose task of resolving all potential conflicts is in any case infeasible with a large set of criteria. In addition, new criteria would necessitate a complete restructuring of the preference order. And with respect to the inaccessibility issue, Carver (1988) and Cohen (1985) argue that if it is impossible to reason why a particular fixed ordering has been set, it is impossible to revise satisfactorily and flexibly in the light of new evidence. This is especially the case with numeric representations.

Numeric representations of strength of belief are used with Bayes’ theorem to provide a means of computing the probability of a conclusion given the numeric probability or degree of belief attached to each evidence hypothesis. There are various problems with this ‘conditionalization’ approach (Jeffrey 1983). Firstly, for every proposition whose probability is to be updated in the light of new evidence, there must be already assigned probabilities to various conjunctions of the proposition and one or more of the possible evidence propositions and/or their denials. This leads to a combinatorial explosion. The number of conjunctions is an exponential function of the number of possibly relevant evidence propositions (Harman 1986).

In addition, once the number has been set, its rationale in terms of the multitude of factors from which it is comprised is submerged. There is no means of distinguishing between ignorance and uncertainty, for example (Carver 1988). A low number could imply a lack of evidence or alternatively plenty of dubious evidence. Dempster/Shafer is a numeric approach which does not suffer from this latter problem in representing both a belief’s support and its plausibility (Shafer 1976). Cohen (1985) and Carver (1988) prefer non-numeric representations attached both to data and to rules, to represent all the various aspects appropriate to reasoning about uncertainty. Cohen refers to these as *endorsements*.

The advantage of numbers is ease of manipulation and combination. But for the determination of preferred belief states for ‘real’ problems, the calculation must be based on more than probabilities of truth. As pointed out by both Doyle (1988) and Harman (1986), however probable and well supported or plausible a tautology is, it has little utility. In contrast, epistemic entrenchments are an indication of explanatory power and informational value (Gärdenfors 1988, Gärdenfors 1989). Associated with such an emphasis on the utility of belief as opposed to its certainty, is a very particular viewpoint on the nature of strength of belief, described below.

The probabilistic approach described above considers beliefs as variably certain. Only fully accepted or certain beliefs have a probability of 1. An alternative viewpoint is to consider all beliefs as accepted sentences, fully believed with a probability of 1, but not all of these may be equally *corrigible* in the sense of being more or less 'vulnerable to removal' (Levi 1984). What distinguishes them is their persistence; their relative ease of disbelief. Harman and Gärdenfors take this view. For Gärdenfors (1988) corrigibility is related to usefulness in inquiry and deliberation. He offers an example from modern chemical theory. Knowledge about combining weights is more important than colour or taste; it has more explanatory power. If chemists change their opinion over the combining weight of two substances, this would have more radical effects on chemical theory than if they changed their opinions over tastes. Beliefs about weights are therefore less corrigible or more entrenched than knowledge about tastes, although knowledge about both is certain.

This view that accepted beliefs are certain but variably corrigible, as opposed to all beliefs being variably certain, is an important component of the model of autonomous belief revision described in the next section. In this model of ABR, beliefs are held or not held in a yes/no fashion, but their strength as a pragmatic and purely comparative notion is considered at the point when held beliefs are challenged. Strength is a facet of revision. It also relates to entire belief sets. Preference of cognitive *state* in the light of a particular change is assessed according to relative persistence or comparative *'hardness of revision'* of alternative combinations of belief. Which set or sets are the most persistent or hardest to revise? Doyle similarly considers the ordering of entire belief sets to be more appropriate than for example, Gärdenfors' ordering of individual propositions. In Doyle's work as well as our model of autonomous belief revision, moreover, there may be alternative, equally preferred revisions. Again this differs from Gärdenfors' epistemic entrenchments which determine a unique and correct revision.

In the discussion above, it is suggested that 'hardness of revision' does not relate to varying certainty or probability of truth, but perhaps to utility in terms of explanatory power and informational value. What is the basis of this explanatory power or informational value? For Gärdenfors more useful beliefs are more entrenched (Gärdenfors 1988, Gärdenfors 1989). He offers various postulates for epistemic entrenchment which maintain individual beliefs as more entrenched than others on purely logical grounds. Nebel(1989, 1992) talks about particular sets of beliefs as more 'valuable' than others, these being more epistemically *relevant.* He describes epistemic relevance as a generalisation of epistemic entrenchment, but representing some 'extra-logical, pragmatic preference'. The specificity/generality distinction referred to above as the basis of HAEL (Konolige 1988) could be one such pragmatic preference candidate. A specific belief is preferable over a generality (Poole 1985). It has more explanatory power and informational value. This notion is also incorporated into inferential distance algorithms for inheritance systems (Etherington 1987, Touretzky 1986).

A wider approach in this latter vein is to look generally at the source of beliefs or the evidences from which they were concluded. As well as being specific or general, perceived beliefs can be the result of first hand experience via sensory apparatus, or they may be the result of second hand communications via other agents or documentation. Cohen (1985) attaches various endorsements to data, one type of which is based on source information. A representation of such endorsements and related set of heuristics regarding combinations of endorsements is outlined in the following section's description of ABR. The intuition is that there are general rules with respect to sources of assumptions underpinning beliefs, such as whether information came from a reliable source or was the subject of gossip for example, which are an important factor in determining relative persistence as relative explanatory power and informational value. Doyle (1991) uses decision theory to determine preferred revisions. Assessments of preference involve expected utilities of belief or utility of their consequences, whilst taking the probability of their occurrence into account.

One question is whether it is feasible to deploy general domain-independent principles such as entrenchment, utility, specificity or endorsement to determine preferred revision. Work by Konolige (1989) rejects the use of generalities in favour of 'knowledge-intensive heuristics tailored to a domain'.

2.1.2 Foundation and Coherence Theories

There are currently two approaches to rational belief change. They are *foundation theories* and *coherence theories*. These form alternative contexts within which any ordering or system of priorities for revision is to be accommodated. Generally these are described as competing approaches, although Doyle (1992) suggests that on close examination, the differences and corresponding pros and cons are certainly not as clear cut as has been previously suggested, and perhaps not even that significant.

Foundation theories focus on justified belief. New beliefs are only added on the basis of other justified beliefs, and beliefs no longer justified are abandoned. An example of this approach in practice is the truth (reason) maintenance system of Doyle (1979). Foundation theory takes its name from the emphasis on justification for belief, which obviously is not infinite. Where it ends up is in beliefs which are justified by themselves, and which then justify or are *foundational* to others. These are self-evident beliefs, for example observations.

Coherence theories on the other hand, represent a *conservatism* whereby justification is only a requisite condition of believing if there is a special reason to doubt a belief: current fully accepted beliefs are justified in the absence of any challenge to them (Harman 1986). If there is such a challenge, for example a new belief making one's belief set inconsistent, the guiding principles are those of *minimal change* and *maximal coherence*. The principle of minimal change states that in revising one's view one should make minimal changes in both adding new beliefs and eliminating old ones (Harman 1986). The notion of changes of state being restricted to keep as much as possible of the previous state, is generally accepted as a good thing, both in philosophy and AI. The competing notion is coherence. This prevents such conservatism resulting in tenacity of belief regardless of evidence to the contrary. Harman (1986) states: "changes are allowed only to the extent that they yield sufficient increases in coherence."

Coherent beliefs are mutually supporting. P can be justified because it coheres with Q and Q be justified because it coheres with P. But the *nature* of this mutual support is of interest. According to Harman, coherence includes not only a consistency relation, but relations of implication and explanation too. Coherence is *connections*, and the connections are of *intelligibility*, in particular intelligible deductive and non-deductive explanation of why or how it is that something is the case. For example, if one believes P, Q and R, but also R *because* P *and* Q. Part of one's view makes it intelligible why some other part should be true. The 'because' can be deductive in P and Q implying R, or it could be statistical as in P and Q generally implies R 'if other things are equal', or it could be based in commonsense psychology (Harman 1986). Believing R is explained by the beliefs P and Q. The connection offers intelligibility and makes the set more coherent than if P, Q and R were consistent but unrelated.

There are various computational examples of foundation theories in the form of reason maintenance systems, such as those of Doyle (1979), McAllester (1980) and de Kleer (1986a). There is only one computational example of a coherence theory (Georgeff and Lansky 1986), but several formal models, the foremost of which is AGM-theory (Alchourron, Gärdenfors and Makinson 1985). One formal hybrid model is described by Rao and Foo (1989a), Rao and Foo (1989b).

Models of coherence theory generally model minimal change amongst sets of consistent beliefs with no justification relations. Maximal coherence is the retention of the maximum possible *logically consistent* beliefs during belief change. These approaches therefore leave out much of Harman's intuitions on the nature and role of coherence. They cannot express that some beliefs are reasons for or explanations of others. However, Gärdenfors' epistemic entrenchments are an attempt to include some of the justificational information available in foundation theory into a formal coherence model. He describes how epistemic entrenchments can be used to reconstruct justifications when needed (Gärdenfors 1989).

The major criticism of foundation-based theories concerns the maintenance or explicit representation of justifications for beliefs, and also then the propagation of disbelief. Harman (1986) and Gärdenfors (1989) cite debriefing studies which demonstrate experimentally that people do not keep track of the justifications for their beliefs. It may therefore not be known when the sole reasons for beliefs have been discredited, and as a consequence unjustified beliefs are retained. Doyle (1992) criticises this as a psychological argument against foundation models. He distinguishes well-founded support from all arguments as well-founded, and separates the issue of recording reasons from that of their use. He also counters the economic arguments raised by Gärdenfors (1989) who suggests that regardless of the psychological plausibility issue, the benefits from keeping track of justifications are outweighed by the computational costs. Gärdenfors' view in this matter is echoed by Rao and Foo (1989b), who justify their own model by claiming RMS's as very inefficient.

All sides agree that justifications as reasons for belief are important, however. The conclusions from the debriefing studies were that in people, beliefs will eventually be abandoned, but not on the basis of a lack of justification.

Disbelief occurs only on the basis of *positive* beliefs about lack of good reasons for believing. Harman correspondingly expands the principle of conservatism into the following principle of *positive undermining*: only stop believing a current belief if there are positive reasons to do so, and this does not include an absence of justification for that belief (Harman 1986). Positive reasons are believing one's reasons for believing the belief to be 'nogood'. This is stated as: "It is incoherent to believe both P and also that all one's reasons for believing P relied crucially on false assumptions" (Harman 1986).

It is this particular version of conservatism, (discussed also by Doyle in (Doyle 1992)) which has been adopted in the model of autonomous belief revision. The following sections describe firstly the logical framework and then the computational model for ABR.

2.2 A Model of Autonomous Belief Revision (ABR)

The project builds on the specific theory of belief revision proposed by Galliers in (Galliers 1989, Galliers 1991, Galliers 1992). This starts from the position that an intelligent agent is obliged, in a changing world of which any agent has only partial knowledge, to operate autonomously. An agent, that is, cannot rely on predictable states of the world, or on predictable behaviour by other agents within the world, and therefore has to do the best with the knowledge and powers it does have in setting its goals and in planning and acting to achieve these. An agent also seeks to behave rationally by maximising its own outcomes, so in a context of uncertainty this implies adaptation. In particular, a continuously changing environment stimulates changes of mental state in agents, i.e. since all knowledge is actually belief, changes in the environment stimulate the revision of beliefs. This revision depends on the agent's goals, but as the environment changes, the goals can change too. Equally, having or adopting goals, which is a fundamental property of agents, implies action in and reaction to the world motivated by planning, and especially by strategic planning, to effect changes in other agents' mental states.

Our model of autonomous belief revision therefore determines preferred cognitive states rather than just beliefs at times of change. Of particular interest are instances of change caused by communicative acts, and where the content of an utterance contradicts an existing belief. In such cases, the principles upon which preferred cognitive states are determined from the logically equivalent possibilities are employed to reason about *whether* to adopt the recognised intended belief via an utterance, as well as *how* to do this. The model embodies a qualitative approach to the strength of belief issue. All-or-nothing beliefs comprise belief sets. If the world changes, new evidence is incorporated such that the resulting belief state is the most persistent of the logically equivalent alternatives. This includes the belief state where the new evidence is not incorporated. The preferred belief set(s) after revision are those retained, and beliefs in all of these are believed. The mechanisms for determining which revisions are preferred are described below. The preferred set(s) are the most coherent or most persistent or 'hardest to disbelieve'. We first introduce a general logical framework for belief revision before turning to ordering relations for preferred revisions.

2.2.1 A Logical Framework for ABR

Galliers and Reichgelt (1990) presented a general logical framework for a theory of autonomous belief revision, which we give here with the revisions the project has shown to be necessary. The framework supports both coherence and foundation theories of belief revision. It can be compared with epistemic entrenchments (Gärdenfors 1992) and epistemic relevance (Nebel 1989, Nebel 1992) as well as with Doyle's framework for rational revision (Doyle 1991). Primarily, it differs from these in allowing revision to be no revision at all; the preferred revision may not include the new evidence. We follow Nebel in assuming finite belief sets of consistent propositions, or belief bases in the sense in which Gärdenfors uses the notion (Gärdenfors 1992). However, our more-coherence ordering relation or *mc* compares these as whole entities, as opposed to either epistemic entrenchments or epistemic relevance which are local notions and order the different propositions within belief sets. In addition, we agree with Doyle (1991)'s criticism of epistemic entrenchments and epistemic relevance in their requirement for unique revisions; we permit equally acceptable alternative revisions.

We assume that an agent's belief state is represented as a finite set, $\mathcal{K}$, of (finite) sets of beliefs K, which are consistent and closed under negation but not necessarily closed under deduction. Beliefs are represented as propositions. A proposition ϕ is said to be pervasive if, for all $K \in \mathcal{K}$, $\phi \in K$. A proposition is believed iff it is pervasive. If there exist belief sets $K, K' \in \mathcal{K}$ such that $\phi \in K$ and $\neg\phi \in K'$ the agent is uncertain about ϕ. There are therefore three states an agent can be in with respect to a proposition ϕ of which it is aware: it can believe ϕ, i.e. ϕ is pervasive; it can disbelieve ϕ, i.e. $\neg\phi$ is pervasive; or it can be uncertain about ϕ, i.e. neither ϕ nor $\neg\phi$ are pervasive.[1] Before and after revision, an agent may be uncertain whether to believe ϕ or $\neg\phi$. For example, if an agent believes $p, \neg(p \wedge q)$ and hence believes $\neg q$, and is told that q by an authoritative source, the agent may be undecided whether to reject q, to abandon p or to abandon $\neg(p \wedge q)$ or both.

We postulate two operations on belief sets, addition and deletion. The addition of a proposition ϕ to $\mathcal{K}$, $\mathcal{K}^{+}\phi$, is then defined as:

$$\mathcal{K}^{+}\phi = \{K' | K \in \mathcal{K}, K' \subseteq K \cup \{\phi\}, cons(K') \text{ and } K' \vdash \phi\} \cup \mathcal{K}$$

where $cons(K)$ intuitively means "K is consistent" and can be defined as "there is a ψ such that $K \not\vdash \psi$". Thus, the addition operator defines a set of possible revised belief sets, i.e. a belief state. Note that, because $K \in \mathcal{K}^{+}\phi$, $K' \in \mathcal{K}^{+}\phi$ does not always imply that $K' \vdash \phi$. This is because we are interested in

---

[1] The details of the belief revision framework are insensitive to changes in the logic of beliefs with the sole exception that the belief sets are closed under negation. Without a modal operator we can only represent three belief states. With the addition of a modal 'belief operator' we can represent seven states (see chapter 5). This is what we mean when we say the model of *belief revision* is largely insensitive to the way in which beliefs are represented.

autonomous belief revision in which an agent may decide to ignore a new piece of evidence. Also, the members of $\mathcal{K}^{+}\phi$ are not necessarily maximal subsets of the members of $\mathcal{K}$. This reflects the intuition expressed by Doyle (1991) that belief revision is not always minimal in the sense that we keep as many of our old beliefs as possible. The only restriction that we have is that, if one decides not to engage in belief revision, nothing changes.

The deletion of a proposition ϕ from a belief set $\mathcal{K}$, $\mathcal{K}^{-}\phi$, can be defined in a similar vein as:

$$\mathcal{K}^{-}\phi = \{K' | K \in \mathcal{K}, K' \subseteq K \text{ and } K' \nvdash \phi\} \cup \mathcal{K}$$

Our addition and deletion operators define a set of potential new belief sets. In order to decide which belief set will actually be adopted, our logic requires an ordering relation between belief sets. Different orderings can be regarded as defining different logics for autonomous belief revision. Ideally, these orderings should define, for every set of belief bases, one maximal member. This belief base is the one that will be adopted after revision.

2.2.2 Ordering Relations for ABR

In the terms of our framework, we can reconstruct the difference between the two types of belief revision theories, foundation and coherence, as a difference between the types of ordering between belief sets imposed by the different approaches. For example, in a foundational approach, the ordering relies on some notion of well-founded support. First, we restrict ourselves to set-theoretically maximal members of $\mathcal{K}^{+}\phi$ or $\mathcal{K}^{-}\phi$ minus $\mathcal{K}$, where the belief state $\mathcal{K}$ contains a single belief set. Moreover, we assume that there is a set of self-evident beliefs, E. We then say that the belief set K is foundationally preferred to the belief set K', $K' <_f K$, if for all $\phi \in K, K \cap E \vdash \phi$, and there is a $\psi \in K', K' \cap E \nvdash \psi$. This means that K is foundationally preferred to K' if all sentences in K ultimately depend on self-evident beliefs, whereas there is at least one sentence in K' that is not supported in this way. It is unlikely that this ordering will produce one maximal member in $\mathcal{K}^{+}\phi$ or $\mathcal{K}^{-}\phi$, and in general it will have to be combined with some other ordering. One such additional ordering is McAllester's proposal to divide propositions into likelihood classes and to prefer those belief sets whose members are in the higher likelihood class (McAllester 1980).

The criteria used in coherence theories are described above. Firstly there is minimal change. The belief sets in $\mathcal{K}^{+}\phi$ or $\mathcal{K}^{-}\phi$ that are closest to $\mathcal{K}$ are preferred to those that make more radical changes. The competing notion is maximal coherence, or connectivity. The tension between the principles of minimal change and maximal coherence is most clearly illustrated in our general framework. Since $\mathcal{K}^{+}\phi$ includes the (single) belief set in $\mathcal{K}$, the principle of minimal change would produce $\mathcal{K}$ itself as the maximal member of $\mathcal{K}^{+}\phi$. However, the maximal coherence principle may produce other results, depending what factor is chosen as the operationalization of the notion of coherence. For example, another member of $\mathcal{K}^{+}\phi$ may have greater explanatory power than $\mathcal{K}$. Harman provides a synthesis out of this clash in his principle of Positive Undermining.

2.2.3 Increased Coherence

We suggest a new coherence ordering, particularly suited to ABR for modelling communication. This is increased or more-coherence, hereafter referred to as *mc*. *mc* orders logically consistent sets according to maximal derivability of *core* beliefs. This is based on the intuition that for a particular context, an agent has a number of central beliefs and that any piece of evidence that increases the agent's confidence in these central beliefs will be adopted—the more support a new belief offers for these central beliefs, the more useful it is. Thus in evaluating alternative revisions of a set of beliefs as responses to an input, it is necessary to consider how these improve the derivational, and hence explanatory, justification for beliefs as this is embodied in the connectivity among the beliefs in a set.

We say that a proposition ϕ increases the coherence of K with respect to some core belief ψ if adding ϕ to K would generate a new proof of ψ.[2] In order to establish whether this is the case, we first remove all proofs for ψ from K, after which we add ϕ to each resulting belief set. The aim is to establish whether ψ can then be proved in at least one of the resulting belief sets. Thus, we define $mc(K, \psi, \phi)$ (ϕ increases the coherence of K with respect to ψ) as

$$mc(K, \psi, \phi) \text{ iff there is a } K' \in (K \downarrow \psi)^{+}\phi, K' \vdash \psi$$

where $K \downarrow \psi$ "K less ψ" is defined as[3]

$$K \downarrow \psi = \{K' | K' \subseteq K \text{ and } K' \nvdash \psi\}$$

i.e. the set of all subsets of K consistent with ψ. We can then start preferring belief sets that have an increased coherence with respect to some core belief ψ. Thus, we define the ordering $\leq_\psi$ as

$$K' \leq_\psi K \text{ iff for all } \phi \in K', \text{ if } mc(K', \psi, \phi) \text{ then } mc(K, \psi, \phi)$$

We can then define a strict ordering in the normal way as $K' <_\psi K$ iff $K' \leq_\psi K$ and $K \not\leq_\psi K'$.

The above describes a more-coherent belief base relative to some core belief, as the harder to disbelieve because there are more justifications, more *proofs* of that core belief.

For example, I believe that I have to pay £50 for the repairs to my car when I collect it from the garage. This is a core belief; it is central to my concerns at the time of collection. I believe it as a self-evident belief because I was told so by the mechanic when I left the car. In addition I believe that if I believe

---

[2] We regret the fact that for historical reasons and reference to other publications the term "coherence" is used in two different (though closely related) senses in this report. It refers both to a broad notion as in coherence theory subsuming, as indicated, both connectivity and minimal change, and more particularly to connectivity alone, which we also define as *mc* or maximal coherence. As noted, Galliers' theory is essentially a coherence theory, but one incorporating foundational considerations as well. In most of this report "coherence" is used in its more detailed specific sense referring to connectivity, and we assume that the context makes the intended use of the term clear.

[3] Note that '$K \downarrow \phi$' is not a belief set, since the resulting set of propositions is not closed under negation.

the mechanic has completed the job, and also that there is a bill for 50 pounds, then I do indeed have to pay £50. When I get to the garage I can directly perceive that the mechanic has completed the job and I therefore believe he has completed the job. However, there is no bill evident as yet. At this point I believe that I will have to pay £50 for the one, self-evident reason as above. However, then I am given some new communicated evidence. I am told there is a bill going in the post for £50. If I believe this, then I have additional proof of my core belief. By taking on the belief that there is a bill for £50, it is 'harder' to disbelieve that I have to pay £50, given I believe the job has been completed and I believe the rule above. This is because, in order to *now* disbelieve that I have to pay £50, either I would have to disbelieve both what the mechanic said to me earlier and the fact that the job has been completed, OR I would have to disbelieve both what the mechanic had said earlier and the existence of a 50 pound bill, OR I would have to disbelieve both what the mechanic had said earlier and the rule that if the mechanic has completed the job, and there is a bill for £50, then I do indeed have to pay £50. On the other hand, before hearing about the bill, I would only have had to disbelieve what the mechanic had said to me earlier in order to disbelieve that I have to pay £50. So, it is more coherent for me to revise my belief set by adopting the communicated belief. I believe that there is a bill in the post for £50.

The more-coherent set does not have to be one including the new evidence. Each potential state is compared equally and autonomously. For example, another time maybe I also believe that if the mechanic has completed the job but there is no bill, then I just pay the £50 he quoted earlier. So, when I get to the garage and there is no bill I still have additional proof of my owing £50. A belief that there is no bill is inconsistent with the belief that there is a bill. When I get told there is a bill in the post, do I adopt this new belief or not? Is the preferred belief state one where there is a bill and I believe the new evidence, or one where I reject the new evidence and stay believing there is no bill? In fact, both are preferred, more coherent states according to the definition of *mc*. They both offer additional proof of the core belief. I still have to pay 50 pounds. The only issue here is whether I also believe there is a bill for £50 or not.

In reality we have good heuristic and intuitive guides which may assist in discriminating between such alternative belief states. In general, if we are told something by someone considered knowledgeable about the matter in hand and who is also considered to be reliable, we will tend to believe it in preference to something contradictory that may have been believed on the basis of less 'persistent' evidence. In the example above, all else being equal, evidence communicated from an employee of the garage is more persistent than contradictory evidence based on previous experience. Such heuristics can form the basis of additional ordering relations which can be employed in conjunction with *mc* to determine the relative persistence of one belief state over another. These are described in more detail below.

2.2.4 Endorsements and Minimal Change

In general the ordering relation *mc* will be insufficient to determine a unique revision. We therefore also employ modified forms of both the foundational and minimal change orderings described above.

The foundational approach was described in the terms of our logical framework in section 2.2.2 as an ordering relation in which one belief state is foundationally preferred to another only if there is some sentence in the latter unsupported by some self-evident or self-justified belief(s). Following de Kleer (de Kleer,1986), we call these self-evident beliefs *assumptions.* Assumptions are *endorsed* according to their source. The intuition is that there are general rules related to the sources of information which are relevant when considering how relatively 'hard' that information is then to give up. For example, whether they came from a reliable source such as directly from sensory apparatus, or alternatively, indirectly via another agent, or if they were assumed on the basis of generalised knowledge in the absence of anything more specific, and so on. Agents will be more unwilling to give up more strongly endorsed assumptions.[4]

Each founding assumption is endorsed as:

Communicated either *first-hand* (sensory information) or *second-hand* (via another agent or text). These assumptions are also very roughly graded as 'pos' if they are communicated with conviction or from a very reliable source, or 'neg' if they are communicated from a spurious source or without conviction, giving a total of four endorsements: *1c-pos*, *1c-neg*, *2c-pos*and *2c-neg*.

Given either as *specific* information widely believed and without any particular source, for example 'James Dean was a film star', or as *default* generalities similarly widely believed. For example, 'birds fly'.

Hypothetical with no evidence at all other than as a possible grounding for a belief under consideration [hypoth]. All beliefs are endorsed at least [hypoth] and may have additional endorsements.

Combinations of endorsed assumptions underlying competing revisions can then be compared using a set of very simple guiding heuristics:

1. Belief states founded upon first-hand evidence are harder to disbelieve than those founded on any other combination of assumptions. (This does not take the possibility of faulty sensors into account). Prefer belief states grounded with more [1c-pos] assumptions.

2. The more positive communicated assumptions or specific assumptions, that ground a belief state, the harder the process of disbelief, regardless of the number of 'neg' or default assumptions. Prefer belief states grounded with more [2c-pos] and [spec] assumptions.

---

[4] Other works concentrating on the role of the source of evidence when reasoning in situations of uncertainty are Thost (1989) and Garigliano, Bokma, and Long (1988).

3. Combinations of ‘neg’ endorsed assumptions and defaults are then considered, preferring belief states with more [1c-neg] over [2c-neg] or [def].

In our model, preferred belief states on revision comprise only self-evident beliefs and beliefs derived from these.[5] In this sense our model employs foundationalism as an ordering relation. It is important to note that there is a difference between our notion of foundationalism and that found in the ATMS or other RMS’s. In our framework, when beliefs become unsupported, their disbelief is only propagated backwards to founding assumptions so that a disbelieved belief cannot immediately be rederived. Disbelief propagation does not occur forward. Beliefs justified by the removed beliefs are retained, unless they are themselves the subject of challenge. This is the conservatism of coherence models in which beliefs are retained unless there is reason to not believe them, which is different from saying there is an absence of justification. This is Harman’s principle of positive undermining or positive disbelief. In our approach, derived beliefs which lose all their justifications in this way become new assumptions, although endorsed only as *hypoth.* ABR therefore represents a blend of coherence and foundationalism.

Finally, we also employ an ordering based on minimal change of pervasive beliefs, i.e a belief set K is preferred to another belief set K' if K contains more of the beliefs that were pervasive before revision than K'.

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

---

[5] Endorsement is not propagated directly to derived beliefs since it is not obvious how derived endorsement values can be calculated from multiple different input values. However endorsements do provide an indirect means of discriminating among derived beliefs via the notion of ‘commitment’ to beliefs (see chapter 5).

2.3 Revising Intentions

The agent's beliefs are only part of the agent's cognitive state. Intentions are also necessary for goal directed problem-solving behaviour. Such intentions depend on beliefs—we cannot intend what we believe to be the case and what we intend depends on what we believe. Revising beliefs therefore entails revising the intentions that depend on them.

Previous accounts of intention revision in the literature (e.g. Pollack, Israel and Bratman (1987), Singh (1991) and Rao and Georgeff (1991)), although recognising the dependence of intentions on belief, have assumed different mechanisms for belief and intention revision. Our work, instead, views intention revision as an aspect of 'belief' revision in general, governed by considerations of consistency between, support for, and minimal change of the resulting beliefs and intentions. Different propositional *attitudes* combine to make up an agent's cognitive state, with derivational links between beliefs, intentions and predicted future states. An agent's *intentions* are determined via 'belief' revision, so in developing our basic model of belief revision to cover intentions we have had to develop the notion of commitment already applied to beliefs so that it also applies to intentions.

Our extended model of autonomous belief and intention revision determines preferred cognitive states at times of change. Of particular interest are instances of change caused by communicative acts, and where the content of an utterance contradicts an existing intention, either directly (the speaker communicates an intention which is inconsistent with one of the hearer's intentions) or indirectly (the speaker communicates a belief which is inconsistent with the beliefs supporting one of the hearer's current intentions). We constrain intentions so that a state cannot be intended if the agent believes it already achieved, or that it will never be achieved: by definition, such states result in inconsistencies. When the intended state is achieved, the intention is abandoned as are any intentions derived from it. As with belief revision above, the principles upon which preferred cognitive states are determined from the logically equivalent possibilities are employed to reason about *whether* to adopt the recognised belief or intention via an utterance, as well as *how* to do this. The preferred attitude set(s) after revision are retained: beliefs which are in all preferred sets are believed, and intentions which are in all preferred sets are intended

We extend our model of belief revision to include intentions as follows. We assume that an agent has a *cognitive state*, a finite set, $\mathcal{K}$, of finite sets of beliefs and intentions, K, which are consistent and closed under negation but not necessarily closed under deduction. We write $B_a p$ to mean that the that the agent a believes that p and $I_a p$ to mean that a intends that p. We redefine the addition and deletion operators in terms of propositional attitudes rather than simple propositions. The addition of a propositional attitude α to $\mathcal{K}$, $\mathcal{K}^+\alpha$, becomes:

$$\mathcal{K}^+\alpha = \{K' | K \in \mathcal{K}, K' \subseteq K \cup \{\alpha\}, cons(K') \text{ and } K' \vdash \alpha\} \cup \mathcal{K}$$

and the deletion of a propositional attitude α from $\mathcal{K}$, $\mathcal{K}^-\alpha$, becomes:

$$\mathcal{K}^-\alpha = \{K' | K \in \mathcal{K}, K' \subseteq K \text{ and } K' \not\vdash \alpha\} \cup \mathcal{K}$$

The revised addition and deletion operators define a set of potential new cognitive states. In order to decide which cognitive state will actually be adopted by the agent we must extend our ordering relation to include preferences over intentions.

The increased coherence ordering mc can be applied to sets of beliefs and intentions with only minor modifications. We say that a propositional attitude α increases the coherence of a belief and intention set K with respect to some core belief or intention β if adding α to K would generate a new proof of β. Thus, we define $mc(K, \beta, \alpha)$ (α increases the coherence of K with respect to β) as

$$mc(K, \beta, \alpha) \text{ iff there is a } K' \in (K \downarrow \beta)^{+}\alpha, K' \vdash \beta$$

and the ordering $\leq_\beta$ as

$$K' \leq_\beta K \text{ iff for all } \alpha \in K', \text{ if } mc(K', \beta, \alpha) \text{ then } mc(K, \beta, \alpha)$$

Note that because of the inherent asymmetry between beliefs and intentions—intentions can be derived from intentions and/or beliefs but not vice versa—the introduction of an intention can never lead to an increase in coherence with respect to a core belief.[6]

In considering an agent's *commitment* to its attitudes, its commitment to its intentions depends in part on its commitment to the beliefs that 'support' that intention. However, commitment also depends on the *importance* of the intended state, and the likelihood and difficulty of achieving that state. Here we borrow from decision-theoretic approaches to action choice (e.g. Gmytrasiewicz, Durfee, and Wehe (1991)), though we will make no assumptions about the availability of numerical estimators of utility. Instead, we extend our notion of *endorsements* on beliefs to apply to intentions by including heuristic descriptions of the *utility* of goal states and the *effort* required to perform the actions leading to that state. New goals to achieve particular states are therefore assigned a description of the expected utility of the intended state, while actions have a crude heuristic description of their associated expected effort.[7]

The belief endorsements are extended to include the following intention endorsements (as with beliefs, all intentions are endorsed *hypoth*):

States are endorsed *desire-pos* if the agent strongly desires the goal, or *desire-neg* if the agent only weakly desires the goal.

Actions are endorsed either *effort-pos* or *effort-neg*, depending on how much effort is required to perform the action given that the preconditions of the action are true.

---

[6] This is an oversimplification—for example it does not apply to predicted beliefs which can have intentions as antecedents, see chapter 5.

[7] The principle of positive undermining also holds for intentions. If the beliefs justifying an intention are dropped the intention is retained so long as there are no actions associated with the intention or its sub-intentions. Actions require effort and, as a result, belief sets containing them are dispreferred. This is discussed in more detail in chapter 5.

Note that the endorsement associated with intentions is related to a heuristic assessment of their expected outcome. This contrasts with those associated with beliefs, where it is the source of the beliefs that are represented. In general intended states which have a high expected utility and intended actions which have low expected effort and uncertainty are preferred. To discriminate between competing revisions on the basis of the intentions and actions they contain, we add the following additional rule to the heuristic ordering, which is applied after the belief rules:

4. intention states containing strongly desired goals are preferred to those containing weakly desired goals; and intention states containing actions which require less effort to perform are preferred to those which require more effort to perform.

These heuristics are not guaranteed to produce a single most preferred cognitive state. At any one time there may be several current alternative preferred belief and intention sets which the agent has no information for choosing among and for which decision information is required. However this is a natural consequence of the fact that agents have only partial knowledge, and this lack of knowledge is reflected in the agent's alternative hypotheses about the state of the world and its corresponding alternative goals and plans (see below).

It is useful to evaluate our approach to intention revision in the light of Bratman's desiderata (Bratman 1987). By defining appropriate logical inference rules we ensure that an agent believes that an intended action can eventually be achieved, and that it has not already been achieved—if either of these conditions fails to hold the intention will be dropped. Because of the basic belief revision mechanism, an agent's intentions will be consistent with their beliefs, and minimal change ensures that intentions have some, though limited, stability. New intentions must be consistent with old ones—if they are preferred they force intention revision—and the system will attempt to determine ways of achieving existing intentions via its normal inference mechanisms. However, although most of this is consistent with Bratman's characteristics, it is unclear whether our notion of minimal change provides *sufficient* stability of intentions (and possibly beliefs too), to ensure that, through the predictability of action, agent actions and interactions may be coordinated.

2.4 Embedding Inference and Planning within the ABR Framework

The uniform approach to belief and intention embodied in our theory of ABR provides a suitable framework within which the inferential and planning capabilities required by an autonomous agent can be embedded.

Embedding inference within the ABR framework is straightforward. Beliefs can be derived from existing beliefs using an appropriate inferential mechanism. Derived beliefs are simply *added* to the current cognitive state.[8] There are four

---

[8] Note that this means that the agent may choose not to believe the logical consequences

possibilities: the inference may result in an inconsistency; it may increase the coherence of a belief and intention set or sets by providing a new justification of a core belief; or it can result in a new justification for an existing belief or a new belief, possibly increasing the endorsement of one or more belief sets. In all cases inference gives rise to belief revision. However this is handled by the implementation of the ABR framework and no modification of the inference mechanism itself is required. The lack of commitment to a particular logic of belief allows us to choose the representation and inference mechanism most appropriate to the domain in question. In chapter 5 we describe the implementation of a simple rule-based inferential procedure for an automated information retrieval agent.

Planning is also straightforward. We can use the inference mechanism to apply intention generation rules, given existing beliefs, intentions and planning operators, to derive new intentions. This is pretty much like standard planning, but the system continually reassesses which mutually realisable sets of intentions are currently preferred (these being the agent's current intentions). In general, intentions may be conflicting (not mutually realisable) because of competing temporal, physical or computational resources. Such intentions will never be jointly intended. However, preference also applies between different sets of intentions which represent alternative ways of (i.e., plans for) satisfying a higher level intention. In this case, the justificatory links between intentions are set so that if one alternative is chosen, the other will lose its support. Sub-intentions in dispreferred sub-plans need not be inconsistent with existing intentions, and so may become preferred (and intended) given an independent reason for that sub-intention to be satisfied. An agent's intentions depend on the expected effort required to achieve the intended state from the current state, the utility of the intention, and the strengths of its associated beliefs about the world. These different types of strength are all captured in the heuristic endorsements associated with beliefs, intentions and actions. An agent is most committed to, and hence will act on, the action it finds hardest to 'disintend'. Thus planning and acting are interleaved in the framework.

The basic agent action cycle therefore involves: incorporating new information and requests into the agent's current cognitive sate; firing some deductive inference rules (including intention generation rules) to generate further (possible) beliefs and intentions; performing belief and intention revision in the face of conflict (or jointly unrealisable intentions) to obtain preferred beliefs and intentions; then executing the most preferred intended action(s). This is discussed in more detail in chapter 5.

---

of its beliefs. The S4 axiom (A6) $B_a p \wedge B_a(p \supset q) \supset B_a q$ therefore does not hold. However, since A6 does not hold for any finite agent in any case, this is no great loss.

Chapter 3

Testing the Model: The Information Retrieval Task

This chapter presents the view of the information retrieval task we have used as the basis for our work. The material is essentially that used in Cawsey et al. (1992a, 1992b). We used this domain partly because it seemed ideal as a testbed for Galliers' theory, since it involves autonomous agents each with their own knowledge engaged in a task which is not independently well defined, and partly because it is important in its own right, so 'automating the librarian' is a worthwhile endeavour.

Our aim in the project was thus to see how to provide the power needed for an automated intermediary by exploiting the general theory of belief revision described in the last chapter as a mechanism motivating both the system's external interaction with the user and its internal problem solving. Intelligent agents are continually revising their beliefs, and this applies to interaction between library users and librarians as much as to other dialogues. Interaction on literature seeking is not driven by fixed goals or manifested in a unidirectional flow of data from one party to the other. The dialogue fragment reproduced in Figure 3.1, of the kind recorded for actual sessions, clearly shows both parties revising their beliefs about what is wanted. The librarian, having started by

Dialogue

`User:       I want a book on cacti.`
`Librarian:  On growing them?`
`User:       No, on the diseases they cause.`
`Librarian:  Other house plants as well?`
`User:       Maybe.`

Search Specification

HOUSE PLANT $\wedge$ HUMAN DISEASE

Figure 3.1: Fragment of dialogue between a user and a librarian and the resulting search specification to be applied to a bibliographic or literature file

assuming that when people ask for books on plants they want books on growing plants, is obliged to revise this belief to accommodate a request for books on other aspects of plants. Equally the user, having started by saying he wants a book on cacti, revises this belief to accept that books on other sorts of plants may be appropriate. Both parties collaborate to arrive at the actual search specification aimed at retrieving literature references from the file, and hence ultimately the literature itself, to meet the user's real need. For this illustration we may envisage the eventual output of the mutual belief revision process as a submitted search request of the conventional sort for online services, in the form of a Boolean combination of terms in some controlled indexing language of the kind exemplified by the Medline system's Medical Subject Headings (MeSH), which is used to characterise the items in the search files, normally of literature references rather than end full texts.

We are therefore concerned on the one hand with appropriate general mechanisms for agents manipulating beliefs and conducting dialogue, and on the other with deploying these mechanisms within the framework supplied by the literature searching task and by a model of the librarian's characteristic knowledge and actions. In this chapter we focus on the information retrieval task. In chapter 4 we return to the problem of dialogue.

3.1 The Information Retrieval Task

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

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

As the above example suggested, the typical situation is the topic or subject search for unknown items, of the kind associated with online search services. The example of Figure 3.1 assumed a subject-based search of an online book catalogue, rather than the more common subject search of journal literature, but the generic situation is the same. Searching in these contexts is of course usually iterative: our initial simplification for experimental purposes is to treat the point at which the first actual search formulation is submitted to the online system as a stopping point; but this does not affect the general form of the agent-user interaction, and iteration can be incorporated later, as it is clearly essential for a realistic and effective system. The situation being modelled will be referred to for convenience as the library situation, regardless of whether there is an actual library with literature to hand, and of whether books or papers are in question.

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

3.2 The BBD Model

We have adopted as a starting point work by Belkin, Brooks, and Daniels (Belkin et al. 1983, Belkin, Hennings and Seeger 1984, Brooks, Daniels and Belkin 1985, Brooks 1986, Daniels 1987)—hereafter referred to as BBD.[1] This work was based on real library dialogues, but it must be emphasised that everything has had to be ruthlessly simplified for our project. This applies whether the real library situation is one where the literature is to hand and the usual means of access is via a conventional catalogue, or where references to literature are obtained via an online search service. The BBD model is completely general, and is intended indeed to apply to all types of information-seeking situation and not just library or literature search services. It is also intended to cover the range of enquiries stretching all the way from quite definite requests for known items to very indefinite, barely formulated needs for unknown items.

BBD have suggested that an appropriate way of modelling the librarian is as a set of subtask processors, or functional experts, each with their own specific resources and each satisfying their own data-gathering goals, but in doing this collectively contributing the data required to achieve the overall system goals, namely to enable the user to satisfy his information need. In general, this may

---

[1] Other research by, e.g. Pollitt (1986), Vickery et al. (1987) and Brajnik, Guida and Tasso (1990), has also been concerned, in different ways, with the problem of constructing information retrieval systems as models of the intermediary: see further below.

Central Processors

Problem Description: cactus cause disease, ...
Problem State: starting finding out, ...
Problem Mode: reading, ...
User Model: householder, ...
Retrieval Strategy: CACTUS $\vee$ SUCCULENT

Support Processors

Dialogue Mode: talking
Explanation Provision: little on plants
Input Analysis: "No, on diseases ..."
Response Generation: non-cacti?
Output Synthesis: "Other house plants ..."

Figure 3.2: Librarian model processors with illustrative information states for the dialogue example of Figure 3.1

be done either directly, or indirectly by providing pointers to documents. But in some cases it may prove impossible to help the user: thus the outcome for the system is more correctly characterised as satisfying the goal of doing the best for the user, as mutually agreed. For the simple experimental case being studied by the project, however, this is taken as agreement on an initial search specification.

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

The complete set of processors BBD propose is quite large. It includes both what may be thought of from the global task point of view as central processors and support processors. The complete set, embodying some compromise between BBD's various publications, and with some renaming for present convenience is shown in Figure 3.2, along with very simple illustrations of the kinds of state they might be in at about (though not necessarily precisely simultaneously) the end of the dialogue fragment of Figure 3.1. These illustrations are simply indicative, however, and are not intended to make any claims about the proper way of representing processor results. The central processors are those bearing directly on the user's information need. They include the Problem Description expert, intended to capture the user's topic and its broader conceptual context or subject area, deemed in the example to be conflated as the notion represented by 'cactus cause disease'; the Problem State expert, showing the status of the user's progress with his subject and topic, in this case just starting finding out; the Problem Mode expert, characterising the manner of information gathering taken as appropriate for the user to supply his need, in this case reading (as opposed to, say, talking to someone); the User Model expert, giving the relevant properties of the user, e.g. householder (not horticulturalist); and the Retrieval Strategy expert which produces the means of access to the description or document file, in this case taken as a Boolean request in a controlled indexing language.

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

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

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

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

3.3 Problems with the BBD Model

While the BBD model provides a useful starting point for an automated intermediary, from our point of view it suffers from a number of problems. The first, most obvious, problem is that it fails to address the issue of belief revision, either at the level of the individual processors or that of the system as a whole. As we saw in Figure 3.1, both parties must continually revise their beliefs about what constitutes an acceptable problem description, and the kind of retrieval strategy which will best meet the user's need. It is not clear from BBD's descriptions how their model should respond to conflicting input either from the user or from the various subtask processors. Our solution to this problem is presented in chapter 6, where we show how a revised version of the BBD model (described in section 3.6 below) can be implemented within the framework for belief and intention revision outlined in chapter 2. (An alternative to this approach, in which each of the functional experts or processors are responsible for their own belief revision is discussed in appendix C.)

However there are also a number of problems internal to the BBD model itself, which must be resolved before any computational implementation, however simple, can be attempted. There are of course questions about the message language used for internal communication, and about the way individual processors interact with the blackboard. But the serious issue is overall control, and in particular control of the external dialogue with the user. The way BBD appear to see control operating is essentially opportunistic, applying 'syntactic' criteria relating, for example, to message or sender status, rather than 'semantic' criteria relating to message content, to determine which messages require responses from the user and when the response should be sought. Thus the notion in Belkin et al. (1983) seems to be that output is triggered when there is enough pressure from the data state (indicating hypotheses to be tested or information to be sought) on the blackboard.

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

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

If the model is redesigned for a dominating controller with subordinated sub-processors, it is not clear how far these can operate autonomously in parallel and in a data-driven way. Even if they can, it is not obvious how control and dialogue management as a whole are to be achieved, given three critical features of the task situation being modelled. These are first, the weakness of the notion of satisfaction for sub-processors, especially key processors like the Problem Description one. Data gathering cannot be driven, as it can for many other tasks, by a check-list approach, certainly not at the level of offering a range of specific choices, but even of generic ones. With a menu system the relevant variables (slots) would be given, and perhaps even the potential values (possible slot fillers). Limited implementations of the automated intermediary like Pollitt's are able to operate effectively with known slots and filler possibilities, and this may be feasible for e.g. simple versions of the User Model. But it is not possible in general, for example, to capture topic information by a menu approach because the range of possibilities is too large, unless the menu is more notional than real, with generic slots like 'Concept1', 'Concept2', and so forth, and the notion of satisfaction applied is minimal, e.g. three concepts is by definition enough. BBD's presumption is that obtaining a proper or adequate topic description is a serious matter, and this implies a sophisticated approach to determining whether a given topic characterisation is adequate, which can only be based on a number of criteria which are individually weak. (This is setting aside the fact that it is hard to establish what the set of criteria is or how they work together, and also the fact that the criteria may be very hard to apply.) It is also difficult to get mileage out of a notion of obligatory data. For example it is not necessary to have any individual information in the User Model at all (and the default user characterisation may be very simple indeed). The weakness of the satisfaction criteria applies everywhere, but is especially awkward as far as the crucial Problem Description processor is concerned: what is the right, or a good, problem description? It is clearly naive to suppose that effective dialogue can be conducted simply by the system applying a 'tell me more' strategy, but when satisfaction is weak it is difficult to determine what a system's output should be. It will certainly require the informed self-evaluation capability mentioned earlier. The satisfaction problem of course also applies at the level of the system as a whole: what, in the likely absence of clear indications from individual processors, determines whether the entire 'information need problem' has been satisfied? It is not evident that relying on the user to declare this, especially without constructive system suggestion, is efficient or effective.

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

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

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

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

3.4 Simulating the BBD Model

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

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

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

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

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

3.5 Distributed Architectures for Information Retrieval

An examination of previous attempts to build an automated intermediary using a distributed architecture tends to reinforce these reservations.

The CODER System

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

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

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

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

The I3R System

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

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

The IR–NLI II System

The architecture of both of these systems is thus less distributed in practice than in principle, and is much like that used in Brajnik et al. (1990)'s IR–NLI II. This prototype intelligent intermediary essentially combines a sophisticated version of Vickery et al. (1987) as a rule-based expert for handling search formulation and reformulation with an ambitious user modelling component. The retrieval subsystem exploits knowledge about search strategies and tactics of a professionally established kind with domain terminological knowledge, and is designed to develop an adequate characterisation of the user's need and appropriate search specification: this may involve iteration using retrieved output. The user modelling component, starting from stereotypes dealing with user experience, background and retrieval history etc, constructs and maintains a current model. Both subsystems may thus involve inference. However IR-NLI's operation is essentially system driven through a well-defined, possibly iterative, sequence of steps from initial request capture to final search specification, with communication with the user modulated by the user-modelling component (and not yet in free natural language). The system design makes control relatively straightforward, and the prototype implementation gains by being able to rely heavily on a quite restricted application domain and user community.[2]

---

[2] Huhns et al. (1987)'s work on distributed artificial intelligence for document retrieval (using a blackboard architecture) focusses on a very different problem from ours: this is how to retrieve effectively from different data sets, for instance distinct personal files.

3.6 Revising the BBD Model

The analysis presented above and the experience with CODER, IR3 and IR-NLI II suggest that there are significant problems with the BBD model. However it is important to understand what sort of problems they are. We must distinguish between BBD's knowledge level analysis of the IR task, which is probably the most complete in the literature (particularly in their emphasis on the development of the problem description) and the distributed architecture they propose for implementing the model, which has a number of significant problems.[3] BBD tend to obscure the distinction between the knowledge level and the symbol level in their published work, but there is no necessary connection between the two and we are free to adopt the former while rejecting the latter. In our case the problem of dialogue control is compounded by the need to integrate the BBD model into the belief revision framework. (The problems of dialogue control and belief revision for a set of highly specialised functional experts are discussed in appendix C.)

We therefore retain BBD's knowledge level analysis into tasks, but abandon the distributed architecture they propose. Instead we envisage a single agent, consisting of a number of individual *modules* which implement the functions identified by BBD. To simplify the problem sufficiently for initial implementation we consider only five Central Processors: Problem State, Problem Mode, User Model, Problem Description and Retrieval Strategy. The two Support Processors, Input Analysis and Response Generation, are replaced by a 'Dialogue' module capable of handling the kinds of dialogues found in the IR transcripts. (We ignore Output Synthesis and the NL parts of Input Analysis and assume the agents communicate in a simple propositional language.)

The architecture we have adopted is essentially rule-based. The automated intermediary consists of a collection of facts and rules, a rule interpreter and a database or working memory which represents the agent's cognitive state. Each processor or module is implemented as a collection of facts and rules. These modules are strongly inter-related within an overall task decomposition and with the dialogue rules. This basic architecture has been implemented within the belief revision framework presented in the previous chapter. Dialogue in information retrieval are discussed in the next chapter. The basic system architecture is described in chapter 5, the IR rules are presented in chapter 6 and the dialogue rules in chapter 7

---

[3] The notion of the knowledge level as a separate level of analysis was introduced by Newell Newell (1981).

Chapter 4

The Library Dialogues

As we noted in the previous chapter, the BBD model does not address the problem of dialogue. The model simply assumes that each of the functional experts can achieve their data-gathering goals, and specifically their data gathering from the user as opposed to from one another. However if we are to implement the BBD model, we must be able to explain how the functional experts go about their data-gathering. There are two problems here: achieving a particular dialogue goal; and maintaining coherence in the dialogue. These two problems are related. In this chapter we extend the BBD model to include dialogue. We discuss the problem of dialogue planning and how this is related to dialogue focus and plan repair in task-oriented dialogues. In the next section we develop the model of belief revision in dialogue sketched in chapter 2 and in subsequent sections we embed a conventional model of task-oriented dialogues, based on speech acts, within this framework.

For the purposes of analysing the library dialogues we limit the cases we consider to simple assertions and questions. We ignore promises, commands and other forms of request that actions be performed in the 'real world', e.g. "Please close the door". We also assume that we will not attempt to model indirect speech acts, e.g. "Do you know the time of the next train to Cambridge" / "Yes" / "4:15" / "4:15 from platform 5" etc., or anaphora, e.g. "Why do you believe that?".[1] While simple declarative assertions such as "It is raining" are relatively unproblematic, further simplification is necessary in the case of questions (and of assertions which form answers).

4.1 Belief Revision in Communication

One of the ways an agent can obtain new information is as a result of communication from another agent. According to Grice (1957, 1967), an utterance is a perceived event that conveys an intention; the speaker's intention that the hearer recognise an intention on the part of the speaker to cause a certain effect in the hearer's mental state. Since agents can be assumed to always have some mental state, then this can be alternatively stated as the recognition of

---

[1] We also have to rule out helpful responses based on common knowledge, e.g. "Everybody says so" or "Everybody says so, but I am not sure".

a speaker intention for a particular *change* in the attendee's cognitive states. Any perceived change in the environment, including the recognition of a communicative intention via an utterance, changes an agent's mental state, and can be dealt with as an incident of revision. Viewed in this way, a particular revision of another's cognitive state is the motivating force for communicative behaviour. Utterance planning concerns desired change of state, not simply a desired end effect. Utterance planning therefore involves an understanding of the principles of belief revision; how beliefs are gained and lost in order to accommodate new evidence.

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

Beliefs about other agents are clearly important in the interaction, but not just because they are part of the furniture of the world. As any agent has only limited powers to effect action, it needs cooperation to achieve its goals. This, however, in turn requires that it be cooperative. Thus dialogue is a process of negotiating and mutually accepting beliefs and hence intentions to act. Dialogue is a public manifestation of pervasive, goal-motivated belief revision in each participating agent, operating at every grain level in the characterisation of mental states.

Autonomous agents may or may not comply with the recognised intended effects of an utterance on their cognitive states. There are no specialised rules dictating what is a cooperative response. Rational communicative action must therefore be planned not only as purposive, but as *strategic* (Galliers 1989, Galliers 1991). One important implication of our approach for task-oriented dialogues is that there is no need for separate axioms describing helpful agents as those that always adopt other's recognised goals, for example to believe p, unless they conflict with an existing belief, such as already believing not p (Cohen and Levesque 1987, Perrault 1987). Similarly, there is no need to dictate either adoption or persistence of belief, or to treat contradictions in any way as a special case.

Strategic interaction acknowledges all participants as sharing control over the effects of a communication. Strategic action is that which maximises one's own outcome. Maximising one's own outcome in a situation of shared control requires that the outcome be maximal for the other agent(s) too. Achieving a desired change in another's belief states is therefore a matter of creating a context such that the general rules of rational belief revision would dictate that change anyway. The aim in utterance planning is to determine one's own actions according to one's own goals and the context. This context includes the other agent and their presumed existing mental states, and a prediction of the changed context which will result in their preferring the intended belief state according to the principles of rational, autonomous belief revision. Cooperative behaviour can therefore emerge autonomously, without being imposed from explicitly stated descriptions of how to behave 'helpfully'.

It might be argued that this is just replacing prescribed acceptance with cooperation as artful persuasion; strategically getting another to want (autonomously) to agree? But this is not so. All agents have autonomy over their belief states; all employ preference orderings based in maximal coherence in contexts of choice. If an utterance is unsuccessful, it may be that there is insufficient evidence that its adoption on the part of the hearer would result in as coherent a belief state as not adopting it. In this case the speaker may offer extra evidence, in order to persuade. But on the other hand, it may be the case that there is some evidence which the hearer has and the speaker does not, which causes the difference in coherence of this item of evidence with other beliefs for the two parties. A conversation aimed at achieving some joint or collaborative venture, which is not yet achieved, would then continue with an appropriate contribution by the hearer. This is *adaptive* cooperation in a distributed environment. Cooperation is achieved over a series of utterances, motivated by this as an ultimate joint goal and by an understanding that all concerned are operating according to a rationality specified by general principles of *autonomous* belief change.

4.2 Speech Acts

The goal in task-oriented dialogues is often for one agent to communicate information to another agent. It seems reasonable that in achieving this goal, the agent should plan to change the other agent's beliefs using what it knows about the other agent's current belief state. Dialogue planning can be seen as a plan to induce a change in an agent's beliefs. These may be not only others' beliefs, but the beliefs of the agent doing the planning, for example when the agent is trying to discern another agent's belief state by asking a question or for an explanation, or to discover why its plan to change the other agent's belief state failed. How the agent constructs its plan depends on what it thinks the other agent believes and how it thinks the other agent will revise its beliefs in response to new information.

As noted in the introduction, we handle dialogue within the conventional framework of speech acts. Thus the primitive actions of the plan are the agent's utterances or *speech acts*. A speech act is the performance of an *illocutionary act* which has the effect of bringing about a change (determined by the act) in the beliefs and intentions of the hearer. Each speech act has a number of *preconditions* which must be true for the successful performance of the act and a number of *guaranteed effects* which will be true after any performance of the act in a situation in which the the preconditions are true. In addition, the performance of the act may have a number of other effects both intended and unintended (perlocutionary effects) depending on the contents of the utterance and in particular the cognitive state of the hearer. For example an INFORM speech act (Allen 1987)

(INFORM x y p)
Preconditions:
(BEL x p)

Effects:
(BEL y p)

has the precondition that the speaker x believes p and the effect that the hearer y believes p. All speech acts have the same role structure, namely the two conversants and the semantic content of the utterance, which is called the propositional content of the utterance. The actual type of the speech acts is often referred to as the illocutionary force. The illocutionary force and the propositional content are independent of each other (Allen 1987).

4.2.1 Speech Acts for Planning

Unfortunately, there are problems with much of the work in the literature for our purposes. These problems can be illustrated by reference to Allen (1987) since, while there have been later developments in the field (see e.g. Cohen et al. (1990)), these do not clearly overcome the difficulties we consider below at the level of detail we require for our implementation or in a style that fits well with our approach to belief revision.

For example, Allen defines a number of *discourse acts* involving actions by both agents in the dialogue as compounds of four primitive speech acts: REQUEST, INFORM, INFORMIF and INFORMREF. Thus the dialogue act ASKIF is defined as a REQUEST that the hearer perform an INFORMIF act, i.e.

(REQUEST x y (INFORMIF y x p))

rather than as communicating the speaker's intention. The REQUEST act has the (guaranteed) effect that

(WANT y (INFORMIF y x p))

which is an implicit precondition of

(INFORMIF y x p)

which when performed has the effect[2]

(KNOWIF y p)

---

[2] Allen defines (KNOWIF x p) as (BEL x p) $\vee$ (BEL x $\neg p$) where p is a proposition or an attitude. For example, (KNOWIF *Sue* (OWN *Sue Fido*)) means 'Sue knows whether she owns Fido or not' and is equivalent to (BEL *Sue* (OWN *Sue Fido*)) $\vee$ (BEL *Sue* $\neg$ (OWN *Sue Fido*)). Note that this is distinct from (BEL x $p \vee \neg p$), which is a tautology and is almost certainly believed by any rational agent. This would typically form part of another agent's belief set, as it has little utility in Sue's own belief set. Similarly, (KNOWREF x $p(u)$) is defined as $\exists u$ (BEL x $p(u)$) For example, (KNOWREF *Sue* (NAMEOF *Mother*(*Sue*) u)) means 'Sue knows her mother's name' and is equivalent to $\exists x$ (BEL *Sue* (NAMEOF *Mother*(*Sue*) u)). Note that this is different from x's belief that there exists a u, without knowing what it is: (BEL x $\exists u\,(p(u))$).

However, Allen's speech acts make a number of assumptions which are not valid in our context, for example, that the agents are both sincere and cooperative. Because these assumptions form part of the definition of the act itself, they make analysis of failures of communication which give rise to (interesting) belief revision more difficult.

Moreover, in common with much of the work in the literature, Allen's speech acts are intended to serve as a basis for advanced dialogue planning rather than as a basis for the actual detailed conduct of dialogue, involving interpretation as well as generation, between computational agents. In other words, they are not *computationally effective*. For example

$$(\textsc{bel}\ \mathit{Jack}\ \exists x\ (\textsc{eq}\ (\mathit{Price}(\mathit{Ticket}\ \mathit{TR1}))\ x))$$

means Jack believes that the clerk knows the price of a ticket to Rochester (and hence that the ticket to Rochester has a price), but Jack himself doesn't know what the price is. Similarly:

$$(\textsc{knowref}\ \mathit{Jack}\ (\textsc{eq}(\mathit{Price}(\mathit{Ticket}\ \mathit{TR1}))\ x))$$

means there exists an x such that Jack believes it is equal to the price of a ticket to Rochester, i.e. Jack knows the price of a ticket to Rochester. This is a description of a state Jack would like to be in, but it doesn't explicitly represent the price itself. Allen's speech acts and the dialogue acts defined in terms of them, allow an agent to reason about what it knows and doesn't know, but they don't result in the agent acquiring the information.[3] The same is true of KNOWIF and INFORMIF. The agent knows whether a proposition is true or false, and an INFORMIF results in the hearer coming to know if the proposition is true or false, but no information is communicated.

$$(\textsc{informif}\ \mathit{Sue}\ \mathit{Jack}\ (\textsc{knowif}\ \mathit{Sue}\ (\textsc{own}\ \mathit{Sue}\ \mathit{Fido})))$$

results in

$$(\textsc{knowif}\ \mathit{Jack}\ (\textsc{own}\ \mathit{Sue}\ \mathit{Fido}))$$

i.e. Jack knows whether Sue owns Fido or not. However we can't get from this to either $(\textsc{bel}\ \mathit{Jack}\ (\textsc{own}\ \mathit{Sue}\ \mathit{Fido}))$ or $(\textsc{bel}\ \mathit{Jack}\ \neg(\textsc{own}\ \mathit{Sue}\ \mathit{Fido}))$ and hence to some inference about who Jack should talk to if Fido is digging up his lawn. For planning purposes this is adequate: if I ask for the price of the ticket I will come to know the price of the ticket which allows me to do further planning by satisfying the preconditions of other operators. (Planning to stack block A on block B does not mean that A is on B or even that A will be on B unless the plan is successfully executed, but it nevertheless allows me to continue planning to build a stack three blocks high.)

However for the analysis of belief revision in dialogue we must be able to model the flow of information between the agents. We are not asking if another agent believes there exists an x such that $P(x)$ or if it believes p or $\neg p$ is true.

---

[3] Allen finesses this point on p. 445, when the INFORMREF by the clerk results in Jack KNOWREFing the the price itself and hence giving the clerk the price of the ticket. However at 2.5 Jack has no more information than he does at 2.1—x does not get bound to anything.

The speech acts must communicate my desire to be in the state such that I know what x is. I can only be in this state if I actually know what x is. This in turn requires that when the plan is executed,the other agent must communicate a *value* for x which results in me being in this state.

4.2.2 Speech Acts for Dialogue

The point is that the view of speech acts appropriate to the analysis of dialogue is somewhat different from that suited primarily to planning. We therefore define our own speech acts to make as few assumptions as possible about the effects of the act and whether the agents are co-operative etc; and we characterise those additional assumptions we do make as preconditions on the speaker's beliefs, or on the speaker's beliefs about the hearer's beliefs and intentions arising from the dialogue context. For example, the speaker assumes only that its intention will be understood and infers, on the basis of its beliefs about the current context and in particular its beliefs about the hearer, that the hearer will come to believe the content of the utterance, or answer the speaker's question or whatever.

To simplify the problem, we assume that the agents communicate using the same propositional language they use to represent their beliefs. As stated in chapter 1 we do not attempt a realistic treatment of NLP issues. However our adoption of a propositional language does not mean that all the agents 'know' the same things; some agents may believe propositions which are not only not believed by other agents, but which the other agents are not even aware of. In addition, we allow the meanings of the propositions to vary from agent to agent. In reality communication is often ambiguous—what the speaker says and the hearer understands him to mean may be two different things. This may be a result of lexical ambiguity, incorrectly resolved anaphora or unrecognised irony. If we view the agents purely as formal systems, the possible interpretations for a proposition p are determined by the agent's other beliefs and intentions. Thus when an agent 'utters' p another agent will 'hear' "p"but may understand the speaker as meaning something other than p. For example, the speaker may believe that $p \supset q$ whereas the hearer may believe that $p \supset r$ (and possibly even $p \supset \neg q$). The hearer may have another belief, say p', which *does* correspond to what the speaker means by p, i.e. $p \equiv p'$—all interpretations which make p true for the speaker make p' true for the hearer. This notion of 'relative interpretation' allows us to model the misunderstandings which occur in the library dialogues, for example when one agent uses a concept which is unknown to the other agent or uses a term with a narrow technical or domain specific meaning and the other agent is only aware of the common or general meaning.[4]

(Saying propositions have different meanings may appear loose talk, but is not so here. What we are doing, as we are working with logical form representations of utterances rather than directly with utterances, is just treating a proposition as a generic stand-in for a family of more specific propositions each

---

[4] Note that understanding must be limited to the 'relevant' aspects of meaning. For example, how the agent came by its *own* belief in p is part of the meaning of p for the agent but is not relevant when considering whether two agents mean the same thing by p.

with their own individual, contextually determined, denotation. Our slightly crude language is conveniently simple and should not lead to any difficulties.)

We use three speech acts *tell*, *ask* and *answer*, which together account for many of the utterances found in the BBD transcripts.

Tell

The simplest speech act is *tell*:

(*tell* s h p)
Preconditions:
$I_s B_h p$
$B_s p$
$B_s \neg B_h p$
$\Box B_s B_h p$

Effects:
$B_s B_h p$
$B_h I_s B_h p$

where s and h are the speaker and hearer respectively, and p is a schematic variable denoting an attitude.

The preconditions for the *tell* act are that s intends h to believe that p, s believes that p and believes that h does not believe that p, and s predicts that the utterance will be successful in getting h to believe that p ($\Box$ is a modal operator denoting future belief). The effects are that s believes that h believes that p (s has faith in its prediction) and that h believes that s intends for h to believe that p.[5]

Ask

The *ask* speech act is a little more complicated:

(*ask* s h p)
Preconditions:
$I_s \exists x B_s p(x)$
$\exists x B_s B_h p(x) \wedge \neg B_s p(x)$
$\Box \exists x B_s p(x)$

Effects:
$B_s B_h I_s \exists x B_s p(u)$
$B_h I_s \exists x B_s p(x)$

The preconditions for the utterance are that s intends that there should be an x such that $p(x)$, where s believes that it currently does not know this but that h does, and s predicts that the utterance will result in s coming to know x. The

---

[5] There is a problem here if a primitive act has multiple effects, only one of which is intended. In this case it is only necessary that the desired effect is predicted to hold; for any additional effects we also require that they do not give rise to inconsistencies.

(guaranteed) effects are that s believes that h believes that s wants to know x, and that h believes that s wants to know x. As with *tell*, there may be other effects, for example h coming to intend to *tell* $p(x)$ to s (if h has nothing better to do).

It is the last effect which is critical for communication. h must come to believe that s wants to know something. If h has the information s requires and is feeling co-operative then h will communicate the information to s.

Answer

We also need a way to answer questions:

$$
\begin{array}{l}
(\mathit{answer}\ s\ h\ p) \\
\qquad \textit{Preconditions:} \\
\qquad\qquad I_h \exists x B_h p(x) \\
\qquad\qquad \exists y B_s p(y) \\
\qquad\qquad B_s \neg B_h p(y) \\
\qquad\qquad \Box B_s B_h p(y) \\
\\
\qquad \textit{Effects:} \\
\qquad\qquad B_s B_h p(y) \\
\qquad\qquad B_h I_s B_h p(y)
\end{array}
$$

The preconditions for the utterance are that the hearer intends that there should be an x such that $p(x)$, that the speaker believes there exists a y such that $p(y)$ and that the hearer does not believe this, and that the utterance will be successful in getting h to believe that $p(y)$. The effects are that the speaker believes that the hearer believes that $p(y)$ and that the hearer believes that the speaker intends for the hearer to believe that $p(y)$. For this to work, we must arrange that in establishing the precondition, y gets bound to a value. It is this act which makes asking questions computationally effective. Unlike Allen's INFORMIF and INFORMREF which are defined in terms of KNOWIF/KNOWREF in their effects, it turns an existentially quantified request $I_h \exists x B_h p(x)$ into a fully instantiated belief to be passed back to the agent which asked the question.

Propositions are handled in the same way as for *ask* acts.

4.2.3 Successful Speech Acts

The effects of the speech acts defined above are guaranteed. Indeed communication is only possible because the hearer is guaranteed to believe the speaker's intent. What the hearer does with this information depends on what else it believes. In particular the hearer may *adopt* the communicated attitude itself.

We thus distinguish between successful communication and a successful speech act. Following, e.g. (Levinson 1983), we define *successful communication* as the speaker's communicative intention becoming mutual knowledge to both the speaker and hearer. In this sense, the speech acts defined above always result in successful communication (so long as there are no misunderstandings). For example, the effects of the *tell* speech act include $B_s B_h p$, which is an approximation to the speaker's half of mutual knowledge, and $B_h I_s B_h p$, which is an approximation to the hearer's half of mutual knowledge.

We define a *successful speech act* (i.e. one in which the illocutionary act is successful) as one in which the speaker's communicative intent is achieved, for example that the hearer comes to believe that p or intend that q, and does so as a result of the speaker's utterance. More precisely we say that a speech act $(S\ s\ h\ \phi)$ is successful iff

$$\exists K \in \mathcal{K}_h,\ \phi \notin K \ \text{ and } \ \forall K \in \mathcal{K}_h^+\ I_s\phi,\ \phi \in K$$

i.e. if ϕ, the attitude the speaker intends the hearer to adopt, is not a member of the hearer's belief and intention state $\mathcal{K}_h$ prior to the speaker's utterance, but is a member after revision by the speaker's communicated intention $I_s\phi$.[6] For example, if an agent performs the speech act

$$(\mathit{tell}\ s\ h\ p)$$

the result of the utterance arises when the hearer revises it beliefs by

$$B_h I_s B_h p$$

If h believes the speaker is honest, it will come to believe (i.e. its new belief state will contain)

$$B_h B_s p$$

We can make this a guaranteed effect of the *tell* speech act if we are willing to assume that the agents are honest, however this is not essential. If h has no conflicting beliefs about p (e.g. belief in $\neg p$), it will also come to believe

$$B_h p$$

all other things being equal, and the speech act will have been successful.

4.3 A Typology of Communicative Outcomes

It is not enough to define speech acts in themselves. We also have to define their consequences in terms of our specific characterisation of belief revision. The aim of a speech act is to change another agent's beliefs or intentions. What change, if any, actually occurs is is predicted by the theory of belief and intention revision and depends on the hearer's current beliefs and intentions. While we can't say which outcome will occur without knowledge of the agents' cognitive states, we can enumerate the possible *outcomes* for some speech act S. Each

---

[6] Note that the first clause of the definition is necessary to capture the notion that the hearer comes to believe ϕ as a *consequence* of the speaker's utterance. Indeed we define the case in which $\forall K \in \mathcal{K}_h$, $\phi \in K$ prior to the speaker's utterance as an unsuccessful speech act, see above.

outcome is characterised by a particular distribution of beliefs between the two agents.[7]

We make a number of assumptions: that the agents are honest, i.e. they are not lying or dissembling (we have no way of representing a proposition presented as true but known to be false—such an agent would simply be incoherent); that the agents are consistent in their beliefs (if an agent has a pervasive belief in p and subsequently comes to have a pervasive belief in $\neg p$, then the agent has changed its mind about p—if this is not the case there is no point in asking the agent what it believes); and that the agents are experts in their respective domains (an agent may be mistaken about some aspect of their problem or about possible retrieval strategies, but they are honest and consistent in their error).

To simplify the discussion, we will assume that an agent is trying to change another agent's beliefs about something and that it has constructed a (partial) plan involving one or more speech acts based on its assumptions about the other agent's current belief state. From the analysis in the previous section, we can see that three conditions must hold for the speaker's intention to be realised:

1. correct ascription of the other agent's beliefs and intentions and their endorsements, i.e. the preconditions of the act must actually hold;
2. the communication must be successful, i.e. the speaker's intention must be successfully communicated; and
3. correct prediction of the effects of communicated beliefs or intentions, i.e. the hearer must actually adopt the attitude.

Whenever any of these is missing, the speech act will be unsuccessful. However correct prediction requires both correct ascription of relevant beliefs and that the communication be successful. The agent may be incorrect in its ascription of beliefs and endorsements to the other agent. For example, it may be wrong about what the other agent believes: it believes the other agent believes p when in fact it believes $\neg p$ or is uncertain. Such errors may render the communication pointless (we do not consider indirect speech acts). For example, I may believe, incorrectly, that you believe $\neg p$, and conceive a plan to convince you of p, a fact which you already believe. Moreover even if the agent gets the polarity of the belief right, it may get the endorsement wrong.[8]

In addition the speaker's intention must be successfully communicated. While the effects of the speech act are guaranteed and unambiguous, the relativisation of concept definitions means that failure and misunderstandings can occur if the hearer does not possess the concept, or means something else by it.

---

[7] The possible outcomes in the list below are not speech acts in the conventional sense as more than one agent is involved, nor are they discourse relations, since they are characterised by their effect on the agent's belief states. Unlike Searle's analysis of speech acts based on their felicity conditions, our classification is intensional, being based on changes in the agents' beliefs and intentions.

[8] This is inevitable to a certain extent, as all the subtleties of endorsement are reduced to 'strong' and 'weak' commitment to belief when beliefs are communicated, see chapter 5.

Finally the agent may also be incorrect in its prediction of the effects of the beliefs it communicates, i.e. in the belief state induced in the other agent. Even if the agent's model of the other agent's beliefs is accurate as far as it goes, the prediction of the effect of a communicated belief may be in error for two reasons. First, the beliefs ascribed to an agent are typically a small subset of the beliefs actually held by that agent, and consequently the agent may fail to revise its beliefs as predicted because of other unascribed beliefs. Secondly, an agent may use simple heuristics to predict how another agent will revise its beliefs instead of applying the theory of belief revision to their model of the agent. In both cases, the issue is one of efficiency. Even if it were possible for an agent to discover what another agent's beliefs are, a complete and accurate model is unnecessary in may cases and it is more efficient for the agent to concentrate on those beliefs and intentions of the other agent which are relevant in the current context. Similarly, planning is expensive. The agent may have to consider several candidate plans, and predicting the outcome of each alternative using a detailed simulation of the other agent's belief revision would incur a considerable computational overhead.[9]

4.3.1 Case Analysis

A speech acts fails when one or more of these conditions is not met. The failures we can detect depend on the point of view we adopt. We can identify three cases: first person (the speaker); second person (the hearer); and third person (the viewpoint of an omniscient observer with access to the belief and intention states of both the speaker and hearer).

First person

From the speaker's point of view there is only one possible outcome: success. The preconditions are guaranteed to hold (otherwise the utterance would never have been made). Problems arise when the result of the hearer's belief revision doesn't match the speakers belief about what this will be.

Second person

From the hearer's point of view there are two possible outcomes: success or failure relative to the hearer's interpretation of the speaker's intention, where correct interpretation at the level of logical form is guaranteed. However the intended meaning the hearer ascribes to the speaker may be incorrect. The recognition of the speaker's presumed intent (for example that the hearer should believe p), and that the speaker believes that its act has been successful, is guaranteed. However whether the hearer adopts the speaker's intention, i.e. comes to believe p, depends on what it prefers to believe after revision. In this sense the hearer's belief revision is autonomous.

---

[9] In fact agents do not use the ICM mechanism to predict the effects of communicated beliefs; they use a simple heuristic to predict the belief state which results from communication, see chapter 7.

The hearer believes the speech act has been successful if it believes the speaker's beliefs about it own beliefs are correct, and if the speaker has correctly predicted how it will revise its beliefs as a result of the speaker's utterance. There are two possible causes for failure: the speaker is incorrect about the hearer's beliefs; or the speaker's prediction about how the hearer will revise their beliefs was incorrect, i.e. the hearer finds the speaker's utterance insufficient grounds for revising its beliefs in the intended manner.

Third person omniscient

From the point of view of an omniscient observer there are three possible outcomes: success, failure and misunderstanding. Success and failure are similar to the second person case above, except that from the third person viewpoint there is no possibility of failure of communication. A speech act is successful (from this point of view) when the speaker intends that the hearer adopt an attitude ϕ, this intention is successfully communicated to the hearer (i.e. both speaker and hearer mean the same thing by ϕ), and the hearer adopts ϕ. A speech act is unsuccessful when the speaker intends that the hearer adopt an attitude ϕ, and this intention is successfully communicated to the hearer, but the hearer does not adopt ϕ.

The third case is the most interesting and arises when the speaker and hearer mean different things by the same proposition. We can identify two sub-cases of misunderstanding:

1. apparent success: the speaker believes p and believes the hearer believes p, but the hearer ascribes a different meaning to p (which the speaker does not believe) and also believes the speaker has this meaning; and

2. apparent failure: the speaker believes p and believes the hearer believes p, but the hearer ascribes a different meaning to p (which the speaker may or may believe) and as a result does not to believe 'p'. However the speaker has another belief, p', which has the same truth conditions as p and if substituted for p in the speaker's utterance would cause the speaker to agree.

The second type of misunderstanding can be viewed as a kind of failure (see above) and typically leads to immediate disagreement (if the belief is relevant to the task). The former kind of misunderstanding is more insidious. Both agents believe the communication was successful, but they believe different things. This may not become apparent until much later in the dialogue. In general, we define a *misunderstanding* as a conflict that can be resolved by a change in one agent's beliefs about the other's beliefs, and specifically by the hearer changing their beliefs about the speaker's beliefs, in particular the hearer's beliefs about what the speaker's utterance *meant*. A *disagreement* arises when the conflict can only be resolved by one or both of the agents changing their beliefs about something other than the other agent's beliefs. A disagreement is recognised as a misunderstanding when the hearer realises that only its beliefs about the speaker's beliefs are affected.

The third person viewpoint allows us to look 'inside' the agents to see what they cannot. While this viewpoint is useful in analysing failures of communication, it does not determine an agent's behaviour since it is in principle not available to any agent. Rather the agent's behaviour is determined by the first person viewpoint (for the speaker) and the second person viewpoint (for the hearer). Note that in this model, neither agent can detect a misunderstanding immediately (by definition), since this requires knowledge of both agent's cognitive states as opposed to the beliefs the agents ascribe to the other agent. From the limited viewpoint of the agents, a misunderstanding will appear initially as either a successful or unsuccessful speech act. However they can come to approach this omniscient viewpoint as a result of dialogue, by building up models of each other's beliefs.

From the hearer's point of view, which is the one we are interested in since it determines the hearer's response to the speaker's utterance, there are three cases for each speech act.

In all these cases, the speech acts work as intended, i.e. the speaker's intent is recognised, but in some cases the speaker has got the hearer's belief state wrong, and as consequence the hearer 'does the wrong thing' with the content of the utterance (from the speaker's point of view).

4.3.2 Successful Plans

In general, an agent's plan to change the beliefs and/or intentions of another agent will succeed so long as

1. the agent is correct about what the other agent currently believes, both in terms of the polarity of its beliefs (believed, uncertain, disbelieved) and their endorsement; and

2. each step in its plan is successful, i.e. the agent's intention is successfully communicated and it changes the other agent's beliefs in the predicted ways.

For example, suppose John's goal is for Mary to believe some proposition r, and John evolves a plan to achieve this as follows:

	Speech Act	Predicted Effect
1.	$(\textit{tell John Mary}\ p)$	$B_{Mary}p$
2.	$(\textit{tell John Mary}\ q)$	$B_{Mary}q$
3.	$(\textit{tell John Mary}\ p \wedge q \supset r)$	$B_{Mary}p \wedge q \supset r \wedge B_{Mary}r$

If John has made errors in either ascription or prediction of the effects of a plan step, that plan step will fail and John must replan to sort out the problem (or we abandon the plan altogether if the goal is unachievable). Suppose, for example, John starts to execute the plan and gets as far as step 2:

	Speech Act	Predicted Effect	Result
1.	$(\mathit{tell\ John\ Mary}\ p)$	$B_{Mary}p$	$B_{Mary}p$
2.	$(\mathit{tell\ John\ Mary}\ q)$	$B_{Mary}q$	$B_{Mary}\neg q$
3.	$(\mathit{tell\ John\ Mary}\ p \wedge q \supset r)$	$B_{Mary}p \wedge q \supset r \wedge B_{Mary}r$	

John's prediction that Mary would believe q after he told her q turns out to have been incorrect. John's beliefs about Mary's beliefs were incorrect, or his prediction of how she would change her beliefs after being informed of q was incorrect, and John must therefore re-plan to convince Mary of q, perhaps by presenting a justification for q or perhaps by devising a new plan to convince Mary of r. John uses feedback from the failure to revise his model of Mary.[10]

A plan consisting of a chain of speech acts is successful if it achieves the intended change in belief. Any successful chain must be terminated by a successful revision. For example, a *tell* (prompted by a failure of ascription or prediction) will itself only be successful if the new ascription/prediction is correct.

For dialogue management it is important to allow not only for the fact that a plan may need a chain of several speech acts to achieve the intended change of belief, but that interactive 'interruption' may occur before the complete chain is communicated. This is illustrated by the 'foci', i.e. dialogue segments or sequences of topic-related turns, that occur in the BBD dialogues, as analysed in Brooks' and Daniels' theses (Brooks 1986, Daniels 1987). Moreover when we study the dialogue transcripts, we find a number of recurring patterns or unit types in the dialogue. Although the topic of each unit instance is different, the pattern of the agent's beliefs and their revision is common across instances of the unit. This distribution of beliefs and intentions can be viewed as the 'cause' of a particular kind of belief revision, and a better understanding of these causes would be of assistance in modelling the dialogues.

4.4 Maintaining Coherence in Dialogue

Discourse, or dialogue, coherence is a large and complex subject, and we cannot consider it in detail here. It is however necessary to incorporate a minimal treatment of dialogue coherence in our modelling, as this is required both by our general interest in effective communicative action as one form of agent behaviour and by our specific literature-seeking task. We also referred, earlier, to dialogue control as a problem for the original BBD version of the librarian expert system that needed to be tackled.

We are not concerned, in defining dialogue coherence for our purposes, with presentational cohesion at the utterance level. Dialogue coherence presupposes relatedness of propositional content, but we define *dialogue coherence* here in terms of functional relations between communicative acts. Dialogue coherence is required both for communicative efficiency, since contextual reference facilitates semantic interpretation, and for effectiveness, so communicative purposes can be properly recognised, e.g. that one dialogue contribution is a follow-up to a previous one. Maintaining dialogue coherence thus depends not only on

---

[10] At present an agent can't revise its prediction heuristics.

maintaining content linkage in the narrow sense, but on maintaining functional relations, as in answering a question. We therefore assume that our agents, as social beings, possess and apply rules ensuring dialogue coherence. (It is unfortunate that "coherence" is the normal term for dialogue when we are already using it for the relations between beliefs, but there is no obvious alternative - since "connectivity" is similarly preempted, - so we assume that where we do not explicitly label it "dialogue coherence" this meaning of "coherence" will be clear from the context.)

Now, given the way we are operating with speech acts and have identified the possible communicative outcomes for a speech act, we can define coherent dialogue in terms of whether and how an agent should respond to a speech act, for instance realising that the utterance it receives is the answer to a question it asked a minute ago. Our simple and uncontroversial approach, in the spirit of dialogue games (Kowtko, Isard and Doherty 1992, Carletta 1992), is as follows.

We assume that agents take conversational *turns* in strict rotation. For each of the three communicative outcomes for each speech act we can identify one or more *legal responses* (in the spirit of dialogue games). So long as the hearer makes a legal response, the dialogue will be coherent. We distinguish between *continuation* and *repair* responses. If the hearer believes the speech act to have been successful it makes a continuation response e.g. answers a question or moves to a new topic (continuing the current segment or starting a new segment at the same level). If it believes the speech act to to have been unsuccessful, it makes a repair response and we get a new segment. The legal continuations are a function of the dialogue context and the legal repairs are a function of the perceived failure. For example, if the act is a question, the only legal response might be to answer the question and a legal repair might be a statement that the hearer doesn't know the answer; in neither case should the hearer simply switch to a different topic or answer a different question for example.[11] Similarly, a failure of ascription might lead to a an attempt to correct the speaker's model of the hearer, whereas failure of prediction might result in an attempt to resolve the conflict. Note that misunderstandings can only become apparent after a minimum of two turns. For example a typical case might be an apparent failure of ascription, leading to an attempt to repair the speaker's model of the hearer, at which point the speaker realises the hearer has misunderstood. In this sense, a misunderstanding is a special case of failure. Transitions from game to game are determined by the agent's goal structure as specified by the BBD model (see chapter 6).

We assume that the for any given set of speech acts there is a transition network or table which gives the legal continuations/repairs as a function of the previous act (and the dialogue context). The acts could be implemented as a set of primitives, e.g. question/answer, or in terms of some set of primitive acts such as *tell*, together with the appropriate machinery to keep track of the dialogue context, for example to distinguish a continuation from a repair. The transition table for our simple three act model, outlined above, is:

---

[11] In reality, of course, there are other possibilities, for example it might be permissible to ask why the question was asked in the first place.

Speech Act	Continuation	Repair
tell	*tell*	*tell*
	ask	*ask*
answer	*tell*	*tell*
	ask	*ask*
ask	*answer*	*tell*

While repair responses may give rise to a new segment, they always preserve the subject of the dialogue. Continuation responses on the other hand always result in another segment at the same or higher level but may change the subject of the dialogue.

We can use these observations as the basis of a simple model of task-oriented dialogue structure which we believe is adequate to model the structure of the dialogues in the BBD transcripts.

We get a new *segment* whenever agent's disagree or whenever the subject of the dialogue changes. We define the *focus* of the new segment is the disputed belief or the new subject which has just been introduced. The focus of a segment is often defined, in the literature on discourse structure, as "the prime candidate for pronominal reference". However we are not interested in using segmentation/focus to control the resolution of anaphora: we want to control inference and recall in such a way as to ensure that the resulting dialogue is coherent. For our purposes this non-standard approach to focus seems quite appropriate, and indeed a thoroughgoing analysis of the notion of focus, or centre, of attention (cf Grosz, Joshi and Weinstein (in press)) as a whole would almost certainly allow for our interpretation of focus as appropriately more substantive than just that of prime reference candidate.[12]

It would be difficult to extend this rather limited model of discourse structure to incorporate plan recognition. At present the system makes no use of prototype plans or scripts in the conventional sense when attempting to infer the intention of other agents or in deciding what to do. While it would be possible to represent plans as objects in the database, there is no easy way to link these to the planning process which is driven by the prime belief(s). There is no concept of a 'standard plan' which is appropriate in a given type of situation, where all that has to happen is that the slots in the plan are filled in. It is clearly desirable, in the longer run, to develop a more comprehensive and sophisticated treatment of plans in relation both to discourse and to our view of belief revision (cf e.g. (Cohen et al. 1990, Lambert and Carberry 1991)). The obvious comparison for our treatment of dialogue is with (Carletta 1992), but our approach is different because for our particular investigation we have to keep the operation of belief revision untrammelled. Thus while we cannot precompile the possible response structure and therefore need to have some discourse constraints, we have deliberately kept these minimally constraining.

---

[12] It would be nice to equate our notion of 'focus' with that of the current *prime belief* (see chapter 5), but at present the prime belief is defined as the most preferred intention.

Chapter 5

Implementing Belief and Intention Revision

In this chapter we flesh out the theory presented in chapter 2 in enough detail for computational implementation. We show how, given some new input from an observation or communicated by another agent, an agent decides whether and how to revise its existing beliefs to accommodate the new information, and how the agent chooses what to reject when a conflict arises. At this more detailed level, our theory of belief revision must provide three things: a way of representing beliefs; a set of algorithms or heuristics for preferring some revisions to others; and a mechanism for applying these criteria to identify the preferred set of beliefs. Specifically, as beliefs form webs of related beliefs, what is needed is a means of handling the way individual beliefs contribute to the structure and solidity of a whole web, and a means of taking account of the propagation effect of changes at the level of individual beliefs, whether the change modifies an existing belief, adds a new one, or deletes an old one.

Implementing this theory of belief revision computationally therefore requires a specific mechanism for constructing and evaluating all the belief sets which constitute alternative ways of dealing with some new input. To do so we have to extend the theory presented in chapter 2 in a number of ways, most notably the introduction of prediction endorsements.[1] In the remainder of this chapter we present the Increased Coherence Model (ICM): this is an implementation of our framework for autonomous belief and intention revision based on de Kleer's ATMS (de Kleer 1986a). We also describe the implementation of the inferential and planning capabilities required by the agent, about which the theory says nothing.

5.1 Agent Architecture

An agent consists of two parts: a *message interception unit* (MIU) and a *cognitive unit* (CU).

The MIU, which operates concurrently with the CU, handles the mechanics

---

[1] Prediction endorsements are discussed in detail in chapter 7.

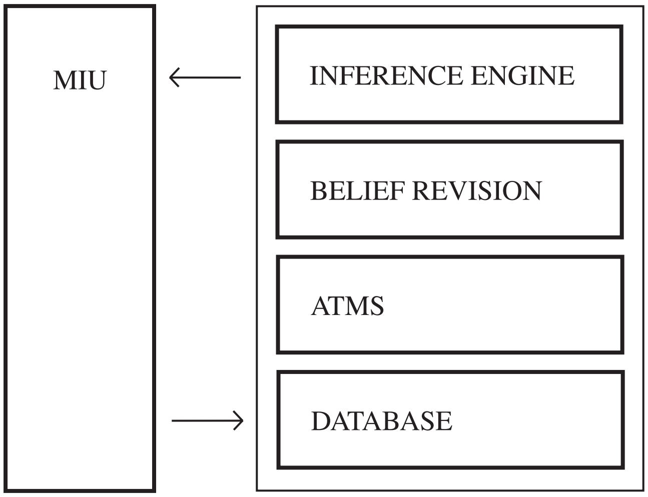


Figure 5.1: Agent Architecture and flow of information

of communication between agents (i.e. the TCP socket connections). It receives outgoing messages from the CU and stores them chronologically until they can be passed onto other agents. It also intercepts communications from other agents and posts these, in order of interception, onto a message board which is local and private to the agent. The CU reads this board at intervals and processes all the messages it finds there.

The CU can be further subdivided into four main components or 'layers': a database, an assumption-based truth maintenance system (ATMS); an attitude revision component; and an inference engine.[2]

- The *Database* records the agent's *cognitive state*—the agent's current beliefs and intentions: what is believed, intended, or uncertain, and the commitment the agent has to these attitudes.
- The *ATMS* computes all possible consistent sets of beliefs and intentions using given and inferred inconsistencies between beliefs, intentions and beliefs and intentions.[3]
- The *Attitude Revision* component computes a preference ordering over the belief and intention sets generated by the ATMS and the commitment to the most preferred attitudes.
- The *Inference Engine* uses planning, belief and prediction rules of the form $\forall x P(x) \supset Q(x)$ which operate on the domain information stored in the database to infer either one or more new beliefs, a new plan step or

---

[2] To simplify the exposition, we have simplified the architecture somewhat. In reality, the boundaries between the layers are rather more blurred than the simple picture outlined here. For example, the ATMS uses information from the Attitude Revision layer to avoid generating belief sets which will never be preferred. We return to the issue of optimisation in chapter 9.

[3] For a description of the ATMS see (de Kleer 1986a, de Kleer 1986b).

a new prediction. Inferences and their *justifications*, for example minor premises, are recorded in the database.

Given a database, the agent's task is to determine its beliefs and intentions and its commitment to these. To do this it compares the relative coherence and endorsement of consistent sets of possible attitudes. Since the agent chooses its intentions based on its beliefs and on its commitment to these beliefs it is evident that the agent's commitments to its beliefs must be determined prior to computing the preference orderings of its intentions. For example, suppose an agent has a strong belief in p and a plan to convince another of p. Later, if x revises its commitment in p (to weak) this might effect the predicted outcome of the plan and consequently lead to plan revision (i.e. the abandonment of the plan). In effect, the agent first decides what it believes (its belief state) and the decides what to do (its intention state) based on what it believes. The belief state of an agent is defined as the most preferred of the sets of possible beliefs entertained by the agent. For each proposition or state of affairs, s, of which the agent is aware, the agent must decide whether it believes that s or disbelieves that s and how strongly it is committed to its belief. These beliefs form the basis of the agent's intention sets which represent the agent's various intentions and and the alternative ways of achieving these intentions being entertained by the agent. The most preferred intention sets constitute the agent's intention state and form the basis for action.

The *agent action cycle* can therefore be summarised as: first compute the most preferred belief sets and the agent's commitment to the beliefs they contain, and then determine a rational course of action based on these beliefs and the agent's intentions. Although this is an oversimplification, it serves to motivate the discussion below. In the remainder of this chapter we describe the agent architecture and action cycle in more detail before presenting a worked example of how the agent performs a simple task in the blocks world domain. In what follows, our aim has been to provide enough information to allow anyone reading the report to implement the architecture. We have adopted this approach for two reasons: we claim a certain success for the model presented below and without a detailed description of the implementation it would be impossible for the interested reader to repeat our experiments and validate our claims or to test the model in other domains; and without a reasonably detailed understanding of the implementation it is impossible to fully appreciate the implications of the problems discussed in chapter 9.

5.2 Representing the Cognitive State

The notation used in chapter 2 on the theory of belief revision is inadequate to describe the implementation. We therefore introduce a new notation which records the agent's *attitude* towards a proposition or state of affairs, the *time* at which the agent held the attitude and the *endorsement* of or *commitment* to the attitude. Unlike the notation employed in chapter 2, our new notation is intended both as a representation language for an agent's beliefs and as a means of allowing us to analyse belief conflicts between agents in a straightforward way. We therefore retain the 'agent' argument in each belief and intention, even though the notation is subjective or 'agent centred'. In particular, there is no notion of 'objective truth'—it is impossible to express that a state of affairs s is true 'in the world'. All the representation allows us to express is how things appear to a particular agent. Within this framework, we can only represent the agents' beliefs about the state of the world and the beliefs of other agents. To represent our (privileged) knowledge of the world, for example to explain the failure of an agent's plan due to mistaken beliefs about the world, we use an informal meta-language.

This new notation also introduces a number of attitudes which have no counterpart at the theoretical level, such as possible belief *p-bel* and possible intention *p-int*, as it is more convenient to distinguish the agent's attitude to individual propositions rather than to sets of such propositions.

5.2.1 Belief-Type Attitudes

Although the treatment of beliefs and intentions within our framework is essentially the same, it is convenient for implementation purposes to distinguish between beliefs and intentions. At the implementation level, we have the following data structures representing beliefs.

(*p-bel* $x\,s\,t\,e$) : x believes that s may be true at time t with endorsement e. (Alternatively there is a possible world in which x believes that s at time t with endorsement e.)

(*bel* $x\,s\,t\,c$) : x believes that s is true at time t with commitment c. (Alternatively x believes that s is true at time t in all possible worlds with commitment c.)

where s is either a state of affairs, i.e. a triple of the form ($p\,l\,t$) and read as 'p is true at location l and time t', or the pervasive belief of another agent (pervasive is defined below). p is either an atomic proposition or an n-ary predicate expression, possibly containing existentially quantified variables.[4] For example ((*on block1 block2*) *table yesterday*) denotes the state of affairs in which *block1* was on *block2* on the table yesterday, and ((*exists !x on !x block1*)) *table yesterday*) denotes the state of affairs in which there was something on *block1* on the table yesterday. The '*!*' operator can be thought of as introducing an existential quantifier: (*exists !x* (*on !x block1*)) is equivalent to $\exists x$(*on* x *block1*).[5]

A *belief set* is a maximally consistent set of p-beliefs. Belief sets contain only possible beliefs and logical combinations of these. (Specifically they do not contain pervasive beliefs, introspective beliefs, intentions or pervasive intentions.)

---

[4] We assume that propositions make no reference to a particular time or location.

[5] Our use of existentially quantified variables is slightly non-standard, in that the variable can be qualified by a list of possible values. For example,

(*p-int* x (*int* y (*exists !x* (*greek roman*) (*bel* x (*pd !x*)))))

is interpreted to mean that agent x possibly intends that y intend that x should believe that either *greek* or *roman* or both are problem descriptors for the current retrieval problem.

Whereas an agent may arrive at possible beliefs about other agents' beliefs as a result of observation or communication (both of which are fallible), it must *compute* its own beliefs from its possible beliefs (see section 5.2.2 below). For example we write

(p-bel x (bel y s t′ strong) t 2c-pos)

to mean that x possibly believes that y strongly believes that s with endorsement *2c-pos*,[6] but it makes no sense to write

(p-bel x (p-bel x s t′ e′) t e)

since an agent never entertains a possible belief about a possible belief or

(p-bel x (p-bel y s t′ e′) t e)

where $x \neq y$, since x has no access to y's p-beliefs. Nor can we state something like

(p-bel x (bel x s t' c) t e)

Although this makes sense (for example the agent may reasonably consider coming to believe s when it currently believes $\neg s$), at this stage the agent has yet to determine what its beliefs are.

This notation allows us to distinguish between the time at which a particular proposition is believed to be true and the time at which the agent comes to believe that the state of affairs is true in the world. For example we can write

(*p-bel x* ("it is raining" Tuesday Cambridge) Monday *2c-pos*)

to represent that on Monday x possibly believed that it will rain in Cambridge on Tuesday. At present, the system uses two time points, *now* and *eventually*. Belief sets contain only p-beliefs which the agent currently holds or has held at some time in the past, since what the agent will believe depends on what it currently believes.

There are seven endorsement types: *spec*, *def*, *1c-pos 1c-neg 2c-pos*, *2c-neg* and *hypoth*. *Spec* and *def* apply to *a priori* or innate information, *1c-pos* and *1c-neg* apply to first hand experience and *2c-pos* and *2c-neg* apply to communicated information. Each possible belief can have multiple endorsements of the same type (for example a belief may have two *2c-pos* endorsements from different sources). An hypothesis is a belief with no other endorsement. There is a preference ordering over endorsements, denoted $>_e$, with *1c-pos* being most preferred and *hypoth* the least preferred. The full ordering is given by

1c-pos $>_e$ *2c-pos*, spec $>_e$ *1c-neg*$>_e$ *2c-neg*, *def*$>_e$ *hypoth*

---

[6] When the '*!*' operator is used within the context of an ascribed belief, for example (*p-bel x* (*bel y* (((*exists !u* (*on !u block1*)) *table yesterday*) *t′ strong*) *t 2c-pos*), it functions analogously to Allen (1987)'s 'KNOWREF' operator, i.e. x believes that y (strongly) believes that there was something on *block1* on the table yesterday, but doesn't know what.

An additional endorsement, *definite*, is used mainly for ascribed beliefs. A *p-bel* endorsed *definite* has the properties of a premise; it appears in every belief set. The difference being that a *definite* proposition can cease to be such. Whereas a premise can never become a non premise and be disbelieved the *definite* endorsement can be revoked and the proposition may then be disbelieved. *definite* endorsements increase the efficiency of the ATMS mechanism by distinguishing propositions which cannot appear in ATMS *candidates* (where a candidate is a set of beliefs which, if removed from an inconsistent belief set, leaves it consistent).

There are three levels of commitment: strong, weak and uncertain. The *strength* of an agent's belief in a proposition s is computed using a heuristic which involves endorsing the negation of the belief $\neg s$ with an extra *2c-neg* endorsement. If this results in the agent's belief in the proposition being undermined, then the statement is only weakly believed, otherwise it is strongly believed. Uncertainty is effectively the midpoint between a strong belief in s and a strong belief in $\neg s$; and arises when the agent has reason to believe both s and $\neg s$. An agent is said to have a *pervasive* belief in s if the agent's commitment to s is not uncertain.

If an agent x, communicates a strongly held belief s to agent y, then agent y will believe that x believes that s. This belief is endorsed as *definite*, i.e. agent y definitely believes that x believes that s, represented as (*p-bel y* (*bel x s t' c*) *t definite*). If x's belief in s was strong, i.e. if $c =$ *strong*, then the resulting justification for y's belief in s as a consequence of x's belief in s is endorsed *2c-pos*, i.e. (*p-bel y s t 2c-pos*). However if x's belief in s was weak, the justification for y's belief in s is endorsed *2c-neg*. This allows a primitive form of belief modelling; an agent can believe that another agent believes a proposition while remaining uncommitted itself.

More complex beliefs are formed by compounding these terms using the logical connectives $\wedge \vee \neg$ and $\supset$. For example, to express the fact that an agent x has a possible belief that s and a possible belief that y believes that s', we write

$$(p\text{-}bel\ x\ s\ t\ e) \wedge (p\text{-}bel\ x\ (bel\ y\ s'\ t\ c)\ t\ e)$$

and that if an agent has a possible belief that s it may have a possible belief that s' we write[7]

$$(p\text{-}bel\ x\ s\ t\ e) \supset (p\text{-}bel\ x\ s'\ t\ derived)$$

Note that whatever the endorsement of the antecedent, the consequent of a rule is always endorsed as *derived*. For example, future p-beliefs and beliefs (i.e. beliefs of the form (*p-bel x s t e*) where $t > t_c$ where t_c is the current time)

---

[7] Note that since x is resource bounded we cannot say that x *does* have a possible belief that q, since x may not have inferred that q. For an ideal agent we also have that

$$\begin{aligned}(p\text{-}bel\ x\ p \wedge q) &\equiv (p\text{-}bel\ x\ p) \wedge (p\text{-}bel\ x\ q)\\ (p\text{-}bel\ x\ p \vee q) &\equiv (p\text{-}bel\ x\ p) \vee (p\text{-}bel\ x\ q)\end{aligned}$$

are always endorsed derived, as they can only arise as a result of prediction by the agent.[8]

In cases where the location is understood and the endorsement of the belief does not matter, we write $(\textit{p-bel}\, x\, p)$ for $(\textit{p-bel}\, x\, (p\; l\; t)\; t'\; e)$ where $t = t' = t_c$ and $(\textit{f-p-bel}\, x\, p)$ when either $t > t_c$ or $t' > t_c$. Similarly, in cases where the agent's commitment to the belief does not matter, we write $(\textit{bel}\, x\, p)$ for $(\textit{bel}\, x\, (p\; l\; t)\; t'\; c)$ where $t = t' = t_c$ and $(\textit{f-bel}\, x\, p)$ where either $t > t_c$ or $t' > t_c$.

5.2.2 Intention-Type Attitudes

In addition, we have the following datastructures representing intentions.

$(\textit{p-int}\; x\; s\; t\; e)$: at time t x has a possible intention that s with endorsement e. (Alternatively there is a possible world in which x intends that s at time t with endorsement e.)

$(\textit{int}\; x\; s\; t\; c)$: at time t x intends that s with commitment c. (Alternatively x intends that s at time t in all possible worlds with commitment c.)

Intentions are endorsed *desire-pos* or *desire-neg* or *hypoth*. An intention is endorsed *desire-pos* if the agent believes that the communicating agent strongly desires the goal, and *desire-neg* if the agent believes that the communicating agent only weakly desires the goal. As with beliefs, an endorsement is a reason for preferring an intention over its negation. Intended actions are endorsed either *effort-pos* or *effort-neg*, depending on how much effort is required to perform the action given that the preconditions of the action are true. The endorsement associated with intentions is related to a heuristic assessment of their expected outcome. This contrasts with those associated with beliefs, where it is the source of the beliefs that are represented. In general intended states which have a high expected utility and intended actions which have low expected effort and uncertainty are preferred.

As with beliefs, there is a preference ordering over intention endorsements, denoted $>_i$, with *desire-pos* being most preferred and *effort-pos* being least preferred. The full ordering is given by

$$\textit{desire-pos} >_i \textit{desire-neg} >_i \textit{hypoth} >_i \textit{effort-neg} >_i \textit{effort-pos}$$

As above, where the location is understood and the endorsement of a possible intention is unimportant, we write $(\textit{p-int}\, x\, p)$ for $(\textit{p-int}\, x\, (p\; l\; t)\; t'\; e)$ where $t = t' = t_c$ and $(\textit{f-p-int}\, x\, p)$ when either $t > t_c$ or $t' > t_c$. Similarly, in cases where the agent's commitment to the intention is unimportant, we write $(\textit{int}\, x\, p)$ for $(\textit{int}\, x\, (p\; l\; t)\; t'\; c)$ where $t = t' = t_c$ and $(\textit{f-int}\, x\, p)$ where either $t > t_c$ or $t' > t_c$.

Further, we introduce four equivalence relations relating (p-) beliefs and intentions

$$(\textit{p-bel}\, x\, (\textit{bel}\, x\, p)) \quad \equiv \quad (\textit{p-bel}\, x\, p)$$

---

[8] In reality, the system maintains a list of endorsements for each possible belief, so that, for example, belief in a proposition or state of affairs can be endorsed both *1c-pos* and *derived*.

$$\begin{aligned}(p\text{-}int\, x\, (int\, x\, p)) &\equiv (p\text{-}int\, x\, p)\\ (p\text{-}int\, x\, (bel\, x\, p)) &\equiv (p\text{-}int\, x\, p)\\ (p\text{-}bel\, x\, (int\ x\ p) &\equiv (p\text{-}int\, x\, p)\end{aligned}$$

These are analogues in our system of Hintikka's axiom of positive introspection (Hintikka 1962).

5.2.3 Database

The database contains conventional ATMS data structures: nodes (representing beliefs and intentions) and justifications (representing their derivational supports, if any) augmented with endorsement information. For n attitudes:

$$Database \overset{def}{=} \{n_1, n_2, \ldots, n_n\}$$

Each node n_i has the form $\langle p_i, l_i, j_i, e_i \rangle$ where l_i is the node label, j_i is its justifications and e_i the endorsements. The ATMS datum p_i is one of the propositional attitudes described above, i.e. one of

(*p-bel x p*)	possible belief that p
(*f-p-bel x p*)	possible future belief that p
(*bel x p c*)	commitment to p (c is either *strong*, *weak* or *uncertain*)
(*bel x p*)	belief that p
(*p-int x p*)	possible intention that p
(*f-p-int x p*)	possible future intention that p
(*int x p c*)	commitment to p (either *strong*, *weak* or *uncertain*)
(*int x p*)	intend that p

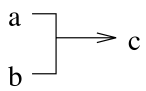
where the argument p can be either a belief-type or an intention-type attitude or a state of affairs. In all cases, note that the propositional content of the datum is unique, i.e. each proposition appears at most once for each attitude type.[9]

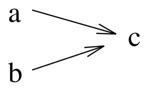
The ATMS maintains dependencies between 'innate' and derived beliefs by computing the *environments*, the consistent sets of innate beliefs or assumptions, in which a belief holds. For example, the expressions a, b, $a \wedge b \supset c$ and $b \supset d$ with c endorsed *2c-pos* and a *def* produce the following ATMS nodes

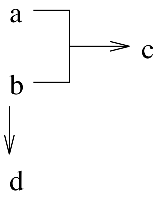
$$\begin{array}{llll}\langle\ \mathrm{a}, & \{\{\mathrm{a}\}\} & \{\{\mathrm{a}\}\} & \{def\}\ \rangle\\ \langle\ \mathrm{b}, & \{\{b\}\} & \{\{b\}\} & \{hypoth\}\ \rangle\\ \langle\ \mathrm{c}, & \{\{\mathrm{a,}b\},\{\mathrm{c}\}\} & \{\{\mathrm{a,}b\},\{\mathrm{c}\}\} & \{2c\text{-}pos\}\ \rangle\\ \langle\ d, & \{\{b\},\{d\}\} & \{\{b\},\{d\}\} & \{hypoth\}\ \rangle\end{array}$$

In this report justifications are often represented schematically by a network diagram. Conjunction is represented by a joined arrow and disjunction by multiple arrows. For example $a \wedge b \supset c$ is represented as

and $a \vee b \supset c$ is represented as







Thus the justification diagram for the above example looks like

In this implementation all attitude propositions are ATMS *assumptions* and the agent's reasoning is confined to maximal ATMS *contexts*.

With the exception of the endorsement field these nodes are identical to those of the ATMS. The endorsement field e_i is a sequence $< k_1, \ldots, k_n >$ of endorsements where each k_i is one of *definite*, *1c-pos* *1c-neg* *2c-pos*, *2c-neg*, *spec*, *def*, *hypoth* for belief-type datum nodes and *desire-pos*, *desire-neg*, *hypoth*, *effort-neg* and *effort-pos* for intention-type datum nodes. Each node may have many endorsements and these need not be of different types.

5.3 Deriving the Belief State

The action cycle begins with the agent computing its current belief state. The information in the database is used by the ATMS to compute the consistent sets of beliefs and intentions. A set is consistent relative to a database D when it contains no nogoods[10]

$$A_D \overset{def}{=} \{S \subseteq D \mid cons_D(S) \wedge \forall S' \subseteq D\ S \subseteq S' \supset \neg cons_D(S')\}$$

where

$$cons_D(S) \overset{def}{=} \neg\, \exists S' \in N_D \wedge S' \subseteq S$$

and N_D is the set of nogoods for the database D.

We require that the belief sets computed by the ATMS are closed under negation: if (*p-bel* x p) is present in the database then so must $\neg$(*p-bel* x p), and

---

[9] Note that all beliefs are ATMS assumptions.

[10] A nogood is a conjunction of assumptions which have been shown to entail falsity (de Kleer 1986a).

every belief set must contain either $(\textit{p-bel}\, x\, p)$ or $\neg\ (\textit{p-bel}\, x\, p)$ (i.e. $(\textit{p-bel}\, x\, p) \vee \neg\ (\textit{p-bel}\, x\, p)$ is true). For example, suppose the following statements are believed by an agent

$$
\begin{array}{r}
(\textit{p-bel}\, x\, p) \wedge (\textit{p-bel}\, x\, q) \supset \perp \\
\neg(\textit{p-bel}\, x\, p) \wedge (\textit{p-bel}\, x\, r) \supset \perp \\
(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, p) \supset \perp
\end{array}
$$

Without the principle of closure under negation the following would be belief sets:

$$
\begin{array}{l}
\{(\textit{p-bel}\, x\, q),\ (\textit{p-bel}\, x\, r)\} \\
\{(\textit{p-bel}\, x\, p),\ (\textit{p-bel}\, x\, r)\} \\
\{(\textit{p-bel}\, x\, q),\ (\textit{p-bel}\, x\, p)\}
\end{array}
$$

However, closure disallows $\{(\textit{p-bel}\, x\, q),\ (\textit{p-bel}\, x\, r)\}$ as a belief set. The reason for enforcing closure under negation becomes obvious when we consider an example. Suppose Steve is reasoning about Brian's nationality and believes Brian can be either Welsh or Scottish. Steve knows that Brian cannot be both Welsh and Scottish and he knows that if Brian was born in Edinburgh then he must be Scottish.

$$
\begin{array}{l}
(\textit{p-bel}\, \textit{Steve}\, (\textit{Welsh Brian})) \wedge (\textit{p-bel}\, x\, (\textit{Scots Brian})) \supset \perp \\
\neg(\textit{p-bel}\, \textit{Steve}\, (\textit{Welsh Brian})) \wedge (\textit{p-bel}\, x\, (\textit{born Brian Edinburgh})) \supset \perp
\end{array}
$$

Now, Steve has good reason to believe that Brian is Welsh and that he was born in Edinburgh. He was told both these facts and they are both well endorsed. Without closure Steve would generate a belief set containing both (*Welsh Brian*) and (*born Brian Edinburgh*) which is clearly nonsense.[11]

The set of possible belief sets is therefore given by the cross product of the possible and predicted belief attitudes less the inconsistencies:[12]

$$
\begin{array}{lcl}
(\textit{p-bel}\, x\, p) \wedge (\textit{p-bel}\, x\, \neg p) & \supset & \perp \\
(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, p) & \supset & \perp
\end{array}
$$

---

[11] For reasons of efficiency not all beliefs are closed under negation at the implementation level, but closure is guaranteed whenever a proposition appears in a candidate. Preferred sets are computed by first finding the least endorsed candidates and then taking the set difference with the database for each candidate. New attitude polarities are generated only when the attitudes appear in the least endorsed candidates (i.e. when an attitude is no longer believed). However, the behaviour of the system is not affected by this and there is no loss of information: the absence of an attitude in a belief set means that the negation of that attitude holds in the set.

[12] Alternatively we can define the set of belief sets as the logical closure of each possible assignment of *True* and *False* to each proposition s and belief $(\textit{p-bel}\, x\, p)$ given the axioms:

$$
\begin{array}{rcl}
(\textit{p-bel}\, x\, p) & \supset & \neg(\textit{p-bel}\, x\, \neg p) \\
\neg(\textit{p-bel}\, x\, p) & \supset & \neg(\textit{p-bel}\, x\, \neg p) \vee (\textit{p-bel}\, x\, \neg p) \\
\neg((\textit{p-bel}\, x\, p) & \wedge & (\textit{p-bel}\, x\, \neg p)) \\
\neg((\textit{p-bel}\, x\, p) & \wedge & \neg(\textit{p-bel}\, x\, p))
\end{array}
$$

This gives three belief sets for each proposition p, namely

$$\begin{array}{rcl} (p\text{-}bel\,x\,p) & \wedge & \neg(p\text{-}bel\,x\,\neg p) \\ (p\text{-}bel\,x\,\neg\, p) & \wedge & \neg(p\text{-}bel\,x\,p) \\ \neg(p\text{-}bel\,x\,p) & \wedge & \neg(p\text{-}bel\,x\,\neg p) \end{array}$$

Note that the ATMS computes consistent sets of beliefs (and intentions) given the justifications encountered thus far, *not* with respect to the logic of the axioms

5.3.1 Computing the Preferred Sets

Given a set of consistent belief sets, the agent must then determine which of these sets are most preferred. We can define the preferred sets Pf_D^X of a database D with ordering X as:

$$Pf_D^X \stackrel{def}{=} \{S \in A_D \mid \forall S' \in A_D\ S \geq_X S'\}$$

The ordering X is defined over endorsement $>_e$, $mc >_c$) and then minimal change $>_m$

$$S \geq_{e,c,m} S' \stackrel{def}{=} [S >_e S'] \vee [S =_e S' \wedge S >_c S'] \vee [S =_e S' \wedge S =_c S' \wedge S \geq_m S']$$

The Endorsement Ordering $>_e$

The endorsement preference ordering $\geq_e$ is computed using the following algorithm. We construct two sets of endorsements e and e' by concatenating the endorsements for all the propositions in the sets S and S'. For belief sets the endorsement ordering relations are:

1. if e contains more *definite* endorsements than e' then $S \geq_e S'$, else if e' contains more *definite* endorsements than e then $S' \geq_e S$ else

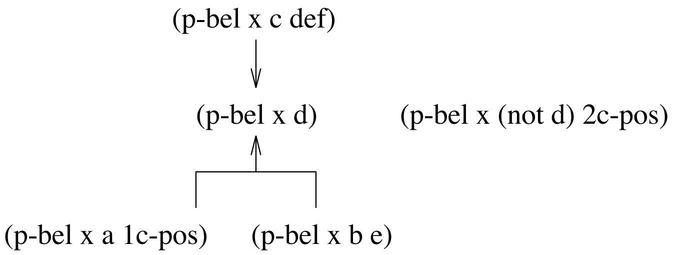
2. if e contains more *1c-pos* endorsements than e' then $S \geq_e S'$, else if e' contains more *1c-pos* endorsements than e then $S' \geq_e S$ else

3. if e contains more *2c-pos* and *spec* endorsements combined than e' then $S \geq_e S'$, else if e' contains more *2c-pos* and *spec* endorsements than e then $S' \geq_e S$ else

4. assign values to *1c-neg*, *2c-neg* and *def* endorsements in e and e'

$$\begin{array}{rcl} \mathit{1c\text{-}neg} & = & 3 \\ \mathit{2c\text{-}neg} & = & 2 \\ \mathit{def} & = & 1 \end{array}$$

   and sum these in each to find the value of the endorsements; if the value of e is greater than that of e' then $S \geq_e S'$ else if the value of e' is greater than that of e then $S' \geq_e S$ else



Endorsements are not propagated through to derived attitudes. However, the corrigibility of a derived attitude is linked to the endorsed attitudes through the ATMS justification network. In order to disbelieve an attitude the agent must disbelieve all proofs for that attitude. Ultimately, this means that the agent must disbelieve endorsed attitudes. The more endorsed these attitudes the harder it is to disbelieve the derived attitude. For example

The possible belief sets are:

Belief Set 1	Belief Set 2	Belief Set 3
(*p-bel* x $\neg d$ *2c-pos*) (*p-bel* x a *1c-pos*)	(*p-bel* x $\neg d$ *2c-pos*) (*p-bel* x b e)	(*p-bel* x d) (*p-bel* x a *1c-pos*) (*p-bel* x b e) (*p-bel* x c *def*)

It is evident that if $e = def$ then Belief Set 2 is preferred and d is not believed. However, if $e =$ *2c-pos* then Belief Set 3 is preferred and d is believed.

This approach exhibits the weakest link property: in order to disbelieve an attitude it is necessary to disbelieve one member from each proof of the attitude. The set preference ordering ensures that the least endorsed assumptions are disbelieved in each proof.

The Increased Coherence Ordering $>_c$

The theory of increased coherence (i.e. *mc*) was presented in chapter 2. The model essentially prefers belief sets which offer the most proof for a priori chosen core beliefs. A set S is preferred over another set S' only if S contains all the proofs for the core belief that are in S' and more. Note that there is no reason to prefer one set if they both contain different proofs for the core beliefs.

A belief is *mc* with respect to a core belief if it is a necessary component of the proofs for the core belief. When $\alpha \land \beta \rightarrow \psi$ and $\alpha \rightarrow \psi$ the truth value of ψ is independent of β and β is unnecessary in the proof for ψ. In this case to remove the proof for ψ we must remove α. If the set does not entail α, adding β to the set will not create a new proof for ψ. Increased coherence is a function of the minimal proofs for core beliefs and this property lends itself to an efficient implementation of *mc* in the ATMS framework which maintains minimal proofs in its label environments. Essentially, a proposition is *mc* with respect to a core belief ψ in a belief set K if there is an environment in ψ's label which is both a subset of K and includes α as a member. In our system all propositions are ATMS assumptions and the proposition labels represent all proofs for all possible environment extensions.

To show this, we consider $mc(K, \psi, \phi)$ where, ψ is a core belief, K a belief set and ϕ is in the database, and label propagation is complete. Let E by the set of ψ's label environments which are subsets of K. We will show that $mc(K, \psi, \phi)$ if and only if ϕ is a member of some environment in E: that is

$$mc(K, \psi, \phi) \equiv \exists env \in l_{\psi}\, env \subseteq K \ \wedge \ \phi \in env$$

The predicate $mc(K, \psi, \phi)$ is true if and only if there is a subset of K in which ψ cannot be derived but, by adding ϕ to this, produces a proof for ψ given the justifications inferred so far. All proofs for ψ can be removed by subtracting from K at least one member from each environment in E. There may be many ways of doing this if the environments in E are large. Let K' be any set formed by subtracting a set of propositions W from K.

If ϕ is not in any environment in E then no matter what choice of W, there are no environments in E which are subsets of $K' \cup \{\phi\}$. Hence ψ is not derivable in $K' \cup \{\phi\}$ and $mc(K, \psi, \phi)$ is false in this case.

If ϕ is a member of one or more environments in E then choose Z to be any one of these. We can choose at least one proposition in each $E \setminus Z$ which is not in Z. This follows from the property that all ATMS label environments are minimal.[13] Let

$$W' \overset{def}{=} \bigcap E \setminus Z$$

and $W = \{\phi\} \cup W'$. It follows that $K \setminus W \not\vdash \psi$ and so there is a $K' = K \setminus W$ in which ψ is not derivable, though it is derivable in $\{\phi\} \cup K'$. Hence $mc(K, \psi, \phi)$ is true in this case.

The foregoing can be used to implement the mc preference ordering:

$$S \geq_c S' \leftrightarrow \bigcup E \cap S' \subseteq \bigcup E \cap S$$

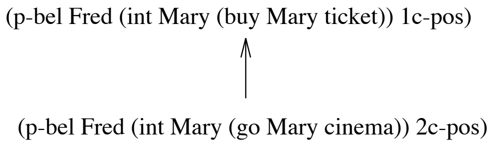
$S \geq_c S'$ if and only if every proposition which is mc with a core belief in S' is also mc in S.

We illustrate this with the following example. Fred is uncertain as to Mary's intentions. In one possible belief set he has the belief the Mary wants to go to the cinema. In another equally preferred set he has the belief that she wants to eat instead. Fred may have numerous reasons for preferring both sets (i.e. he may have been told by two separate but reliable sources) but this is immaterial. The fact that Fred is uncertain is all that matters. Suppose that Fred is outside the cinema and he sees Mary go in and buy a ticket. Although buying a ticket is not inconsistent with going out to eat (for example Mary could have bought a ticket for tomorrow evening's performance) it is more coherent with the intention of going to the cinema. Quantitatively, Fred has the following propositions

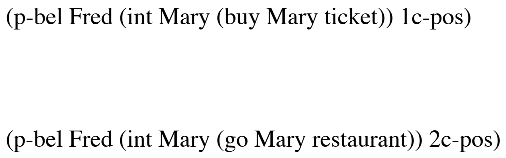
(*p-bel Fred* (*int Mary* (*go Mary restaurant*)) *2c-pos*)
(*p-bel Fred* (*int Mary* (*go Mary cinema*)) *2c-pos*)
(*p-bel Fred* (*int Mary* (*buy Mary ticket*)) *1c-pos*)

---

[13] Minimal in that within a label no environment is a subset of any other.


(p-bel Fred (int Mary (buy Mary ticket)) 1c-pos)
(p-bel Fred (int Mary (go Mary cinema)) 2c-pos)

(1) Mary intends to go to the cinema


(p-bel Fred (int Mary (buy Mary ticket)) 1c-pos)
(p-bel Fred (int Mary (go Mary restaurant)) 2c-pos)

(2) Mary intends to go to the restaurant

He believes that Mary cannot intend to go to both the cinema and the restaurant, and that if Mary intends to go to the cinema she will have to buy a ticket. Fred thus has two belief sets:

Both are equally preferred with respect to endorsement but the set (1) is more coherent. If (*p-bel Fred* (*int Mary* (*buy Mary ticket*))) is made a core belief then, although common to both sets, the core belief has more proofs in set (1). Hence (1) is preferred by *mc*.

Note that due to the difficulty of identifying core beliefs, the *mc* ordering was not used in the examples described in chapter 8.

The Minimal Change Ordering $>_m$

The minimal change ordering, $\geq_m$, is computed from the change in pervasive attitudes. Suppose the agent is currently in a cognitive state with beliefs Pf_D, then

$$S >_m S' \equiv \| S \cap Pf_D \| > \| S' \cap Pf_D \|$$

When revising its attitudes the agent prefers those attitude sets which contain the greatest number of pervasive attitudes from the previous cognitive state (see below). Beliefs are self justified (i.e. ATMS assumptions) and can be believed even when the original justification for the belief has been undermined. Our minimal change criterion ensures that belief revision of undermined pervasive beliefs occurs only when they are positively undermined. That is, when they conflict with other, preferred beliefs.

5.3.2 The Agent's Pervasive Beliefs

A p-belief is said to be *pervasive* if it is a member of all the preferred belief sets. A proposition is believed, written (*bel x p*), if (*p-bel x p*) is pervasive. An agent has some preferred (though not necessarily pervasive) belief-type attitude towards each proposition.[14] We will use the term *belief state* to refer to the set of preferred belief sets.

---

[14] An agent's preferred attitude towards a belief set need not be pervasive, for example if an agent is unable to decide between a proposition and its negation.

Below we list the possible contents of the preferred belief sets for a proposition p. With the exception of (3), which is a special case, the following belief states are organised in order of specificity, most specific first.

1. $(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, \neg p)$

   The agent believes that p and does not believe that $\neg p$. Since these p-beliefs are mutually consistent, both are pervasive $(\textit{bel}\, x\, p) \wedge \neg(\textit{bel}\, x\, \neg p)$. This is the state which would normally be described as 'belief in p'. Note that the agent cannot both pervasively believe that p and believe that not p.

2. $(\textit{p-bel}\, x\, \neg p) \wedge \neg(\textit{p-bel}\, x\, p)$

   The agent does not believe that p and believes that $\neg p$. Since these p-beliefs are mutually consistent, both are pervasive $(\textit{bel}\, x\, \neg p) \wedge \neg(\textit{bel}\, x\, p)$. This is the state which would normally be described as 'disbelieving or not believing that p'.

3. $\neg(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, \neg p)$

   The agent does not believe that p and does not believe that $\neg p$. Again these two p-beliefs are mutually consistent and therefore are pervasive, i.e. $\neg(\textit{bel}\, x\, p) \wedge \neg(\textit{bel}\, x\, \neg p)$. This state could be described as 'uncertain about p', where the uncertainty is understood to be due to lack of knowledge or reasons for believing in either p or $\neg p$. This could be termed 'pervasive uncertainty' as, unlike some of the other form of uncertainty described below, the resulting belief set(s) are consistent. This state would normally never arise. It can arise when p is an absurdity, e.g. "the present King of France is bald".

4. $(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, \neg\ p) \wedge \neg(\textit{p-bel}\, x\, p)$

   The agent is certain it does not believe $\neg p$ (i.e. $\neg(\textit{bel}\, x\, \neg p)$), but is uncertain whether it believes p. (For example, the agent is certain it does not believe it is not raining, but is uncertain whether it believes it is raining.) In some possible worlds it believes that p and in others it is uncertain about p. In no world does it believe that $\neg p$.

5. $(\textit{p-bel}\, x\, \neg p) \wedge \neg(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, \neg p)$

   The agent is certain it does not believe p (i.e. $\neg(\textit{bel}\, x\, p)$), but is uncertain whether it believes $\neg p$. In some possible worlds the agent does not believe that p and in others it is uncertain about p. In no world does it believe that p.

6. $(\textit{p-bel}\, x\, p) \wedge \neg(\textit{p-bel}\, x\, \neg\ p) \wedge (\textit{p-bel}\, x\, \neg p) \wedge \neg(\textit{p-bel}\, x\, p)$

   The agent has no pervasive beliefs about p. The agent is uncertain about p, but in this case the uncertainty is based on 'too much' knowledge rather than its absence; the agent has good reason both for believing that p and disbelieving that p. In one possible world, the agent believes that p an in another possible world it believes that $\neg p$, i.e. it believes either p or $\neg p$.

7. $(p\text{-}bel\,x\,p) \wedge \neg(p\text{-}bel\,x\,\neg p) \wedge \neg(p\text{-}bel\,x\,p) \wedge (p\text{-}bel\,x\,\neg p)$

   The agent has no pervasive beliefs. All the possible attitudes towards p are accessible from the current world. In some worlds the agent believes that p, in others it disbelieves p and in still others it is uncertain about p. There are therefore no conclusions we can draw about the agent's beliefs. The agent is seriously confused.

In cases (1)–(3) the agent's beliefs about p are pervasive. Note that we need an explicit representation of pervasive belief. We need to be able to determine which beliefs an agent doesn't have as well as those it does, otherwise all (belief-type) inference rules will fire in cases (4) and (7). While cases (5) and (6) each have a pervasive belief, this belief also appears in one or more of cases (1)–(3). To select just case (5), for example, we would need a rule with a LHS of the form:

$$\neg(bel\,x\,\neg\ p) \wedge \neg(bel\,x\,(bel\,x\,p)) \wedge \neg(bel\,x\,\neg(bel\,x\,p))$$

where '*bel*' is interpreted as above as a pervasive belief. This requires iterated belief.[15]

5.3.3 Commitment to Pervasive Beliefs

An agent's *commitment* to an attitude reflects the corrigibility of that attitude and the agent can be strongly committed or weakly committed. An agent has reasons for believing (through *mc*, justifications and endorsements) and for disbelieving (through reason for believing a contradictory attitude). Commitment is a relative measure of the reasons for believing against those for disbelieving. An agent's commitment to a belief-type attitude is '*strong*' if it is hard to disbelieve and '*weak*' if it is relatively easy to disbelieve.

As stated above, an agent x *believes* a belief if the belief appears in all the preferred belief sets and disbelieves it otherwise

$$(bel\,x\,p\ \ strong) \vee (bel\,x\,p\ \ weak) \equiv \forall S \in Pf_D^{\{e,c\}}\ p \in S$$

An agent x is *uncertain* when the belief appears in *some* but not all the preferred belief sets.

$$(bel\,x\,p\ \ uncertain) \equiv \exists S' \in Pf_D^{\{e,c\}}\ \neg p \in S' \wedge \exists S \in Pf_D^{\{e,c\}}\ p \in S$$

To implement this notion of commitment we introduce a new endorsement, *bel-comm* to calculate the commitment to beliefs. This endorsement belongs to the following ordering

1c-pos > *2c-pos*, *spec* > *bel-comm*> *1c-neg* > *2c-neg*, *def* > *hypoth*

---

[15] This raises an interesting point: should the behaviour of the agent be different for different kinds of uncertainty? Should, for example, the agent try and resolve pervasive and non-pervasive uncertainty in different ways.

To determine the commitment to a pervasive belief p, the agent temporarily attaches an extra *bel-comm* endorsement to $(\neg p)$. If p remains pervasive then the agent is strongly committed to p, otherwise it is only weakly committed. Formally,

$$(\mathit{bel}\ \mathit{agent}\ p\ \ \mathit{strong}) \equiv$$
$$p \in \bigcap Pf^{\{e,c\}}_{\{\ldots,\{\langle\neg p, l_i, j_i, e_i\rangle\},\ldots\}} \cap \bigcap Pf^{\{e,c\}}_{\{\ldots,\langle\neg p, l_i, j_i, e_i \oplus_p \langle \mathit{bel\text{–}comm}\rangle\rangle,\ldots\}}$$

$$(\mathit{bel}\ \mathit{agent}\ p\ \ \mathit{weak}) \equiv$$
$$p \in \bigcap Pf^{\{e,c\}}_{\{\ldots,\{\langle\neg p, l_i, j_i, e_i\rangle\},\ldots\}} \setminus \bigcap Pf^{\{e,c\}}_{\{\ldots,\langle\neg p, l_i, j_i, e_i \oplus_p \langle \mathit{bel\text{–}comm}\rangle\rangle,\ldots\}}$$

where we define the concatenation operator $\oplus_p$ as:

$$< k_1, \ldots, k_n > \oplus_p < k_{n+1}, \ldots, k_m > \overset{def}{=} < k_1, \ldots, k_n, k_{n+1}, \ldots, k_m >$$

To explain why we have chosen the above endorsement orderings we must consider the role of commitment in dialogue, where agents communicate their commitments to attitudes. If we consider a simple, almost trivial scenario we can see why the *bel-comm* endorsement appears in the endorsement orderings above. An agent (the hearer) with no initial attitudes (i.e. a completely empty cognitive state) is communicated a belief p with commitment c by a fellow agent (the speaker). Naturally, since the hearer does not already have reasons for believing p or $(\neg\ p)$, he should share his commitment to p with the speaker.

When the speaker is strongly committed to p then, in this scenario, the hearer should also become strongly committed to p. Now, in section 5.2.1 we noted that a *strong* commitment to p by the speaker is converted to a *2c-pos* endorsement for the hearer believing p. The hearer's commitment is calculated by endorsing $\neg p$ *bel-comm* and comparing, in this case, the hearer's two possible belief sets

1. { (*p-bel* $x\ p$ *2c-pos*) }
2. { (*p-bel* $x \neg p$ *bel-comm*) }

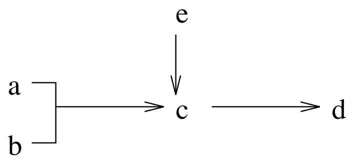
The possible belief (*p-bel* $x\ p$) remains pervasive and is therefore a strong belief only if *2c-pos* $>_e$ *bel-comm*.

Similarly, when the speaker is weakly committed to p then the hearer should also become weakly committed to p. A *weak* commitment to p by the speaker is converted to a *2c-neg* endorsement for the hearer believing p. Again, the hearer's commitment is calculated by endorsing $\neg p$ *bel-comm* and comparing the hearer's two possible belief sets

1. { (*p-bel* $x\ p$ *2c-neg*) }
2. { (*p-bel* $x \neg p$ *bel-comm*) }

The p-belief (*p-bel* $x\ p$) is no longer pervasive and is therefore a weak belief only if *bel-comm* $\geq_e$ *2c-neg*.

Combining these results we find that



a [*hypoth*, *1c-pos*]	$\neg a$ [*hypoth*]
b [*hypoth*, *def*]	$\neg b$ [*hypoth*]
c [*hypoth*]	$\neg c$ [*hypoth*, *spec*]
d [*hypoth*]	$\neg d$ [*hypoth*]
e [*hypoth*, *2c-pos*]	$\neg e$ [*hypoth*]

$$\textit{2c-pos} >_e \textit{bel-comm} \geq_e \textit{2c-neg}$$

The ordering relations for attitude sets given in 5.3.1 are augmented to include the two commitment heuristics. In that section we considered comparing the endorsements e and e' of two sets S and S' respectively. In the belief set ordering relations the following is inserted between steps 3 and 4

3a. if e contains more *bel-comm* endorsements than e' then $S \geq_e S'$, else if e' contains more *bel-comm* endorsements than e then $e' \geq_e e$

As a demonstration of commitment, consider the following justifications where a, b, c, d and e represent possible beliefs with the following endorsement assignments

The ordering heuristics produce the following set preference hierarchy

$$\begin{array}{c}
\{a, b, c, d, e\} \\
\vdots \\
\{a, \neg b, c, d, e\} \\
\vdots \\
\{a, \neg b, \neg c, d, \neg e\} \quad \{a, \neg b, \neg c, \neg d, \neg e\} \\
\vdots \\
\{a, b, c, d, \neg e\} \\
\vdots
\end{array}$$

There is only one preferred set $\{a, b, c, d, e\}$ and a, b, c, d and e are all believed to be the case. However, the commitment to c, for example, is only *weak*. This can be seen by giving $\neg c$ an extra *bel-comm* endorsement in which case the new set ordering becomes

$$\{a, \neg b, \neg c, d, \neg e\} \quad \{a, \neg b, \neg c, \neg d, \neg e\}$$
$$\vdots$$
$$\{a, b, c, d, e\}$$
$$\vdots$$
$$\{a, \neg b, c, d, e\}$$
$$\vdots$$
$$\{a, b, c, d, \neg e\}$$
$$\vdots$$

and c is no longer pervasive. If, however, b had been endorsed *2c-pos* instead of *def* then the commitment to c would have been *strong* since $\{a, b, c, d, e\}$ would still have been preferred after $\neg c$ was extra endorsed *bel-comm*.

The result of the revision process is a consistent set of beliefs and disbeliefs and a commitment to each belief (one of strong, weak or uncertain) which form the agent's belief state for this cycle. Note that, unlike p-beliefs, beliefs are not closed under negation. If (*p-bel* $x\, p$) is preferred, the agent's belief state will contain only (*bel* $x\, p$) and not $\neg$(*bel* $x\, p$) (although it will contain $\neg$(*bel* $x \neg p$)).

5.4 Deriving the Intention State

The procedure for calculating the agent's intentions follows essentially the same pattern as beliefs. The agent's intention sets are based on the agent's preferred belief sets. Having determined the preferred belief sets from the consistent sets of belief-type attitudes (i.e. the *p-bel*s) and its commitment to these beliefs, the agent then

1. assigns a *definite* endorsement to any pervasive beliefs (*bel* nodes) in the most preferred belief sets[16] endorsement is attached to these *bel* nodes only;
2. computes the consistent sets of beliefs (*bel*s) and p-intentions;
3. determines the preferred intention sets;
4. calculates its commitment to the *p-int*s and assigns a *definite* endorsement to the *int* nodes found in the previous step; and
5. find the extension of the intention sets found in step 3 and choose the most preferred of these.[17]

The belief revision relationship between these various attitude types is shown in figure 5.2. The arrows indicate the direction of influence of belief

---

[16] A *definite* endorsement can be assigned temporarily to attitudes. While assigned a *definite* endorsement an attitude cannot be disbelieved. Equivalently, candidates containing definite attitudes cannot be preferred.

[17] This stage is necessary to determine the agent's attitude to its predicted beliefs and intentions, if any (see chapter 7).

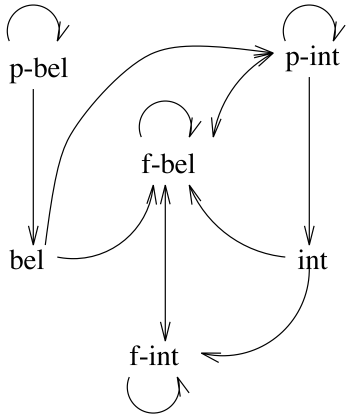


Figure 5.2: Attitude influence relationships

revision. For example revising a possible belief (i.e. *p-bel*) can induce the revision of other possible beliefs, and revising a possible belief can induce the revision of a pervasive belief. Note, however, that the agent's commitment to a possible intention (i.e. *p-int*) does not influence its commitment to its beliefs.

Unlike beliefs, intentions are not closed under negation. The only requirement therefore is that the agent should not simultaneously intend both p and $\neg p$, i.e.

$$(p\text{-}int\ x\ p) \wedge (p\text{-}int\ x\ \neg p) \supset \perp$$

In addition we have the following rule linking the agent's beliefs and p-intentions:

$$(bel\ x\ p) \wedge (p\text{-}int\ x\ p) \supset \perp$$

i.e. the agent cannot intend a state it believes to be true. A consequence of this is that a possible intention, for example $(p\text{-}int\ x\ p)$, which conflicts with a pervasive belief, for example $(bel\ x\ p)$, will not form part of any intention set. However if the agent is uncertain about p or disbelieves that p the possible intention will be included in the intention sets.

5.4.1 Computing the Preferred Sets

The computation of the agent's most preferred intention sets follows basically the same pattern as beliefs. However for intention sets the endorsement ordering relations are:

1. assign values to *desire-pos*, *effort-pos*

$$\begin{array}{rcl} desire\text{-}pos & = & 4 \\ effort\text{-}pos & = & \text{-1} \end{array}$$

and sum these in each to find the value of the sets.[18]. If the value of e is greater than that of e' and $e \geq 0$ then $S \geq_e S'$ else if the value of e' is greater than that of e and $e' > 0$ then $S' \geq_e S$ else

2. if e contains more *desire-neg* endorsements than e' $S \geq_e S'$ else if e' contains more *desire-neg* endorsements than e then $S' \geq_e S$ else

3. if e contains more *effort-neg* endorsements than e' $S \geq_e S'$ else if e' contains more *effort-neg* endorsements than e $S' \geq_e S$ else $e \geq_e e'$

The definitions of increased coherence and minimal change orderings are identical to those for beliefs

5.4.2 The Agent's Pervasive Intentions

A p-intention is said to be pervasive if it is a member of all the preferred intention sets. A proposition is intended, written $(\mathit{int}\, x\, p)$, if $(\mathit{p\text{-}int}\, x\, p)$ is pervasive. Since intention sets are not closed under negation, an agent need have no preferred intention-type attitude towards a proposition of which it is aware, for example when an agent has a possible intention that p and a pervasive belief that p, since the presence of $(\mathit{p\text{-}int}\, x\, p)$ by itself does not give rise to $(\mathit{p\text{-}int}\, x\, \neg p)$. We use the term *intention state* to refer to the set of preferred intention sets.

After intention revision, an agent's intention-type attitude towards a state of affairs or proposition p is one of the following. With the exception of (3), which is a special case, the following intention states are organised in order of specificity, most specific first.

1. $(\mathit{p\text{-}int}\, x\, p) \wedge \neg(\mathit{p\text{-}int}\, x\, \neg p)$

   The agent intends that p and does not intend that $\neg p$. Since these p-intentions are mutually consistent, both are pervasive $(\mathit{int}\, x\, p) \wedge \neg(\mathit{int}\, x\, \neg p)$. This is the state which would normally be described as 'intending that p'. Note that the agent cannot both pervasively intend that p and intend that $\neg p$.

2. $(\mathit{p\text{-}int}\, x\, \neg p) \wedge \neg(\mathit{p\text{-}int}\, x\, p)$

   The agent does not intend that p and intends that $\neg p$. Since these p-intentions are mutually consistent, both are pervasive $(\mathit{int}\, x\, \neg p) \wedge \neg(\mathit{int}\, x\, p)$. This is the state which would normally be described as 'not intending that p'.

3. $\neg(\mathit{p\text{-}int}\, x\, \neg p) \wedge \neg(\mathit{p\text{-}int}\, x\, p)$

   The agent does not intend that p and does not intend that $\neg p$. Again these two p-intentions are mutually consistent and therefore are pervasive $\neg(\mathit{int}\, x\, \neg p) \wedge \neg(\mathit{int}\, x\, p)$. This state could be described as 'indifference to p', where the indifference is understood to be due to lack of reasons

---

[18] A *desire-pos* utility can support plans of up to 4 *effort-pos* effort actions. The value 4 is an arbitrary number but it is necessary to have a limit on the size of plans to reduce the possibility of deadlock.

(interpreted as goals or desires rather than knowledge) for intending either that p or that $\neg p$, i.e. the agent has no reason for either wanting that p or wanting that $\neg p$. Note that, unlike belief, this is a non dominated state, since intentions have negative endorsement in the form of effort. Case (3) is therefore preferred to both cases (1) and (2) which require effort on the part of the agent to achieve the intentions. [19]

4. $(\textit{p-int}\ x\ p) \wedge \neg(\textit{p-int}\ x\ \neg p) \wedge \neg(\textit{p-int}\ x\ p)$

   The agent is certain it does not intend that $\neg p$ (ie $\neg(\textit{int}\ x\ \neg p)$), but is uncertain whether it intends that p. (For example, the agent is certain it does not intend the door to be open, but is uncertain whether it intends to close the door; the agent may be indifferent whether the door is open or not.) In some possible worlds it intends that p and in others it is uncertain in its intention wrt p. In no possible world does it intend that $\neg p$.

5. $(\textit{p-int}\ x\ \neg p) \wedge \neg(\textit{p-int}\ x\ p) \wedge \neg(\textit{p-int}\ x\ \neg p)$

   The agent is certain it does not intend that p (i.e. $\neg(\textit{int}\ x\ p)$), but is uncertain whether it intends that $\neg p$. In some possible worlds the agent does not intend that p and in others it is uncertain in its intention wrt p. In no possible world does it intend that p.

6. $(\textit{p-int}\ x\ p) \wedge (\textit{p-int}\ x\ \neg p) \wedge \neg(\textit{p-int}\ x\ p) \wedge \neg(\textit{p-int}\ x\ p)$

   The agent has no pervasive intentions wrt p. In other words the agent is uncertain about whether it should try and achieve p, but in this case the uncertainty is based on too many goals rather than too few; the agent has good reason both for intending that p and not intending that p. In one possible world, the agent intends that p, and in another it does not intend that p, i.e. it intends either p or $\neg p$.

7. $(\textit{p-int}\ x\ p) \wedge \neg(\textit{p-int}\ x\ \neg p) \wedge (\textit{p-int}\ x\ \neg p) \wedge \neg(\textit{p-int}\ x\ p)$

   The agent has no pervasive intentions. All the possible attitudes towards p are accessible from the current world. In some possible worlds the agent has good reason to p-intend that p, in others it has reason not to p-intend that p, not to p-intend that p and not to p-intend that $\neg p$ There are therefore no conclusions we can draw about the agent's intentions. The agent is seriously confused.

5.4.3 Commitment to Pervasive Intentions

The same definitions apply to intentions as beliefs: an agent x *intends* an intention if the intention appears in all the preferred intention sets and disintends it

---

[19] If a proposition is an intention to act (i.e. is endorsed *effort-pos* or *effort-neg*) then the negation of the proposition must be in the database otherwise the agent has no choice but to intend the action. The fact that an action involves effort introduces a further reason to prefer not to perform an action even if the intention to act is consistent. To achieve this within the ATMS we must ensure that the intention is in a nogood.

otherwise. An agent's commitment to an intention-type attitude is *strong* if it is hard to disintend and *weak* if it is relatively easy to disintend.

$$(\mathit{int}\ x\, p\ \ \mathit{strong}) \vee (\mathit{int}\ x\, p\ \mathit{weak} \equiv) \forall S \in Pf_D^{\{e,c\}}\ p \in S$$

An agent x is uncertain when the intention is in some but not all of the preferred intention sets.

$$(\mathit{int}\ x\, p\ \ \mathit{uncertain}) \equiv \exists S' \in Pf_D^{\{e,c\}}\ \neg p \in S' \wedge \exists S \in Pf_D^{\{e,c\}}\ p \in S$$

As with beliefs, we introduce a new endorsement, *int-comm*, to calculate the commitment to intentions which belongs to the following ordering

$$\mathit{desire\text{-}pos} > \mathit{int\text{-}comm} > \mathit{desire\text{-}neg} > \mathit{hypoth} > \mathit{effort\text{-}neg} > \mathit{effort\text{-}pos}$$

Formally,

$$\begin{aligned}
&(\mathit{int}\ \mathit{agent}\, p\ \ \mathit{strong}) \equiv \\
&\quad p \in \bigcap Pf^{\{e,c\}}_{\{\ldots,\{\langle \neg p, l_i, j_i, e_i\rangle\},\ldots\}} \cap \bigcap Pf^{\{e,c\}}_{\{\ldots,\langle \neg p, l_i, j_i, e_i \oplus_p \langle \mathit{int\text{-}comm}\rangle\rangle,\ldots\}} \\
&(\mathit{int}\ \mathit{agent}\, p\ \ \mathit{weak}) \equiv \\
&\quad p \in \bigcap Pf^{\{e,c\}}_{\{\ldots,\{\langle \neg p, l_i, j_i, e_i\rangle\},\ldots\}} \setminus \bigcap Pf^{\{e,c\}}_{\{\ldots,\langle \neg p, l_i, j_i, e_i \oplus_p \langle \mathit{int\text{-}comm}\rangle\rangle,\ldots\}}
\end{aligned}$$

where the concatenation operator $\oplus_p$ is defined as for beliefs and a similar argument for intentions yields

$$\mathit{desire\text{-}pos} >_e \mathit{int\text{-}comm} \geq_e \mathit{desire\text{-}neg}$$

In the intention set ordering relations the following replaces step 1

1. assign values to *desire-pos*, *effort-pos*

$$\begin{aligned}
\mathit{desire\text{-}pos} &= 4 \\
\mathit{effort\text{-}pos} &= \text{-}1 \\
\mathit{int\text{-}comm} &= 2
\end{aligned}$$

   and sum these in each to find the value of the sets; if the value of e is greater than that of e' and $e \geq 0$ then $S \geq_e S'$ else if the value of e' is greater than that of e and $e' > 0$ then $S' \geq_e S$.[20]

The result of the revision process is a consistent set of beliefs and intentions and a commitment to each intention (one of strong, weak or uncertain) which form the agent's intention state for this cycle.

---

[20] The choice of 2 for the value of *int-comm* is arbitrary. The effect is that an intention is strongly committed to unless it requires two or more actions to satisfy; then the commitment is reduced to weak.

5.5 Inference

The Inference Engine uses the output of the belief and intention revision phase to decide what to infer. The Inference Engine uses *inference rules* to generate new justifications for beliefs and intentions from existing attitudes. There are two kinds of inference rules—logical rules of the form

$$a_1 \mathbin{\&} a_2 \mathbin{\&} \ldots \mathbin{\&} a_n \mapsto_k c$$

and defeasible rules of the form:

$$a_1 \mathbin{\&} a_2 \mathbin{\&} \ldots \mathbin{\&} a_n \Rightarrow_k c$$

In both cases $a_1 \mathbin{\&} a_2 \mathbin{\&} \ldots \mathbin{\&} a_n$ is the antecedent and c is the consequent. The only difference between the two types of rule is that defeasible rules give rise to rule instances (see section 5.5.2) which form part of the justification of the consequent, whereas logical rules do not. In what follows, our comments about defeasible rules should be taken to apply to both types.

Each antecedent or consequent term ϕ_i may be either a (possibly non-ground) attitude, a conjunction of attitudes (written $(and\ (\phi_1\ \phi_2 \ldots \phi_j)))$, a disjunction of attitudes (written $(or\ (\phi_1\ \phi_2 \ldots \phi_j)))$, or a function which evaluates to an attitude, a conjunction or a disjunction of attitudes. Free variables are denoted by symbols prefixed with a question mark, e.g. ?x, which must be instantiated before the rule can fire. In addition, the antecedent may contain predicates evaluating to *True* or *False*. k is called the *confidence* of the rule and is a measure of the support the rule gives to the consequent. This can be, for example, '*def*' which indicates relatively weak support or '*spec*' (i.e. specific) which is stronger. Rules are not beliefs and do not form part of the agent's database. However by using explicit representations of the bindings of terms to free variables and confidence factors we can implement defeasible inference (see section 5.5.2).

The following functions and predicates can appear in rules.

Function: Forall

The function '*forall*' attempts to unify pattern *pat1* with each member of *set*. If each unification is successful then these bindings are used, in turn, to instantiate pattern *pat2* and the conjunction of these instantiations is returned. False is returned otherwise.

$$(\textit{forall pat1 set pat2}) \rightarrow (\text{and}\ (inst_1 \ldots inst_n))$$

Examples

$$\begin{array}{lcl}
(\textit{forall}\ ?x\ (a\ b\ c)\ (t\ ?x)) & \rightarrow & (\textit{and}\ ((t\ a)\ (t\ b)\ (t\ c))) \\
(\textit{forall}\ (t\ ?x)\ ((t\ a)\ (t\ b))\ ?x) & \rightarrow & (\textit{and}\ ((a\ b))) \\
(\textit{forall}\ (t\ ?x)\ ((t\ a)\ (s\ b))\ ?x) & \rightarrow & False
\end{array}$$

Function: Intorbel

The function '*intorbel*' takes an agent identifier and a literal and returns the disjunction of the literal and the intention for an agent to achieve this literal.[21]

$$(\text{intorbel } ?agent\ ?literal) \rightarrow (\text{or } ((\text{bel } ?agent\ ?literal)\ (\text{p-int } ?agent\ ?literal)))$$

Examples

$$\begin{array}{lcl}
(\text{intorbel } agent1\ (\text{bel } agent1\ p)) & \rightarrow & (\text{or } ((\text{bel } agent1\ p)\ (\text{p-int } agent1\ p))) \\
(\text{intorbel } agent1\ (\text{bel } agent2\ p)) & \rightarrow & (\text{or } ((\text{bel } agent1\ (\text{bel } agent2\ p)) \\
 & & \quad (\text{p-int } agent1\ (\text{bel } agent2\ p)))) \\
(\text{intorbel } agent1\ (\text{bel } agent1\ p)) & \rightarrow & (\text{or } ((\text{bel } agent1\ p) \\
 & & \quad (\text{p-int } agent1\ (\text{bel } agent1\ p)))) \\
(\text{intorbel } agent1\ (int\ agent1\ p)) & \rightarrow & (\text{or } ((\text{bel } agent1\ p) \\
 & & \quad (\text{p-int } agent1\ p)))
\end{array}$$

Function: Some

The function '*some*' attempts to unify pattern *pat1* with each member of *set*. The bindings for those that are successful are then used, in turn, to instantiate pattern *pat2*, and these instantiations are returned as a disjunction. False is returned otherwise.

$$(\text{some } pat1\ set\ pat2) \rightarrow (\text{or } (inst_1 \ldots inst_n))$$

Examples

$$\begin{array}{lcl}
(\text{some } ?x\ (a\ b\ c)\ (t\ ?x)) & \rightarrow & (\text{or } ((t\ a)\ (t\ b)\ (t\ c))) \\
(\text{some } (t\ ?x)\ ((t\ a)\ (s\ b))\ ?x) & \rightarrow & (\text{or } ((a\ b))) \\
(\text{some } (t\ ?x)\ ((w\ a)\ (s\ b))\ ?x) & \rightarrow & False
\end{array}$$

Predicate: Binds

The predicate '*binds*' takes a variable *var*, a set of allowable bindings *allowables* for this variable, a pattern *pattern* which contains the variable, and an expression *expression*

$$(\text{binds } var\ allowables\ pattern\ expression) \rightarrow \{True, False\}$$

Binds evaluates to True if *pattern* unifies with *expression*, binding *var* to a member of *allowables*. Note that *var* and its occurrence in *pattern* need not be

---

[21] The rule

$$a_1\ \&\ \ldots\ \&\ a_n \rightarrow (\text{intorbel agent } p)$$

is equivalent to the more efficient ATMS justification

$$a_1\ \&\ \ldots\ \&\ a_n\ \&\ (\text{not } (\text{bel } agent\ p)) \rightarrow (\text{p-int } agent\ p)$$

which is what is actually used in the database. Expressions are also simplified using the introspection relations discussed in section 5.2.

prefixed with a query ?.

Examples

$$\begin{array}{lll}
(binds\ ?x\,(a\ b\ c)\ (s\ t\ ?x)\ (s\ t\ a)) & \rightarrow & True, ?x/a \\
(binds\ !x\,(a\ b\ c)\ (s\ t\ !x)\ (s\ t\ a)) & \rightarrow & True \\
(binds\ x\,(a\ b\ c)\ (s\ t\ x)\ (s\ t\ a)) & \rightarrow & True \\
(binds\ !x\,(a\ b\ c)\ (s\ t\ !x)\ (s\ u\ a)) & \rightarrow & False \\
(binds\ !x\,(a\ b\ c)\ (s\ t\ !x)\ (s\ t\ d)) & \rightarrow & False
\end{array}$$

5.5.1 The Rule Binding Algorithm

The system binds the variables in a rule

$$a_1\ \&\ a_2\ \&\ \ldots\ \&\ a_n \Rightarrow_k c$$

by instantiating the rule antecedent terms, a_i, in the order in which they appear in the rule specification. For each antecedent term

1. if the antecedent term matches an attitude bind variables in the antecedent (if any);
2. if the antecedent term instantiates to a function, evaluate the function and append the result onto the antecedent of the rule; and
3. if the antecedent term instantiates to a predicate, evaluate the predicate and, if the result is True then continue, otherwise abandon this particular rule firing.

To illustrate, consider the following scenario in the library domain. The librarian has asked the user for confirmation of his status: whether the user is student or a member of staff: "Are you a student or a member of staff?" The user believes that the librarian wants to know.

$$(bel\ user\ (int\ lib\ (exists\ !a\ (student\ staff)\ (bel\ lib\ (bel\ user\ (status\ user\ !a))))))$$

This attitude means that the user believes that the librarian intends to have a belief that the user either believes he is a student or believes he is a member of staff. *!a* is an existentially quantified variable and the statement

$$(exists\ !a\ (student\ staff)\ (bel\ lib\ ((bel\ user\ (status\ user\ !a))))$$

means that when *!a* is bound to either 'student' or 'staff' then

$$(bel\ lib\ (bel\ user\ (status\ user\ !a)))$$

is *True*. The user adopts this intention as his own

$$(p\text{-}int\ user\ (exists\ !a\ (student\ staff)\ (bel\ lib\ (bel\ user\ (status\ user\ !a)))))$$

Now, consider the following rule from the point of view of the user

(*p-int ?agent1* (*exists ?x ?y* (*bel ?agent2* (*status ?agent1 ?x*)))) &
(*bel ?agent1* (*status ?agent1 ?p*)) &
(*binds ?x ?y ?x ?p*)
$\Rightarrow_{[\text{desire-pos}]}$ (*intorbel ?agent* (*bel ?agent2* (*status ?agent ?x*)))

The variables *?agent1* and *?agent2* should instantiate to agent names (e.g. *user* or *lib*), *?x* instantiates to an existentially quantified variable (e.g. *!a*) and *?y* instantiates to a set of possible statuses (e.g. *student* or *staff*).[22] The rule states that if an agent knows his status and this status is represented in the list of options *?y*, and the agent intends a fellow agent to know about his status, then he will intend to communicate his status.

We apply the above rule to the following attitudes:

(*p-int user* (*exists !a* (*student staff*) (*bel lib* (*status user !a*))))
(*bel user* (*status user student*))

The antecedent members are bound in order.

(*p-int ?agent1* (*exists ?x ?y* (*bel ?agent2* (*status ?agent ?x*))))

binds to

(*p-int user* (*exists !a* (*student staff*) (*bel lib* (*status user !a*))))

producing the following bindings

$?agent1$ / *user*
$?agent2$ / *lib*
$?x$ / !*a*
$?y$ / (*student staff*)

The second antecedent term

(*bel ?agent1* (*status ?agent1 ?p*))

binds to

(*bel user* (*status user student*))

augmenting the bindings to the following

$?agent1$ / *user*
$?agent2$ / *lib*
$?x$ / !*a*
$?y$ / (*student staff*)
$?p$ / (*status user student*)

---

[22] Note that *?x* in the rule definition is not an existential term; it binds to an existential term.

Finally, the third antecedent term (*binds ?x ?y ?x ?p*) is instantiated to produce

(*binds !a* (*student staff*) *!a* (*status user student*))

which is a predicate which evaluates to *True* (see section 5.5). The consequent instance is constructed from the above bindings

(*intorbel lib* (*bel lib* (*status user student*)))

In addition, we introduce a a notion of *relevance* to decide between competing rule bindings. We define an attitude b as *relevant* to an attitude c if the commitment to c is dependent on the commitment to b, that is, if the commitment to c changes when the commitment to b does. Specifically, an attitude b is relevant to an attitude c iff

1. if b is believed/intended, making $\neg b$ believed/intended results in a change in the commitment to c;
2. if b is uncertain, making it either believed/intended or $\neg b$ believed/intended results in a change in the commitment to c.

More precisely, a attitude b is relevant to c, written (*Rel* $b\,c$), if and only if

$$c \in \bigcap S_P \uplus \left(\bigcap S_{P_b} \cup \bigcap S_{P_{\neg b}}\right)$$

where $\uplus$ denotes set exclusive union, ie $A \uplus B \equiv (A \cup B) - (A \cap B)$, S_P is the set of preferred beliefs, and S_{Pb} denotes the set of most preferred belief sets containing b, ie

$$Pf^{\{e,c\}}_{\{\ldots,<b,l_i,j_i,e_i \oplus_p <definite>>,\ldots\}}$$

Note, relevance is not transitive in all cases.

$$(\mathit{Rel}\ a\ b) \wedge (\mathit{Rel}\ b\ c) \not\rightarrow (\mathit{Rel}\ a\ c)$$

Consider, for example, the case where a, b and c are all believed and suppose that changing the commitment of a so that $\neg a$ is believed makes b uncertain. The change of commitment of b from believed to uncertain may be insufficient to change the commitment to c.

In the example in section 5.3.3 there is only one preferred set $\{a, b, c, d, e\}$ and a, b, c, d and e are all believed. Now, we will consider those beliefs relevant to c. Since c is believed a belief x is relevant to c if changing the status of x makes c disbelieved. So, for example, if the status of b were changed so that $\neg b$ was believed (i.e. by temporarily making $\neg b$ a *definite* belief), the most preferred set would be $\{a, \neg b, c, d, e\}$, in which case c would still be believed. Hence, b is not relevant to c. However, the most preferred sets which contain $\neg e$ also contain $\neg c$ (i.e. sets $\{a, \neg b, \neg c, d, \neg e\}$ and $\{a, \neg b, \neg c, \neg d, \neg e\}$). Hence, e is relevant to c. If, however, a, b, c, d and e were endorsed

a [*hypoth*, *1c-pos*]	$\neg a$ [*hypoth*]
b [*hypoth*, *def*]	$\neg b$ [*hypoth*]
c [*hypoth*]	$\neg c$ [*hypoth*, *spec*]
d [*hypoth*, *2c-pos*]	$\neg d$ [*hypoth*]
e [*hypoth*, *spec*]	$\neg e$ [*hypoth*, *def*]

c is uncertain and the belief set hierarchy looks like this

$$\begin{array}{c} \{a,b,c,d,e\} \quad \{a,\neg b,\neg c,d,\neg e\} \\ \vdots \\ \{a,\neg b,c,d,e\} \\ \vdots \\ \{a,\neg b,\neg c,\neg d,\neg e\} \\ \vdots \\ \{a,b,c,d,\neg e\} \\ \vdots \end{array}$$

There are now two preferred sets $\{a,b,c,d,e\}$ and $\{a,\neg b,\neg c,d,\neg e\}$, and b,c and e are uncertain. We can see that adding the definite endorsement to b or e will result in just one preferred set $\{a,b,c,d,e\}$ in which c is believed, and b and e are therefore relevant to c. Again, changing the belief status of d to induce the belief in $\neg d$ by endorsing $\neg d$ definite would cause $\{a,\neg b,\neg c,\neg d,\neg e\}$ to be preferred and $\neg c$ to be believed. Hence, d is also relevant to c.

5.5.2 Defeasible Inferences

When an inference is made the antecedents of a rule are bound to attitudes and bindings are found for variables in the antecedent. These bindings are then used to instantiate the consequent. For example, the antecedent in the following rule can fire on (*p-bel Fred* (*cat Tiddles*)), binding *?agent* to *Fred* and *?animal* to *Tiddles.*

(*p-bel ?agent* (*cat ?animal*))
$\Rightarrow_{[def]}$ (*p-bel ?agent* (*likes ?animal fish*))

The entire expression can be instantiated to yield

(*p-bel Fred* (*cat Tiddles*))
$\Rightarrow_{[def]}$ (*p-bel Fred* (*likes Tiddles fish*))

This is modus ponens except for the fact that the rule is not certain, only default. This is interpreted to mean that it is not necessarily the case that when Fred believes that Tiddles is a cat he must also believe that Tiddles likes fish. He will do so only if there is no stronger reason to disbelieve. Fred can reason about the applicability of the rule under various bindings.

For each inference, the Inference Engine creates a *rule instance* which is a statement about the applicability of a rule under a set of bindings. It has the form

(*rule-inst rule ant con*)

where *rule* is a quoted facsimile of the rule, *ant* is the *antecedent instance* and *con* is the *consequent instance.* The antecedent instance is the instantiated rule antecedent and the consequent instance is the instantiated consequent. Agents reason about the applicability of the rule by considering their belief about the rule instance.

(*p-bel agent* (*rule-inst rule ant con*))

If this *p-bel* is pervasive then the agent believes that the rule applies with the given bindings. If the negation of the *p-bel* is pervasive then the agent believes that the rule does not apply, and he is uncertain otherwise. (*p-bel agent* (*rule-inst rule ant con*)) inherits its endorsement from the confidence measure of the rule. In our example a *def* endorsement is assigned to the *p-bel*.

(*p-bel Fred* (*rule-inst* "((*p-bel ?agent* (*cat ?animal*)))
⇒ (*p-bel ?agent* (*likes ?animal fish*))"
((*p-bel reasoner* (*cat Tiddles*)))
(*p-bel reasoner* (*likes Tiddles fish*)))
def)

This reads that Fred believes that any reasoner, if he has the rule about cats liking fish and the belief that Tiddles is a cat, should also have the belief that Tiddles likes fish. The rule instance includes the notion of an abstract reasoner so that agents can negotiate over the applicability of rules.

A rule instance r with an antecedent instance comprising n attitudes $a_1 \ldots a_n$ and a consequent-instance c, generates the following ATMS justification when the rule is fired

$$a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \ \& \ (\textit{p-bel agent } r) \Rightarrow c$$

In general, an instance of a conjunctive rule

$$a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \Rightarrow_k (\textit{and}((c_1 \ldots c_n)))$$

with associated rule instance r is equivalent to the set of instantiated rules (i.e. justifications)

$$\begin{array}{l} a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \ \& \ (\textit{p-bel agent } r) \Rightarrow c_1 \\ a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \ \& \ (\textit{p-bel agent } r) \Rightarrow c_2 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \ \& \ (\textit{p-bel agent } r) \Rightarrow c_n \end{array}$$

with (*p-bel agent r*) endorsed k. An instantiated disjunctive rule

$$a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \Rightarrow_k (\textit{or}((c_1 \ldots c_n)))$$

is equivalent to the following justification

$$a_1 \ \& \ a_2 \ \& \ \ldots \ \& \ a_n \ \& \ (\textit{not}c_1) \ldots (\textit{not}c_n) \ \& \ (\textit{p-bel agent } r) \Rightarrow \perp$$

with (*p-bel agent r*) endorsed k.

```
(p-bel Fred (penguin Tweety) 2c-pos)
(p-bel Fred (bird Tweety) 2c-pos)
(p-bel Fred (rule-inst "((p-bel ?a (bird ?x))) ⇒ (p-bel ?a (can-fly ?x))"
                       ((p-bel reasoner (bird Tweety)))
                       (p-bel reasoner (can-fly Tweety))) def)
(p-bel Fred (can-fly Tweety))
```

Figure 5.3: Belief set A in which Tweety can fly

```
(p-bel Fred (penguin Tweety) 2c-pos)
(p-bel Fred (bird Tweety) 2c-pos)
(p-bel Fred (rule-inst "((p-bel ?a (penguin ?x))) ⇒ (p-bel ?a (not (can-fly ?x)))"
                       ((p-bel reasoner (bird Tweety)))
                       (p-bel reasoner (not (can-fly Tweety)))) spec)
(p-bel Fred (not (can-fly Tweety)))
```

Figure 5.4: Belief set B in which Tweety cannot fly

The following example demonstrates the application of the rule instance. Fred believes that Tweety is a bird (i.e. (*p-bel agent* (*bird Tweety*)) and that birds can fly by default, i.e.

$$(\textit{p-bel ?agent}\,(\textit{bird ?x})) \Rightarrow_{[def]} (\textit{p-bel ?agent}\,(\textit{can-fly ?x}))$$

then Fred may infer that, by default, Tweety can fly. The rule instance corresponding to the application of this default rule to Tweety is

```
(rule-inst  "(p-bel ?agent (bird ?x)) ⇒ (p-bel ?agent (can-fly ?x))"
            ((p-bel reasoner (bird Tweety)))
            (p-bel reasoner (can-fly Tweety)))
```

However Fred also believes that Tweety is a penguin (i.e. (*p-bel Fred* (*penguin Tweety*))) and Fred has a specific rule which states that penguins cannot fly

$$(\textit{p-bel ?agent}\,(\textit{penguin ?x})) \Rightarrow_{[spec]} (\textit{p-bel ?agent}\,(\textit{not}\,(\textit{can-fly ?x})))$$

which leads Fred to infer that Tweety cannot fly. This results in a belief conflict which must be resolved by the belief revision mechanism. Belief sets which contain all the attitudes in the antecedent instance and the rule instance must also contain the consequent. In order to disbelieve the consequent the agent must either disbelieve some part of the antecedent instance or the rule instance itself. Consequently, ABR is applied to rule instances. Fred, having fired both rules on Tweety, has two belief sets; the first set, (A), containing the belief that Tweety can fly and being a penguin is immaterial (see Figure 5.3); and a second set, (B), in which Tweety cannot fly and the default rule does not apply in Tweety's case (see Figure 5.4).

We can see that the overall endorsements of the belief sets (A) and (B) favours (B), the correct assertion that Tweety cannot fly. In effect the *spec* rule overrides the *def* rule and Fred comes to believe that Tweety cannot fly. Fred comes to believe that the *def* rule does not apply to Tweety. That is, the particular instance of the rule with *?agent* bound to *Fred* and *?bird* to *Tweety* is not believed to be the case.

Logical inference rules of the form:

$$a_1 \mathbin{\&} a_2 \mathbin{\&} \ldots \mathbin{\&} a_n \mapsto_k c$$

also give rise to rule instances, ATMS justifications and labels. However, the difference is that whenever one or more a_i is disbelieved then the rule instance must also be disbelieved. In effect the rule instance does not form part of the justification of the consequent.

When both types of rule are used the following algorithm should be used to calculate the ATMS candidates:

1. Calculate the candidates from the nogoods using the algorithm in de Kleer (1986a). Given an inconsistent set of attitudes S, a candidate X is a minimal subset of S such that $S \setminus X$ is consistent.

2. For each candidate add those '$\mapsto$' type rule instances when either the consequent instance is in the candidate or some member of the antecedent instance is in the candidate.

5.5.3 The Inference Algorithm

In this section we address the problem of conflict resolution for inference, whereby an agent chooses one out of many possible inferences it can draw. Ideally agents should be able identify beliefs that are salient to their current task, and inferences relating to these beliefs which will have the most assured outcome. For example, an agent should be concerned more with robust justifications.

The principle of *recency* is used to constrain inference. It is naturally modelled by a stack-oriented approach to reasoning and dialogue, and this is used here to define the current task. An agent maintains a stack, the *inference stack*, of *sets* of attitudes, and pays more attention to attitudes towards the top of the stack. Each *stack set* comprises believed or uncertain attitudes, and when the agent decides to infer it starts at the top of its inference stack and looks for an inference not already drawn. When inferences are drawn the cognitive state is revised. Beliefs may change status, for example by becoming uncertain when they were previously believed. Any attitudes that are no longer believed or are uncertain are deleted from the stack. Attitudes which become believed or uncertain are added to the top of the stack and duplicates, if any, are deleted from lower down so that attitudes appear in only one stack set. Agents therefore infer from the most recently revised attitudes which are still either pervasive or uncertain.

The system chooses which rule to fire next in the context of its current cognitive state and inference stack. It searches down the inference stack taking each inference stack set A in turn until it finds an appropriate binding to a rule, and then fires the rule. The ordering over preferred rule bindings within an inference stack set is given by the inference algorithm. The inference algorithm has several steps:

1. For each literal in a stack set A find all rules and associated bindings for which the literal binds with a member of the antecedent or with the consequent.
2. Attempt to bind the remaining uninstantiated antecedent members of the incomplete rule instances returned in the previous stage with attitudes from the database. Discard those rule instances with incomplete instantiated antecedents and those rule instances which have already been fired.
3. Find those attitudes that are relevant to any member in A. Partially order the rule instances found in the previous stage, preferring those with more relevant attitudes in their antecedent instantiations.
4. Extend the order of rule instances returned by stage 3 to include the preference for rule instances built from more confident rules, e.g. a binding for a rule endorsed *spec* will be preferred to a binding for a rule endorsed *def*. Rules with belief set endorsement confidences (i.e. *1c-pos*, *2c-pos* etc) are preferred over those with intention set type (i.e. *desire-pos*, *effort-pos* etc).
5. If there is more than one preferred rule instance, choose one at random and execute it.

5.6 Planning

Inference is also used to implement planning. Planning is implemented at the rule level and plans are subject to revision in the same manner as beliefs. Rules generate possible intentions and possible future beliefs from other intentions, beliefs and future beliefs. The intention sets comprise consistent sets of intentions, i.e. sets of intentions which are mutually achievable.

Agents plan according to their beliefs. They have *desire rules* which take beliefs as their antecedents and intentions (called *leading intentions* of the plan) in their consequents and endorse the utility of the intention. They fire opportunistically when the beliefs in the cognitive state are not desirable and are of the form

$$(\textit{bel agent}\, p_1)\ \&\ \ldots \&\ (\textit{bel agent}\, p_n) \\ \Rightarrow_{[\textit{desire-pos/desire-neg}]} (\textit{p-int agent i})$$

The rules which give rise to the intention to resolve conflicting beliefs between agents are of this type (see chapter 7). For example, a state in which two agents disagree over a belief generates the intention to resolve this conflict.

$$(\textit{bel ?agent1 ?p})\ \& \\ (\textit{bel ?agent1 (bel ?agent2 (not ?p)))} \\ \Rightarrow_{[\textit{desire-pos}]} (\textit{or ((int ?agent1 (bel ?agent1 (not ?p)))} \\ \textit{(int ?agent1 (bel ?agent1 (bel ?agent2 ?p)))))}$$

The utility of this desire and the fact that the undesirable state exists endorses the intentions in the consequent. Our system considers two degrees of utility (i.e. *desire-pos* or *desire-neg*) and these are assigned as endorsements to the rule instances arising from desire rule.

5.6.1 Action Schemata

Agents also have *planning* rules and *action schemata* on which these operate. Action schemata describe ways in which a cognitive state can be revised. When the preconditions (i.e. a set of attitudes) of an instance of an action schema are satisfied the action can be performed to realise the effects (i.e. a set of attitudes) of the action.

Action schemata describe the primitive actions that can be performed by the agent. Each schema has six 'slots' and is of the form

(**action**
preconditions
action descriptor
effects
constraints
effort)

where

action the name of this action;

preconditions are the necessary preconditions for performing an action (if any), for example (*clear ?block*) states that *?block* should be clear;

action descriptor a description of the action that will be performed, for example (*stack ?block1 ?block2*) describes the action of stacking *?block1* on *?block2*;

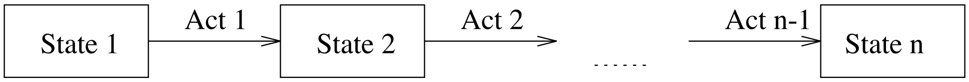
effects the effects of performing the action, for example (*on ?block1 ?block2*);

constraints the constraints that must hold before the action can be performed (if any) and that cannot be achieved through planning, for example (*cuboid ?block1*) describes the condition that *?block1* is a cuboid (spherical blocks form unstable stacks); and

effort the effort required to perform the action, for example *effort-pos*.

Constraints and preconditions are distinguished by the fact that it is possible to achieve a precondition which does not currently hold, whereas a constraint is not achievable if it does not already hold. For example, the fact that the speaker and hearer of a dialogue action instance must be different is a constraint.

An instantiated schema is called an action instance and consists of a set of bindings for the free variables in the slots of the schema. It is assumed that *executing* an instantiated schema has some effect in the real world, such as moving a block, sending a message or performing an inference.



5.6.2 Planning Rules

Planning rules deconvolve intentions into actions and other intentions to satisfy the action preconditions. Planning rules decompose existing intentions into sub-intentions.

$$
\begin{aligned}
&(\textit{p-int agent}\, i_1)\ \&\ (\textit{not}\ (\textit{bel agent}\, p_1))\ \ \&\ \ldots \&\ \ (\textit{not}\ (\textit{bel agent}\, p_n))\\
&\qquad \Rightarrow_{[premise]}\ (\textit{or}\ ((\textit{p-int agent}\, p_1)\ \ldots (\textit{p-int agent}\, p_n)))
\end{aligned}
$$

Intentions generated in this way are not assigned explicit utility but inherit the influence of utility endorsements through their labels in the same manner as derived beliefs inherit the influence of belief endorsements. Intentions to act are assigned negative effort endorsements (i.e. *effort-pos* or *effort-neg*).[23] The utility endorsements and effort endorsements govern the preference ordering of the consistent intention sets and ultimately contribute to the agents attitudes towards these intentions. The agent prefers minimal plans, is committed to intentions associated with minimal plans, and disintends those states for which the effort of achieving the state far outweighs the utility.

In accordance with section 5.5, planning rules have corresponding rule instances. A rule instance which justifies sub-intentions from a higher intention can be read as the belief that, in the particular instance, the agent believes that the rule applies. Again, these rule instances are corrigible and should naturally appear within *p-bel* propositions. However, in section 5.2.3 we demonstrated the need to separate possible intentions and beliefs in order to plan with belief commitments. Since our intention rule instances must be reasoned about within the context of intentions we introduce a *p-bel** intention-type attitude to accommodate intention set rule instances. These propositions appear in intention sets and not belief sets, and have a corresponding pervasive attitude *bel**, which is also constrained to intention sets. For example,

$$
\begin{aligned}
&(\textit{bel* fred}\ (\textit{rule-inst}\ \text{“}((\textit{p-int ?a}\ (\textit{happy ?a})))\Rightarrow(\textit{p-int ?a}\ (\textit{drink ?a beer}))\text{”}\\
&\qquad\qquad\qquad ((\textit{p-int reasoner}\ (\textit{happy reasoner})))\\
&\qquad\qquad\qquad (\textit{p-int reasoner}\ (\textit{drink reasoner beer})))\\
&\qquad\quad \textit{strong})
\end{aligned}
$$

Agents plan using the STRIPS planning paradigm with notions of sequences of states and actions which transform one state into the next (see figure 5.6.2). Each action has preconditions which are conditions which must hold in the preceding state of the action, and effects which hold true in the succeeding state if the action is successful. Agents plan to satisfy goal states, and seek sequences of actions which would transform the current state into the goal

---

[23] Recall that the conventional interpretation of endorsements is reversed for effort endorsements, i.e. *effort-pos* $<_e$ *effort-neg*.

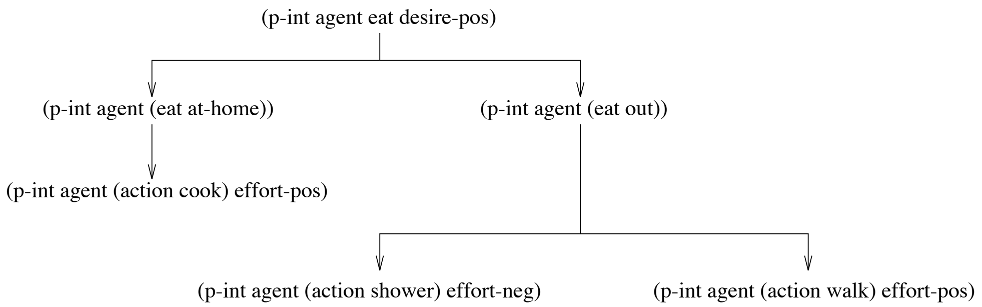

(p-int agent eat desire-pos)
(p-int agent (eat at-home))
(p-int agent (eat out))
(p-int agent (action cook) effort-pos)
(p-int agent (action shower) effort-neg)
(p-int agent (action walk) effort-pos)

Figure 5.5:

state. In our model the agent plans to satisfy leading intentions by intending a sequence of actions which would, if successful, transform the agent's current belief and intention state into the desired state.

Intentions give support to their sub-intentions. If an intention is dropped then this reduces the reason for maintaining its sub-intentions. An agent abandons an intention under any of the following conditions:

1. it is inconsistent with a preferred intention;
2. it requires too much effort to achieve;
3. it is no longer supported by any other attitudes and is positively undermined; or
4. it has been realised.

Conditions 1 to 3 arise out of the intention set preference mechanism. Condition 4 is a result of the consistency constraints on beliefs and intentions described in section 5.2.2.

5.6.3 Minimal Plans

When alternative plans are available the intention set preference orderings ensures that the least effortful alternative is preferred.

For example, if an agent wants to satisfy his desire to eat then he may have two plans available: to eat at home or to eat out at a restaurant. Eating at home requires cooking the food which is effortful (the preparation and washing up afterwards). Eating out requires the agent taking a shower (effortless) and walking into town. Walking requires lots of effort since our agent lives quite a distance from town. The resulting justifications for this scenario are shown in figure 5.5.

The ATMS mechanism generates three intention sets; one containing the intention to eat at home, another the intention to eat out and the third, not to eat at all.

1. { (*p-int agent eat desire-pos*), (*p-int agent* (*action cook*) *effort-pos*) }

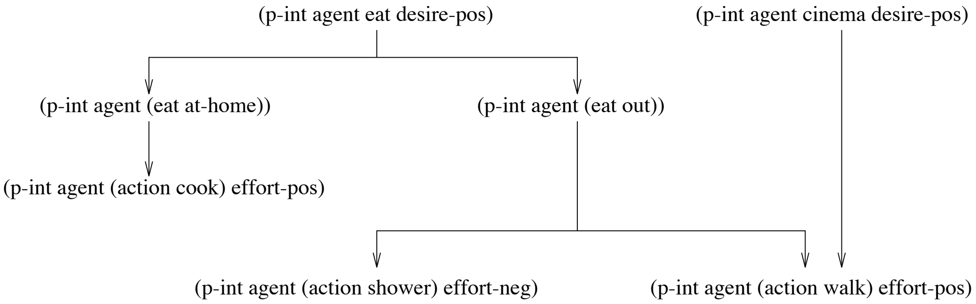

(p-int agent eat desire-pos)
(p-int agent cinema desire-pos)
(p-int agent (eat at-home))
(p-int agent (eat out))
(p-int agent (action cook) effort-pos)
(p-int agent (action shower) effort-neg)
(p-int agent (action walk) effort-pos)

Figure 5.6:

2. { (*p-int agent eat desire-pos*), (*p-int agent* (*action shower*) *effort-neg*), (*p-int agent* (*action walk*) *effort-pos*) }
3. { (*not* (*p-int agent eat desire-pos*)) }

The most endorsed set involves the plan to eat at home and this becomes the chosen plan.

If there is more than one leading intention then the preferred plan is a minimal (wrt effort endorsements) subset of actions which can achieve the leading intentions. For example, suppose our agent also has the intention to visit the cinema after he has eaten. The cinema is in town. The justification net with this extra intention is in figure 5.6.3. The intention sets with both leading intentions are

1. { (*p-int agent eat desire-pos*), (*p-int agent cinema desire-pos*), (*p-int agent* (*action cook*) *effort-pos*), (*p-int agent* (*action walk*) *effort-pos*)}
2. { (*p-int agent eat desire-pos*), (*p-int agent cinema desire-pos*), (*p-int agent* (*action shower*) *effort-neg*), (*p-int agent* (*action walk*) *effort-pos*)}

In this particular case the agent chooses to eat in town since his plan involves the least effortful actions to satisfy both his intention to eat and visit the cinema.

5.6.4 Alternative Plan Branching Points

This is a technical note concerned with the implementation of rules for generating alternative plans. When STRIPS planning it is often impractical or impossible to generate all possible alternative sub-intentions s_i from an intention S.

- $(int\, x\, s) \supset (int\, x\, s_1) \vee \ldots \vee (int\, x\, s_n)$ is equivalent to $(int\, x\, s) \wedge\neg\, (int\, x\, s_1) \ldots \wedge\, (int\, x\, s_n) \supset\perp$. This generates $n+1$ candidates and for m branch points we have $m(n+1)$ candidates.
- Non-omniscient agents may not be aware of all the alternatives that can form the disjunction. For example, if an agent intends to buy food then he might consider going to Sainsbury's or to the Co-op.

(*p-int agent* (*buy agent food*))
$\Rightarrow_{[premise]}$ (*or* ((*p-int agent* (*go agent sainsburys*))
(*p-int agent* (*go agent co-op*))))

However, suppose later he is told that there is a market in town then he would want to fire the rule

(*p-int agent* (*buy agent food*))
$\Rightarrow_{[premise]}$ (*or* ((*p-int agent* (*go agent sainsburys*))
(*p-int agent* (*go agent co-op*))
(*p-int agent* (*go agent market*))))

Since the ATMS maintains minimal labels, the second rule would not supersede the first. If both (*p-int agent* (*go agent sainsburys*)) and (*p-int agent* (*go agent co-op*)) were dropped then (*p-int agent* (*buy agent food*)) would also be dropped even though a further alternative exists (i.e. the market).

We solve these problems by introducing a new type of rule (i.e. the *dynamic disjunction rule* identified ' $\rhd$'), which operates by gradually building a disjunction via multiple rule firings. Rules of this type operate by replacing rule instances for disjunctions of n alternatives with new rule instances of $n + 1$ alternatives. Consider for example, firing the following rule

(*p-bel ?agent* (*a 0*)) &
(*p-bel ?agent* (*a ?x*))
$\rhd_{[desire\text{-}pos]}$ (*p-bel ?agent* (*b ?x*))

with *?agent* bound to *A* and initially *?x* bound to 1 and then *?x* bound to 2. The first firing is equivalent to

(*p-bel ?agent* (*a 0*)) &
(*p-bel ?agent* (*a 1*))
$\Rightarrow_{[desire\text{-}pos]}$ (*p-bel ?agent* (*b 1*))

The corresponding rule instance inherits its endorsement from the confidence assigned to the rule.

(*p-bel A* (*rule-inst* "(*p-bel ?agent* (*a 0*)) (*p-bel ?agent* (*a ?x*))
$\rhd$ (*p-bel ?agent* (*b ?x*))"
((*p-bel reasoner* (*a 0*))
(*p-bel reasoner* (*a 1*)))
(*p-bel reasoner* (*b 1*)))
desire-pos)

Call this rule instance *R1*. The second rule firing references the first and adds (*p-bel agent* (*b 2*)) to the disjunction in the consequent. This is equivalent to firing

(*p-bel ?agent* (*a 0*)) &
(*p-bel ?agent* (*a 1*)) &
(*p-bel ?agent* (*a 2*))
$\Rightarrow_{[desire\text{-}pos]}$ (*or* ((*p-bel ?agent* (*b 1*)) (*p-bel ?agent* (*b 2*))))

and endorsing the belief in (*p-bel A* (*not R1*)) premise. As a final remark, the first entry in the antecedent (called the *alternative dependent*) denotes the disjunction to which a particular rule firing belongs. For example, the rule

(p-bel ?agent (a ?y)) &
(p-bel ?agent (b ?y ?x))
$\qquad \rhd_{[premise]}$ *(p-bel ?agent (c ?x))*

when applied until closure to the following set of propositions.

(*p-bel A* (*a 1*))
(*p-bel A* (*a 2*))
(*p-bel A* (*b 1 1*))
(*p-bel A* (*b 1 2*))
(*p-bel A* (*b 2 1*))

will be equivalent to the following rules.

(p-bel A (a 1)) &
(p-bel A (a 1 1)) &
(p-bel A (a 1 2))
$\qquad \Rightarrow_{[premise]}$ (or (*(p-bel A (c 1)) (p-bel A (c 2))*))

(p-bel A (a 2)) &
(p-bel A (a 2 1))
$\qquad \Rightarrow_{[premise]}$ *(p-bel A (c 2))*

Alternative dependent intentions are not removed from the inference stack when they are disintended. This allows agents to construct alternative plans even when the existing plan has been abandoned. How the agent plans is discussed in more detail in appendix B.

5.7 The Agent Action Cycle

In this section, we attempt to draw together the components of the agent to give a picture of its overall operation, which we illustrate with an example. The operation of the agent comprises an initialisation stage followed by iterations of a read, infer, write cycle. From a predetermined set of assumptions the agent constructs the initial database, chooses preferred attitude sets, and pushes its intentions onto the inference stack.[24]

At each subsequent cycle the system starts at the lowest layer and works its way upwards.

1. if there are any incoming messages on the message board the MIU adds them to the database as premise *pbel* attitudes about the beliefs or intentions of the speaker and pushes the result as a single group onto the inference stack;

---

[24] The agent always infers from intentions to begin with.

2. the ATMS (re)computes the possible consistent belief sets, taking into account any new messages just added and any new beliefs or inconsistencies introduced at the last cycle;
3. the Attitude Revision component orders the sets and works out commitments to the beliefs and intentions in the most preferred set;
4. the Inference Engine searches down the inference stack, applying the inference algorithm to each group of attitudes until an inference can be drawn. If an inference is drawn, the new information (i.e. rule instance, consequent labels) is added to the database for use at the next cycle.

Only one inference is drawn per cycle. This can be either a belief inference, a plan inference or a prediction. The Inference Engine chooses whether to fire a plan type rule (i.e. $bel \Rightarrow int$ or $int \Rightarrow int$) or a belief type rule (i.e. $bel \Rightarrow bel$) according to recency and the endorsement values assigned to the rules. Belief rules are preferred to planning rules. If no new beliefs can be inferred, the Inference Engine attempts to extend the current plan given the set of most preferred beliefs and intentions—any new plan steps (i.e. intentions) are added to the Database together with their supporting justifications. Hence, the agent tends to infer a whole bunch of beliefs and then a whole bunch of intentions depth first. If neither a belief or a plan steps can be inferred, the Inference Engine tries to fire a prediction rule. Actions can be viewed as a special kind of inference which results in a message being sent via the MIU to another agent.

The resulting system is modular, with each 'layer' acting on the output of the layer below. The Inference Engine knows about inference stack, leading intentions, and the last utterance. The Attitude Revision system knows about endorsements and commitments and how to compute them. The ATMS knows about manipulating sets of nodes (representing beliefs and intentions) and the justifications linking them (representing their supports) stored in the database.

For example, the Inference Engine isn't told and doesn't know about belief revision. If the plan it was working on is invalidated by belief revision, the intentions, beliefs and rule instances representing the plan are simply no longer in the most preferred belief set(s) generated by the Attitude Revision component and any references to the old plan (in the form of leading intentions etc.) are silently removed. The Inference Engine does not grieve over lost plans, for it has no independent memory of the plan it was working on. At each cycle, its task is simply to take the leading intention(s) and most preferred belief/intention sets it is 'given', find the best planning rule to apply in the circumstances and apply it, putting the result – a new intention and its justification – back in the database. It is in this sense that the Attitude Revision system 'plans', by throwing away plans which are too costly.[25] The role of the Inference Engine is therefore limited to proposing candidate plans (or rather adding new plan steps)—the decision on how good a plan is is taken as a 'side effect' of the BR process.

---

[25] Actually the Attitude Revision system doesn't know it is doing this, only that the sets which happen to include the plan are not the most preferred sets according to its preference rules.

5.7.1 A Blocks World Example

The following example is taken from the blocks world and demonstrates how the system would plan in a dynamic environment. We have deliberately chosen a non-dialogue example at this point as we do not consider the implications of dialogue management until chapter 7.

There are three blocks labelled a, b and c. An agent observes that block c is on b and block a is on its own to the left and subsequently has the possible belief

$$(\textit{p-bel agent}\ \begin{array}{ccc} & & \boxed{c} \\ \boxed{a} & & \boxed{b} \\ \hline \end{array}\ \textit{1c-pos})$$

The agent also has the desire that the blocks should be stacked 'a on b on c' which is represented by the desire rule

$$(\textit{not}(\textit{bel agent}\ \begin{array}{c} \boxed{a} \\ \boxed{b} \\ \boxed{c} \\ \hline \end{array})) \Rightarrow_{\textit{desire-pos}} (\textit{p-int agent}\ \begin{array}{c} \boxed{a} \\ \boxed{b} \\ \boxed{c} \\ \hline \end{array})$$

From the desire rule the agent generates the leading intention

$$(\textit{p-int agent}\ \begin{array}{c} \boxed{a} \\ \boxed{b} \\ \boxed{c} \\ \hline \end{array})$$

The agent can also move a single block at a time using the *transform* action which transforms the *precondition* state of the blocks into the *effect* state of the action. For example, if the blocks are in the following configuration

$$\begin{array}{cc} & \boxed{c} \\ \boxed{a} & \boxed{b} \\ \hline \end{array}$$

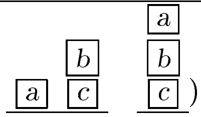
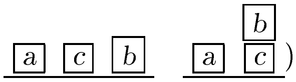
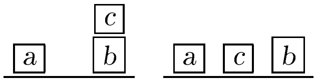
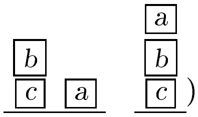
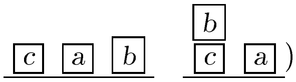
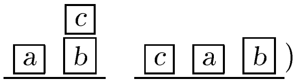
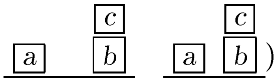
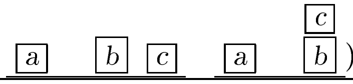
then the action

$$(\textit{transform}\ \begin{array}{cc} & \boxed{c} \\ \boxed{a} & \boxed{b} \\ \hline \end{array}\quad \begin{array}{ccc} \boxed{a} & \boxed{b} & \boxed{c} \\ \hline \end{array})$$

will unstack the c block from the b block which is then placed to the right of the b block producing the effect

$$\begin{array}{ccc} \boxed{a} & \boxed{b} & \boxed{c} \\ \hline \end{array}$$

$\begin{array}{cc} & \boxed{c} \\ \boxed{a} & \boxed{b} \\ \hline \end{array}$ is the precondition of the transform and $\begin{array}{ccc} \boxed{a} & \boxed{b} & \boxed{c} \\ \hline \end{array}$ the effects. In order for an action to be performed the blocks configuration must correspond to the preconditions of the action. The blocks are heavy and actions which involve lifting are effortful (i.e. *effort-pos*) where those that require dragging or lowering a block require less effort (i.e. *effort-neg*). The following are legal moves:

	Action	*Effort*
act1	(*transform* )	*effort-pos*
act2	(*transform* )	*effort-pos*
act3	(*transform* 	*effort-neg*
act4	(*transform* )	*effort-pos*
act5	(*transform* )	*effort-neg effort-pos*
act6	(*transform* )	*effort-neg effort-neg*
act7	(*transform* )	*effort-neg*
act8	(*transform* )	*effort-pos*

Since the blocks are believed to be in state "c on b and a to the left" then all other beliefs about the state of the blocks (*not* (*bel agent* . . .)) must be asserted. The network divides into two alternative plans to achieve the leading intention as illustrated in Figure 5.7 below

Plan	*Effort*
act3 → act2 → act1	*effort-neg effort-pos effort-pos*
act7 → act6 → act5 → act4	*effort-neg effort-neg effort-neg* *effort-neg effort-pos effort-pos*

Figure 5.7: Alternative plans for stacking the blocks

The top three intention sets in the preference hierarchy contain

1. leading intention and plan *act*3 → *act*2 → *act*1;
2. no leading intention and no plan;

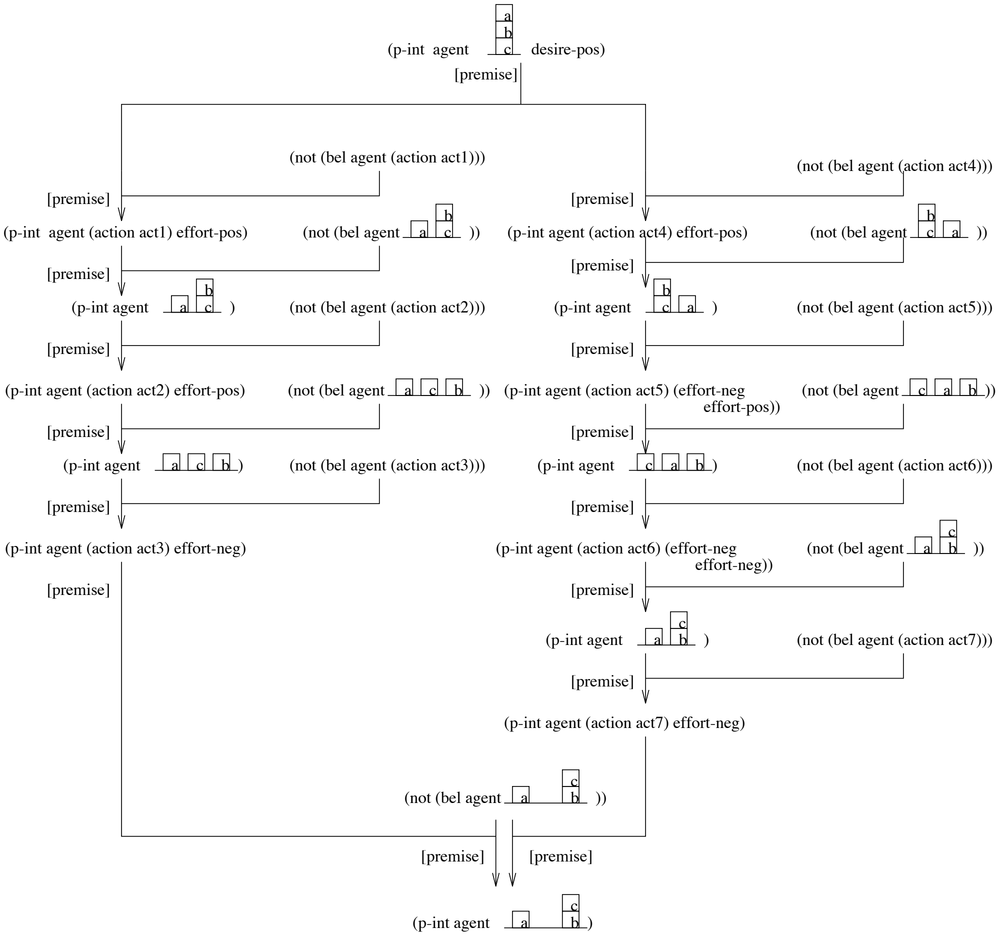
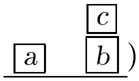

(p-int agent a b c desire-pos)
[premise]
(not (bel agent (action act1)))
(not (bel agent (action act4)))
[premise]
(p-int agent (action act1) effort-pos)
(not (bel agent a b c))
(p-int agent (action act4) effort-pos)
(not (bel agent b c a))
(p-int agent a b c)
(not (bel agent (action act2)))
(p-int agent b c a)
(not (bel agent (action act5)))
(p-int agent (action act2) effort-pos)
(not (bel agent a c b))
(p-int agent (action act5) (effort-neg effort-pos))
(not (bel agent c a b))
(p-int agent a c b)
(not (bel agent (action act3)))
(p-int agent c a b)
(not (bel agent (action act6)))
(p-int agent (action act3) effort-neg)
(p-int agent (action act6) (effort-neg effort-neg))
(not (bel agent a c b))
(p-int agent a c b)
(not (bel agent (action act7)))
(p-int agent (action act7) effort-neg)
(not (bel agent a c b))
(p-int agent a c b)

Figure 5.8: Blocks world justification network

3. leading intention and plan $act7 \rightarrow act6 \rightarrow act5 \rightarrow act4$

The preferred set contains the leading intention and plan $act3 \rightarrow act2 \rightarrow act1$. The plan $act7 \rightarrow act6 \rightarrow act5 \rightarrow act4$ requires more effort than the utility of the leading intention and the agent would prefer to drop the leading intention than adopt this plan.

The agent can perform $act3$ as the precondition (*bel agent* ) is satisfied. Once the action has been executed, (*bel agent* (*action act3*)) is made a premise (i.e. the agent believes the action has been done) and (*p-int agent* (*action act3*)) is dropped by the intention/belief consistency rule. The agent also infers by default that the action has been successful and that its effects will eventually hold true. The precondition for $act2$ is now satisfied and this action can be performed. Finally, $act1$ is performed and the default belief that the leading intention has been achieved is asserted. The justification network for these plans is shown in figure 5.8

The stack characteristic of the inference algorithm results in plans being constructed top down. If only one plan is considered and this is dispreferred then the leading intention is automatically dropped. Since inferences are only drawn from pervasive and uncertain intentions, alternative plans may not be considered. There are two mechanisms to alleviate this problem:

1. generate all alternative sub-intentions simultaneously; and
2. mark all branching points and allow the inference algorithm to infer from intentions with alternative sub-intentions even when the intention has been dropped.

Related to this is the fact that agents only explore alternative plans when either the effort associated with the current plan exceeds the utility of the leading intention, or the preconditions of the current plan have no actions to solve them. The former case is explored in chapter 7 and is shown to exhibit stability of intentions. The latter case removes a potential planning deadlock situation, and allows the agent to consider alternative plans when it has no rules to fire to satisfy the preconditions of the current plan. Also, the agent draws single inferences from an attitude before moving on. This may be inadequate since two or more inferences from an attitude may be immediately relevant. An agent may not explore the pros and cons for its commitment to an attitude. However, since agents prefer to draw strong justifications this problem is not crucial.

5.7.2 Revising the Plan

So far we have considered planning in a perfect world where actions are always successful. This, however, is a default assumption given that no other process has contrived to reduce the success of the action. In the remainder of this section considers a number of scenarios derived from the above example in which the world varies from this assumption and in which the agent chooses to revise his plan.

Failed action

After action *act*3 the agent observes that the state of the blocks has not changed. The agent has asserted (*bel agent* (*action act3*)), and has dropped (*p-int agent* (*action act3*)), but still believes (*not* (*bel agent* $\underline{\boxed{a}\ \boxed{c}\ \boxed{b}}$)). From figure 5.8 it is evident that the entire plan current plan is dropped and plan *act*7 → *act*6 → *act*5 → *act*4 is asserted in its place.

Choosing alternative plan

If we imagine that the blocks are moved when the agent is not looking and he observes the blocks state has changed to $\underline{\boxed{c}\ \boxed{a}\ \boxed{b}}$ .and (*bel agent* $\underline{\boxed{c}\ \boxed{a}\ \boxed{b}}$) is now believed and the top three intention sets in the preference ordering become

1. the leading intention and the plan *act*5 → *act*4;

2. the leading intention and the plan *act*3 → *act*2 → *act*1; and
3. no leading intention and no plan.

The agents alternative plan involves less effort than his current plan and he revises his intentions to accommodate this. The preferred intention set now contains the subpart of the alternative plan *act*5 → *act*4.

Dropping both plans and the leading intention

If, prior to acting, the agent comes to believe that the blocks are actually in state $\underline{\boxed{a}\quad\boxed{b}\ \boxed{c}}$ then he must augment both plans to include action *act*8. This introduces a further *effort-pos* endorsement to both plans. Since the effort of both plans exceeds the utility then the leading intention is dropped.

Chapter 6

Implementing the BBD Architecture

In this chapter we outline a partial implementation of the extended BBD model described in chapter 3. The modules comprising the librarian are implemented as a collection of rules which are interpreted by the extended belief revision system described in the previous chapter. In section 2 we motivate the facts and rules forming the librarian database and the satisfaction conditions for each of the modules comprising the librarian agent. In section 3 we describe the overall architecture of the system and the goal hierarchy resulting from the task decomposition.

It is important to stress that much of the work in this chapter is largely illustrative of the kind of librarian we could build within the framework presented in chapter 5. As such, it attempts to address the issues of belief revision, dialogue management and control identified as critical in chapter 3. Indeed this is the librarian we intended to build until we ran into a number of problems which are discussed in more detail in chapter 9. These problems imposed severe limitations on the kind of system it was feasible for us to build. The actual implementation outlined in section 6.3 is thus extremely rudimentary, consisting of little more than a fragment of the system outlined below and exists purely to drive the dialogue fragments presented in the next chapter.[1]

6.1 The Functional Experts

We make the following basic assumptions about the information retrieval task:

1. the user has a problem description;
2. there is an overlap in the user's and librarian's knowledge;
3. there is a retrieval strategy for some from of the problem description;
4. the user will ultimately accept the retrieval strategy proposed by the librarian; and

---

[1] The 'Obtaining the Problem Description' task has been simulated using OPS5 without belief revision.

5. we don't engage in a literature search, i.e. we are at the pre-search stage.

As described in chapter 3 we use a simplified form of the BBD model consisting of five modules: Problem State; Problem Mode; User Model; Problem Description; and Retrieval Strategy. Each module is associated with a *goal*. We distinguish between goals which should be satisfied (wants to know) and goals which must be satisfied (needs to know), as follows.

1. The *Problem State module* wants to know the *problem-state*. The *problem-state* is either *early* (the user is just beginning) or *middle* (the user has already done some work on the problem.
2. The *Problem Mode module* wants to know the *document-type*. The *document-type* is one of *books* (the user is looking for books) or *photographs* (the user is looking for photographs).
3. The *User Model module* wants to know the *user-status*. The *user-status* is one of *novice* or *expert*.
4. The *Problem Description module* needs to know the *problem description*, where a description is a collection of problem *descriptors*.
5. The *Retrieval Strategy module* needs to know the *search request*, where a request is a collection of search *terms*.

The Problem State, Problem Mode and User Model modules are relatively straightforward, requiring only the choice of a value from a fixed set of alternatives. The main objectives of the system are therefore twofold: to build a *problem description* in the Problem Description module; and to build a *search request* in the Retrieval Strategy module. As indicated in chapter 3, one of the major problems in modelling the retrieval task is in defining usable satisfaction conditions for descriptions and requests. In the remainder of this section we discuss the goals associated with the Problem Description and Retrieval Strategy modules in detail and describe their satisfaction conditions. We introduce the notions of a *valid*, *minimal* and *good* description/request as a basis for the system satisfaction conditions. As mentioned earlier, we have taken Architectural History as our test domain and all our illustrations are therefore drawn from this. In what follows, problem descriptors are denoted by an intensional description of the corresponding set, e.g. $\{x \mid church(x)\}$ or in abbreviated form as the predicate in single quotes in italic, e.g.'*church*'. 'Entry terms' are denoted by the corresponding English word in double quotes, e.g. "church". Search terms are written in italic within double quotes, e.g. "*church*". Terms representing the agent's cognitive state, the rules used by the agent etc. are denoted in the usual way, e.g. $(bel\, user\, p)$.

6.1.1 The Problem Description

A problem description is a complex concept. To simplify our task we shall assume that the problem description consists of four components: a *topic* component; a *subject area* component; a *document type* component; and a *document* *level* component. The topic component consists of a disjunction of problem descriptors derived from the user's description of their information need. A *problem descriptor* is a concept or a predicate which describes the information the user needs to solve their problem.[2]

A problem description is *minimal* if the topic component consists of at least three descriptors and if the descriptors collectively are neither too general nor too specific. A problem description is *good* if it consists of a topic component, a document type component and a document level component. The subject area component is inferred from the topic component for both minimal and good problem descriptions. We need a minimum description and want a good description.

The Topic Component

The *topic* is the critical part of the problem description. The user requires information to solve a problem. For example, the user might be a student who has to write an essay on Wren's London churches. The user seeks the librarian's assistance in identifying those documents which (hopefully) contain the information the user requires. In doing so, the user must describe their problem and the information they require to the librarian. (Note that the user describes the information they require, not the documents themselves, although in some situations they may specify how they information is to be represented, e.g. as pictures or text.) However the user often does not know which precisely which information they need to know to solve their problem ("I want the construction date of Christ Church Newgate") and must instead describe an object which is to be the target of the information retrieval. This object may be an individual (e.g. St. Martin-in-the-Fields), a class (e.g. Wren's churches), an idea or relationship (e.g. the influence of liturgy on church design) or a process (e.g. the construction of the dome of St. Paul's Cathedral).

It is important to distinguish between the description of the user's problem as an information need and the the retrieval strategy designed to meet this need. While both deal with objects and processes in the real world, the former is likely to be much richer and more complex than the latter, and an accurate description of the user's problem is independent of whether it can subsequently be converted into a retrieval strategy. We may understand the user's problem, even if the best we can do if offer books on churches in general. Conversely, there is no point in building a problem description which is too detailed.

If we attempt to represent the user's information need, we get something like:

$$\{x \mid \mathit{document}(x) \land \mathit{describes}(x, q)\}$$

i.e. the set of documents which describe the object of the user's query q. The problem is that these objects can themselves refer, resulting in complex nested descriptions (e.g. "I am looking for books on paintings of Baroque churches").

---

[2] Note that other components of the problem description and the retrieval strategy refer to the *representation* of the information or to the documents containing the information, e.g. text *vs.* photographs, books in English *vs.* books in French etc.

To avoid ambiguities, it is necessary to explicitly represent the object(s) to which the descriptors apply, otherwise we can't tell if the problem description "*baroque* $\wedge$ *church land painting*" refers to paintings of baroque churches or painting in baroque churches such as altar pieces or frescoes.[3] Individuals are represented as a singleton class by specifying descriptors which uniquely identify the object concerned e.g. $eq(x, \textit{St. Martin-in-the-Fields})$, or indexically as in "I am looking for more information on the technique described in this paper". Descriptions can be nested to any level required so that it becomes possible to specify properties of the representation and reference, e.g. "I am looking for books on frescoes depicting the Resurrection". The latter would be represented as:

$$\{b \mid \mathit{book}(b) \wedge \mathit{describes}(b,' \{f \mid \mathit{fresco}(f) \wedge \mathit{depicts}(f, r) \wedge \mathit{resurrection}(r)\})\}$$

and reviews of books describing Wren's churches as:

$$\begin{aligned}&\{r \mid \mathit{review}(r) \wedge \\ &\quad \mathit{describes}(r,' \{b \mid \mathit{book}(b) \wedge \\ &\qquad\qquad\qquad \mathit{describes}(b,' \{c \mid \mathit{church}(c) \wedge \mathit{architect}(c, \mathit{Wren})\})\})\}\end{aligned}$$

using a first-order language extended with quotation (e.g. (Perlis 1985), (Hadley 1990)). All descriptors with the exception of those which accept quoted descriptions as arguments such as *describes*, *depicts* etc. are assumed to take a single (free) variable as an argument. However, to simplify the problem, we shall assume that the problem description contains no nested terms, i.e. that all the descriptors refer to a single implicit object or set of objects which are not themselves descriptors. Unless care is taken, this can result in ambiguous descriptions. For example, there are at least two objects that have the property of being St. Pancras: the church and the station. Similarly '*church frescoes*' must be viewed as a single compound descriptor denoting a a type of fresco rather than the (empty) intersection of the set of things which are churches and the set of things which are frescoes (see below). To simplify the representation, we shall denote problem descriptors by simple (quoted) terms, for example the term '*church*' represents the set of all churches $\{x \mid \mathit{church}(x)\}$ and '*St. Martin-in-the-Fields*' represents the class $\{x \mid \textit{St. Martin-the-Fields}(x)\}$; x is (has the property of being) St. Martin-in-the-Fields.

We assume that the user communicates to the librarian an unambiguous intensional description of a class of objects, which may be individuals, relations or processes. We also assume that both the user and the librarian use similar sets of problem descriptors, i.e. there are no translation problems. We allow the user and librarian's sets of descriptors to be partially disjoint, but we assume that they have enough descriptors in common to permit the definition of any descriptors they do not have in common. The components of the problem description are represented as propositional attitudes in the agent's database.

The set of problem descriptors recognised by the system form a hierarchy (or several hierarchies) of concepts. For example, '*St. Martin-in-the-Fields*' is

---

[3] Note that we may not be able to distinguish these two cases at the Retrieval Strategy level.

defined to be a subclass of '*church*', '*review*' to be a subclass of '*article*' etc. Each hierarchy of descriptors has a top node or most general descriptor. When attempting to build a problem description our objective, in accordance with conventional retrieval strategies, is always to narrow the intersection of the existing descriptors.[4] One way to do this is to ask about most general descriptors which have subclasses that could narrow the intersection. If descriptors share a more general descriptor, they must have have objects in common. However for this to work, the most general descriptors must be thought of as defining sorts or types which constrain the arguments of the descriptors below them in the hierarchy. This in turn means that the same descriptor cannot occur in more than one hierarchy. For example, we must distinguish between '*baroque architecture*' (a kind of architecture) and '*baroque music*' (a kind of music); although architectural and musical styles overlap, neither is a proper subset of the other. This is arguably more natural than having an 'artificial' class of all baroque things, but there are many cases where multiple inheritance would be useful.

We assume that each problem descriptor comprising the topic is associated with a list of one or more words or synonyms which has the descriptor as their denotation. One of these words is selected as the primary expression of the concept and is tried first when attempting to map the concept to a search term. If this mapping fails, the synonyms are tried. If no such mapping exists, we replace the descriptor with the descriptor which forms its smallest superclass and try again, starting with the primary expression of the new concept. For example the smallest superclass of '*St. Martin-in-the-Fields*' is the class '*church*' and the smallest superclass of '*review*' is '*article*'. Similarly, if the retrieval strategy is too general, we can consider replacing the each descriptor with the descriptor(s) corresponding to its largest subclass(es).

The Subject Area Component

The top node in each hierarchy of problem descriptors has associated with it one or more *subject areas*. The subject area for each problem descriptor is found by following the 'kind-of' or 'is-a' links to the top of the descriptor hierarchy and looking up the subject areas for the most general descriptor. Note that each descriptor in the topic specification of the problem description may have its own set of subject areas.

Whereas the topic is derived from the user's information need, the subject area is part of a generally agreed classification scheme which serves to characterise the contents of a database containing appropriate references. (Having said this, a given subject area may be covered by two or more databases e.g. the subject area 'chemistry' may appear in both 'science abstracts' and 'manufacturing technology'; a database may cover more than one subject area, e.g. physics and chemistry; or the subject area may be split across several databases, e.g. organic and inorganic chemistry.) Even when the topic is a subject area, e.g. 'cosmology', the required material would be about cosmology *as a subject*

---

[4] Unless it turns out to be too narrow.

and the appropriate subject area might be 'the history of science'. When the same topic falls within several different subject areas, the subject area implies a particular perspective on the topic. For example, in a molecular biology database the subject area may focus on the structure and properties of DNA, whereas for a genetics database the subject area would focus on the role of DNA in reproduction.

Subject areas can be thought of as meta-classes. For example, the subject area 'architectural history' can be thought of as dealing with the relationships between building types, styles periods and architects; 'building technology' with the class of construction technologies, and so on. As with simple descriptors there are relationships between subject areas. For example, the subject area 'architecture' would normally be thought of as subsuming the subject area 'architectural history'. (On the other hand, the subject area 'buildings' might or might not subsume the subject area 'architectural history', depending on how it is construed: architectural history is often concerned with un-executed projects which would not be subsumed by 'buildings-in-the-world'.) Again, there are problems when the natural most general descriptor doesn't fall neatly within a subject area. For example, not all architects would fall within the subject area 'architectural history', only the famous dead ones.

For the purposes of the project, we assume a single subject area, 'architectural history'. Ambiguous descriptors are assumed to have an 'architectural' prefix or suffix, e.g. 'architectural-style', 'baroque-architecture'. We assume that there are a number of attributes of the problem which are relevant to all queries: building-type, architect, period, style. Not all attributes are valid in all circumstances, e.g. the 'architect' attribute would not be relevant when seeking information about early Gothic churches, but we allow null values for attributes so this should not be a great problem. Many combinations of attributes have no intersection, e.g. 16th century Gothic railways stations; however it is anticipated that this will be a fruitful source of problems which can be resolved by negotiation.

The Document Type and Level Components

The *document type* consists of a disjunction of document types. The legal types are *books* and *photographs*. (Note that this information also forms part of the retrieval strategy specification.) The *document level* is one of *introductory* or *advanced*. The document type and document level serve primarily to further characterise the possible database(s).

6.1.2 Retrieval Strategy

A search request consists of four components: a *term* component; a *subject area* component; a *database* component; and a *document type* component. A search request is *minimal* if it consists of a least a search term component. A search request is *good* if it consists of a term component, a database component and a document type component. As with the problem description, in reality there would be additional constraints on a good request. For example too many terms which are too specific may not return all the relevant documents (or any at all)—too few terms or terms which are too general will return too many documents. (We use the position of the term in the search term hierarchy rather than the number of occurrences of each term in the database as an estimate of how discriminating the term will be and hence how useful it will be in controlling the search.) Nor is a binary classification into acceptable and non-acceptable requests realistic. A request may be ambiguous, requiring clarification to determine the user's intent, or may suggest possibilities that the user had not considered. For example, if the user is is looking for information on 'the implementation of UN resolutions', the existence of two databases covering the topic, say 'current affairs' and 'international law', may suggest additional ways of looking at the user's problem such as 'the standing of UN resolutions in international law' or 'the political consequences of their (non) implementation'.

The Term Component

The *term* component consists of a disjunction of search terms derived from the problem description. A search term serves to label a class of documents. There are a number of indexing strategies in common use, including: indexing on a particular morphological norm for a set of variant word forms (e.g. the singular noun for plural noun, verb etc.); indexing on a preferred word in a group of synonyms or related words (e.g. the 'entry words' "kirk", "church", and "churches" all map to the index term "*church*"); and indexing using a list of legal index terms—(note that terms in the legal list may be restricted to a single sense, e.g. in one database the term "*bank*" may mean a financial institution and in another a type of building; note also legal terms re often supplied with entry words). In the general case, a search term stands for a complex English language description of a document class. As with problem descriptions, the components of the retrieval strategy are represented as propositional attitudes in the agent's database. The set of search terms recognised by the system form a hierarchy (or several hierarchies) of index terms distinct from the descriptors used by the problem description. The top node in each hierarchy of search terms has associated with it one or more *subject areas.* As with problem descriptors, the subject area for each search term is found by following the 'kind-of' or 'is-a' links to the top of the term hierarchy and looking up the subject areas for the most general term. Note that these subject areas are not necessarily the same as those used to classify the problem descriptors. However, for the sake of simplicity, we will initially assume a null mapping, i.e. a 'problem descriptor' subject area maps to an identical 'search term' subject area.

In reality, a term hierarchy may be unique to a singe database, or common to a number of databases. In addition a given database may support more than one indexing scheme. However to simplify matters, we shall assume that although the search terms used by the system cannot be organised into a single hierarchy they are all drawn from a single controlled indexing language used by all the databases.

The Database and Document Type Components

The *database* component consists of a disjunction of database names. The *document type* component consists of a disjunction of document types.

6.2 System Architecture

In this section, we discuss the how the architecture outlined in chapter 5 can be used to achieve the librarian's goals. We outline the decomposition of the high-level goals identified in the previous section to lower-level goals using goal decomposition or planning rules. Actions to achieve these lower-level goals, i.e. speech acts, are discussed in the next chapter, as is the problem of maintaining coherence in the dialogue (at least on the librarian's part). As mentioned at the beginning of the chapter, this in an inteded architecture: the more limited version we actually implemented is described in section 6.3 below.

The facts and rules comprising the librarian are loosely organised into a number of *modules*. Information from one module may be useful to other modules. In particular the Retrieval Strategy module uses information from all the other modules to derive a retrieval strategy. This dependency is reflected in the task decomposition outlined below. For example, the user model and problem state provide information about the user (student, researcher etc.) and any searches the user may have requested prior to the current request. This information can be used to infer the document level and hence the databases which may be appropriate. It is anticipated that there will be a particularly strong relationship between the Problem Description module and the Retrieval Strategy module. Failure to derive a minimum search request can result in revision of the search request, the problem description or both.

6.2.1 Task Decomposition

In reality, the librarian wants to help the user solve their problem. This assistance may take many forms: providing advice about the available on-line services, suggesting the user try a specialised library or a library at another institution etc, as well as locating references to documents likely to contain the information the user requires. Even in the latter case, finding the documents the user requires may involve many searches spread over a number of sessions, the librarian consulting their colleagues or information about the on-line services under consideration.

However, for simplicity, we assume that the top-level goal is one of producing a retrieval strategy which meets the user's problem description. A top-level decompisition of this goal is shown in Figure 6.1[5]

---

[5] Unfortunately there are a number of problems with this simple decomposition. Of the subgoals listed above, only the problem description is necessary to derive a retrieval strategy. While we can distinguish between goals with differing degrees of commitment, we cannot represent deontic attitudes such as 'must', 'should' and 'may'. For an AND node to succeed, all its sub-goals must succeed, no matter how weak the commitment to the sub-goals.

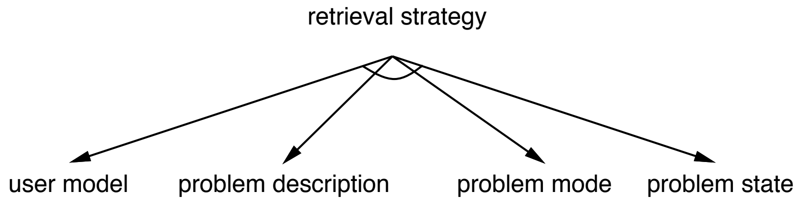


Figure 6.1: Task Decomposition

In general subgoals would be attempted in left-to-right order. Daniels' (1987) analysis suggests that dialogues between users and information retrieval specialists typically follow the pattern: information about the user; the user's motivation; the background to the problem; and the problem in detail, and we follow this.[6]

The strategy adopted in this system is one of top-down goal reduction interleaved with bottom-up solution evaluation. In general, the goal decomposition rules are heuristics, i.e. they are abductive: $(int\, x\, p) \Rightarrow (int\, x\, q)$ does not imply that $q \supset p$ (this is only true when p and q are logically equivalent). Whether the heuristic is successful in the current context is determined by the 'deductive' rules which evaluate the proposed solution to see if it does, in fact, satisfy the goal. The acceptability or otherwise of a solution is determined by the rules which constitute the system's knowledge or theory of the domain. In many cases such evaluation is immediate, however in the problem description and retrieval strategy cases a number of conditions must be shown to hold before the solution is deemed to be satisfactory.

A solution fails when it leads to a state which is inconsistent with the intended state. All unsatisfactory evaluations are, by assumption, inconsistent with the intention and lead to part of the solution being abandoned. For example, given the intention $(int\, x\, p)$ and the possible solution q, if $q \supset \neg p$, q is inconsistent with the intention to achieve p. Which part of the solution is abandoned is determined by heuristics that provide justification for a particular component not being part of the solution. In situations where there is no single culprit, the system considers all possible ways of restoring consistency and selects that which is most preferred. If all are equally preferred it picks one at random.[7] For example, if the three descriptors d_1, d_2 and d_3 are *collectively* too vague to form the basis of good retrieval strategy, it is unreasonable to pick one as the culprit. The sensible thing to do is to ask the user to be more specific, or to suggest specialising one of the descriptors (perhaps chosen at random if we have no other grounds for choice).

---

[6] At present there we have no means of specifying default orderings on plan steps unless one plan step is a precondition for the next.

[7] There is a problem here with uncertain intentions which may cause the system to do nothing at all if no course of action is clearly better than the alternatives.

Obtaining the User Status

This is achieved using a simple *ask* speech act, possibly supplemented by a *tell* act if the user responds with a question asking why the system wants to know. (Action schemata to implement the speech acts are described in the next chapter.) The justification would be that the information will assist the system in inferring the level of document the user requires.

Obtaining the Initial Problem Description

It is assumed that when the system asks the user for a problem description the user will volunteer an *initial problem description* consisting of one ore more problem descriptors. This is accomplished using a simple *ask* speech act. Since there is no natural way of obtaining a a set of n descriptors from the user, we do not attempt to. Instead we build up the problem description incrementally, looking for a good set of problem descriptors. We continue elaborating the problem description until we get such a set or until we run out of questions to ask. Typically we won't be able to use everything the user tells us anyway. 'Unsatisfactory' problem descriptors remain part of the problem description—after all they are probably still true—unless they are explicitly negated by the user. Indeed they may prove useful if we get stuck, by suggesting further questions to ask. However this should only happen as a last resort because of the way recency influences dialogue (see section 5.5.3).

The Problem Description module uses three main types of rule:

1. Elaboration rules: these rules attempt to elaborate an existing problem description by asking for other 'missing' bits of the description which might be relevant. The sorts of things which might be relevant can be determined from the subject area. For example, in the architectural history domain, we might ask about building types, periods, styles and locations, as these are often useful in narrowing down the architects or buildings the user is interested in.

2. Evaluation rules: these rules attempt to evaluate an existing partial problem description, and the individual problem descriptors forming this description, to guide and ultimately terminate the elaboration of the problem description. The problem description is also subject to the general acceptability tests outlined above.

3. Repair rules: these rules attempt to fix an existing broken problem description. Of course the problem description doesn't exist as a single entity, it's just that we have been given a lot of descriptors and we can't make a good description using them. We make a number of assumptions: we don't try to repair the description until we have run out of elaboration rules; and that in our problem domain the most likely problem is that the user is being too specific (this is also the easiest to do).

In addition we need some domain knowledge, in our case about architectural history, concerning architects, periods, styles etc. For example, if the subject area is ‘architecture’ or ‘architectural history’ and one of the problem descriptors is a person, then assume that the person is an architect and moreover that they are the architect of whatever building or class of building the user is interested in. This is just a default and could result in misunderstandings when the person is e.g. the owner of the building (Hadrian’s villa).[8]

If the problem description is acceptable the problem description goal is deemed to be satisfied and the system switches to the next subgoal. If the initial problem description is not acceptable, the failure gives rise to subgoals to inform the user of the problem and to revise the problem description. For example, the initial problem description ‘*architects* $\wedge$ *buildings* $\wedge$ *styles*’ would fail the acceptability tests as it is too general to be useful. If we exit without a good description, we tell the user what’s wrong with it and why we can’t generate an acceptable retrieval strategy from it. Initially the evaluation is null.

Obtaining the Problem Mode

If this cannot be inferred from the information about the user’s status or the system’s default assumptions result in conflicts, it may be necessary to ask the user what type of documents they require. This is achieved by a simple *ask* speech act, possibly supplemented by a *tell* act if the user responds with an *ask*. The system’s response would depend on the justification for the gaol, e.g. missing information or conflicting assumptions.

Obtaining the Problem State

It is assumed that the problem state cannot be inferred from the system’s knowledge of the user. While problem state information is not essential, it provides an alternative means of deriving the document level if this cannot be inferred from the user status and has not been volunteered by the user as part of the problem description. The problem state goal is achieved by a simple *ask* speech act, possibly supplemented by a *tell* as above. The justification for seeking the information is that it assists the system in determining the document level and hence the appropriate databases to search.

Deriving the Retrieval Strategy

The critical step in deriving the retrieval strategy is mapping from the space of problem descriptors to the space of search terms. In generating the term component of the retrieval strategy, we attempt to map each descriptor in the topic onto a search term or terms, while preserving the logical structure of the topic component.

In a conventional library on-line service the librarian (notionally and often actually) have to look search terms up in the indexing documentation and explicitly consider and compare them as concept representations, individually or conjointly, with the concept(s) forming the problem description. This is not

---

[8] Note that we don’t count saints as people.

practical in our case, so we take an extreme 'innatist' position—the librarian has already read or remembered all the the relevant search terms and thus has has already formed their conceptual representations. All we have to do is find the closest match to the concepts in the problem description. If the result is unsatisfactory (as it typically will be) we explain this to the user and, if the retrieval strategy is too general, we search for a general problem descriptor and suggest that the user specialise it; or, if the retrieval strategy is too specific, we look for a specific problem descriptor and suggest that the user generalise it. In either case we take the first match, since we have no means of selecting 'the' culprit descriptor.

There are three mapping rules which are tried in order. The first rule is the simplest; if this fails we try the second rule; if this fails we try the third. We assume that the third rule will catch all cases (see assumption 3 above).

1. Attempt to find the descriptor among the 'entry words' for a search term. We assume that the descriptor (concept) maps to the lexically equivalent entry word; thus '*cat*' maps to "cat" which in turn gives the search term "*cats*". Such mappings give the 'approved' morphological form of the corresponding English word, and handle simple cases, e.g. where the search term is slightly more general or specific than the corresponding descriptor/entry word, or where the descriptor maps into two or more search terms, e.g. '*cartography*' might give the search terms "*maps* $\wedge$ *drawing*". Note that the resulting search terms need not be in the same term hierarchies: in this example '*map*' may be a specialisation of '*document*', while "*drawing*" is grouped with "*painting*" and "*sculpture*".

2. Attempt to find a synonym for the descriptor which matches one of the entry words for a search term. The synonym classes embody both the librarian's linguistic knowledge and their knowledge of the domain. For example, there may be no direct mapping for '*monkey*', but we know that "ape" is a synonym for "monkey" and that "ape" is an entry word with the corresponding search term "*primates*". Note that the mapping has altered the sense of the descriptor (or at least the perspective or embedding context); "*primates*" is a much more 'technical' term than the original descriptor '*monkey*'. This becomes important when we are looking for, e.g. "the monkey in mediaeval literature".

3. Attempt to find a mapping for the next most general (or specific) descriptor in the descriptor hierarchy using rules (1) and (2) above. If this succeeds, return one or all of the next most specific (or general) search terms for that mapping. For example, there may be no mapping from the descriptor '*football*' (or any of its synonyms). However '*football*' is a kind of '*game*' and '*game*' maps to the search term "*sports*" which has specialisations "*soccer*" and "*rugby*". It is not clear how far up (or down) the descriptor hierarchy we should go in an attempt to get a match. For the sake of simplicity, we shall assume that rule (3) is only every applied once.

The mapping can either be based on the set of all search terms (i.e. all the search term hierarchies) or it can be restricted to those search term hierarchies associated with particular databases. The databases in turn are selected from information about the the subject area derived from the problem description. The former approach is simpler but relies on the fact that we are using a common controlled indexing language.

A retrieval strategy can be rejected for one of two reasons:

1. it fails to cover the problem description, i.e. no search term can be found for one or more of the problem descriptors; or

2. the retrieval strategy is too general or too specific.

When we cannot find a search term for a given problem descriptor, we can use information about the subject area to identify other problem descriptor hierarchies dealing with the same subject. The descriptors in these hierarchies might serve as a basis for reformulating the problem description.

We need a minimum request and want a good request. However, if we simply interpret need as *desire-pos* and want as *desire-neg*, we will never get a good request, given the system's preference for minimum effort plans. Making both a good and a minimum request *desire-pos* will not solve the problem, since deriving a minimum request will always require less effort than a good request and the system plans to minimise effort. We therefore need some motivation for the system to pursue good requests. Presumably, a more developed request will result in more relevant documents being presented to the user. We therefore exploit the notion of 'document relevance' and assume that if the search request is 'good' the chance of any individual document returned by the request being relevant to the user's problem is high, whereas if we only have a minimum request, the chances of any document being relevant to the user's problem is low. This gives two conflicting goals: to derive a search request with least effort for the librarian; and to derive a search request which returns relevant documents to the user. How the conflict is resolved depends on the problem context and how the system trades off the conflicting goals.

6.3 Information Retrieval Rules

As stated in the introduction, for the purpose of testing the architecture outlined in chapter 2, we have considered only the simpler goals to change the beliefs or intentions of another agent (see chapter 8). This is principally because of the computational demands of the underlying belief revision system, particularly when processing extended intention structures, and because the simple rule-based system used to implement the librarian is unable to cope with some of the constructs described above, for example AND nodes in plans, ordering of goals etc. (These problems are discussed in more detail in chapter 9.) We have therefore concentrated on the simpler sub-goal of eliciting the problem description.

The following rules are assigned to the librarian agent (*libr*). Each agent has its own notion of how problem descriptors fit together in a descriptor hierarchy. The *pdtree* predicate relates descriptors to all their sub-descriptors. For example, *classical* architecture can be either *greek* or *roman* and the problem descriptors have the relation (*pdtree* (*greek roman*) *classical*). The *class* predicate describes individual *pdtree* pairs: *greek* is a type of *classical* architecture and so is *roman*. Thus we have (*class greek classical*) and (*class roman classical*).

The problem description comprises a set of problem descriptors (i.e. *pd*s) which are drawn from a single descriptor hierarchy. In adition to communicating the *pd*s constituting their problem description, an agent can also communicate details of the structure of their problem descriptor hierarchy. For example, an agent can suggest "Michelangelo the artist" which not only includes the problem descriptors (*pd michelangelo*) and (*pd artists*) but also information describing the way these terms are related: that Michelangelo is a member of the class of artists. This would be communicated as (*pdh artist michelangelo*).

6.3.1 Problem Description Rules

The following rules are used by the librarian:[9]

(D–1) (*bel libr* (*pd ?DESC*))
$\Rightarrow_{[desire-pos]}$ (*intorbel libr* (*exists !x ?_* (*bel libr* (*class ?DESC !x*))))

When an agent comes to believe a problem description it must make sure that it understands where the problem description lies within the descriptor hierarchy.

(D–2) (*p-bel libr* (*bel user* (*pdh ?DESC1 ?DESC2*) *?COMMITMENT*))
$\Rightarrow_{[premise]}$ (*and* ((*p-bel libr* (*bel user* (*pd ?DESC1*)) *?COMMITMENT*)
(*p-bel libr* (*bel user* (*class ?DESC1 ?DESC2*)
?COMMITMENT))))

The user can offer a sub-descriptor/descriptor section of their problem descriptor hierarchy as part of the problem description. The librarian may conclude that the user's problem description includes both descriptors and that they are related as a descriptor/sub-descriptor pair in the user's problem descriptor hierarchy. For example, (*bel user* (*pdh michelangelo architect*))) would suggest to the librarian that the user believes that *michelangelo* is of the class *architect* and that the user believes *michelangelo* is a good descriptor of their needs.

(D–3) (*p-bel libr* (*pdh ?DESC1 ?DESC2*)) &
(*p-bel libr* (*not* (*class ?DESC1 ?DESC2*)))
$\Rightarrow_{[premise]}$ *false*

---

[9] The user agent (see chapter 8) uses broadly similar rules. However, while the user lacks much of the librarian's knowledge about search requests, descriptors etc. they have more specific knowledge of the architectural history domain, for example that Wren designed St Paul's cathedral. The functions *intorbel*, *some*, *forall* and the predicate *binds* are defined in section 5.5.

If the librarian knows that the descriptor *?DESC1* is not a sub-descriptor in the problem descriptor hierarchy for *?DESC2* then (*pdh ?DESC1 ?DESC2*) cannot be a part of the problem description.

(D–4) (*p-bel libr* (*bel user* (*pd ?DESC*)))
$\Rightarrow_{[def]}$ (*p-bel libr* (*exists !x ?_* (*bel user* (*class ?DESC !x*))))

If the librarian believes that the user believes that *?DESC* is a good problem descriptor for the problem then it also believes, by default, that the user also knows what class in the problem descriptor descriptor hierarchy *?DESC* belongs to.

(D–5) (*p-bel libr* (*pd ?DESC*)) &
(*p-bel libr* (*pdtree ?SUBDESCS ?DESC*))
$\Rightarrow_{[premise]}$ (*some ?D ?SUBDESCS* (*p-bel libr* (*pd ?D*)))

When the librarian believes that a problem descriptor is appropriate then some of the sub-classes of that descriptor in the descriptor hierarchy must also be appropriate.

(D–6) (*p-bel libr* (*pd ?DESC1*)) &
(*p-bel libr* (*pdtree ?SUBDESCS ?DESC2*)) &
(*binds ?DESC1 ?SUBDESCS ?DESC1 ?DESC1*)
$\Rightarrow_{[def]}$ (*p-bel libr* (*pd ?DESC2*))

When the librarian believes that a problem descriptor is appropriate then its super-class (assumed to be unique) of that descriptor in the descriptor hierarchy must also be appropriate.

(D–7) (*p-bel libr* (*bel user* (*pd ?DESC*))) &
(*p-bel libr* (*bel user* (*pdtree ?SUBDESCS ?DESC*)))
$\Rightarrow_{[premise]}$ (*forall ?W ?SUBDESCS*
(*exists !x* ((*pd ?W*) (*not* (*pd ?W*))) (*bel user !x*)))

When the librarian believes that the user believes that a problem descriptor is appropriate then the librarian believes that, for each sub-descriptor in the user's apparent descriptor hierarchy, the user knows whether it is an appropriate pd descriptor or not.

(D–8) (*bel libr* (*dt ?P*))
$\Rightarrow_{[spec]}$ (*bel libr* (*exists !x* ((*dt ?P*) (*not* (*dt ?P*))) (*bel user !x*)))

If the librarian believes that *?P* is an appropriate document type then he can conclude that the user knows whether be wants this type or not.

(D–9) (*p-bel libr* (*status user ra*))
$\Rightarrow_{[def]}$ (*p-bel libr* (*dt advanced-books*))

By default, a research assistant will want literature on advanced material.

6.3.2 Domain Rules

Finally, we have a couple of rules which are specific to the domain of architectural history.

(D–10) (*p-bel libr* (*class michelangelo artist*))
$\Rightarrow_{[def]}$ (*p-bel libr* (*not* (*class michelangelo architect*)))

The librarian believes that if Michelangelo was an artist then he could not have been an architect. For this rule it was necessary to introduce a new endorsement *def-pos* which is equal to *spec* in the endorsement ordering but is interpreted as a strong default. This is discussed in section 8.2.

(D–11) (*p-bel libr* (*pd classical-arch*))) &
(*p-bel libr* (*pd classical-revival-arch*)))
$\Rightarrow_{[premise]}$ *false*

The librarian believes that the descriptors "classical architecture" and "classical revival architecture" refer to distinct periods of architecture and that the choice of both is perhaps too general.

In chapter 8 we illustrate how this simple model can be used to drive the example dialogues.

Chapter 7

Implementing Dialogue

In this chapter we describe the action schemata and production rules used by the agents for dialogue planning. Agents plan to change their own or other agent's attitude states. By considering their own and others beliefs they generate leading intentions and then plan to satisfy these intentions using instances of action schemata. For example an agent has *desire rules* (see chapter 5) which are triggered by belief or intention conflicts with other agents and result in an intention to resolve the conflict. To achieve its goals, an agent plans to communicate its commitments to attitudes and actions, in order to solicit attitude commitments from other agents. These plans consist of intentions to communicate attitudes to other agents and instances of action schemata. For example, if an agent is uncertain about a belief it might well plan to determine the commitment of another agent to this belief. Alternatively, if two agents disagree about a belief one agent might plan to communicate its commitment to a justification for its belief. As described in chapter 5 plans are generated in a STRIPS manner using production rules which generate intentions to act and intentions to satisfy the preconditions of these actions. Actions have associated effort and the agent chooses the least effortful plan for its tasks.

In the description of the action schemata and rules below we adopt the following conventions for the name of variables:

Variable	*Interpretation/Allowable Bindings*
?A1, *?A2*, *?A3*	Agent identifiers (e.g. *user*, *lib*)
?P, *?Q*	Propositions
?ATT	Attitude type (i.e. *int* or *bel*)
?X, *?Y*	Existential arguments (e.g. *!x*)
?W, *?Z*	Working variables (i.e. function variables)
?C	Attitude commitment
?ANT	Rule antecedent instance
?CON	Rule consequent instance
?ACT	Action schema identifier (e.g. (*tell A B* (*bel A P*)))
?PRECS	Action schema preconditions
?EFFS	Action schema effects
?CST	Action schema constraints

In general in this chapter, explanations of or comments on schemata and rules *follow* the presentations of the schemata and rules themselves.

7.1 Action Schemata

We use three action schemata: `tell`, `adopt` and `infer`, which are common to all agents. Unlike, for example Allen (1987), we distinguish between the communication of an attitude and its adoption by the hearer. In our system the agents are autonomous and during communication the hearer can believe that the speaker believes the content of its message without necessarily itself believing in the content of the communication. Hence, a `tell` action can be successful (i.e. achieve its effects) independent of the success of the `adopt` action. There are numerous examples in human dialogue where a speaker wishes to inform the interlocutor of his commitment to an attitude without necessarily intending the interlocutor to adopt the attitude.

In the description of the action schemata we adopt the following conventions for the name of variables: *?A2* and *?A3* denote the actors (e.g. the speaker and the hearer). The variable *?A1* denotes the reasoner, i.e. the agent building the plan and from whose point of view beliefs and intentions are attributed. Note that neither actor (e.g. the speaker or the hearer) in the action schemata described below need be the agent constructing the plan. For example, suppose *agent1* wants *agent2* to tell *agent3* that *agent2* believes that *p* (suppose *agent1* is trying to increase *agent3*'s commitment to *p* by demonstrating that *agent2* believes that *p*:

(*bel agent1* (*int agent1* (*bel agent3* (*bel agent2 p*))))

Agent1 therefore plans for *agent2* to tell *agent3* that *agent2* believes that *p*. This requires two `tell` actions, one for *agent1* to communicate its intention to *agent2* and, assuming *agent2* adopts *agent1*'s intention, one for *agent2* to tell *agent3* that *p*. While *agent1* may 'intend' both these actions in the sense that it wishes them performed, it can only execute one of them directly, and must rely on *agent2* to perform the other. A similar situation arises in the case of two agents, when *agent1* wants *agent2* to tell *agent1* something.

7.1.1 The Tell Schema

The `tell` action is used to communicate an agent's commitment to a belief or intention, and is of the form:

(*action-schema* ((*?ATT ?A2 ?P*))
(*tell ?A2 ?A3* (*?ATT ?A2 ?P*))
((*bel ?A3* (*?ATT ?A2 ?P*)))
((*bel ?A1* (*?A2 can talk to ?A3*)))
effort-neg)

The `tell` action informs the hearer *?A3* that the speaker *?A2* has the belief or intention that *?P*. The message is a structure called "tell" with four slots: the name of the speaker, the name of the intended recipient, and the attitude the speaker wishes to communicate and its commitment to that attitude. The precondition of the `tell` action is that the speaker believes the communicated attitude and the effect is that the hearer comes to believe that the speaker is committed to the attitude *?P*. This effect is guaranteed. Agents in our current system are cooperative and truthful about their attitudes and commitments and communications are always successful. Whether the hearer adopts the communicated attitude is another matter. The constraint is that the reasoner *?A1* actually believes that *?A2* can communicate with *?A3*. The sending of the utterance requires relatively little effort.

For example, consider the message

(*tell user lib* (*bel user* (*status user RA*) *strong*))

This message conveys to the librarian *lib* that the user is strongly committed to the belief that the user is a research assistant. From this the librarian may conclude by default that the user wants advanced books. In which case he could respond by attempting to verify this belief

(*tell lib user* (*int lib* (*exists !x* ((*doc-type advanced-books*) (*not* (*doc-type advanced-books*))) (*bel lib* (*bel user !x*))) *strong*))

This message conveys to the user that the librarian intends to know whether the user wants advanced books or does not want advanced books.

7.1.2 The Adopt Schema

The `adopt` action is the process of an agent adopting, as its own, an attitude held by another agent, and is the agent's (abstract) representation of the process of belief revision.

(*action-schema* ((*bel ?A3* (*?ATT ?A2 ?P*)))
(*adopt ?A2 ?A3* (*?ATT ?A3 ?P*))
((*?ATT ?A3 ?P*))
((*bel ?A1* (*?A2* $\neq$ *?A3*)))
effort-pos)

The adoption action describes the process of agent *?A3* adopting an attitude with propositional content *?P* held by another agent *?A2*. It has as a precondition that the agent adopting the attitude *?A3* believes that the other agent *?A2* has the belief or intention *?P* and the effect that *?A3* believes or intends that *?P*. This requires belief revision and is therefore an effortful process. The constraint (*?A2* $\neq$ *?A3*) prevents the agent trying to plan to adopt an attitude from itself. This is an oversimplification, but it works in the two agent case where the expertise of the agents is implicit in the formulation of the problem. For example, if the librarian wants to know the user's status, we have the intention $\exists x$ (*bel lib* (*status user x*)), which becomes the effect of an `adopt` action with the precondition $\exists x$ (*bel lib* (*bel user* (*status user x*))) which the librarian can plan to achieve. In this case a plausible constraint would be something like (*bel lib* $\exists$ *x* (*bel user* (*status user x*))), i.e. that the librarian believes that the user knows what his status is is, which must be true for the `adopt` action to make sense. However, when the librarian simply wants to know if the user knows his own status (ignoring indirect speech acts), the intention (and hence the effect of the `adopt` action) is $(\mathit{bel\,lib}\,\exists\ x\ (\mathit{bel\,user}\,(\mathit{status\ user}\ x)))$. The precondition would then be $(\mathit{bel\,lib}\ (\mathit{bel\,user}\,\exists\ x\ (\mathit{bel\,user}\,(\mathit{status\ user}\ x))))$ with the only constraint that the agents are distinct. If I want to know if you know something, then presumably I have reasons for wanting this which are embodied in my higher-level intentions, and the only constraint that can sensibly be applied is that I am not talking to myself.

Thus, when the action sequence `tell` → `adopt` is successful the hearer comes to share the communicated attitude with the speaker. However the tell action can be successful when the `adopt` action fails. This means that the hearer recognises the speaker's commitment to the communicated attitude but refuses to adopt the attitude himself.

7.1.3 The Infer Schema

The `infer` action is the action of an agent drawing an inference from a rule and is of the form:[1]

```
(action-schema ((bel ?A2 (rule-inst ?R ?ANT ?CON))
                (forall ?W ?ANT (bel ?A2 ?W)))
               (infer ?_ ?A2 (rule-inst ?rule ?ANT ?CON))
               ((bel ?A2 ?CON))
               ((bel ?A1 (rule-inst ?R ?ANT ?CON)))
               effort-pos)
```

The schema has as preconditions that the agent *?A2* must believe the rule *?R* to be applicable in this case (i.e. there must be a rule instance) and the agent must believe all the antecedents *?ANT* of the rule. The effect is that the agent *?A2* believes the consequent. The constraint on the infer action is that the rule instance is already believed by the planning agent *?A1*. Since an agent will inevitably consider communicating a rule instance in order to change its interlocutor's mind about some other attitude, the action can lead to belief revision and is therefore effortful.

The infer action schema is used when planning explanations and we restrict our agents' explanations to rule instances (i.e. rules already fired by the agent). Since our agents do not perform abduction we do not allow them to plan to discover new rule instances. They must have already created the instance through some other means when they consider including an infer steps in their plans.

7.2 Dialogue Rules

There are five types of rules. In addition to the desire and planning rules described in chapter 5, there are three additional types required for dialogue

---

[1] The dummy variable ?_ in the action descriptor is void but is present to conform with the three argument structure of the other action descriptors.

planning: *ascription rules*, *adoption rules* and *prediction rules*. These rules are possessed by all agents and apply to planning, prediction, belief and intention adoption and belief/intention modelling. Below we briefly discuss the role of each type in the production of dialogue plans. A full list of rules required for the example in section 7.4 and for the test dialogues in chapter 8 are given in this chapter. At the end of the chapter we comment on the set of rules as a whole.

Rules refer to an agent's **agent** point of view with known interlocutor **fellow-agent**. We assume the following beliefs are asserted

(*p-bel *agent** (**agent** $\neq$ **fellow-agent**) *premise*)
(*p-bel *agent** (**fellow-agent** $\neq$ **agent**) *premise*)
(*p-bel *agent** (**agent* can talk to *fellow-agent**) *premise*)
(*p-bel *agent** (**fellow-agent* can talk to *agent**) *premise*)

where **agent** and **fellow-agent** are set to the identifiers of the reasoning agent and its interlocutor. The variable *?_* matches everything but is never bound. It is used to identify variables in rule antecedents whose binding are immaterial.

7.2.1 Ascription Rules

Agents model other agents' believed beliefs and intended intentions using the following rules.

(R–1) (*p-bel ?A1* (*action* (*tell ?A2 ?A1* (*?ATT ?A2 ?P ?C*))))
$\Rightarrow_{[definite]}$ (*and* ((*p-bel ?A1* (*?ATT ?A2 ?P ?C*))
(*p-bel ?A1* (*?ATT ?A2 ?P*))))

When the hearer (i.e. *?A1*) receives a communication from the speaker (ie *?A2*) the hearer assumes that the speaker believes that the preconditions to the dialogue action hold. As stated above, agents assume that all agents share the same dialogue actions.

(R–2) (*p-bel ?A1* (*bel ?A2 ?P*)) &
(*p-bel ?A1* (*bel ?A2* (*not ?P*)))
$\Rightarrow_{[premise]}$ *false*

Agents assume that all agents are consistent reasoners.[2]

(R–3) (*p-bel ?A1* (*bel ?A2* (*rule-inst ?_ ?ANT* (*?ATT1 reasoner ?CON*))) &
(*forall* (*?ATT reasoner ?W*) *?ANT* (*p-bel ?A1* (*?ATT ?A2 ?W*)))
$\Rightarrow_{[def]}$ (*p-bel ?A1* (*?ATT1 ?A2 ?CON*))

If *?A1* believes *?A2* believes a rule instance and also the antecedents of the rule instance then *?A1* will conclude, by default, that *?A2* also believes the consequent.

---

[2] This rule works when one of the antecedents is the result of a default inference about the modelled agent's beliefs but does not work when this agent communicates his change of mind (see section 9.4).

7.2.2 Adoption Rules

Whether an agent adopts the communicated attitude depends on what else it believes. An agent uses the following rules to decide whether to adopt a communicated belief or intention. These rules infer from utterances, extracting the content of the message and transforming it into reasons for the hearer to adopt the communicated intention or belief. The endorsements *2c-pos* and *2c-neg* are assigned to communicated beliefs and give reason for the hearer to believe the communicated proposition. (The corresponding endorsements for communicated intentions are *desire-pos* and *desire-neg*.) We argue that such endorsements arise out of other agents' commitment to the communicated attitude.

(R–4) (*p-bel ?A1* (*bel ?A2 ?P strong*))
$\Rightarrow_{[2c-pos]}$ (*p-bel ?A1 ?P*)

If an agent is strongly committed to a belief, this gives the hearer reason to believe the belief.

(R–5) (*p-bel ?A1* (*bel ?A2 ?P weak*))
$\Rightarrow_{[2c-neg]}$ (*p-bel ?A1 ?P*)

If, on the other hand, the speaker's commitment is to the communicated attitude is weak then the hearer has less justification for believing/intending the attitude.

(R–6) (*bel ?A1* (*int ?A2 ?P strong*))
$\Rightarrow_{[desire-pos]}$ (*intorbel ?A1 ?P*)

and

(R–7) (*bel ?A1* (*int ?A2 ?P weak*))
$\Rightarrow_{[desire-neg]}$ (*intorbel ?A1 ?P*)

The fact that the speaker is committed to an intention gives the hearer reason to intend that intention (i.e. a leading intention). Note that, unlike rules R–4 and R–5, R–6 and R–7 are defined over the agent's pervasive beliefs and can therefore only fire in the intention sets.

7.2.3 Prediction Rules

Agents predict their future belief states from their current beliefs and intentions. If an agent has an intention to act then it has reason to believe that the effects of the action will be true in the future. Conversely, agents assume that nothing will change unless they or others act—what is true now will continue to be true in the future.[3] For example, an intention to perform a `tell` act is reason to

---

[3] This is in our simple world where nature is frozen, the agents politely await their turn to act, and the only actions are communicative ones. We discuss the problems presented by these heavy and unrealistic constraints, even for quite restricted modelling applications, in Chapter 9.

believe the hearer will come to believe that the speaker is actually committed to the communicated attitude. This in turn is reason to believe the hearer will adopt the communicated attitude, all other things being equal. The success of a `tell` → `adopt` pair will depend on the speaker's commitment to its belief when the utterance is made, which in turn depends on the speaker's current commitment to its attitudes. (It is assumed that justifications are conserved over time and an agent's commitment to an attitude is guaranteed not to change as long as it does not infer or receive communications.)

When an agent communicates an attitude and its commitment it does so with the knowledge that the hearer will convert this communicated commitment into an endorsement for adopting the attitude itself.

Beliefs	*strong*	→	*2c-pos*
	weak	→	*2c-neg*
Intentions	*strong*	→	*desire-pos*
	weak	→	*desire-neg*

For example, if the speaker S believes that the hearer H is weakly committed to a belief p then S knows that an additional *2c-pos* endorsement on (*not* p) would be sufficient for H to revise its beliefs and come to believe (*not* p). Alternatively, if S believes that H is strongly committed to p then the communication of the strongly held (*not* p) by S would have unpredictable effects. Alternatively, if S were weakly committed to (*not* p) and H strongly committed to p then communicating (*not* p) would not cause H to revise its beliefs.

The speaker, therefore, can reason about the outcome of its utterances. It can predict the gain of endorsement on the hearer's beliefs and ultimately, given knowledge of the hearer's current cognitive state, predict whether or not the attitude will be adopted. We have the following guidelines

The 'rule of continuity': If an agent is deemed to be strongly committed to an attitude then this is reason to believe that it will be just as strongly committed in the future. Similarly for weak commitment.

The 'rule of action': If a speaker intends to communicate a belief p, to which it is strongly committed then this good is reason to believe that the hearer will believe p in the future. Alternatively, if the speaker intends to communicate a weak belief then this gives less reason to believe that the hearer will adopt p.

The rule of continuity is is based on the intuition that an agent's current reasons for an belief/intention will contribute towards its believing/intending this attitude in a future cognitive state. If an agent believes that p with strong commitment then it can conclude that its future state will inherit relatively incorrigible justifications for p relative to (*not* p). The action property is that an action (such as the receiving of utterances or the performance of inference) can introduce new justifications in the future state.

To implement prediction in dialogue we need eight new endorsements

auto-predict-bel-pos
auto-predict-bel-neg
auto-predict-int-pos
auto-predict-int-neg
alter-predict-bel-pos
alter-predict-bel-neg
alter-predict-int-pos
alter-predict-int-neg

with ordering

$$\textit{auto-predict-bel-pos} >_e \textit{auto-predict-bel-neg} >_e \textit{auto-predict-int-pos} >_e \textit{auto-predict-int-neg}$$

$$\textit{alter-predict-bel-pos} >_e \textit{alter-predict-bel-neg} >_e \textit{alter-predict-int-pos} >_e \textit{alter-predict-int-neg}$$

These give a qualitative measure for an agent's reasons for predicting what will be believed or intended in a future state. They are divided into two groups: *auto-predict* and *alter-predict*. The *auto-predict* endorsements capture the potency of the predicted future justifications and endorsements for the reasoner's attitudes. In the case of inertia, where an agent currently believes p and is strongly committed to this belief, the relative merit of his reasons for believing p over disbelieving p will contribute positively in a future state. Thus we have the following rule.

(*bel A p strong*)
$\Rightarrow_{[\textit{auto-predict-bel-pos}]}$ (*f-p-bel A p*)

A current weak commitment to a belief means that a future state will inherit relatively corrigible reasons for believing p.

(*bel A p weak*)
$\Rightarrow_{[\textit{auto-predict-bel-neg}]}$ (*f-p-bel A p*)

An agent can also apply inertia to predict his future intentions.

(*int A p strong*)
$\Rightarrow_{[\textit{auto-predict-int-pos}]}$ (*f-p-int A p*)

The *alter-predict* endorsements capture the modelled agent's justifications and endorsements in its future state. An agent A can predict the future state of another agent B using the inertia property of B's attitudes. If A believes that B is strongly committed to p then A can conclude that B will have strong reason to continue believing p in a future state.

(*bel A* (*bel B p strong*))
$\Rightarrow_{[\textit{alter-predict-bel-pos}]}$ (*f-p-bel A* (*bel B p*))

Similarly, for a weak commitment by B to p. When an agent has non-certain beliefs about other agents beliefs we can combine *auto* and *alter* endorsements to allow agents to reason about other agents' future states. For example, if agent A has a weak commitment to the belief that agent B is strongly committed to p then this gives a relatively corrigible justification to A that A will believe that B will believe p in the future.

$$\textit{(bel A (bel B p strong) weak)}$$
$$\Rightarrow_{[\textit{auto-predict-bel-neg,alter-predict-bel-pos}]} \textit{(f-p-bel A (bel B p))}$$

If A has weak commitment to the belief that agent B is strongly committed to the intention that p then this gives a relatively corrigible justification to A that A will believe that B will intend p in the future.

$$\textit{(bel A (int B p weak))}$$
$$\Rightarrow_{[\textit{auto-predict-bel-pos,alter-predict-bel-neg}]} \textit{(f-p-bel A (int B p))}$$

Intentions to act can also influence the expected attitude contents of future states. Consider, for example, agent A intends to tell agent B that A is weakly committed to belief p. While this intention holds A will expect to eventually perform the action and predicts that B will gain a weak reason (i.e. an extra *2c-neg* endorsement) for p.

$$\textit{(p-int A (tell A B (bel A p weak)))}$$
$$\Rightarrow_{[\textit{auto-predict-bel-pos,alter-predict-bel-neg}]} \textit{(f-p-bel A (bel B p))}$$

The endorsement pair *[auto-predict-bel-pos,alter-predict-bel-neg]* here means that A will have strong reason to believe that B will gain a weak reason to believe p. The endorsement is *auto-predict-bel-pos* since a tell of a weak committed belief is *guaranteed* to add a *2c-neg* to B's endorsements for p.

The prediction endorsement ordering above can be extended to include *auto/alter* endorsement pairs which endorse predictions about fellow agents' attitudes. If we write $au_{belneg}al_{belpos}$ to mean *[auto-predict-bel-neg,alter-predict-bel-pos]* and in general $au_{wx}al_{yz}$ to mean *[auto-predict-w-x,alter-predict-y-z]* we have

1. A belief with strong justification in a modelled agent's future state will be preferred to a contradictory, but weakly justified belief.

$$au_{belpos}al_{belpos} >_e au_{belpos}al_{belneg}, au_{belneg}al_{belneg}$$

2. An intention with strong justification in a modelled agent's future state will be preferred to a contradictory, but weakly justified intention.

$$au_{belpos}al_{intpos} >_e au_{belpos}al_{intneg}, au_{belneg}al_{intpos}, au_{belneg}al_{intneg}$$

3. If an agent has strong reason to believe that agent B will hold a belief/intention in a future state then A predicts that B will not intend this attitude in the future.

$$au_{belpos}al_{belpos}, au_{belpos}al_{belneg} >_e au_{belpos}al_{intpos}, au_{belpos}al_{intneg}$$

When planning to change another agent's cognitive state, an agent must weigh the predicted effects of its actions against the predicted beliefs of the other agent. For example, if agent A strongly believes p and believes that agent B has a weak belief in (*not* p), then A will predict that performing the act

$$\textit{(tell A B (bel A p strong))}$$

will result in B coming to believe p, all other things being equal. Note that if B currently had a strong belief in (*not* p) and hence was predicted to have a strong belief in (*not* p) in the future, then A's plan will be predicted to be unsuccessful.[4] If an agent predicts that a plan will not be successful, the agent will abandon the plan.

The following steps should be included after the last step (i.e. step 3) in the intention set preference algorithm described in section 5.3.1. The algorithm compares the endorsements e and e' of two sets s and s' respectively.

5. if e contains more *predict-bel-pos* endorsements (i.e. *auto-predict-bel-pos* and *alter-predict-bel-pos*) than e' then $s \geq_e s'$, else if e' contains more *predict-bel-pos* endorsements than e then $e' \geq_e e$ otherwise
6. if e or e' contains more *auto-predict-bel-neg*endorsements then $s =_e s'$ otherwise
7. if e contains more *predict-int-pos* endorsements (i.e. *auto-predict-int-pos* and *alter-predict-int-pos*) than e' then $s \geq_e s'$, else if e' contains more *predict-int-pos* endorsements than e then $e' \geq_e e$

In section 5.3.3 we described the way agents determine their commitment to their beliefs and intentions. Agents can also determine their commitment to what they believe they will believe or intend in a future state. We introduce two endorsements to calculate commitment: *predict-bel-comm* and *predict-int-comm*.

auto-predict-bel-pos $>_e$ *predict-bel-comm* $>_e$ *auto-predict-bel-neg*

auto-predict-int-pos $>_e$ *predict-int-comm* $>_e$ *auto-predict-int-neg*

A pervasive future belief (e.g. (*f-bel A X*)) is strongly committed to if its negation (*not* (*f-bel A X*))) can be given an extra *predict-bel-comm* endorsement and (*f-bel A X*) remains pervasive.

Although agents do not represent commitment to future beliefs explicitly, they use this information when deciding whether to act. Agents only act if they are strongly committed to the belief that their actions will succeed otherwise they replan.[5] The prediction rules are listed below.

(R–8) (*f-p-bel ?A1 ?P*) (*f-p-bel ?A1* (*not ?P*))
$\Rightarrow_{[premise]}$ *false*

Agents have consistent beliefs in future states.

(R–9) (*f-p-bel ?A1* (*bel ?A2* (*rule-inst ?_ ?ANT* (*?ATT reasoner ?CON*)))) &
(*forall* (*?ATT reasoner ?W*) *?ANT* (*f-p-bel ?A1* (*?ATT ?A2 ?W*)))
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*?ATT ?A2 ?CON*))

---

[4] Of course, were the plan to be executed, B might well come to believe p—A may be wrong in ascribing a strong belief in (*not* p) to B, or B might come to believe p for other reasons.

[5] Acting under prediction in dialogue is described further in section 7.3.

If an agent predicts that a fellow agent will believe a rule instance and also the antecedent to the rule instance, then it predicts that the fellow agent will believe the consequent of the rule instance.

(R–10) (*bel ?A1* (*?ATT ?A2 ?P ?C2*) *?C1*)
$\Rightarrow_{[auto-predict-bel-?E1,alter-predict-?ATT-?E2]}$
(*f-p-bel ?A1* (*?ATT ?A2 ?P*))

where $?Ei = pos$ if $?Ci = strong$ and $?Ei = neg$ if $?Ci = weak$. If an agent is committed to an attitude then, with no further communication, it will be equally committed in the future (this ignores further inferences by the agent).

(R–11) (*bel ?A1* (*action* (*tell ?A2 ?A3* (*?ATT ?A2 ?P ?C*))))
$\Rightarrow_{[definite]}$ (*and* ((*f-p-bel ?A1* (*bel ?A3* (*?ATT ?A2 ?P ?C*)))
(*f-p-bel ?A1* (*bel ?A3* (*?ATT ?A2 ?P*)))))

If an action has been performed then this is reason to believe that the effects of the action will be achieved.

(R–12) (*p-int ?A1* (*action* (*tell ?A1 ?A2* (*?ATT ?A1 ?P*))))
(*?ATT ?A1 ?P ?C*)
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*bel ?A2* (*?ATT ?A1 ?P ?C*)))

If agent *?A1* intends to tell a agent *?A2* that *?P* and the preconditions of this action are satisfied (i.e. *?A1* believes *?P*), then this gives reason to believe that the effects of the action (i.e. that *?A2* will believe that *?A1* believes *?P*) will be achieved.

(R–13) (*f-p-bel ?A1* (*bel ?A2* (*?ATT ?A3 ?P ?C*)))
$\mapsto_{[\text{auto-predict-bel-pos,alter-predict-?ATT-?E}]}$ (*f-p-bel ?A1* (*?ATT ?A3 ?P*))

where $?E = pos$ if $?C = strong$ and $?E = neg$ if $?C = weak$. If *?A1* predicts that *?A2* will believe that *?A3* is committed to an attitude then *?A1* has reason to believe that *?A2* agent will adopt the attitude. A strong commitment is converted to *pos* and a weak to *neg* in the rule instance. Note that the endorsement on the rule instance, which contains either *alter-predict-bel-pos* or *alter-predict-bel-neg* in the belief case, reflects the behaviour of the adoption rules (R–7 and R–8). If the consequent is disbelieved then the rule-instance is also disbelieved and the antecedent can be retained (since this is a $\mapsto$ rule). It is not true that if an agent prefers not to adopt a communicated attitude then it should believe that the communicator no longer has the attitude.

(R–14) (*f-p-bel ?A1* (*action ?ACT*)) &
(*action-schema ?_ ?ACT* (*?EFF*) *?_ ?_*)
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1 ?EFF*)

and

(R–15) (*f-p-bel ?A1* (*exists ?X ?Y* (*action ?ACT*))) &
(*action-schema ?_ ?ACT* (*?EFF*) *?_ ?_*)
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*exists ?X ?Y ?EFF*))

If an agent predicts that an action will be performed then he has reason to believe that the effects of that action will hold in the future state.

(R–16) (*f-p-bel ?A1* (*int ?A2 ?EFF*)) &
(*action-schema ?PRECS ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*) &
(*forall ?Z ?PRECS* (*or* ((*bel ?A1 ?Z*) (*f-p-bel ?A1 ?Z*))))
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*action ?ACT*))

and

(R–17) (*f-p-bel ?A1* (*int ?A2* (*exists ?X ?Y ?EFF*))) &
(*action-schema ?PRECS ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*) &
(*forall ?Z ?PRECS* (*or* ((*bel ?A1* (*exists ?X ?Y ?Z*))
(*f-p-bel ?A1* (*exists ?X ?Y ?Z*)))))
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*exists ?X ?Y* (*action ?ACT*)))

If an agent believes a fellow agent intends a state and there is an action schema with the state as an effect and the preconditions of this action schema are predicted to hold then the agent predicts that the fellow agent will perform an instance of this action schema.

(R–18) (*f-p-bel ?A1 ?EFF*) &
(*action-schema ?PRECS ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*) &
(*forall ?Z ?PRECS* (*or* ((*bel ?A1 ?Z*) (*f-p-bel ?A1 ?Z*))))
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*action ?ACT*))

and

(R–19) (*f-p-bel ?A1* (*exists ?X ?Y ?EFF*)) &
(*action-schema ?PRECS ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*) &
(*forall ?Z ?PRECS* (*or* ((*bel ?A1* (*exists ?X ?Y ?Z*))
(*f-p-bel ?A1* (*exists ?X ?Y ?Z*)))))
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*exists ?X ?Y* (*action ?ACT*)))

If an agent predicts that a fellow agent will believe an attitude and also that this belief might arise from an action performed by the fellow agent then he predicts that this action will be successful.

(R–20) (*f-p-bel ?A1* (*exists ?X ?Y* (*bel ?A1* (*bel ?A2 ?P*))))
$\Rightarrow_{[\text{auto-predict-bel-pos}]}$ (*f-p-bel ?A1* (*exists ?X ?Y* (*bel ?A2 ?P*)))

If an agent *?A1* predicts that it will come to know another agent *?A2*'s belief then *?A1* has good reason to believe that it will adopt this belief itself. There is a slight hack here: agent *?A1* assumes that it will come to believe a belief strongly held by *?A2*.

(R–21) (*f-p-bel ?A1* (*?ATT ?A2 ?P ?C*))
$\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*?ATT ?A2 ?P*))

This is book keeping rule. If *?A1* comes to believe *?A2*'s commitment to an attitude then *?A1* infers *?A2* has that attitude.

(R–22) (*f-p-bel ?A1* (*bel ?A2 ?P*)) &
(*f-p-bel ?A1* (*bel ?A2* (*not ?P*)))
$\Rightarrow_{[premise]}$ *false*

Agents are consistent reasoners.

7.2.4 Desire Rules

These rules generate intentions (i.e. leading intentions) from belief attitudes. Note that the relevance motivation in the inference algorithm dictates that only relevant intentions are undertaken.

(R–23) (*p-int ?A1* (*exists ?X ?Y* (*bel ?A1* (*?ATT ?A2 ?P*)))) &
(*?ATT ?A1 ?P*) &
(*binds ?X ?Y ?X ?P*)
$\Rightarrow_{[desire-pos]}$ (*intorbel* (*bel ?A1* (*?ATT ?A2 ?P*)))

If *?A1* wants *?A2* to believe some binding to an attitude and *?A1* actually has such a binding, then *?A1* would want to communicate the attitude which instantiates the binding. This may arise when *?A2* requests information.

(R–24) (*bel ?A1 ?P*)
(*bel ?A1* (*int ?A2* (*exists ?X* (*?P* (*not ?P*)) (*bel ?A2* (*bel ?A1 ?X*)))))
$\Rightarrow_{[desire\text{-}pos]}$ (*p-int ?A1* (*bel ?A2* (*bel ?A1 ?P*)))

If an agent *?A1* has some belief that *?P* and believes that another agent *?A2* wants to know whether *?A1* believes *?P* or (*not ?P*) then *?A1* infers the intention for *?A2* to come to know that *?A1* beleives that *?P*.

(R–25) (*bel ?A1 ?P weak*) &
(*bel ?A1* (*exists ?X* (*?P* (*not ?P*)) (*bel ?A2 ?X*)))
$\Rightarrow_{[desire\text{-}pos]}$ (*intorbel ?A1* (*exists ?X* (*?P* (*not ?P*))
(*bel ?A1* (*bel ?A2 ?X*))))

If *?A1* is only weakly certain about a belief *?P* and he knows *?A2* has some belief about *?P* then he intends to determine *?A2*'s belief.

(R–26) (*bel ?A1 ?P weak*) (*bel ?A1* (*not* (*bel ?A2 ?P*)))
$\Rightarrow_{[desire-pos]}$ (*intorbel ?A1* (*bel ?A2* (*bel ?A1 ?P*)))

If an agent is only weakly certain, but yet again believes/intends then he communicates his attitude. If there is a conflict then the interlocutor may enter negotiation.

(R–27) (*bel ?A1 ?P unc*)
$\Rightarrow_{[desire-pos]}$ (*intorbel ?A1* (*exists ?X* (*?P* (*not ?P*))
(*bel ?A1* (*bel ?A2 ?X*))))

The agent desires to resolve uncertainty or increase commitment of weakly held beliefs. It does this by finding out what other agents believe.

(R–28) (*?ATT ?A1 ?P strong*) &
(*bel ?A1* (*?ATT ?A2* (*not ?P*)))
$\Rightarrow_{[\textit{desire-pos}]}$ (*p-int ?A1* (*?ATT ?A2 ?P*))

and

(R–29) (*?ATT ?A1* (*not ?P*) *strong*) &
(*bel ?A1* (*?ATT ?A2 ?P*))
$\Rightarrow_{[\textit{desire-pos}]}$ (*p-int ?A1* (*?ATT ?A2* (*not ?P*)))

If an agent recognises a disagreement with a fellow agent then it attempts to resolve the conflict.

7.2.5 Planning Rules

These rules generate sub-intentions from intentions.

(R–30) (*p-int ?A1 ?EFF*) &
(*action-schema ?_ ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*)
$\rhd_{[\textit{premise}]}$ (*intorbel ?A1* (*action ?ACT*))

and

(R–31) (*p-int ?A1* (*exists ?X ?Y ?EFF*)) &
(*action-schema ?_ ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*)
$\rhd_{[\textit{premise}]}$ (*intorbel ?A1* (*exists ?X ?Y* (*action ?ACT*)))

If an agent has an intention and there is an action which satisfies this intention then the agent also intends the action. [6]

(R–32) (*p-int ?A1* (*action ?ACT*)) &
(*action-schema ?PRECS ?ACT ?_ ?_ ?_*)
$\Rightarrow_{[premise]}$ (*forall ?W ?PRECS* (*intorbel ?A1 ?W*))

and

---

[6] Note that each of our action schemata have only one effect.

(R–33) (*p-int ?A1* (*exists ?X ?Y* (*action ?ACT*))) &
(*action-schema ?PRECS ?ACT ?_ ?_ ?_*)
$\Rightarrow_{[premise]}$ (*forall ?W ?PRECS* (*intorbel ?A1* (*exists ?X ?Y ?W*)))

This is the STRIPS planning operator; agents intend to satisfy the preconditions of all intended actions. Note that this is a premise rule; if the preconditions are deemed unsatisfiable then the intention to act is dropped.

(R–34) (*p-int ?A1* (*action* (*tell ?A2 ?A1 ?P*))) &
(*action-schema ?_* (*tell ?A2 ?A1 ?P*) *?EFFS ?_ ?_*)
$\Rightarrow_{[premise]}$ (*some ?W ?EFFS* (*intorbel ?A1* (*int ?A2 ?W*)))

and

(R–35) (*p-int ?A1* (*exists ?X ?Y* (*action* (*tell ?A2 ?A1 ?P*)))) &
(*action-schema ?_* (*tell ?A2 ?A1 ?P*) *?EFFS ?_ ?_*)
$\Rightarrow_{[premise]}$ (*some ?W ?EFFS* (*intorbel ?A1* (*int ?A2* (*exists ?X ?Y ?W*))))

If the intention is for a fellow agent to act then it is necessary to instill the intention to act in the fellow agent.

7.3 Agent Action Cycle Revisited

The agent action cycle was described in section 5.7. Here we extend this in the dialogue domain to include prediction and action. Agents plan by inferring from their intentions and predict by inferring from their intentions and beliefs. They act only when they are confident that their plan will succeed. If an intended action is strongly predicted to succeed and its preconditions are satisfied then that action is executed. The iterative action cycle is extended to include a further step.

1. If there are any incoming messages on the message board then push these as a single group onto the inference stack, add premise *p-bel* attitudes that these actions have taken place to the database, and revise the belief sets accordingly.

2. Search down the inference stack by applying the inference algorithm to each group of attitudes until an inference can be drawn. If an inference is drawn add new information (i.e. rule instance, consequent labels) to the database, and then revise attitude sets accordingly by selecting most preferred sets from the database. [7]

---

[7] The recency part of the inference choice mechanism (see section 5.5.3) causes the agent to fail to draw important predictions (i.e. continuity predictions from ascribed attitudes). This is partly solved by ensuring that inferences with prediction confidence values are preferred over those with desire confidence values.

3. If an intended dialogue action is *relevant* (see section 5.5.1) to some leading intention and this leading intention is *strongly predicted* (see section 7.2.3) to be successful then execute this action and assert the premise that the action has been executed: add this premise to the database and to the inference stack, and revise attitudes accordingly.

7.4 An Example

Consider the scenario in which an agent A has a strong desire to resolve conflicting beliefs between two agents, itself and another agent B. Agent A believes that it is dark outside and he also believes that agent B does not believe that it is dark outside. The plan for A to tell B that it is dark outside is shown in figure 7.1.[8] A believes that, in order for B to adopt the belief *dark-outside B* must be made aware of A's commitment to this belief. One way of achieving this is by A telling B. Notice that this plan has two actions, one involving effort *effort-neg* and the other *effort-pos*: The tell action (it takes time to generate an utterance) and the action of B adopting *dark-outside.* The latter requires B to expend a lot of effort on A's behalf; namely the time spent inferring from the utterance and the effort involved in revising beliefs.[9] The rule instances in 7.1 correspond to the following rules.

Instance	*Rule*
rule-inst-1	R–11
rule-inst-2	R–36
rule-inst-3	R–28
rule-inst-4	R–30
rule-inst-5	R–28
rule-inst-6	R–12
rule-inst-7	R–13
rule-inst-8	R–18
rule-inst-9	R–18

The rule instances between intentions are premises. This is to facilitate back propagation of revision from intentions lower in the tree structure to those above. Thus if (*p-int A* (*action* (*tell A B* (*bel A dark-outside*))) *effort-neg*) had been abandoned then the leading intention (*p-int A* (*bel B dark-outside*)) would have been abandoned also, since there is only one plan.

The future beliefs

(*f-p-bel A* (*bel B* (*bel A dark-outside strong*)))
(*f-p-bel A* (*bel B dark-outside*))
(*f-p-bel A* (*bel B* (*not dark-outside*)))

are used in the predictive process. The prediction endorsements (*alter-predict-bel-pos* and *alter-predict-bel-neg*) capture the force of the endorsement assignments

---

[8] This structure represents the justifications that are held in the database.

[9] In our, system, the planner considers the effort that would have to be expended by all actors.

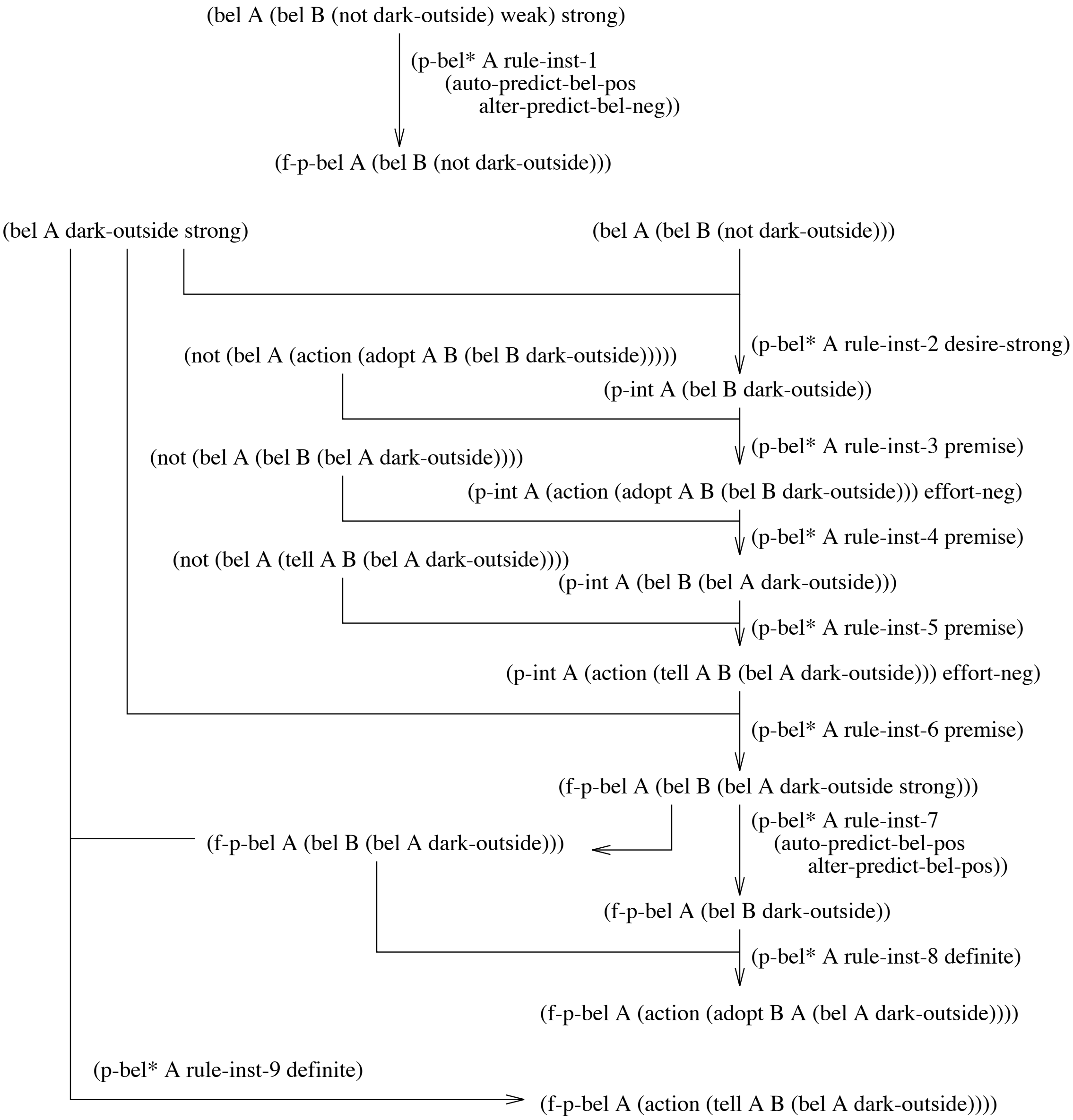

(bel A (bel B (not dark-outside) weak) strong)
(p-bel* A rule-inst-1 (auto-predict-bel-pos alter-predict-bel-neg))
(f-p-bel A (bel B (not dark-outside)))
(bel A dark-outside strong)
(bel A (bel B (not dark-outside)))
(p-bel* A rule-inst-2 desire-strong)
(not (bel A (action (adopt A B (bel B dark-outside)))))
(p-int A (bel B dark-outside))
(p-bel* A rule-inst-3 premise)
(not (bel A (bel B (bel A dark-outside))))
(p-int A (action (adopt A B (bel B dark-outside))) effort-neg)
(p-bel* A rule-inst-4 premise)
(not (bel A (tell A B (bel A dark-outside))))
(p-int A (bel B (bel A dark-outside)))
(p-bel* A rule-inst-5 premise)
(p-int A (action (tell A B (bel A dark-outside))) effort-neg)
(p-bel* A rule-inst-6 premise)
(f-p-bel A (bel B (bel A dark-outside strong)))
(f-p-bel A (bel B (bel A dark-outside)))
(p-bel* A rule-inst-7 (auto-predict-bel-pos alter-predict-bel-pos))
(f-p-bel A (bel B dark-outside))
(p-bel* A rule-inst-8 definite)
(f-p-bel A (action (adopt B A (bel A dark-outside))))
(p-bel* A rule-inst-9 definite)
(f-p-bel A (action (tell A B (bel A dark-outside))))

Figure 7.1: A Simple Plan

```
(bel A dark-outside strong)
(bel A (bel B (not dark-outside) weak))
(not (bel A (action (adopt A B (bel B dark-outside)))))
(not (bel A (bel B (bel A dark-outside))))
(not (bel A (action (tell A B (bel A dark-outside)))))
(p-int A (bel B dark-outside))
(p-int A (action (adopt A B (bel B dark-outside))) effort-neg)
(p-int A (bel B (belA dark-outside)))
(p-int A (action (tell A B (bel A dark-outside))) effort-neg)
(f-p-bel A (bel B (bel A dark-outside strong)))
(f-p-bel A (bel B dark-outside))
(p-bel* A rule-inst-2 desire-pos)
(p-bel* A rule-inst-7 (auto-predict-bel-pos
                       alter-predict-bel-pos))
```

```
(bel A dark-outside strong)
(bel A (bel B (not dark-outside) weak))
(not (bel A (adopt A B (bel B dark-outside))))
(not (bel A (bel B (bel A dark-outside))))
(not (bel A (tell A B (bel A dark-outside))))
(p-int A (bel B dark-outside))
(p-int A (action (adopt A B (bel B dark-outside))) effort-neg)
(p-int A (bel B (belA dark-outside)))
(p-int A (action (tell A B (bel A dark-outside))) effort-neg)
(f-p-bel A (bel B (bel A dark-outside strong)))
(f-p-bel A (bel B (not dark-outside)))
(p-bel* A rule-inst-2 desire-pos)
(p-bel* A rule-inst-1 (auto-predict-bel-pos
                       alter-predict-bel-neg))
```

Figure 7.2: Two intention sets for the Simple Plan

that will exist in B's future state. (*f-p-bel A* (*bel B dark-outside strong*)), the belief that B will adopt the belief that it is dark outside, is justified by the intention that A will tell B his commitment to the belief that it is dark outside. The *alter-predict-bel-pos* endorsement on this rule instance catches the fact that B will convert A's strong commitment to a *2c-pos* endorsement for adopting the belief.

A's belief (*bel A* (*bel B* (*not dark-outside*) *weak*)), that B has a weak belief for (*not dark-outside*), means that B's belief can be overridden by introducing an extra *2c-pos* endorsement to (*p-bel B dark-outside*). The net effect of B's existing endorsements for and against believing (*not dark-outside*) is captured by the *auto-predict-bel-neg* endorsed justification for (*f-p-bel A* (*bel B* (*not dark-outside*))).

Since (*f-p-bel A* (*bel B dark-outside*)) and (*f-p-bel A* (*bel B* (*not dark-outside*))) are inconsistent, there is one intention set with (*f-p-bel A* (*bel B dark-outside*)) and another set with (*f-p-bel A* (*bel B* (*not dark-outside*))). The relative endorsements of these sets determine which is preferred; whether or not to believe that B will come to believe that it is dark outside (see figure 7.2). Referring back to the preference heuristics in section 5.3.1 we see that the set with the plan is preferred.[10] So the agent predicts that the plan will be successful.

If A had had a weak commitment to the belief that it is dark outside then (*f-p-bel A* (*bel B dark-outside*)) would be justified by a *auto-predict-bel-neg* endorsed rule instance and, unlike the previous example, the prediction endorsements would not distinguish a single preferred set. That is, the outcome would be uncertain. Alternatively, if, just prior to A uttering its tell message, B were to volunteer the information that he believes that A believes that it is dark outside (for example, if B anticipates A's plan), then A would accept the belief (*bel A* (*bel B* (*bel A dark-outside*))) and the intention to tell would be dropped.

---

[10] The endorsement *alter-predict-bel-pos* is more potent that *alter-predict-bel-neg*.

7.5 Stability of Intentions in Dialogue

Stability of intentions arises through minimal change of attitudes during revision, from considerations of effort versus utility for the change of an intention, and from the maintenance of agreement of attitudes between agents.[11] Minimal change stops an agent switching back and forth between equally preferred plans. An agent only revises to a different plan when the effort difference between the current plan and the alternative plan exceeds some threshold.

Stability also arises through our agents' desire to maintain agreement of attitudes between agents. They may be loath to revise intentions when this would introduce conflicts. An agent may consider changing its commitment to an intention which it has communicated to another and knows the other has adopted. In the intention set from which it has dropped this intention it has a plan to resolve the disagreement over the intention given the other agent still has this intention. This plan would have associated effort endorsement which would reduce the preference for this set and ultimately discourage the agent from dropping the intention.

For example, suppose an agent *agent1* had had the intention to go to the cinema and had communicated this to another agent *agent2* who subsequently adopted it (*agent2* may then plan to get a ticket for *agent1*). *Agent1* subsequently considers changing his mind. He has gained extra justification for going out to dinner instead and these intentions are mutually exclusive. Given no other information *agent1* prefers the intention to eat. However, in the intention set which contains the intention to eat there is also the conflict between *agent1*'s intention not to go to the cinema and *agent1*'s belief that *agent2* believes that *agent1* intends to do so. If *agent1* has the desire rule to resolve such intention conflicts, then in this intention set he also has an effortful plan to resolve this conflict. *Agent1* can contemplate a number of possible revisions

1. The set which contains the intention to go to the cinema. This set is endorsed by the utility of going to the cinema.

2. The set which contains the intention to eat, not go to the cinema and the intention to tell the other agent of the change of plan. This set is endorsed with the utility of eating, the utility for resolving attitude conflicts and opposed by the effort of the plan to resolve the conflict.

3. The set which contains the intention to eat, not go to the cinema and not to tell the other agent of the change of plan. This set is endorsed by the utility to eat.

The preferred revision depends on the agent's utility for resolving conflicts, the effort involved in the plan to resolve the conflict (*agent2* may have to be contacted indirectly by forwarding a phone message), and the utilities for eating and watching a movie.

---

[11] This section has been motivated by a discussion with Barbara Grosz.

Chapter 8

Testing

User-Librarian dialogues are illustrated in detail in Brooks (1986) and Daniels (1987) and also described in Brooks et al. (1985); we give a brief relevant example on Greek-Turkish relations in appendix A. An analysis of the negotiation which occurs in these dialogues reveals that it consists mainly of simple elaboration. For example, in the Greek-Turkish relations dialogue there are only two instances of explanation. In no cases does negotiation extend beyond the initial belief conflict and its immediate resolution. There are no cases, for example, of an agent attempting to refute the antecedent of a justification for the disputed proposition or of an agent making reference to other disbelieved or dispreferred implications of the disputed belief.

What would be required to reproduce these dialogues? Even if we could do this, it would not be a useful standard for evaluating the system. The dialogues seem non-deterministic; there is no guarantee that the same participants would produce the same dialogue if the experiments were run again. A better way of characterising the problem would be to say that we want *similar* behaviour to that observed in the dialogues. However this simply redefines the problem as one of defining similar dialogues.

Unfortunately, as our current system has no natural language understanding capabilities, has very minimal information retrieval capabilities, and is computationally limited, we cannot do a '*live*' test in which the system attempts to answer real or test queries put by a human user. Fortunately such capabilities are not required to demonstrate our thesis: for our purposes a more modest standard of performance will suffice. What we can do is to demonstrate that the system described in chapters 5, 6 and 7 correctly implements the theory presented in chapter 2, by reproducing the characteristic behaviour of chapter 3 and 4. This implementation has been applied to two-agent systems involved in dialogue in a particular information retrieval subject domain. To produce this behaviour, we take the goal structure from BBD and the belief revision mechanism from ABR.

In chapter 6 we presented the decomposition of the IR task from a belief revision point of view, and in chapter 7 we explained how belief revision underlies dialogue. To model both halves of the dialogue we use two agents, one as librarian (i.e. *libr*) and one as user (i.e. *user*).[1] These agents behave as described in chapter 5 and are implemented as LISP processes on a SPARC Station IPX. They communicate via socket-based TPC/IP about a simple architectural history domain. We focus on a few simple subgoals within the task plan described in chapter 6.

8.1 Belief Revision in the Library Dialogues

We can identify a number of different causes of belief revision in dialogue, each of which can be characterised by a particular distribution of beliefs between the two agents (we assume that any potential change in an agent's beliefs results in 'revision'.) It is important to note that in all these cases, the speech acts work as intended, i.e. the speaker's intent is recognised, but this does not always result in a successful communicative outcome. For example, the speaker may be wrong about the hearer's belief state, with the result that the hearer 'does the wrong thing' with the content of the utterance (from the speaker's point of view). We illustrate the cases with examples for literature seeking in the architecture domain, as follows.

1. INFORM

One agent successfully informs another agent of its belief in some proposition. For example:

```
USER (John) : I am looking for information on churches
LIBR (Mary) : OK
```

John tells *Mary* something about the information he is looking for. *Mary* acknowledges that she has understood *John*'s utterance by making some form of affirmative reply (ok, right, uhu, mmm, etc.). Before his utterance *John* believes that he is looking for information on churches, though *Mary* does not know this, and that *Mary* will be better able to help him if he tells her. After *Mary*'s utterance, *John* believes that *Mary* believes that he is looking for information on churches. *Mary* believes that *John* is looking for information on churches and that *John* believes that she believes this etc. *John's* utterance is successful because *Mary* understands *John*'s intent; she already knows what a church is and what it is to want information about something.

2. QUESTION/ANSWER

One agent asks another agent for information. For example:

```
LIBR (Mary) : Any particular period?
USER (John) : Late Gothic.
```

---

[1] Following convention, and for ease of pronominal differentiation, we call these *Mary* and *John* respectively.

Mary asks if *John* is looking for buildings of any particular period and *John* tells her he is looking for buildings of the Late Gothic period. Before her utterance, *Mary* believes that she doesn't know the period of the buildings that *John* is interested in, that *John* does have this information, and that she would be better able to help *John* if she knew. After *John*'s utterance, *Mary* believes that *John* is interested in Late Gothic buildings, and that *John* believes that she believes this. [Note that this is simply a prediction on *John*'s part since *Mary* has not yet indicated that she understood *John*'s utterance. For example, *Mary* may not understand the term 'Late Gothic'.] *John* believes that *Mary* believes that he is interested in Late Gothic buildings and that *Mary* believes that he believes this. For *John's* utterance to be successful, *John* has to understand *Mary*'s question and be able to answer it.

3. FAILED INFORM

One agent's knowledge is incomplete and the agent knows that it is. For example:

```
USER (John) : I am looking for books on Wren.
LIBR (Mary) : Who is Wren?
USER (John) : He designed St. Paul's cathedral.
```

John asks for books on a subject *Mary* has never heard of before, so she asks for more information. While she believes that *John* is looking for books on Wren, she can't do much with this information since she doesn't know who (or what) 'Wren' is. While the speech act was successful, *Mary* couldn't do anything with the content of the utterance. *Mary*'s question is followed by an answer (a repair response). *John*'s reply provides just enough information to give *Mary* some idea who Wren was. But depending on the information *John* requires, *Mary* may have to ask for more information about Wren. We can view this case as:

```
INFORM segment starts
    QUESTION/ANSWER segment
INFORM segment ends
```

4. FAILED QUESTION/ANSWER

One agent's knowledge is incomplete but the other agent doesn't know this. For example:

```
LIBR (Mary) : Any particular architects?
USER (John) : I don't know.
```

Mary asks *John* if he is looking for information on a particular architect. She asks this because she believes *John* knows which architects he is interested in, and that the answer will help her find the information *John* is looking for. However *John* doesn't know the answer; this may be one of the things *John* is trying to discover.

5. MISUNDERSTANDING

One agent's knowledge is incomplete, but the agent does not realise that it is. This results in a form of 'default reasoning' based on the closed world assumption, i.e. that the agent already possesses all the information necessary to solve the problem. For example:

```
USER (John) : I am looking for books on Classical architecture.
LIBR (Mary) : Are you more interested in Greek or Roman
               architecture?
USER (John) : No, like the British Museum.
LIBR (Mary) : Ah, you mean Classical Revival architecture.
```

In this example *John* is mistaken about the meaning of the term 'Classical architecture'. This involves two changes of belief: *Mary*'s revises her original assumption about the period *John* is interested in; and *John* revises his belief about the meaning of the term 'Classical'. Note that for this to work, *John* must have a 'deeper model' of what he wants (in this case an example) to allow him to recover from his error. If *John* only has the problem description 'Classical architecture' he is stuck.

6. FAILED DEFAULT ASSUMPTION

A default assumption may be made by one agent about the beliefs of the other agent, which turns out to be wrong. For example:

```
USER (John) : I am a research assistant.
LIBR (Mary) : Do you want advanced books?
USER (John) : No, introductory books.
```

Mary assumes that because *John* is a research assistant, he will want advanced books. While a reasonable guess, it turns out to be wrong in this case. *Mary* simply revises her default assumption and those inferences and plans which depend upon it. Note that the utterance which triggers the default may have taken place some time before the attempt at verification.

7. FAILED PREDICTION

A prediction made by an agent about the effects of an utterance turns out to be wrong. For example:

```
USER (John) : I am looking for books on the architecture of
               Michelangelo.
LIBR (Mary) : I thought Michelangelo was an artist.
USER (John) : He was also an architect.  He designed St. Peter's
               in Rome.
LIBR (Mary) : OK, ...
```

This is similar to the case above except that the assumption is about the effects of an utterance on the beliefs of the agent rather than about the agent's current beliefs. It is also similar to a failed inform, but fails for a different reason, namely that the recipient holds a conflicting belief. A failed prediction nearly always results in the agent replanning as actions are the result of unachieved intentions. However belief revision and replanning are usually straightforward unless either the action provokes an unanticipated response, thus simply failing to achieve the intended effect, or the agent tries to analyse the reasons for the failure of the action.

In this case *Mary* thinks *John* has made a mistake but she is not sure. *John* convinces her that he is better informed than she is by incorporating her belief within a wider belief set, and by providing an additional justification for his claim that Michelangelo was an architect, i.e. that he designed St. Peter's. In this case, *John* is mistaken; Michelangelo didn't design St Peter's but *Mary* doesn't know this and accepts that *John* knows what he is talking about.

8.2 Simulation

The belief revision examples in the previous section are typical of the kinds that are found in library dialogues. We demonstrate how the system can model these situations by reproducing four of the dialogues presented above. The first example 'Failed Inform' demonstrates the telling both of a belief and of an intention. We present this example complete with the number of attitude and attitude set candidates, to give the reader a feel for the numbers involved: in appendix B we describe in detail the inferences drawn for part of this dialogue. The remaining examples share much in common with 'Failed Inform' and are not presented as fully.

It should be noted that we were not able to identify core beliefs appropriate to these examples. They do not therefore test and illustrate one element of our belief revision apparatus, namely increased coherence (*mc*). Our difficulty in finding cases that could exercise this important part of our belief revision theory (and its computational implementation) was unexpected, and is clearly unfortunate. We return to this problem in chapter 9.

The agents' databases are initialised with rules and attitudes and they are then left to their own devices. We distinguish *common knowledge* and *local knowledge*. Common knowledge is known by both agents. This includes the dialogue rules in chapter 7 and the following premise constraints (where *agent* is either the librarian (ie *libr*) or the user (i.e. *user*)).

$$(p\text{-bel}\ agent\ (libr\ can\ talk\ to\ user)\ \ premise)$$
$$(p\text{-bel}\ agent\ (user\ can\ talk\ to\ libr)\ \ premise)$$

These are the `tell` action constraints that the librarian and user can communicate.

$$(p\text{-bel}\ agent\ (libr \neq user)\ \ premise)$$
$$(p\text{-bel}\ agent\ (user \neq libr)\ \ premise)$$

These are the `adopt` action constraints that agents may adopt attitudes from other agents.[2] Local knowledge is that which is peculiar to an agent and is not necessarily common knowledge between agents. This includes the librarian's expertise rules presented in chapter 6 and the various initial attitudes that characterise the dialogues.

Dialogue 3: Failed Inform

In this example one agent's knowledge is incomplete and it is aware of this. The user agent (i.e. *John*) asks for books on a subject the librarian agent (i.e. *Mary*) has never heard of before, so she asks for more information. While the librarian believes that the user is looking for books on Wren she can't do much with this information since she doesn't know who (or what) '*wren*' is.

1. `USER (John) :  I am looking for books on Wren.`

2. `LIBR (Mary) :  Who is Wren?`

3. `USER (John) :  He designed St Paul's Cathedral.`

The following is the machine-generated dialogue:[3]

1. USER (John) : (tell user libr (bel user (pd wren) strong))

2. LIBR (Mary) : (tell libr user
(int libr (exists !x ?_ (bel libr (class wren !x)) strong)))

3. USER (John) : (tell user libr (bel user (class wren designed-st-pauls) strong))

The *user* agent is initialised with the following local propositions in its database:

1. (*p-bel user* (*class wren designed-st-pauls*) *spec*)
   He believes Wren designed St Pauls Cathedral.

2. (*p-bel user* (*pd wren*) *spec*)
   He believes Wren is a good description of his problem.

3. (*p-int user* (*bel libr* (*pd wren*)) *desire-pos*)
   He intends to share his description of the problem with the librarian. This could have arisen from a request for such information by *libr* as a partial solution to her goal to find an information retrieval strategy (see chapter 6).

---

[2] This avoids an agent adopting beliefs from itself or even planning to adopt a sequence of ever deeper nested beliefs from itself, i.e. the user adopting (*bel user p*) from (*bel user* (*bel user p*)) etc.

[3] For brevity we shall concentrate on the problem description content of the utterances and ignore the reference to the document type '*books*'.

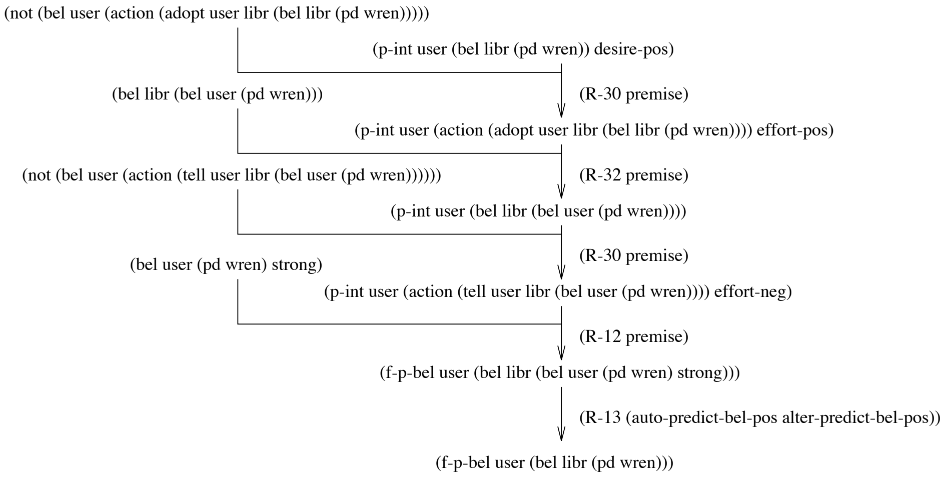

(not (bel user (action (adopt user libr (bel libr (pd wren)))))
(p-int user (bel libr (pd wren)) desire-pos)
(bel libr (bel user (pd wren)))
(R-30 premise)
(p-int user (action (adopt user libr (bel libr (pd wren)))) effort-pos)
(not (bel user (action (tell user libr (bel user (pd wren))))))
(R-32 premise)
(p-int user (bel libr (bel user (pd wren))))
(bel user (pd wren) strong)
(R-30 premise)
(p-int user (action (tell user libr (bel user (pd wren)))) effort-neg)
(R-12 premise)
(f-p-bel user (bel libr (bel user (pd wren) strong)))
(R-13 (auto-predict-bel-pos alter-predict-bel-pos))
(f-p-bel user (bel libr (pd wren)))

Figure 8.1: *User* agent's plan to suggest a problem description

User infers from his intention (*p-int user* (*bel libr* (*pd wren*))) and constructs the plan shown in figure 8.1. This comprises a simple `tell` action by *user* followed by the adoption of the contents of the `tell` message (i.e. (*pd wren*)) by *libr*. From his intention to `tell`, *user* predicts that eventually *libr* will come to believe that *user* believes that *wren* is a good problem descriptor, and also will `adopt` this problem descriptor as her own. Just prior to generating his utterance *user* has 10 belief set type attitudes, 67 intention set type attitudes, 4 belief set candidates and 12 intention set candidates:

User outputs the following message

(*tell user libr* (*bel user* (*pd wren*) *strong*))

and the *premise* endorsed p-belief that he has done so is added to the database

(*p-bel user* (*action* (*tell user libr* (*bel user* (*pd wren*) *strong*))) *premise*)

He subsequently drops his intention to `tell` [4]

(*p-int user* (*action* (*tell user libr* (*bel user* (*pd wren*)))))

Libr agent receives the *user*'s message and creates the premise that that message occurred

(*p-bel libr* (*action* (*tell user libr* (*bel user* (*pd wren*) *strong*))) *premise*)

From this *libr* infers that *user* is strongly committed to (*pd wren*)

(*p-bel libr* (*bel user* (*pd wren*) *strong*))

---

[4] His prediction beliefs are also dropped as a consequence. However, he uses rule R–12 (see chapter 7) to infer, by default, that this `tell` action will eventually be successful and this reaffirms the predicted beliefs (see figure 8.2).

$$\begin{tikzcd}
\text{(bel user (action (tell user libr (bel user (pd wren) strong)))} \arrow[d, "\text{(R-11 definite)}"] \\
\text{(f-p-bel user (bel libr (bel user (pd wren) strong)))}
\end{tikzcd}$$

Figure 8.2:

Libr adopts *user*'s belief that (*pd wren*) since the *strong* commitment above constitutes a *2c-pos* reason for *libr* to do so (*libr* has no conflicting reasons). However, *wren* is not part of *libr*'s problem descriptor hierarchy. *Libr* has accepted (*pd wren*) and is even strongly committed to her belief about it, but she does not know who (or what) *wren* is! Intuitively *libr* should not accept a problem descriptor until she is sure that she agrees that it is an appropriate problem descriptor term (i.e. until she is aware of how *wren* fits in with her preferred problem description). She has no reference to *wren* in her problem descriptor tree hierarchy and attempts to fill this gap in her knowledge by asking for more information. This is captured by rule D–1 (see chapter 6).

(*bel libr* (*pd ?TERM*))
$\Rightarrow_{[desire-pos]}$ (*intorbel libr* (*exists !x ?_* (*bel libr* (*class !x ?TERM*))))

Libr plans to ask *user* what *wren* is (i.e. to which class in *user*'s problem descriptor hierarchy *wren* belongs). Using the above rule *libr* infers the following leading intention

(*p-int libr* (*exists !x ?_* (*bel libr* (*class wren !x*))))

By linking *wren* with *pd* term information she already possesses *libr* can reason about its accordance/consistency with the preferred problem description. Only then might reasons arise to disbelieve (*pd wren*), by the realisation that (*pd wren*) is inconsistent with *libr*'s preferred beliefs (for example, if *wren* is deemed to belong to the least preferred of two mutually exclusive problem descriptor classes).

Libr's plan to satisfy her intention (*p-int libr* (*exists !x ?_* (*bel libr* (*class wren !x*)))) is shown in figures 8.3 and 8.4. She intends to encourage *user* to `adopt` this intention as his own, with the intended outcome that *user* will communicate a class description which *libr* will subsequently `adopt`. This plan adds to the her previous propositions: 20 belief set type propositions and 135 intention set type propositions. There are 4 belief set candidates and 12 intention set candidates.

The precondition for *libr*'s plan is that she believes that *user* actually has a class description for *wren*. Since *user* initially suggested *wren* as a problem description *libr* assumes *user* has a class entry for this through rule D–2

(*p-bel libr* (*bel user* (*pd ?TERM*)))
$\Rightarrow_{[spec]}$ (*p-bel libr* (*exists !x ?_* (*bel user* (*class !x ?TERM*))))

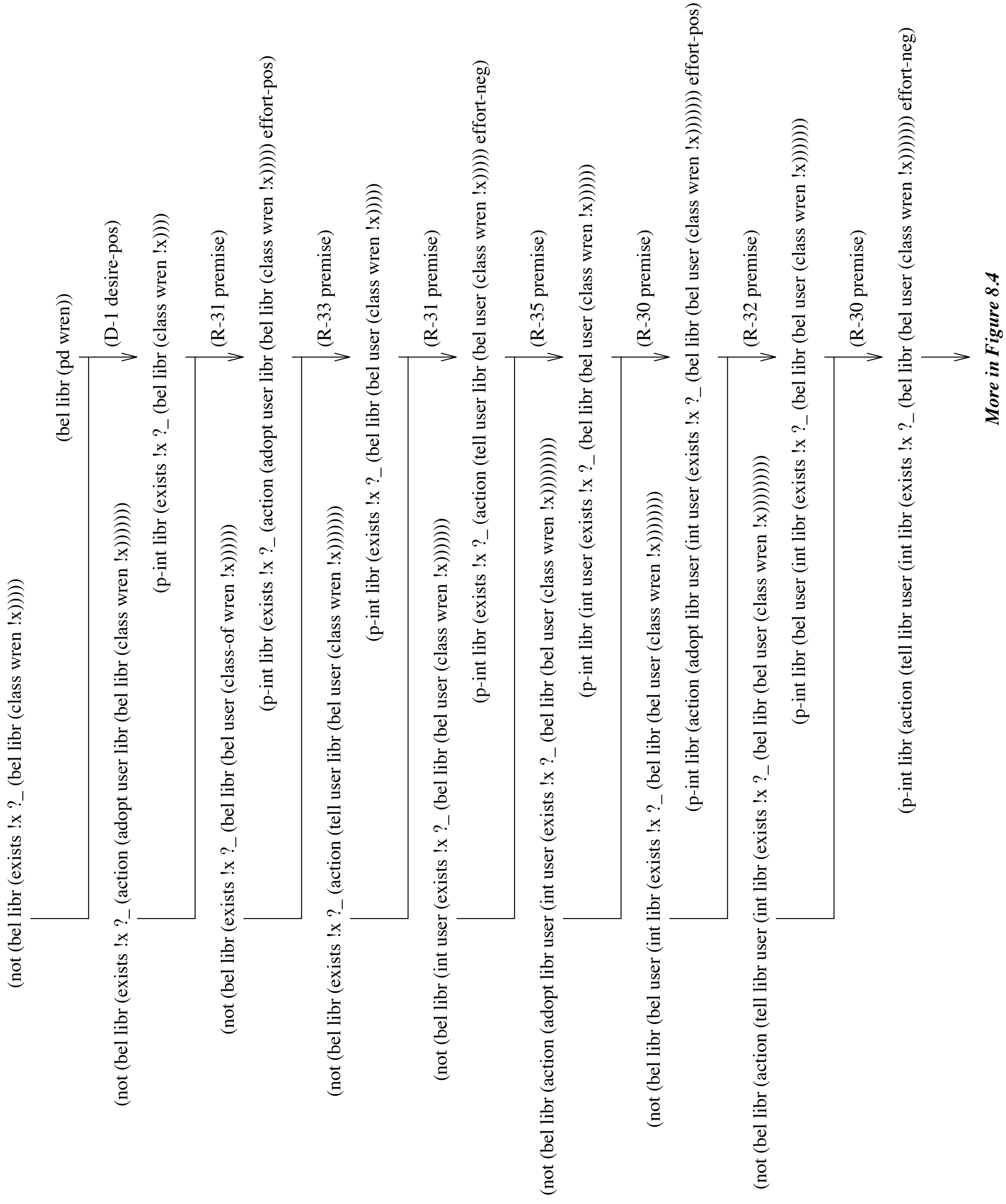

(not (bel libr (exists !x ?_ (bel libr (class wren !x)))))
(bel libr (pd wren))
(D-1 desire-pos)
(not (bel libr (exists !x ?_ (action (adopt user libr (bel libr (class wren !x)))))))
(p-int libr (exists !x ?_ (bel libr (class wren !x))))
(R-31 premise)
(not (bel libr (exists !x ?_ (bel libr (bel user (class-of wren !x))))))
(p-int libr (exists !x ?_ (action (adopt user libr (bel libr (class wren !x)))))) effort-pos
(R-33 premise)
(not (bel libr (exists !x ?_ (action (tell user libr (bel user (class wren !x)))))))
(p-int libr (exists !x ?_ (bel libr (bel user (class wren !x)))))
(R-31 premise)
(not (bel libr (int user (exists !x ?_ (bel libr (bel user (class wren !x)))))))
(p-int libr (exists !x ?_ (action (tell user libr (bel user (class wren !x)))))) effort-neg
(R-35 premise)
(not (bel libr (action (adopt libr user (int user (exists !x ?_ (bel libr (bel user (class wren !x)))))))))
(p-int libr (int user (exists !x ?_ (bel libr (bel user (class wren !x))))))
(R-30 premise)
(not (bel libr (bel user (int libr (exists !x ?_ (bel libr (bel user (class wren !x))))))))
(p-int libr (action (adopt libr user (int user (exists !x ?_ (bel libr (bel user (class wren !x)))))))) effort-pos
(R-32 premise)
(not (bel libr (action (tell libr user (int libr (exists !x ?_ (bel libr (bel user (class wren !x)))))))))
(p-int libr (bel user (int libr (exists !x ?_ (bel libr (bel user (class wren !x)))))))
(R-30 premise)
(p-int libr (action (tell libr user (int libr (exists !x ?_ (bel libr (bel user (class wren !x)))))))) effort-neg
More in Figure 8.4

Figure 8.3: *Libr*'s plan to find out about *wren*: part 1

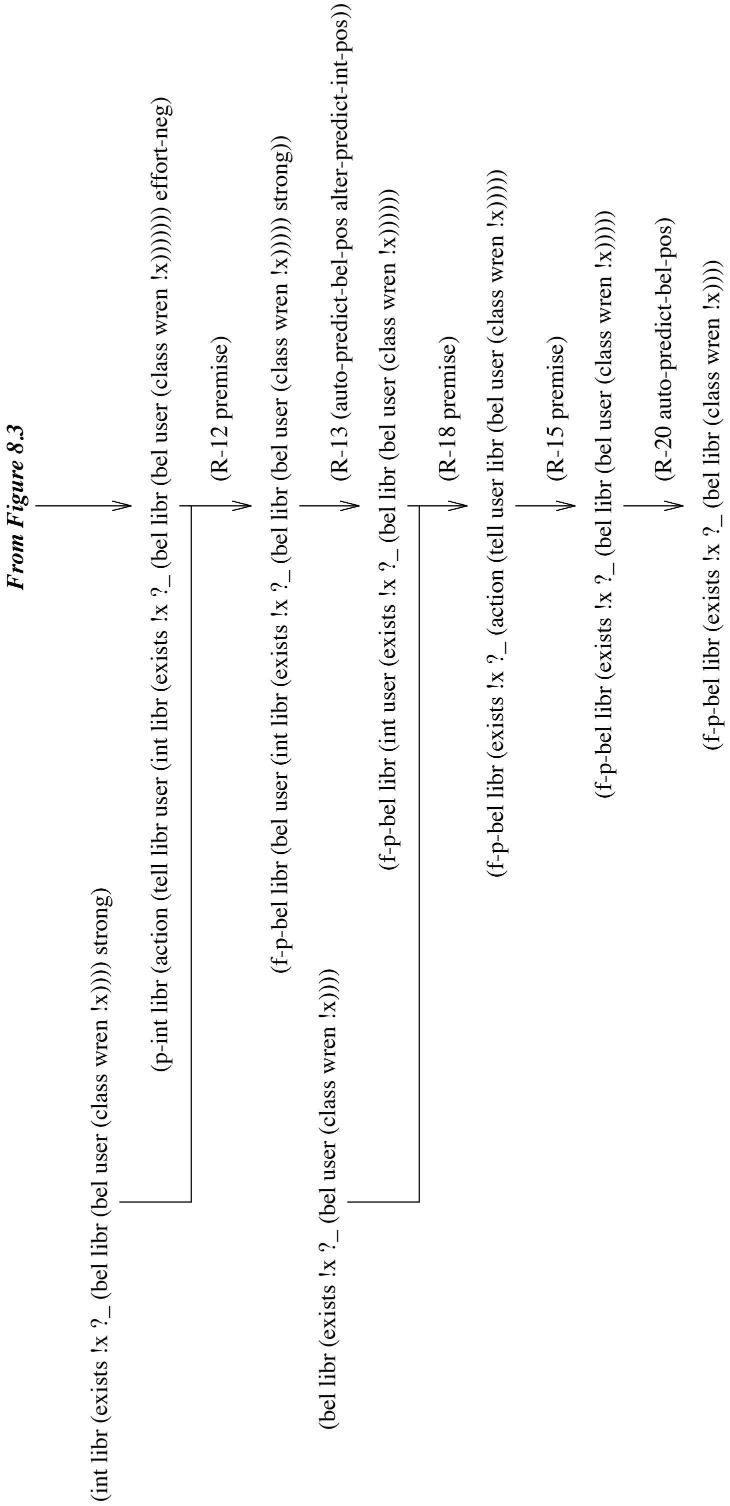

From Figure 8.3
(int libr (exists !x ?_ (bel libr (bel user (class wren !x)))) strong)
(p-int libr (action (tell libr user (int libr (exists !x ?_ (bel libr (bel user (class wren !x))))))) effort-neg)
(R-12 premise)
(f-p-bel libr (bel user (int libr (exists !x ?_ (bel libr (bel user (class wren !x))))) strong))
(bel libr (exists !x ?_ (bel user (class wren !x))))
(R-13 (auto-predict-bel-pos alter-predict-int-pos))
(f-p-bel libr (int user (exists !x ?_ (bel libr (bel user (class wren !x))))))
(R-18 premise)
(f-p-bel libr (exists !x ?_ (action (tell user libr (bel user (class wren !x))))))
(R-15 premise)
(f-p-bel libr (exists !x ?_ (bel libr (bel user (class wren !x)))))
(R-20 auto-predict-bel-pos)
(f-p-bel libr (exists !x ?_ (bel libr (class wren !x))))

Figure 8.4: *Libr*'s plan to find out about *wren*: part 2

Libr outputs his utterance

(*tell libr user* (*int libr* (*exists !x ?_* (*bel libr* (*bel user* (*class wren !x*)))) *strong*))

and *user* infers *libr*'s strongly committed intention. *User* subsequently adopts this intention as his own *desire-pos* endorsed intention

(*p-int user* (*exists !x ?_* (*bel libr* (*class wren !x*))) *desire-pos*)

User already believes (*class wren designed-st-pauls*) and infers his intention to reply using rule R–22

(*p-int ?A1* (*exists ?X ?Y* (*bel ?A2* (*?ATT ?A1 ?P*))))
(*?ATT ?A1 ?Q*)
(*binds ?X ?Y ?P ?Q*)
$\Rightarrow_{[desire-strong]}$ (*intorbel ?A1* (*bel ?A2* (*?ATT ?A1 ?Q*)))

He plans to instill in *libr* the belief that *wren* is of the class of people who designed St Paul's Cathedral (see figure 8.5).

Prior to his final utterance *user* has 19 belief set propositions, 160 intention set propositions, 4 belief set candidates and 1152 intention set candidates.[5]

This example took just over 2 days to run. We repeated the experiment with *p-int*s and *f-p-bel*s as non assumptions (and thus switched off positive undermining of these attitudes) and the example ran in little less than eighteen minutes. The following numbers of propositions and candidates were recorded for this case. Those for the full system above are reproduced for comparison. (We consider the implications of these timing points in the concluding chapter.)

Cog State	*Full system*		*Minimal system*	
	int types	*int cands*	*int types*	*int cands*
User after plan in figure 8.1	67	12	45	4
Libr after plan in figure 8.3	135	12	89	6
User after plan in figure 8.5	160	1152	91	18

Dialogue 5: Misunderstanding

In this example, one agent's knowledge is incomplete, but this agent does not realise that it is. For example, the user is mistaken as to the meaning of *Classical architecture* and believes that the *British Museum* is an example of it. The librarian believes that the concept of *Classical architecture* captures both *Greek* and *Roman* style architecture, that the *British Museum* is an example of *Classical Revival architecture*, and that *Classical* and *Classical Revival* describe totally different architectural styles.

---

[5] These figures represent the combination of *user*'s plans for both of his utterances in this dialogue.

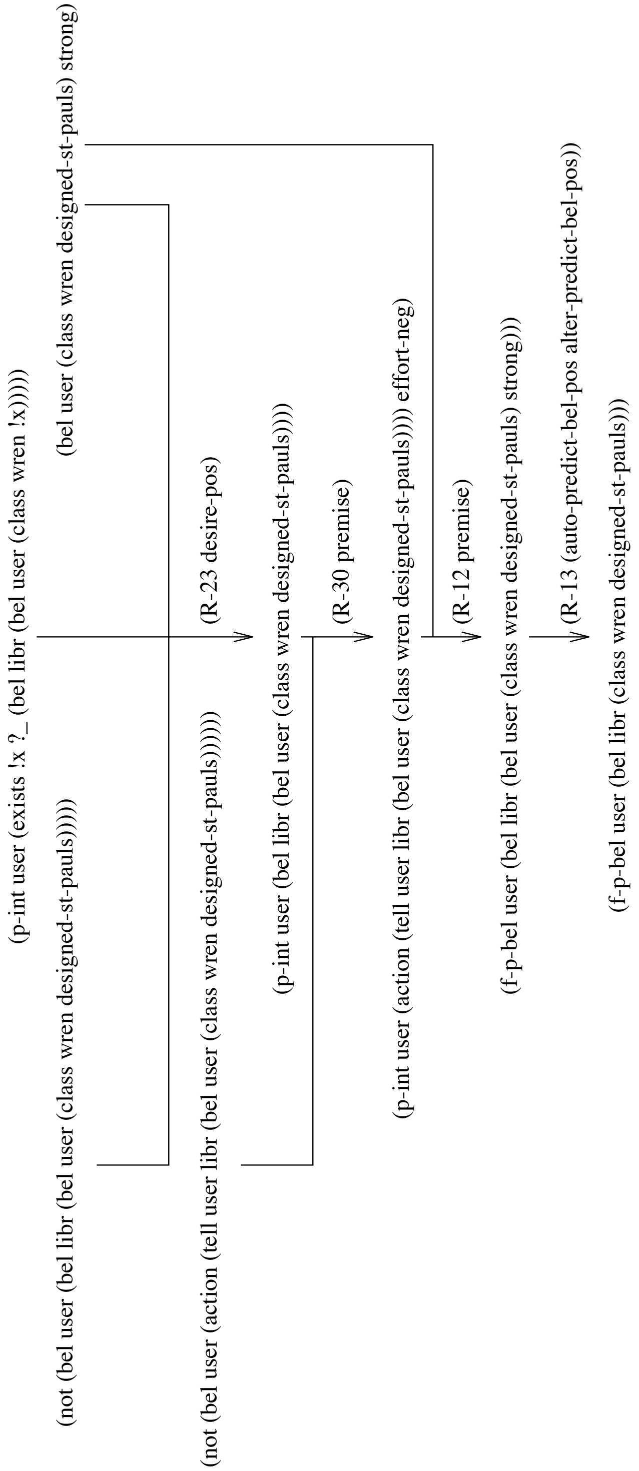
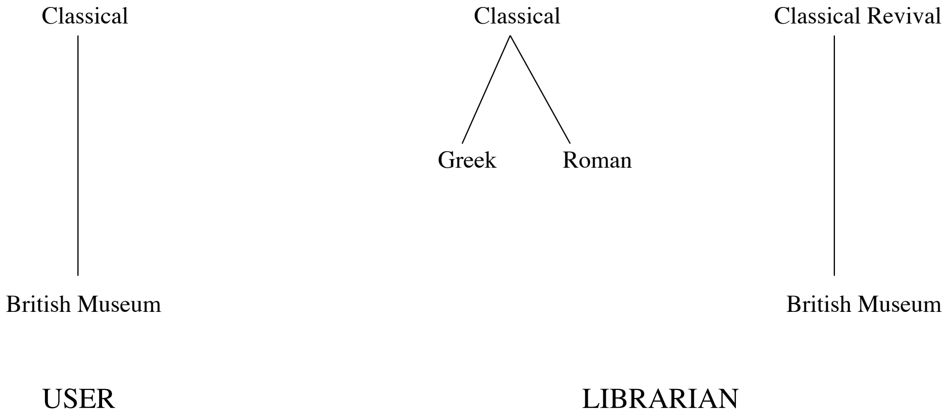

(p-int user (exists !x ?_ (bel libr (bel user (class wren !x)))))
(not (bel user (bel libr (bel user (class wren designed-st-pauls)))))
(bel user (class wren designed-st-pauls) strong)
(R-23 desire-pos)
(not (bel user (action (tell user libr (bel user (class wren designed-st-pauls))))))
(p-int user (bel libr (bel user (class wren designed-st-pauls))))
(R-30 premise)
(p-int user (action (tell user libr (bel user (class wren designed-st-pauls)))) effort-neg)
(R-12 premise)
(f-p-bel user (bel libr (bel user (class wren designed-st-pauls) strong)))
(R-13 (auto-predict-bel-pos alter-predict-bel-pos))
(f-p-bel user (bel libr (class wren designed-st-pauls)))

Figure 8.5: *User*'s plan to offer information about *wren*



The user informs the librarian that he wants literature on *Classical architecture* but the librarian realises that he actually wants *Classical Revival architecture*:

1. USER (John) : `I am a looking for books on Classical architecture.`
2. LIBR (Mary) : `Are you more interested in Greek or Roman architecture?`
3. USER (John) : `No, like the British Museum.`
4. LIBR (Mary) : `Ah, you mean Classical Revival architecture.`

The following is the machine generated dialogue:

1. *USER* (*John*) *:* (*tell user libr* (*bel user* (*pd classical-arch*) *strong*))
2. *LIBR* (*Mary*) *:* (*tell libr user* (*int libr* (*exists !x* ((*pd roman*) (*not* (*pd roman*)))
   (*bel libr* (*bel user !x*))) *st rong*))
   LIBR (*Mary*) *:* (*tell libr user* (*int libr* (*exists !x* ((*pd greek*) (*not* (*pd greek*)))
   (*bel libr* (*bel user !x*))) *st rong*))
3. *USER* (*John*) *:* (*tell user libr* (*bel user* (*not* (*pd roman*)) *strong*))
   USER (*John*) *:* (*tell user libr* (*bel user* (*not* (*pd greek*)) *strong*))
   USER (*John*) *:* (*tell user libr* (*bel user* (*pd british-museum*) *strong*))
4. *LIBR* (*Mary*) *:* (*tell libr user* (*bel libr* (*not* (*pd classical-arch*)) *strong*))
   LIBR (*Mary*) *:* (*tell libr user* (*bel libr* (*pd classical-revival-arch*)) *strong*))

Libr is initialised with the following attitudes:

1. (*p-bel libr* (*pdtree classical-arch* (*greek roman*)) *premise*)
   She believes that *roman* and *greek* are the sub-classes of *classical-architecture*.
2. (*p-bel libr* (*class classical-revival-arch british-museum*) *premise*)
   She believes that the *british-museum* is a member of the class of *classical-revival-architecture*.

3. (*p-bel libr* (*exists !x ?_* (*bel libr* (*class greek !x*))) *premise*)
   She knows what *greek* means (i.e. how *greek* fits into his problem descriptor term hierarchy).

4. (*p-bel libr* (*exists !x ?_* (*bel libr* (*class roman !x*))) *premise*)
   She knows what *roman* means (i.e. how *roman* fits into his problem descriptor term hierarchy).

5. (*p-bel libr* (*exists !x ?_* (*bel libr* (*class british-museum !x*))) *premise*)
   She knows what *british-museum* means (i.e. how *british-museum* fits into his problem descriptor term hierarchy).

User is initialised with the following attitudes:

1. (*p-bel user* (*pd classical-arch*) *spec*)
   He believes that *Classical architecture* is a good description of his problem.

2. (*p-bel user* (*not* (*pd greek*)) *spec*)
   He is not interested in *Greek* architecture.

3. (*p-bel user* (*not* (*pd roman*)) *spec*)
   He is not interested in *Roman* architecture.

4. (*p-int user* (*bel libr* (*pd classical-arch*)) *desire-pos*)
   He intends to inform *libr* of his belief that *Classical architecture* is a good description of the problem.

User initially informs *libr* that he strongly believes that he wants literature on classical architecture. This gives *libr* a *2c-pos* reason to believe that (*pd classical-architecture*) is a good problem descriptor, which she subsequently adopts. She infers that *user* wants to study either *greek* or *roman* architecture, since these are the two sub-types of *classical*. However, *libr* is unsure which of these sub-types is appropriate. She has three preferred belief sets: one containing (*pd greek*), another (*pd roman*) and a third with both. She addresses this uncertainty and infers using

(*bel ?A1 ?P uncertain*)
$\Rightarrow_{[desire-pos]}$ (*intorbel ?A1* (*exists ?X* (*?P* (*not ?P*))
(*bel ?A1* (*bel ?A2 ?X*))))

the intention to know whether the *user* believes (*pd roman*) or not, and whether he believes (*pd greek*) or not. The plan for (*pd roman*) is shown in figures 8.6 and 8.7: the plan for (*pd greek*) is similar.

Libr asks *user* whether he believes (*pd roman*) or (*not* (*pd roman*)) is appropriate

(*tell libr user* (*int libr* (*exists !x* ((*pd roman*) (*not* (*pd roman*)))
(*bel libr* (*bel user !x*))) *strong*))

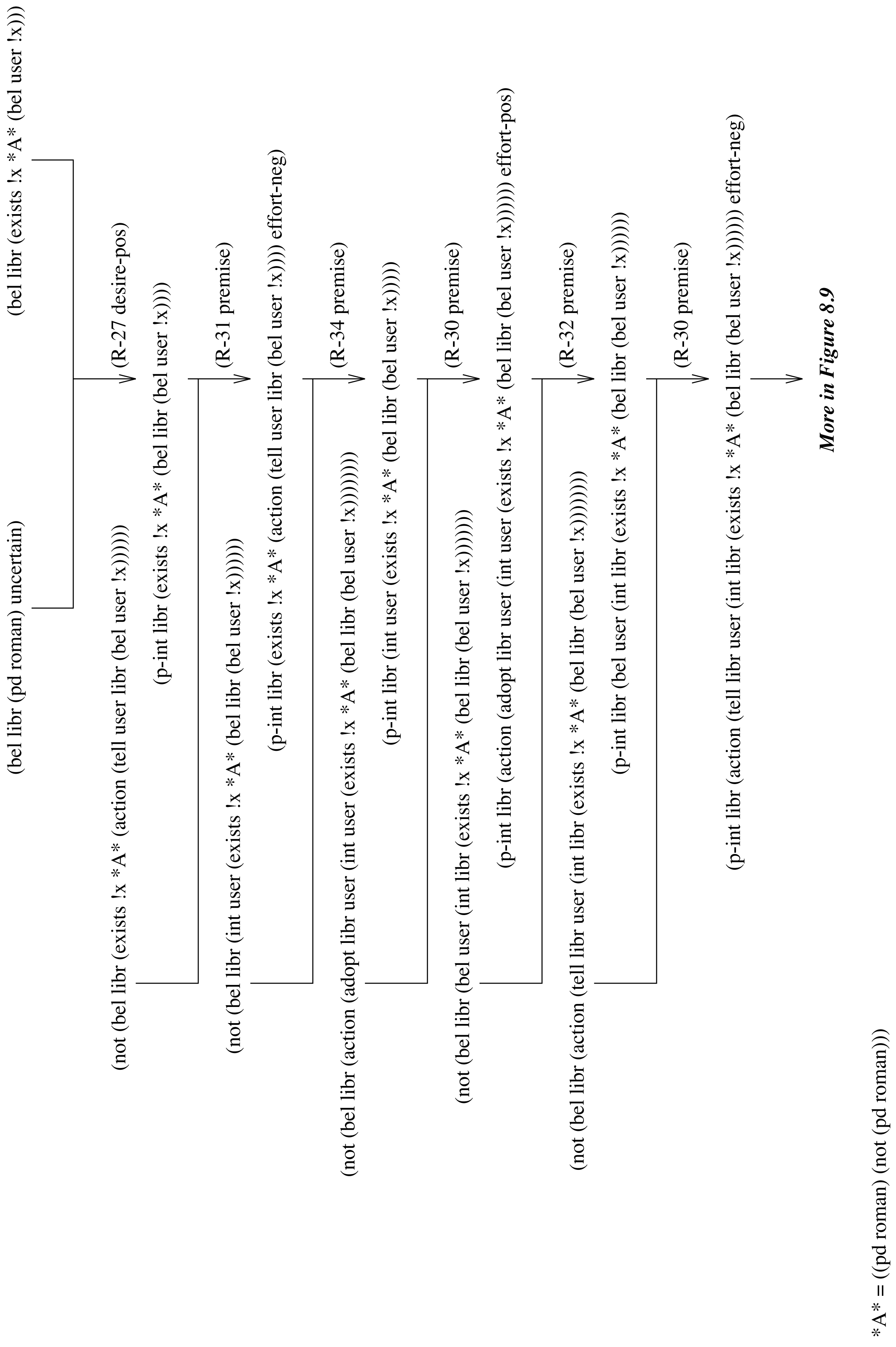

(bel libr (pd roman) uncertain)
(bel libr (exists !x *A* (bel user !x)))
(R-27 desire-pos)
(not (bel libr (exists !x *A* (action (tell user libr (bel user !x))))))
(p-int libr (exists !x *A* (bel libr (bel user !x))))
(R-31 premise)
(not (bel libr (int user (exists !x *A* (bel libr (bel user !x))))))
(p-int libr (exists !x *A* (action (tell user libr (bel user !x)))) effort-neg)
(R-34 premise)
(not (bel libr (action (adopt libr user (int user (exists !x *A* (bel libr (bel user !x))))))))
(p-int libr (int user (exists !x *A* (bel libr (bel user !x)))))
(R-30 premise)
(not (bel libr (bel user (int libr (exists !x *A* (bel libr (bel user !x)))))))
(p-int libr (action (adopt libr user (int user (exists !x *A* (bel libr (bel user !x)))))) effort-pos)
(R-32 premise)
(not (bel libr (action (tell libr user (int libr (exists !x *A* (bel libr (bel user !x))))))))
(p-int libr (bel user (int libr (exists !x *A* (bel libr (bel user !x))))))
(R-30 premise)
(p-int libr (action (tell libr user (int libr (exists !x *A* (bel libr (bel user !x)))))) effort-neg)
More in Figure 8.9
A = ((pd roman) (not (pd roman)))

Figure 8.6: *Libr* asks *user* if he wants to know about *Roman* architecture: Part 1.

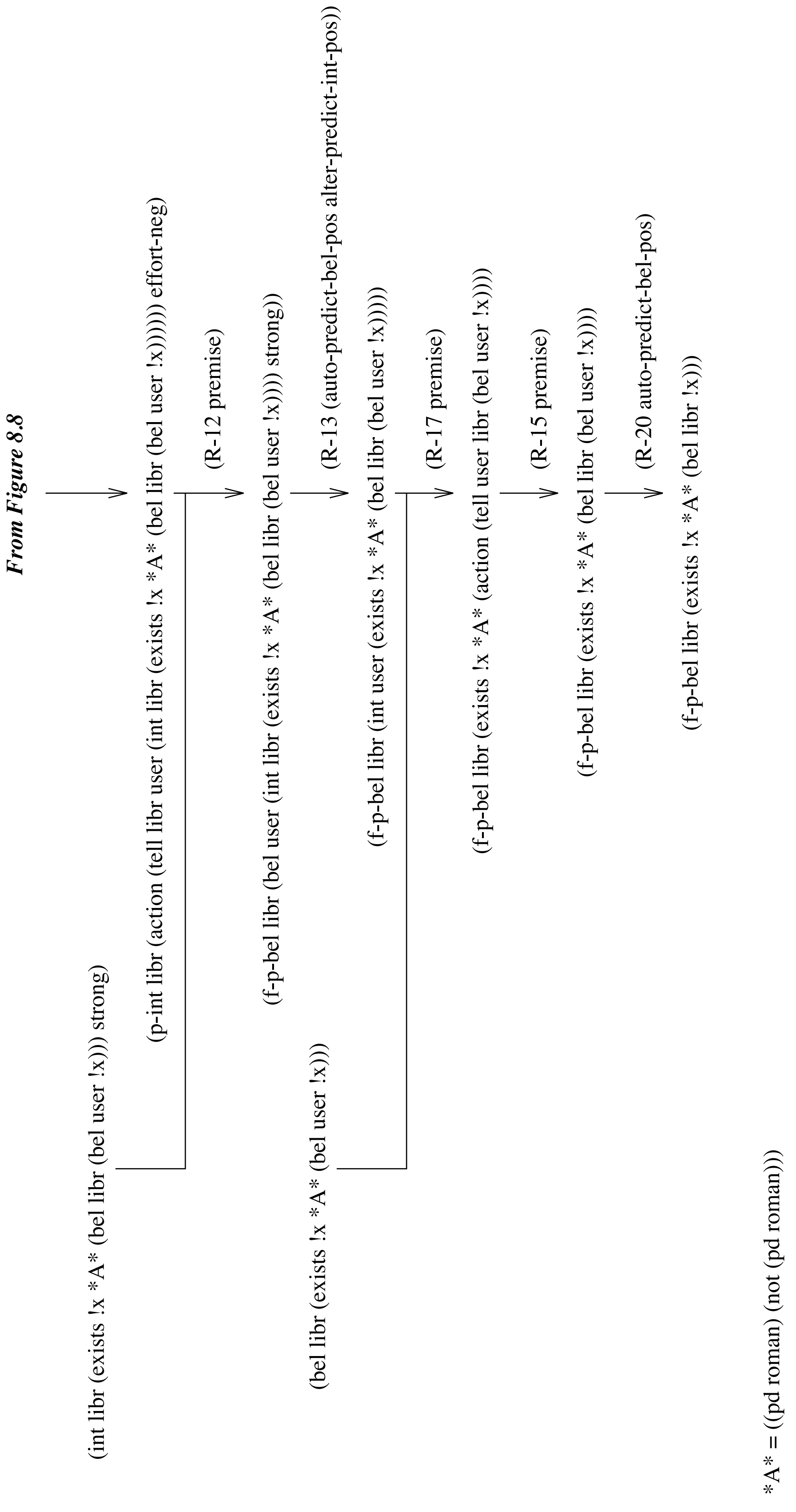

From Figure 8.8
(int libr (exists !x *A* (bel libr (bel user !x))) strong)
(p-int libr (action (tell libr user (int libr (exists !x *A* (bel libr (bel user !x)))))) effort-neg)
(R-12 premise)
(f-p-bel libr (bel user (int libr (exists !x *A* (bel libr (bel user !x)))) strong))
(R-13 (auto-predict-bel-pos alter-predict-int-pos))
(bel libr (exists !x *A* (bel user !x)))
(f-p-bel libr (int user (exists !x *A* (bel libr (bel user !x)))))
(R-17 premise)
(f-p-bel libr (exists !x *A* (action (tell user libr (bel user !x)))))
(R-15 premise)
(f-p-bel libr (exists !x *A* (bel libr (bel user !x))))
(R-20 auto-predict-bel-pos)
(f-p-bel libr (exists !x *A* (bel libr !x)))
A = ((pd roman) (not (pd roman)))

Figure 8.7: *Libr* asks *user* if he wants to know about *Roman* architecture: Part 2.

While *libr* is planning her second utterance to resolve her uncertainty about (*pd greek*) *user* receives the message above. *User* adopts the intention as his own

(*int user* (*exists !x* ((*pd roman*) (*not* (*pd roman*))) (*bel libr* (*bel user !x*))) *strong*)

and since he already believes he is not interested in *roman* architecture, he infers the leading intention to `tell` *libr* this. Next, *user* assumes *libr*'s intention is to discover what example architectures are appropriate. He uses the following rule

(*p-int ?A1* (*int ?A2* (*exists ?X* ((*?TYPE ?P*) (*not* (*?TYPE ?P*))) (*bel ?A1* (*?ATT ?A2 !x*))))) *&*
(*?ATT ?A1* (*not* (*?TYPE ?P*))) *&*
(*?ATT ?A1* (*?TYPE ?Q*))
$\Rightarrow_{[desire-pos]}$ (*intorbel* (*?ATT ?A2* (*?TYPE ?Q*)))

to infer the leading intention to convince *libr* that the *british-museum* is a good example description of his problem. *User* subsequently sends two messages: the first rejecting (*pd roman*), and the second offering an alternative (*pd british-museum*):

(*tell user libr* (*bel user* (*not* (*pd roman*)) *strong*))
(*tell user libr* (*bel user* (*pd british-museum*) *strong*))

Meanwhile the *user* has been planning to resolve his uncertainty in his belief (*pd roman*). There is a potential problem here. If *user* were to reply to *libr*'s initial query about roman architecture before *libr* had asked about *greek* architecture then, on receiving and processing the *user*'s message (*not* (*pd roman*)), *libr* would accept that (*not* (*pd roman*)), consequently come to believe (*pd greek*) strongly, and thus drop her plan to ask about (*pd greek*). Then, when *libr* receives *user*'s second message (i.e. (*pd british-museum*) she will infer equal reason for believing (*pd classical-revival*) as (*pd classical*) and she would be seriously confused. She would have two equally endorsed belief sets, each with a different problem description, namely one containing (*pd classical-revival-arch*) and (*pd british-museum*) and the other containing (*pd classical-arch*) and (*pd greek*):

(*1*) (*p-bel libr* (*pd classical-architecture*))
(*p-bel libr* (*not* (*pd roman*)))
(*p-bel libr* (*pd greek*))

(*2*) (*p-bel libr* (*pd classical-revival-arch*))
(*p-bel libr* (*not* (*pd roman*)))
(*p-bel libr* (*pd british-museum*))

Belief set (1) would be marginally preferred over (2) through minimal change. *Libr* would have many *weak* beliefs and it is not necessarily the case that she would ask about (*pd greek*) as required. We overcome this by freezing the *user* agent's action cycle until *libr* has asked about *roman* and *greek* architecture. *User* then receives the following simultaneously:

(*tell libr user* (*int libr* (*exists !x* ((*pd roman*) (*not* (*pd roman*)))
(*bel libr* (*bel user !x*))) *strong*))
(*tell libr user* (*int libr* (*exists !x* ((*pd greek*) (*not* (*pd greek*)))
(*bel libr* (*bel user !x*))) *strong*))

He replies:

(*tell user libr* (*bel user* (*not* (*pd roman*))) *strong*))
(*tell user libr* (*bel user* (*pd british-museum*)) *strong*))
(*tell user libr* (*bel user* (*not* (*pd greek*))) *strong*))

These give *libr* a *2c-pos* reason to believe (*not* (*pd roman*)) and she comes to prefer belief set (2) above.

The librarian's final utterance in the human text is ambiguous:

"Ah, you mean Classical Revival Architecture."

This could mean one of three things.

1. That the librarian has recognised an inconsistency between herself and the user as to the meaning of *Classical*. The Librarian understands what concept the user wants and her utterance is purely to educate the user of the correct terminology.

2. That the librarian has recognised a conceptual inconsistency between herself and the user. The user believes that the British Museum was built in the classical style and not the classical revival style.

3. That the librarian has changed her belief about the appropriate architecture from *Classical* to *Classical Revivall* and is therefore just exclaiming without attempting to change the user's mind.

It is hard for our *libr* agent to distinguish cases 1 and 2 without having some complex model of the *user* agent's term hierarchy. Generally, agents communicate terms which capture the meaning of a concept. The hearer has to attempt to infer the concept the speaker has in mind. If the hearer comes to believe the concept is inappropriate then does it believe its own mapping from term to concept is at fault, or the speaker's mapping from concept to term (i.e. terminological disagreement), or the speaker's original belief in the concept?

Case 3 above involves exclamation with no desired strategic effect on the hearer's beliefs. We are not concerned with this type of utterance. Consequently, we capture the cases 1 and 2 in our test example. *Libr* believes that (*pd classical-revival-arch*) is appropriate and that (*pd classical-arch*) is not, and she also believes that *user* believes that (*pd classical-arch*) and (*not* (*pd classical-revival-arch*)). *Libr* considers changing *user*'s mind about *Classical* by telling *user* (*not* (*pd classical-arch*)) with *strong* communicated commitment. However, since *libr* believes that *user* is strongly committed to (*pd classical-arch*), she is uncertain as to whether her plan will be successful. The final step in this plan, that *user* accepts *libr*'s suggestion (*not* (*pd classical-arch*)), is the only weakly predicted outcome of the plan. *Libr* subsequently replans from her intention for agreement on (*not* (*pd classical-arch*)), but this time by planning to offer a justification. *Libr* plans to convince *user* of the justification

$$(pd\ classical\text{-}revival\text{-}arch) \Rightarrow (not\ (pd\ classical\text{-}arch))$$

Libr now has two plans to convince *user* that (*pd classical-revival*): one is to simply `tell` the *user* that (*bel libr* (*pd classical-revival*) *strong*), and the second is to offer a justification for (*pd classical-revival*). This justification comprises the following beliefs

(*bel libr* (*rule-inst* "(*bel ?A1* (*pd classical-revival-arch*))
⇒ (*not* (*bel ?A1* (*pd classical-arch*)))"
((*bel reasoner* (*pd classical-revival*)))
(*bel reasoner* (*not classical*))) *strong*)
(*bel libr* (*pd classical-revival*) *strong*)

and gives extra reason for *libr* believing that the *user* will `adopt` (*not* (*pd classical-arch*)) in the future state. *Libr* predicts that *user* will come to believe both beliefs in the justification, and also that he will come to believe the consequent (*not* (*pd classical-arch*)). Each plan when considered separately cannot achieve the leading intention (*p-int libr* (*bel user* (*pd classical-revival-arch*))). However, their combined effect is the prediction that (*bel user* (*pd classical-revival*)) will hold in the future state (this combined plan is shown in figure 8.8). Since *libr* believes that *user* already believes the rule instance, she plans to communicate (*pd classical-revival-arch*) and (*not* (*pd classical-arch*)) only. She subsequently outputs the following utterances

(*tell libr user* (*bel libr* (*not* (*pd classical-arch*)) *strong*))
(*tell libr user* (*bel libr* (*pd classical-revival-arch*) *strong*))

and *user* accepts that *classical-revival-arch* is a more appropriate problem descriptor.

Dialogue 6: Failed Default Assumption

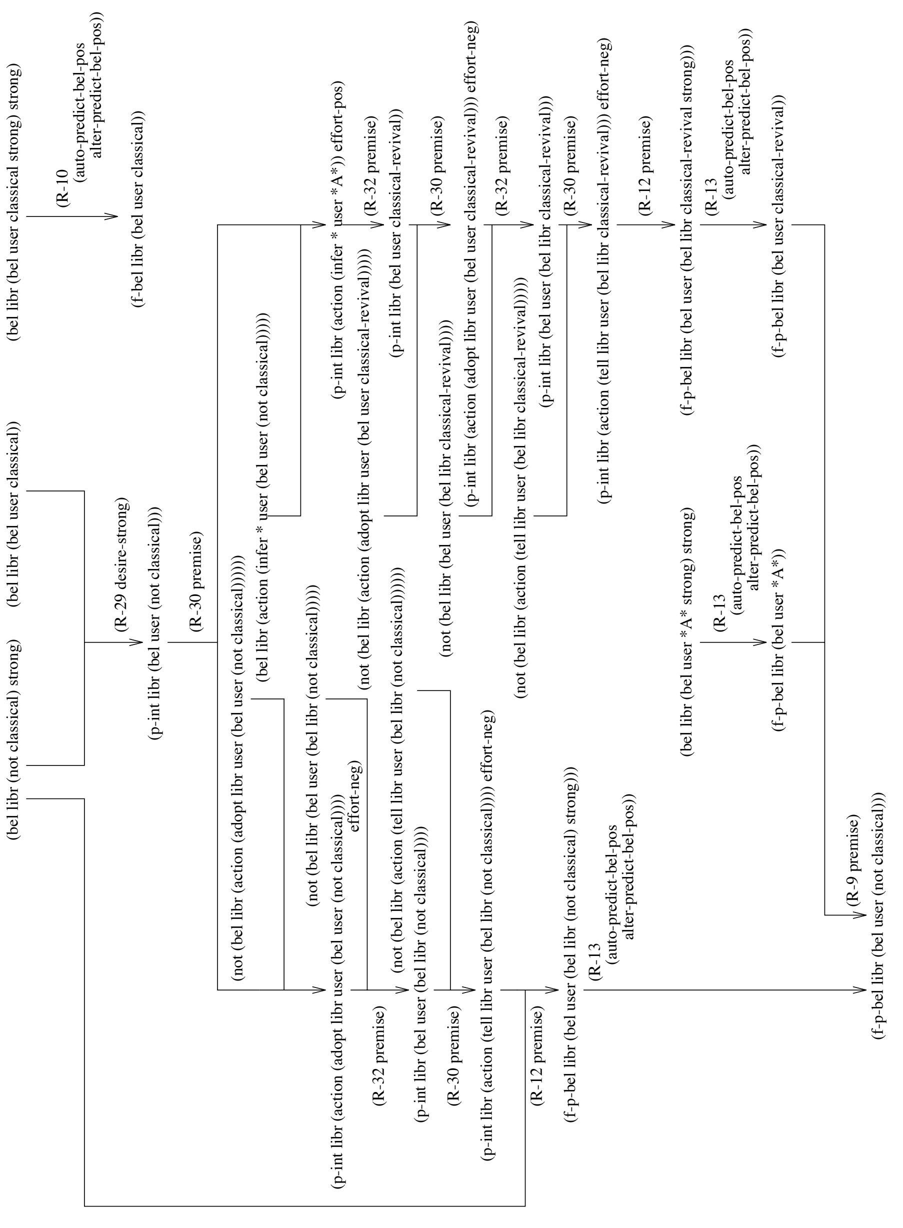
A default assumption made by one agent about the beliefs of another agent turns out to be wrong. *Libr* agent assumes that because *user* is a research assistant, he will want advanced books. While a reasonable guess, it turns out to be wrong in this case. *Libr* revises his default assumption and those inferences and plans which depend on it.

1. `USER (John) :  I am a research assistant.`

2. `LIBR (Mary) :  Do you want advanced books?`

3. `USER (John) :  No, introductory books.`

The following is the machine generated dialogue:


(bel libr (not classical) strong)
(bel libr (bel user classical))
(bel libr (bel user classical strong) strong)
(R-10
(auto-predict-bel-pos
alter-predict-bel-pos))
(f-bel libr (bel user classical))
(R-29 desire-strong)
(p-int libr (bel user (not classical)))
(R-30 premise)
(not (bel libr (action (adopt libr user (bel user (not classical))))))
(bel libr (action (infer * user (bel user (not classical)))))
(p-int libr (action (infer * user *A*)) effort-pos)
(R-32 premise)
(p-int libr (bel user classical-revival))
(not (bel libr (action (adopt libr user (bel user classical-revival)))))
(R-30 premise)
(not (bel libr (bel user (bel libr classical-revival))))
(p-int libr (action (adopt libr user (bel user classical-revival))) effort-neg)
(R-32 premise)
(p-int libr (bel user (bel libr classical-revival)))
(not (bel libr (action (tell libr user (bel libr classical-revival)))))
(R-30 premise)
(p-int libr (action (tell libr user (bel libr classical-revival))) effort-neg)
(R-12 premise)
(f-p-bel libr (bel user (bel libr classical-revival strong)))
(R-13
(auto-predict-bel-pos
alter-predict-bel-pos))
(f-p-bel libr (bel user classical-revival))
(bel libr (bel user *A* strong) strong)
(R-13
(auto-predict-bel-pos
alter-predict-bel-pos))
(f-p-bel libr (bel user *A*))
(not (bel libr (bel user (bel libr (not classical)))))
(p-int libr (action (adopt libr user (bel user (not classical))))
effort-neg)
(R-32 premise)
(not (bel libr (action (tell libr user (bel libr (not classical))))))
(p-int libr (bel user (bel libr (not classical))))
(R-30 premise)
(p-int libr (action (tell libr user (bel libr (not classical)))) effort-neg)
(R-12 premise)
(f-p-bel libr (bel user (bel libr (not classical) strong)))
(R-13
(auto-predict-bel-pos
alter-predict-bel-pos))
(R-9 premise)
(f-p-bel libr (bel user (not classical)))

A = (rule-inst "((p-bel ?A1 (pd classical-revival)) => (not (p-bel ?A1 (pd classical)))"
((p-bel reasoner (pd classical-revival)))
(p-bel reasoner (not (pd classical))))

Figure 8.8: *Libr* plans to inform *user* that (*pd classical-revival-arch*) is more appropriate.

1. *USER* (*John*) *:* (*tell user libr* (*bel user* (*status user ra*) *strong*))
2. *LIBR* (*Mary*) *:* (*tell libr user* (*bel libr* (*dt advanced-books*) *weak*))
3. *USER* (*John*) *:* (*tell user libr* (*bel user* (*not* (*dt advanced-books*)) *strong*))
   USER (*John*) *:* (*tell user libr* (*bel user* (*dt introductory-books*) *strong*))

The *user* agent is initialised with the following local attitudes in its database:

1. (*p-bel user* (*status user ra*) *spec*)
   User believes he is an ra (i.e. research assistant).
2. (*p-bel user* (*dt introductory-books*) *spec*)
   User believes that he wants introductory books.
3. (*p-bel user* (*not* (*dt advanced-books*)) *spec*)
   He also believes that he does not want advanced books.[6]
4. (*p-int user* (*bel libr* (*status user ra*)) *desire-pos*)
   He intends to share his belief that he is an ra with *libr*. This intention may have arisen through a request from *libr* since a sub-goal of *libr*'s information retrieval task is to determine *user*'s status.

User infers a `tell` → `adopt` pair for his intention, which leads to the first utterance:

(*tell user libr* (*bel user* (*status user ra*) *strong*))

Libr accepts that *user* is a research assistant and infers, by default (using rule D–2), that the *user* probably wants advanced material (for research purposes perhaps)

(*bel libr* (*dt advanced-books*) *weak*)

This is a *weak* belief since its only justification arises out of a *def* endorsed rule instance. *Libr* also comes to believe that the user should know whether or not he wants this document type. This is captured by *libr*'s local rule

(*p-bel libr* (*dt ?P*))
$\Rightarrow_{[def]}$ (*p-bel libr* (*exists !x* (*?P* (*not ?P*)) (*bel user !x*)))

with consequent instance

(*p-bel libr* (*exists !x* ((*dt advanced-books*) (*not* (*dt advanced-books*))) (*bel user !x*)))[7]

---

[6] Note that wanting introductory books does not necessarily preclude wanting advanced books.

[7] Although this is a *weak* belief and agents ask about their weakly held beliefs we suppress agents asking second order questions like: "Do you know whether you know that ...".

Libr plans to ask *user* whether he believes that advanced books are appropriate. The plan is shown in figures 8.9 and 8.10 and comprises *libr* telling *user* of her intention

(p-int libr (exists !x ((dt advanced-books) (not (dt advanced-books))))
(bel libr (bel user !x)))

user adopting this intention as his own, and *user* subsequently planning to satisfy this intention by intending *libr* to come to believe that *user* either believes (dt advanced-books) or believes (not (dt advanced-books)).

User receives *libr*'s message

(tell libr user (int libr (exists !x ((dt advanced-books) (not (dt advanced-books)))
(bel libr (bel user !x)))))

and adopts the intention

(p-int user (exists !x ((dt advanced-books) (not (dt advanced-books)))
(bel libr (bel user !x))))

This is *user*'s intention for *libr* to believe that the user believes that either *advanced-books* is appropriate or is not appropriate. Since he already believes (not (dt advanced-books)) he fires the following rule with *?P* bound to (not (dt advanced-books))

(p-int ?A1 (exists ?X ?Y (bel ?A1 (?ATT ?A2 ?P))))
(?ATT ?A1 ?P)
(binds ?X ?Y ?X ?P)
$\Rightarrow_{[desire-pos]}$ (intorbel (?ATT ?A2 ?P))

and infers the intention for *libr* to come to believe that *user* believes (not (dt advanced-books)).

(p-int user (bel libr (bel user (not (dt advanced-books)))))

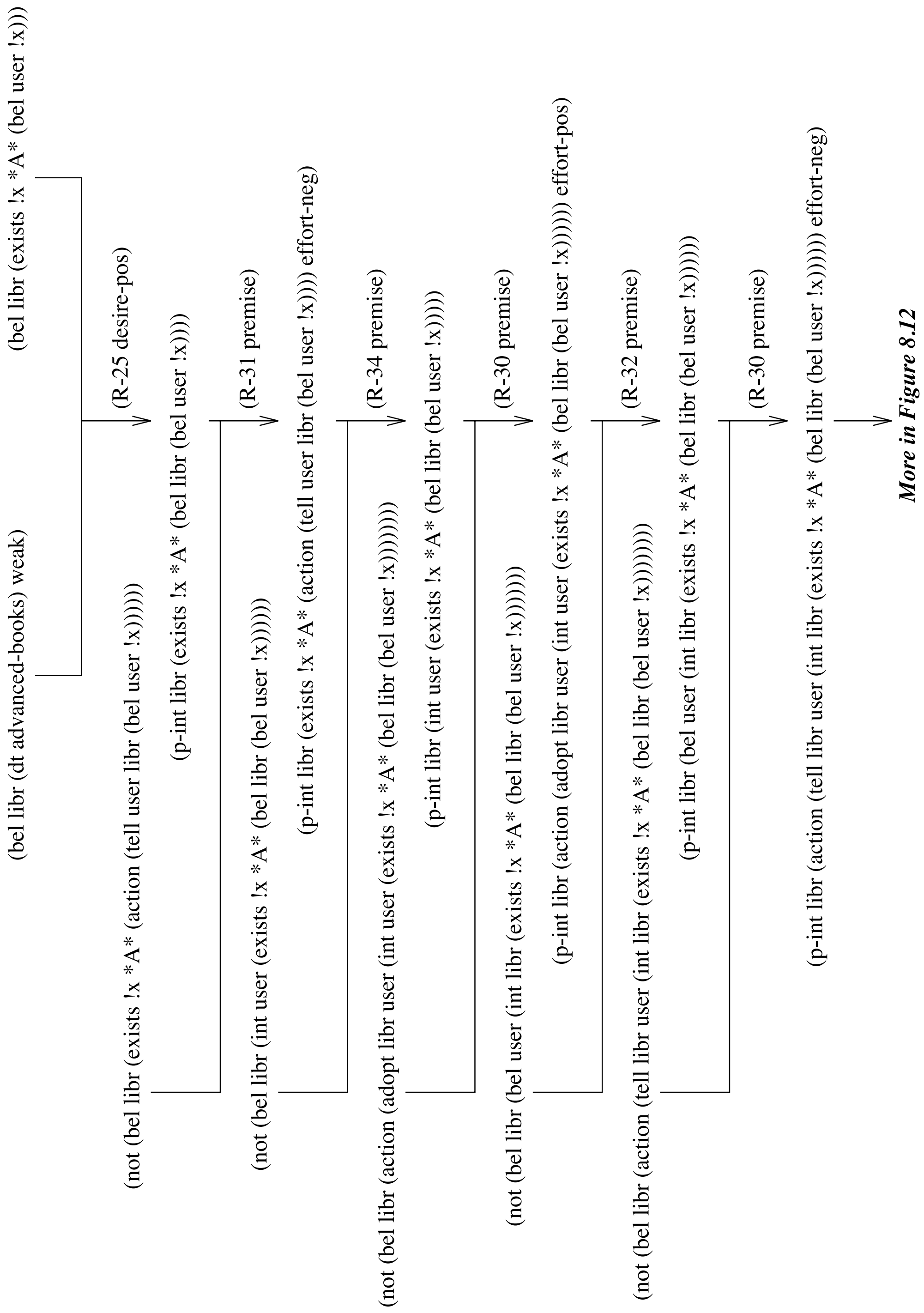
User's plan to satisfy this intention is shown in figure 8.11.

User outputs

(tell user libr (bel user (not (dt advanced-books)) strong))

and predicts that *libr* will eventually come to believe (not (dt advanced-books)). *User* does not wait for *libr*'s response, although all his intentions are now satisfied or predicted to be satisfied in the future. He searches down his inference stack and re-encounters his earlier belief

(bel user (int libr (exists !x ((dt advanced-books) (not (dt advanced-books)))
(bel libr (bel user !x)))))


(bel libr (dt advanced-books) weak)
(bel libr (exists !x *A* (bel user !x)))
(R-25 desire-pos)
(not (bel libr (exists !x *A* (action (tell user libr (bel user !x))))))
(p-int libr (exists !x *A* (bel libr (bel user !x))))
(R-31 premise)
(not (bel libr (int user (exists !x *A* (bel libr (bel user !x))))))
(p-int libr (exists !x *A* (action (tell user libr (bel user !x)))) effort-neg
(R-34 premise)
(not (bel libr (action (adopt libr user (int user (exists !x *A* (bel libr (bel user !x))))))))
(p-int libr (int user (exists !x *A* (bel libr (bel user !x)))))
(R-30 premise)
(not (bel libr (bel user (int libr (exists !x *A* (bel libr (bel user !x)))))))
(p-int libr (action (adopt libr user (int user (exists !x *A* (bel libr (bel user !x)))))) effort-pos
(R-32 premise)
(not (bel libr (action (tell libr user (int libr (exists !x *A* (bel libr (bel user !x))))))))
(p-int libr (bel user (int libr (exists !x *A* (bel libr (bel user !x))))))
(R-30 premise)
(p-int libr (action (tell libr user (int libr (exists !x *A* (bel libr (bel user !x)))))) effort-neg
More in Figure 8.12

A = ((dt advanced-books) (not (dt advanced-books)))

Figure 8.9: *Libr*'s plan to verify his default assumption (*dt advanced-books*): part 1

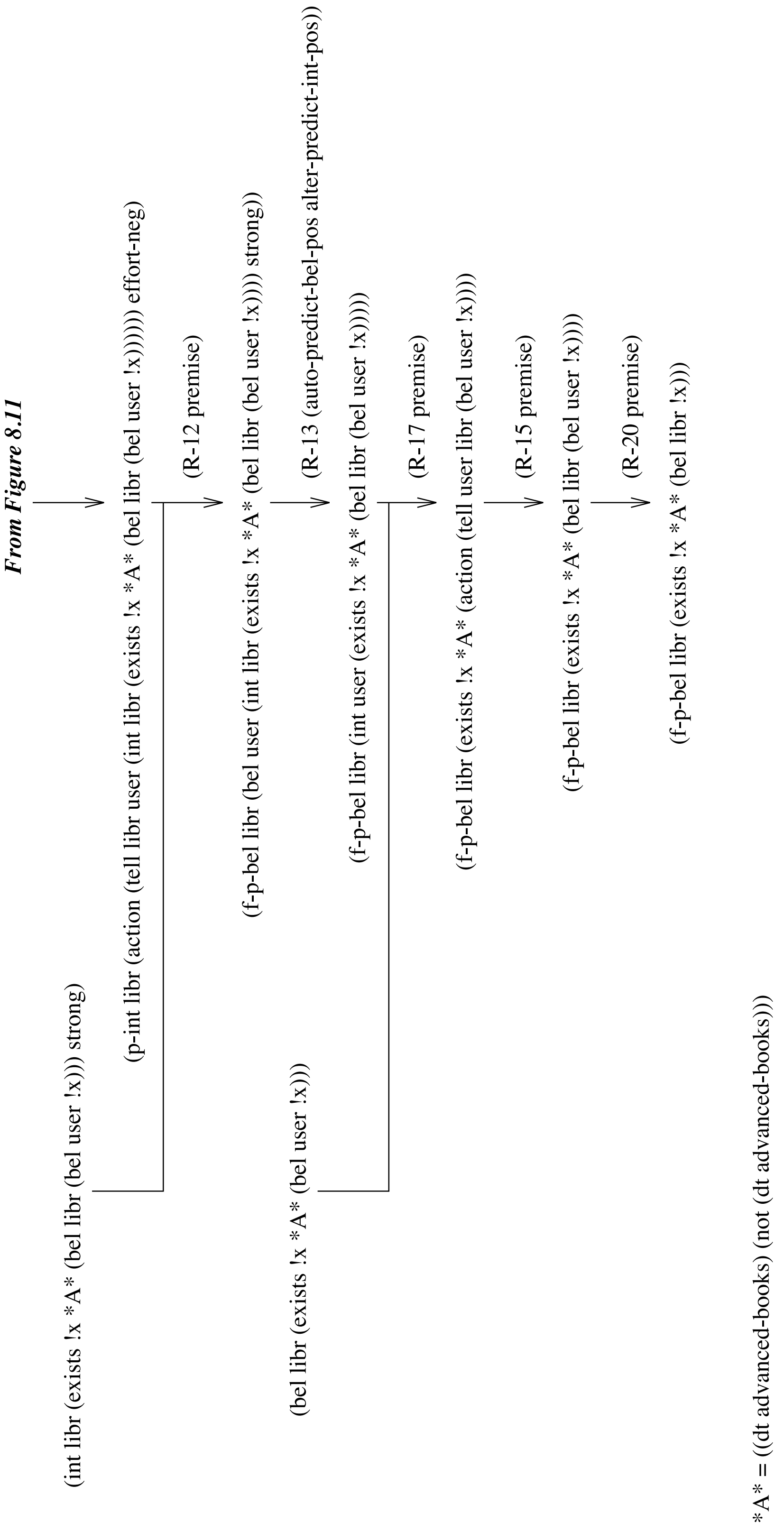

From Figure 8.11
(int libr (exists !x *A* (bel libr (bel user !x))) strong)
(p-int libr (action (tell libr user (int libr (exists !x *A* (bel libr (bel user !x)))))) effort-neg)
(R-12 premise)
(f-p-bel libr (bel user (int libr (exists !x *A* (bel libr (bel user !x)))) strong))
(R-13 (auto-predict-bel-pos alter-predict-int-pos))
(bel libr (exists !x *A* (bel user !x)))
(f-p-bel libr (int user (exists !x *A* (bel libr (bel user !x)))))
(R-17 premise)
(f-p-bel libr (exists !x *A* (action (tell user libr (bel user !x)))))
(R-15 premise)
(f-p-bel libr (exists !x *A* (bel libr (bel user !x))))
(R-20 premise)
(f-p-bel libr (exists !x *A* (bel libr !x)))
A = ((dt advanced-books) (not (dt advanced-books)))

Figure 8.10: *Libr*'s plan to verify his default assumption (*dt advanced-books*): part 2

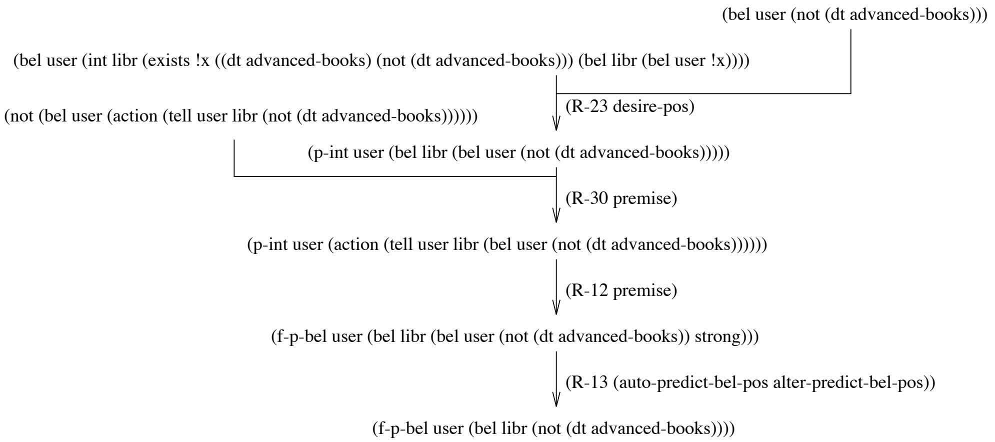

(bel user (not (dt advanced-books)))
(bel user (int libr (exists !x ((dt advanced-books) (not (dt advanced-books))) (bel libr (bel user !x))))
(not (bel user (action (tell user libr (not (dt advanced-books))))))
(R-23 desire-pos)
(p-int user (bel libr (bel user (not (dt advanced-books)))))
(R-30 premise)
(p-int user (action (tell user libr (bel user (not (dt advanced-books))))))
(R-12 premise)
(f-p-bel user (bel libr (bel user (not (dt advanced-books)) strong)))
(R-13 (auto-predict-bel-pos alter-predict-bel-pos))
(f-p-bel user (bel libr (not (dt advanced-books))))

Figure 8.11: *User*'s plan to tell *libr* that *advanced books* are inappropriate

This was inferred directly from *libr*'s utterance. *User* takes the initiative and assumes that *libr*'s query was motivated by the need to find a document type specification. Since *user* has just denied *advanced-books* he offers *introductory-books* as an alternative document type. The following rule

(*p-int ?A1* (*int ?A2* (*exists ?X* ((*?TYPE ?P*) (*not* (*?TYPE ?P*))) (*bel ?A1* (*?ATT ?A2 !x*))))) &
(*?ATT ?A1* (*not* (*?TYPE ?P*))) &
(*?ATT ?A1* (*?TYPE ?Q*))
$\Rightarrow_{[desire-pos]}$ (*intorbel* (*?ATT ?A2* (*?TYPE ?Q*)))

with *?TYPE* bound to *dt* generates the appropriate leading intention

(*p-int user* (*bel libr* (*bel user* (*dt introductory-books*))))

The plan to fulfill this intention is shown in figure 8.12. This is a single `tell` → `adopt` sequence again.

Libr accepts *user*'s belief (*not* (*dt advanced-books*)) as his own. The *2c-pos* endorsement from *libr*'s *strong* communicated commitment is sufficient to over come the *libr*'s *def* justification for (*dt advanced-books*). *Libr* also accepts *introductory books*, as an alternative document type specification.

Dialogue 7: Failed Prediction

In this example, one agent doubts the expertise of another agent. A `tell` only works when the recipient has no reason to doubt the competence of its informant. In this case the librarian thinks the user has made a mistake but she is not sure. The librarian offers a reason intended to convince the user of his mistake but this reason is not accepted. The user believes that Michelangelo was both an architect and an artist and the librarian that people are normally either artists or architects but not both:

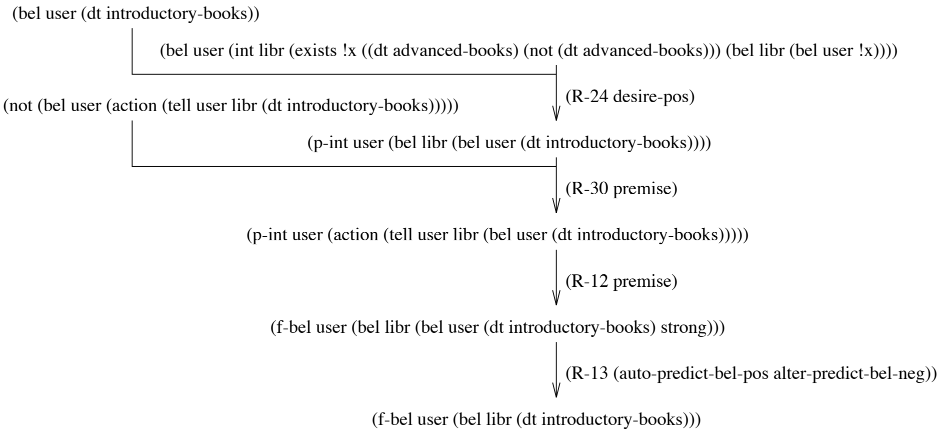

(bel user (dt introductory-books))
(bel user (int libr (exists !x ((dt advanced-books) (not (dt advanced-books))) (bel libr (bel user !x))))
(R-24 desire-pos)
(not (bel user (action (tell user libr (dt introductory-books)))))
(p-int user (bel libr (bel user (dt introductory-books))))
(R-30 premise)
(p-int user (action (tell user libr (bel user (dt introductory-books)))))
(R-12 premise)
(f-bel user (bel libr (bel user (dt introductory-books) strong)))
(R-13 (auto-predict-bel-pos alter-predict-bel-neg))
(f-bel user (bel libr (dt introductory-books)))

Figure 8.12: User suggests an alternative *document type*

1. `USER (John) :` `I am looking for books on the architecture of Michelangelo.`

2. `LIBR (Mary) :` `I thought Michelangelo was an artist.`

3. `USER (John) :` `He was also an architect.`

The following is the machine generated dialogue:

1. *USER (John) :*
(tell user libr (bel user (pd (architect michelangelo)) strong))

2. *LIBR (Mary) :*
(tell libr user (bel libr (not (class michelangelo architect)) weak))

LIBR (Mary) :
(tell libr user
(bel libr (rule-inst
"(p-bel ?A1 (class michelangelo artist))
⇒ (p-bel ?A1 (not (class michelangelo architect)))"
((p-bel reasoner (class michelangelo artist)))
(p-bel reasoner (not (class michelangelo architect))))
weak))

3. *USER (John) :*
(tell user libr (bel user (class michelangelo architect) strong))

USER (John) :
(tell user libr
(bel user (not (rule-inst
"(p-bel ?A1 (class michelangelo artist))
⇒ (p-bel ?A1 (not (class michelangelo architect)))"

((*p-bel reasoner* (*class michelangelo artist*)))
(*p-bel reasoner* (*not* (*class michelangelo architect*)))))
strong))

Libr is initialised with the following attitudes:

1. (*p-bel libr* (*rule-inst*
   “(*p-bel ?A1* (*class michelangelo artist*))
   ⇒ (*p-bel ?A1* (*not* (*class michelangelo architect*)))”
   ((*p-bel reasoner* (*class michelangelo artist*)))
   (*p-bel reasoner* (*not* (*class michelangelo architect*)))))
   def)

   She believes that if Michelangelo was an artist then, by default, he could not have been an architect.

2. (*p-bel libr* (*class michelangelo artist*) *spec*)
   She believes that Michelangelo was an artist.

3. (*p-bel libr* (*bel user* (*class michelangelo artist*)) *def*)
   She believes that, by default, *user* believes that Michelangelo was an artist.

4. (*p-bel libr* (*bel user* (*rule-inst*
   “(*p-bel ?A1* (*class michelangelo artist*))
   ⇒ (*p-bel ?A1* (*not* (*class michelangelo architect*)))”
   ((*p-bel reasoner* (*class michelangelo artist*)))
   (*p-bel reasoner* (*not* (*class michelangelo architect*))))))
   def)

   She believes, by default, that the user believes that if Michelangelo was an artist then he could not have been an architect.

User is initialised with the following attitudes:

1. (*p-bel user* (*class michelangelo architect*) *spec*)
   He believes Michelangelo was an architect.

2. (*p-bel user* (*class michelangelo artist*) *spec*)
   He also believes michelangelo was an artist.

3. (*p-bel user* (*pdh michelangelo architect*) *spec*)
   He believes that ‘*Michelangelo the architect*’ describes his search query.

4. (*p-bel user* (*not* (*rule-inst*
   “(*p-bel ?A1* (*class michelangelo artist*))
   ⇒ (*p-bel ?A1* (*not* (*class michelangelo architect*)))”
   ((*p-bel reasoner* (*class michelangelo artist*)))
   (*p-bel reasoner* (*not* (*class michelangelo architect*))))))

   He does not believe that if Michelangelo was an artist then he could not have been an architect.

5. (*p-int user* (*bel libr* (*bel user* (*pdh michelangelo architect*))) *desire-pos*)
   He intends to inform *libr* of his belief that Michelangelo the architect is a good problem descriptor for his needs.

Initially, both agents contemplate a common rule that Michelangelo cannot be an artist and an architect:

(*p-bel user* (*class michelangelo artist*))
$\Rightarrow_{[def]}$ (*p-bel user* (*not* (*class michelangelo architect*)))

Both agents fire this rule but *user*, with *spec* endorsements on his beliefs that Michelangelo was both an artist and an architect, does not come to believe the rule instance. This creates a belief conflict situation between the agents which, as we will see, will motivate part of the dialogue.

User has the intention to notify *libr* that he is interested in '*Michelangelo the architect*' [8] and he outputs the following utterance:

(*tell user libr* (*bel user* (*pdh michelangelo architect*)) *strong*))

Libr draws the inference that *user* is strongly committed to his communicated belief

(*p-bel libr* (*bel user* (*pdh michelangelo architect*) *strong*) *premise*)

Implicit in this belief is the fact that *user* is strongly committed to the belief that Michelangelo was an architect. This is captured by the following rule:

(*p-bel lib* (*bel user* (*pdh ?TERM1 ?TERM2*)))
$\Rightarrow_{[premise]}$ (*p-bel lib* (*bel user* (*class ?TERM1 ?TERM2*) *strong*))

Libr infers *user*'s belief that Michelangelo was an architect (i.e. (*class michelangelo architect*)) and she believes that *user* is strongly committed to this belief. Generally, a *strong* communicated commitment is converted to a *2c-pos* endorsement reason for the hearer to believe the communicated belief. However, this conversion does not take into account notions of expertise. A non-expert can be strongly committed to a belief, but can be unaware of all relevant information and lack important reasons for disbelieving the belief. The hearer is less convinced by the commitment of a non-expert speaker, and in this case should convert the communicated commitment *strong* to a *2c-neg* reason for adopting the communicated belief itself. In our example the librarian doubts the competence of the user and she does not attach weight to the the user's assertion that Michelangelo is an architect. Thus, in the simulation *libr* agent gains a *2c-neg* reason for adopting (*class michelangelo architect*) from *user*. *Libr* already has a *def* reason for believing (*not* (*class michelangelo architect*)) and, through minimal change, she retains this belief.

So *libr* now has a weak belief in (*not* (*class michelangelo architect*)) and the rule instance supporting this belief:

---

[8] His plan is similar to 8.1.

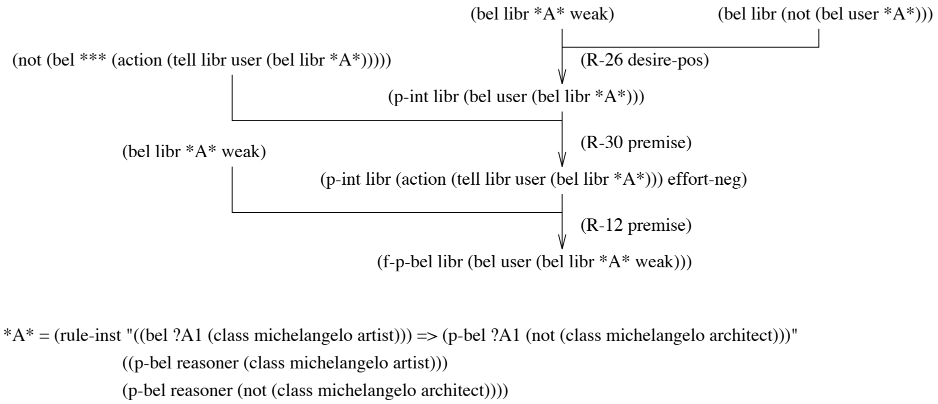

(bel libr *A* weak)
(bel libr (not (bel user *A*)))
(not (bel *** (action (tell libr user (bel libr *A*)))))
(R-26 desire-pos)
(p-int libr (bel user (bel libr *A*)))
(R-30 premise)
(bel libr *A* weak)
(p-int libr (action (tell libr user (bel libr *A*))) effort-neg)
(R-12 premise)
(f-p-bel libr (bel user (bel libr *A* weak)))
A = (rule-inst "((bel ?A1 (class michelangelo artist))) => (p-bel ?A1 (not (class michelangelo architect)))"
((p-bel reasoner (class michelangelo artist)))
(p-bel reasoner (not (class michelangelo architect))))

Figure 8.13: *Libr* agent's plan to investigate his *weak* beliefs

$$
\begin{aligned}
&(\textit{p-bel user } (\textit{rule-inst } \text{“}(\textit{p-bel ?A1 } (\textit{class michelangelo artist})) \Rightarrow \\
&\qquad\qquad (\textit{p-bel ?A1 } (\textit{not } (\textit{class michelangelo architect})))\text{”} \\
&\qquad ((\textit{p-bel reasoner } (\textit{class michelangelo artist}))) \\
&\qquad (\textit{p-bel reasoner } (\textit{not } (\textit{class michelangelo architect})))))
\end{aligned}
$$

Libr plans to verify these weakly held beliefs by informing *user* of her commitment to these, with the expectation that *user* will attempt to resolve any disagreements. Her plan for the rule instances this is shown in figure 8.13.

It is interesting to note that those weakly held beliefs relevant to

$$(\textit{p-bel libr } (\textit{pdh michelangelo architect}))$$

which, in turn, is relevant the overall goal of finding a retrieval strategy, are the two weakly held beliefs discussed above (see the table below). In chapter 9 we discuss a mechanism which allows an agent to focus its ATMS. We propose that agents focus on weakly held attitudes that are relevant to the overall goal of the agent.

Commitment	*Libr's beliefs*
strong	(*pdh michelangelo architect*)
strong	(*rule-inst* "((*p-bel ?A1* (*bel ?A2 ?P ?C*))) ⇒ (*p-bel ?A1 ?P*)" ((*bel reasoner* (*pdh michelangelo architect*) *strong*)) (*bel reasoner* (*pdh michelangelo architect*)))
strong	(*bel user* (*class michelangelo architect*))
strong	(*bel user* (*class michelangelo architect*) *strong*)
strong	(*rule-inst* "((*p-bel ?A1* (*bel ?A2* (*pdh ?TERM1 ?TERM2*)))) ⇒ (*p-bel ?A1* (*bel ?A2* (*class ?TERM1 ?TERM2*) *strong*))" ((*p-bel reasoner* (*bel user* (*pdh michelangelo architect*)))) (*p-bel reasoner* (*bel user* (*class michelangelo architect*) *strong*))
strong	(*bel user* (*class michelangelo artist*))
weak	(*not* (*class michelangelo architect*))
weak	(*rule-inst* "((*p-bel ?A1* (*class michelangelo artist*)) ⇒ (*p-bel ?A1* (*not* (*class michelangelo architect*)))" ((*p-bel reasoner* (*class michelangelo artist*))) (*p-bel reasoner* (*not* (*class michelangelo architect*))))
strong	(*class michelangelo artist*)

Libr chose to verify her weak beliefs because it was the next activity to be inferred from her inference stack. The inference stack mechanism dictates that agents focus their attention on the most recently derived attitudes and infer in a depth first manner. However, this can cause important inferences to be missed, since more than one inference from an proposition can be relevant. In this case a breadth-first inference search would be more appropriate. As an alternative to the inference stack, our focussing mechanism would encourage agents to work with weakly held beliefs relevant to their overall goals and to infer accordingly. In our dialogue example the rule instance is relevant to the overall task of information retrieval and is also weakly held. Thus with our proposed focus mechanism, *libr* would still plan for and generate the above utterance.

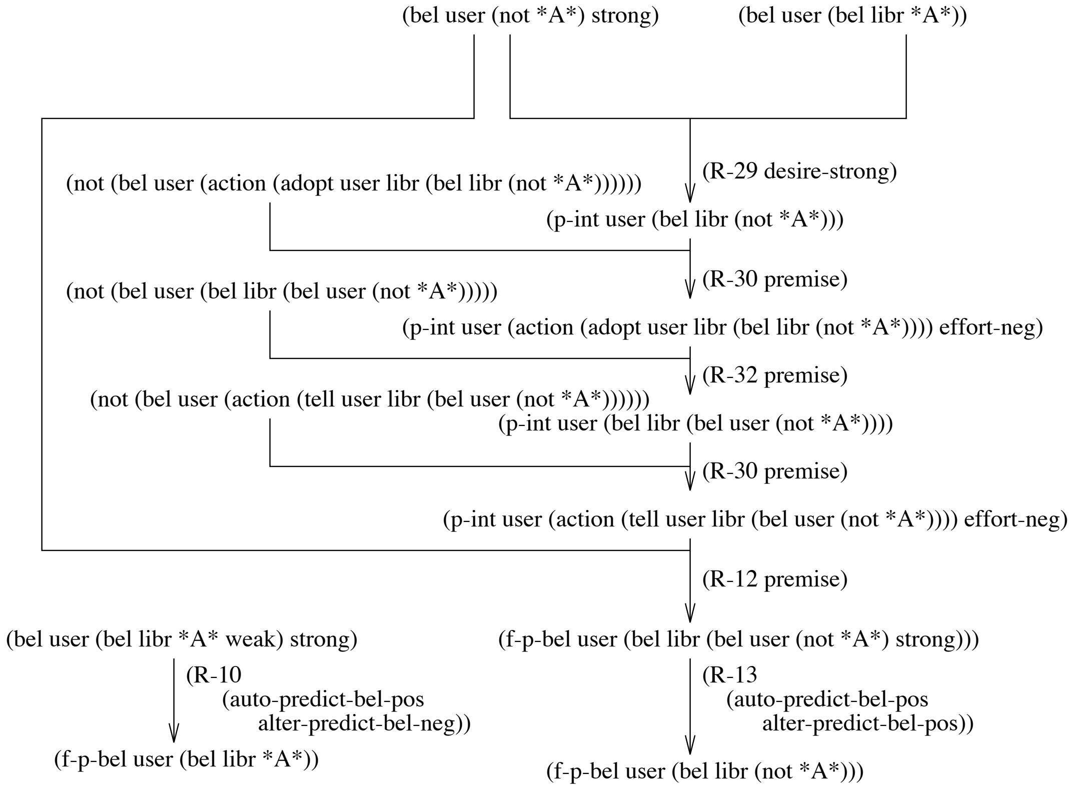
User recognises that *libr* is weakly committed to the rule instance and considers revising his own beliefs with the extra *2c-neg* reason *libr* has given him for adopting each. This is insufficient to change *user*'s mind and he comes to realise that they disagree about the rule instance and about (*class michelangelo architect*). Since he is strongly committed to both his beliefs and he also believes that *libr* is weakly committed to hers, he predicts that simply informing *libr* of his commitments

```
(tell user libr (bel user (class michelangelo architect) strong))
(tell user libr
      (bel user (not (rule-inst "(p-bel reasoner (class michelangelo artist)) ⇒
                                      (p-bel reasoner (not (class michelangelo architect)))"
                                 ((p-bel reasoner (class michelangelo artist)))
                                 (p-bel reasoner (not (class michelangelo architect)))))
      strong))
```

will be sufficient to change *libr*'s mind (see Figure 8.14).[9]

---

[9] Note that the user's reply


(bel user (not *A*) strong)
(bel user (bel libr *A*))
(R-29 desire-strong)
(not (bel user (action (adopt user libr (bel libr (not *A*))))))
(p-int user (bel libr (not *A*)))
(R-30 premise)
(not (bel user (bel libr (bel user (not *A*)))))
(p-int user (action (adopt user libr (bel libr (not *A*)))) effort-neg)
(R-32 premise)
(not (bel user (action (tell user libr (bel user (not *A*))))))
(p-int user (bel libr (bel user (not *A*))))
(R-30 premise)
(p-int user (action (tell user libr (bel user (not *A*)))) effort-neg)
(R-12 premise)
(bel user (bel libr *A* weak) strong)
(R-10 (auto-predict-bel-pos alter-predict-bel-neg))
(f-p-bel user (bel libr *A*))
(f-p-bel user (bel libr (bel user (not *A*) strong)))
(R-13 (auto-predict-bel-pos alter-predict-bel-pos))
(f-p-bel user (bel libr (not *A*)))

A = (rule-inst "((bel ?A1 (class michelangelo artist))) => (p-bel ?A1 (not (class michelangelo architect)))"
((p-bel reasoner (class michelangelo artist)))
(p-bel reasoner (not (class michelangelo architect))))

Figure 8.14: *User* agent's plan to resolve conflict in rule instance

8.3 Results of the Tests

We have identified some causes of belief revision in User-Librarian dialogues and have shown that these dialogue phenomena can be modelled using ABR. There are, however, some cases we cannot model. These include the generation of the affirmative replies such as "OK" in

Dialogue 1: INFORM

```
USER (John) : I am looking for information on churches
LIBR (Mary) : OK
```

and denials such as "I don't know" in

Dialogue 4: FAILED QUESTION/ANSWER

```
LIBR (Mary) : Any particular architects?
USER (John) : I don't know.
```

The librarian uses "OK" to inform the user that she understands what "churches" means and has not found a problem incorporating this into the problem description. That is, the librarian has realised that she has drawn the necessary inferences and that no further inferences should cause problems. Our agents do not introspect about their own action cycle, and they do not reason about what inferences can be drawn in the future, so they cannot reason as to when an affirmative reply is appropriate. In the second example, "I don't know" tells us that the user is aware that he has no architects in mind. Our agents plan from propositions and the lack of information would thus have to be represented explicitly. This is akin to the frame problem and also requires introspection.

$$(tell\ user\ libr\ (bel\ user\ (class\ michelangelo\ architect)\ strong))$$

gives *libr* more reason to believe that *user* believes (*class michelangelo architect*), but does not give *libr* more reason to believe (*class michelangelo architect*) herself since she already believed that *user* is strongly committed to his belief.

Chapter 9

Problems and Future Work

In this chapter we discuss some of the problems with the theory and implementation of belief and intention revision presented in the previous chapters and identify areas for future work. These are the computational complexity problem that arises in using an ATMS as the mechanism for implementing belief revision; the problem that arises with increased coherence (*mc*); the problems of prediction in general and more specifically of handling predicted intentions; the problem of communicating commitment; and that of dialogue management and the need for an appropriate treatment of focus.

9.1 The Problem of Computational Complexity

Perhaps the most important problem is the computational complexity of belief revision. At first sight, this may seem surprising, since the implementation employs techniques similar to those used by de Kleer in his ATMS which has formed the basis of a number of successful applications. de Kleer (1986a) states: "the architecture of the ATMS is such that it is practical to use it even when n is very large, e.g. 1000. The ATMS is then exploring a space of size 2^{1000}." However a superficial comparison between the task of the ATMS and the belief revision system is misleading. On closer inspection, it turns out that a number of simplifying assumptions which are valid for de Kleer's problems do not hold for belief revision.

The theory of belief revision states that when an agent learns something which is inconsistent with its current beliefs, it generates a belief set for all possible ways of removing the inconsistency (Galliers 1992). Which of these set(s) is preferred depends on the properties of the sets: how endorsed they are; their coherence; and how close they are to the agent's current beliefs.

The problem is that there are typically a large number of sets. The number of ways an agent can revise their beliefs depends on the number of inconsistencies in their beliefs. We can identify three possible sources of inconsistency in an agent's beliefs: closure of the agent's belief set under negation; closure of the agent's intention set under negation and indifference; and derived inconsistencies between the agent's beliefs or intentions or both. The first two categories are in a sense 'innate' in that they are enforced by the logic of beliefs and intentions, and they place an upper bound on the number of belief sets the agent must consider.

In relation to the third category, there are four possible belief attitudes an agent can hold towards a proposition: $(p\text{-}bel\, x\, p)$, $(p\text{-}bel\, x\, \neg p)$, $\neg(p\text{-}bel\, x\, p)$ and $\neg(p\text{-}bel\, x\, \neg p)$. The inconsistencies between these attitudes mean that there are only three consistent or *stable* belief states that the agent can be in:[1]

1. the agent believes that p: $(bel\, x\, p) \wedge \neg(bel\, x\, \neg p)$;
2. the agent believes that $\neg p$: $(bel\, x\, \neg p) \wedge \neg(bel\, x\, p)$; or
3. the agent is uncertain about p, where the uncertainty is understood to be due to lack of knowledge or reasons for believing in either p or $\neg p$: $\neg(bel\, x\, p) \wedge \neg(bel\, x\, \neg p)$.

As noted in section 5.2.1, case (3) would normally never arise. We can therefore represent the agent's state using a single ATMS assumption. For example, given $(bel\, x\, p)$ we can derive $\neg(bel\, x\, \neg p)$ and vice versa. If there are b beliefs and no inconsistencies other than p is inconsistent with $\neg p$, then we need 2^b belief sets to model the 3^b possible stable belief states (since case (3) never arises) and the 7^b unstable states that can be constructed from these.

As with beliefs, there are three possible consistent intention states that an agent might be in:

1. the agent intends that p: $(int\, x\, p) \wedge \neg(int\, x\, \neg p)$;
2. the agent intends that $\neg p$: $(int\, x\, \neg p) \wedge \neg(int\, x\, p)$; or
3. the agent is indifferent to p—it neither intends that p nor intends that $\neg p$, where the indifference is understood to be due to lack of reasons (interpreted as goals or desires rather than knowledge) for intending either that p or that $\neg p$: $\neg(int\, x\, p) \wedge \neg(int\, x\, \neg p)$.

Unlike beliefs, we need to represent all three consistent intention states explicitly. If there are i intentions, we need 3^i intention sets to represent the 3^i possible consistent intention states and the 7^i unstable states that can be constructed from these. These intention sets include the belief sets and therefore give an upper bound on the number of belief/intention sets an agent must consider. Adding additional relations to the belief/intention sets, for example that belief in p is inconsistent with belief in q or that an intention p implies an intention q, can only reduce the number of belief/intention sets which must be considered, by ruling out otherwise acceptable candidates.

The efficiency of the ATMS is exponential in the number of assumptions. A task with n assumptions has 2^n environments and at most 2^n contexts. In our case, to encode the possible belief and intention states requires $2b + 3i$ assumptions. While there are at most 2^{2b} consistent maximal environments

---

[1] The inconsistent or unstable states described in chapter 5 only arise when the agent can't decide between these states.

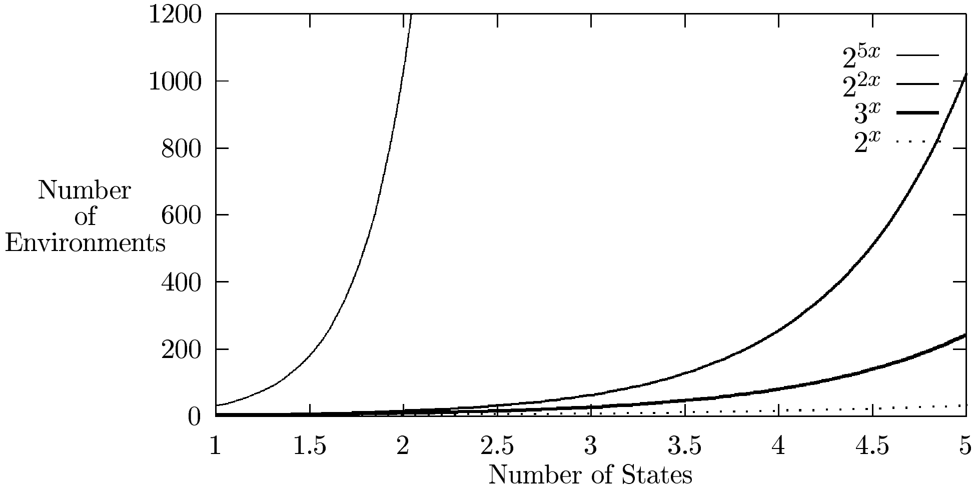


Figure 9.1: States and Environments

(interpretations), the ATMS must search a space of 2^n environments to find them, where n is the number of assumptions required to encode the beliefs and intentions. In the worst case, where all propositions are both believed and intended (i.e. where $b = i$), this gives 2^{5b} environments which must be considered. The lower bound is given by 2^{2b} since all intended propositions must also be either believed or disbelieved. Figure 9.1 shows the upper and lower bounds on the number of environments (2^{5x} and 2^{2x} respectively), the number of intention sets (3^x) and the number of belief sets (2^x) plotted against the number of propositions (states).

In a conventional ATMS-based application, closure is represented implicitly. In a diagnosis problem, for example, an assumption might represent that fact that a component was faulty without there being a corresponding assumption that the component is working properly. If the assumption is in an interpretation computed by the system, the component is faulty, if it is not then the fault lies elsewhere.[2] However in our case, we must explicitly represent both p and $\neg p$, both because they may have different endorsements, and because they may justify other beliefs. In a resource bounded agent, what is derivable from p is not the complement of what is derivable from $\neg p$.

Even so, this doesn't seem too bad. Assuming the number of intentions is small relative to the number of beliefs, de Kleer's assertion seems to imply that we should be able to cope with about 500 beliefs.[3] However this ignores

---

[2] de Kleer also employs a number of heuristics, such as the 'single fault' assumption, to reduce the number of interpretations that must be considered.

[3] Actually we are slightly worse off than this, since we need a node (assumption) for every rule instance as well, whereas de Kleer does not use assumptions to record derived information or rule instances. If each rule has on average 2 antecedents, about one quarter of the database would consist of rule instances. This assumption may also not be true after focussing, since the focus is supposed to contain only those beliefs relevant to the current intention(s).

the difficulty of *finding* the interpretations. The problem of interpretation construction is, in general, NP-complete (de Kleer 1986b). de Kleer argues that interpretation construction can be made more tractable by constructing *minimal interpretations* which do not contain any 'worthless' assumptions. However, in the ICM, all interpretations are minimal, as every datum is represented as an assumption.[4]

Intuitively we can understand the task of interpretation construction as identifying the boundary between the consistent and inconsistent environments in the environment lattice. If the boundary lies near the top or the bottom of the lattice or is extremely convoluted, the middle of the lattice can be ignored. de Kleer (1986b) describes two basic approaches to interpretation construction which take advantage of this fact. The first algorithm starts at the bottom of the lattice searching upwards adding as many assumptions as possible until the environments become inconsistent. The second algorithm starts at the top searching downwards to consistency removing as few assumptions as possible until environments become consistent. Both algorithms find the consistent environments with no consistent supersets, i.e. the interpretations. The first algorithm is best for situations where there are many nogoods (i.e. the boundary is near the bottom of the lattice) and the second algorithm is best in situations where there are relatively few nogoods (i.e. the boundary is near the top of the lattice).

Unfortunately, in the ICM, the number of beliefs and inconsistencies is (very approximately) equal and the boundary between the consistent and inconsistent environments is in the middle of the lattice. This is a consequence of the closure of beliefs and intentions under negation. Every belief is inconsistent with at least one other belief, but typically not many other beliefs. (If every belief was inconsistent with nearly every other belief, the interpretations would be near the bottom of the lattice and we could find them; conversely if nearly all the beliefs were consistent, the interpretations would be near the top of the lattice and again we could find them.) To quote de Kleer: "If the [boundary] between consistent and inconsistent environments lies along the middle of the lattice, then *both algorithms will fail for even small problems*" de Kleer (1986b, p 191) (emphasis added).

If there are n assumptions, there are $n+1$ layers in the lattice, and the interpretations will lie around layer $n/2$. The layers of the lattice enumerate the ways of choosing k assumptions from the set of n assumptions. In the first layer we choose zero assumptions, in the next layer we choose one assumption, then two assumptions, etc. The critical layer is the one in which the environments are of length $n/2$. When an environment contains more than $n/2$ assumptions, it must contain both p and $\neg p$ for some p. Since the belief sets are closed under negation, the interpretations must lie in the bottom half of the lattice. However, unless all the beliefs are pairwise inconsistent, there must be at least one interpretation in the $n/2$ layer. Note that the presence of intentions simply

---

[4] Provan (1988), (1990) argues that the construction of minimal interpretations is in fact NP-hard. He also proves that label construction is exponential for almost all problems. However in the ICM, label construction is dominated by interpretation construction (see below), so this result is of less importance.

No. beliefs	No. environments	Crtical Layer	No. Interpretations
10	1048576	184756	1024
20	1.1×10^{12}	1.37×10^{11}	1048576
30	1.15×10^{18}	1.18×10^{17}	1.07×10^{9}

Table 9.1: Number of interpretations as a function of the number of beliefs

increases the number of inconsistent environments above the $n/3$ layer, but as long as there is at least one non-intended belief which is not pairwise inconsistent with an intended belief, there must be at least two consistent environments in the $n/2$ layer. We can choose a subset of $n/2$ assumptions in $\binom{n}{n/2}$ ways. These numbers grow very quicky. Table 9.1 shows the total number of environments, the number of environments in the 'critical layer' and the number of interpretations for 10, 20 and 30 beliefs.

Of course we don't have to generate all the 2^{2b+3i} environemts below the critical layer. We can generate the consistent environments directly from the nogoods.[5] The basic algorithm consists of three steps (de Kleer 1986a):

1. generate the set of all possible *candidates* consisting of one member of each nogood;
2. form the complement of each candidate with respect to the set of assumptions to generate the set of possible interpretations; and
3. prune the set of possible interpretations to remove subsets.

Step 2 is linear in the number of candidates and should be fast. Step 3 in principle requires testing every set against all the others; however the sets are ordered by length and the number of sets gets smaller as subsets are weeded out. Step 1 is a function of the number of inconsistencies and nogoods and is the expensive one. If we have 10 beliefs and hence 10 nogoods each containing two assumptions we would generate 2^{10} candidates. In general, we have $\prod_{i=1}^{k} l_i$ where l_i is the number of assumptions appearing in the ith nogood.

Lastly, we do all this at every inference cycle, whereas de Kleer does interpretation construction once, at the end of the problem solving process. While it may be acceptable to wait say, an hour, for the results of a diagnosis, spending an hour on each inference is not feasible in a dialogue system.

9.1.1 Definite Beliefs

This problem was anticipated by Galliers and Reichgelt (1990) who proposed a solution based on *definite beliefs*.[6] Galliers and Reichgelt argue that while some progress could be made with clever programming techniques, a general

---

[5] Since the interpretations are not guaranteed to be near the top or the bottom of the lattice, de Kleer's 'efficient' algorithms won't work.

[6] In the draft IJCAI paper, these are called *core* beliefs.

solution requires some limit on the number of belief sets.[7] They suggest that we should only consider belief sets that contain a core set of beliefs as a subset, arguing that "an agent has a number of central beliefs that they will never give up, or only in extreme circumstances."

From a belief revision point of view, such definite beliefs are essentially 'free'—they can't change their belief status and hence incur no belief revision overhead. Each belief we assume to be definite at least halves the number of interpretations. To limit the belief revision overhead, this set should be as large as possible, and to be maximally effective, each definite belief should be inconsistent with as many non-definite beliefs as possible to limit the number of interpretations. At the same time, the intersection of their 'extensions' (the non-definite beliefs with which they are consistent) must be non-empty, otherwise they rule out all interpretations.

While it is undoubtedly true that not all of a reasoning agent's beliefs need to be revisable, this cannot in itself constitute a general solution to the belief revision problem. In effect, it amounts to saying we don't need to revise any beliefs we can assume to be non-revisable. All such beliefs will, by definition, be members of all belief sets, and therefore can't be used to choose between possible belief sets. While definite beliefs implicitly impose an ordering on the possible alternative revisions, they do not reduce the number of revisions. Beliefs which must be revisable include intentions, defaults, and much of the substantive content of a dialogue. The status of these beliefs cannot be explicitly or implicitly determined by definite beliefs: it must be possible for an intention to be abandoned or achieved; a default to be false etc. Otherwise there could be no negotiation.[8] Our results therefore still hold for revisable beliefs.

Though crude, the complexity results presented above give us a worst case upper bound on the size of the focus set. Depending on the number of inconsistencies, we have perhaps 20-30 assumptions available to represent beliefs and intentions. this may not seem like very much. However the situation is not as bad as the worst case analysis suggests. In the next section we describe anumber of techniques which improve the performance of the system for typical cases.

9.1.2 Focus Set and Focussing Algorithm

One way of overcoming the problem of computational complexity which has been widely studied in the ATMS literature is to use a *focus set* (Forbus and de Kleer 1988, Dressler and Farquhar 1990). The function of the focus mecha-

---

[7] Galliers and Reichgelt explicitly reject the approach advocated by Gärdenfors (1988) and others of limiting possible revisions to maximal consistent sets as incompatible with the basic principles underlying the theory.

[8] Note that the proposal by Galliers and Reichgelt to equate the intentions used in the definition of increased coherence with definite beliefs won't work: even if we are certain we want to achieve something, we have to allow for having achieved it. When an intention is achieved, the intended state goes from being disbelieved to being believed and the intention itself becomes disintended. This change is also propagated to other, higher-level, states and intentions which depend on the intended state. Intentions are also dropped when they require too much effort or the agent predicts that an intended action will not achieve its objective.

nism is to restrict inference to the current problem by focussing on a subset of the available information. One of the problems of focus is what to focus on. In chapter 8 we noted that human dialogue can be modelled by allowing the agents to focus on weakly supported attitudes relevant to the overall task of the agent. We therefore propose a focussing mechanism which allows an agent to reason with a limited set of those weakly held beliefs that are relevant to its goals. Attitude nodes can be moved into focus (called 'recall') and can be moved out of focus (called 'focussing out'). The agent focusses on a set of attitudes called the 'prime set' and on those attitudes which are uncertain or weakly held and relevant to these (Cawsey et al. 1993).[9]

The focus set consists of a subset of the revisable beliefs and is distinct from the agent's database. When the agent is first created all attitudes known to the system are in focus (the focus set is a copy of the agent's initial database), and the prime set contains the initial intentions of the system (e.g. to find a retrieval strategy). Those beliefs not relevant to the initial intentions are then focussed out to produce focus set F_0. The focussing algorithm (described below) is invoked after each inference on the current focus space (i.e. once every action cycle), to generate the successive focus sets.

The focussing algorithm has three stages:

1. Identify the prime set. The prime set for focus $t+1$ consists of those attitudes to which the agent is weakly committed or about which it is uncertain, that are relevant to any member of the prime set for focus t.
2. RECALL: recall potential attitudes relevant to the attitudes in the prime set. Nodes are copied from the database to the focus set.
3. FOCUS: focus out those attitudes not relevant to any member of the prime set. These attitude nodes are deleted from the focus set.

Nodes in focus undergo label adjustment so that the environments contain only those attitudes in focus. After attitudes are recalled or focussed out, the system

---

[9] Note that the definition of focus given in (Cawsey et al. 1993) is unsatisfactory. It gives the wrong results in cases where the current intention (or whatever we are focussing on) is not the sole justification for another belief or intention. Suppose we have

$$\begin{aligned} (\mathit{int}\, x\, p) \wedge (\mathit{bel}\, x\, \neg q) &\supset (\mathit{int}\, x\, q) \\ (\mathit{int}\, x\, r) \wedge (\mathit{bel}\, x\, \neg q) &\supset (\mathit{int}\, x\, q) \end{aligned}$$

and $(\mathit{int}\, x\, p)$ is the current intention. Then, using the definition of relevance given in Cawsey et al. (1993), $(\mathit{int}\, x\, q)$ is not relevant to $(\mathit{int}\, x\, p)$, since disbelief in $(\mathit{int}\, x\, p)$ does not lead to the agent disbelieving $(\mathit{int}\, x\, q)$. However, $(\mathit{int}\, x\, q)$ should be relevant to $(\mathit{int}\, x\, p)$, as disbelieving $(\mathit{int}\, x\, q)$ does entail disbelieving $(\mathit{int}\, x\, p)$. The current implementation therefore uses an alternative definition of focus, which includes the weakest consequences as well as the weakest antecedents.

constructs the focus set by copying from the database those nodes corresponding to attitudes in focus and adjusts the labels in the copies, as follows. For each label

1. delete all attitudes which were believe/intended when last in focus; and
2. delete all environments which contain an attitude which was disbelieved/disintended when last in focus.

Inferences are only drawn from those attitudes in focus. Any new attitudes (i.e. the consequents and the rule instances) are added as hypotheses to both the focus set and the database. Next, the justifications are added to the justification lists of the consequents in the focus set and the database. Then the ATMS label propagation algorithm is run both on the database and the focus set.

Attitude sets are constructed from those ATMS nodes in the focus set only. If the focus set is inconsistent (i.e. if false is justified in the focus set) then more than one attitude set is created. Hence, the nogoods are restricted to those beliefs currently in focus.

However one remaining problem is that of recall. Stage 2 of the focussing algorithm requires that the system identifies potentially relevant attitudes to recall. This means that the system not only has to recall attitudes which are currently relevant through their labels but also those which might possibly become relevant through a future inference. Note also that all approaches to focus based on relevance have the disadvantage that they are based on labels rather than logical implication: only inferences which have already been made can be in focus. This problem has yet to be solved.

In general, the smaller the focus set, the fewer belief sets there will be and the faster the system will run. The effective size of the focus set is therefore the number of revisable beliefs it contains. However we cannot make the focus set too small if the system is to be able to reason effectively. Otherwise we would be in the position of a person trying to solve a crossword puzzle who can only look at one or two of the squares at once. At a minimum, we would need one rule (or planning operator) and its antecedents in focus. However this is clearly unrealistic—we cannot possibly know which rule is the 'correct' one to fire at any given point in the inference process. Even if we could, the cost of refocussing after every inference would be prohibitive. Nor would such a small focus set allow us to plan. To evaluate a choice point in a plan we need at least the 'OR' node and enough of each of the alternative sub-plans to estimate their feasibility and utility.

The introduction of the focus set together with a number of other optimisations has a significant effect on system performance. For example, if we analyse the generation of the of the simple `tell` utterance shown in figure 8.1 we obtain the results shown in tables 9.2 and 9.3.

The tables show the number of belief sets, the number of intention sets, the number of belief candidates and the number of intention candidates at each stage of the plan in Figure 8.1. Table 9.2 gives the results for sets closed under negation and table 9.3 shows the values for one such optimisation which results in the system only generating the negations of literals when these are required.

Belief Sets	Intention Sets	Belief Candidates	Intention Candidates
12	39	9	3
12	58	9	7
12	77	9	15
12	96	9	31
12	111	9	62
12	121	9	126
12	131	9	254
–	–	–	–

Table 9.2: Without optimisation

Belief Sets	Intention Sets	Belief Candidates	Intention Candidates
8	16	4	2
8	24	4	3
8	32	4	4
8	46	4	5
8	50	4	5
8	54	4	5
8	59	4	10
8	63	4	10

Table 9.3: With optimisation

The absence of a value indicates that a time or space bound was exceeded. As can be seen, the optimisation results in a marked reduction in the number of intention sets and particularly the number of intention candidates.

While this results in a substantial improvement in overall system performance, the net result of turning on all the optimisations is only enough to run the test examples in chapter 8. This allows us to process simple dialogues consisting of two or three conversational turns; however it seems unlikely that the approach can be extended to allow the fuller dialogues necessary to build a problem description or retrieval strategy.

9.1.3 Possible Solutions to the Complexity Problem

In this section we explore a number of possible solutions to the problem of computational complexity. There appear to be two main approaches to this: reducing the number of beliefs and intentions the system must consider; and reducing the number of belief sets generated from a given number of assumptions. The former approach involves changes in the architecture, the latter approach involves changes in the theory.

The theory is silent about what constitutes a belief or intention, not altogether surprisingly given that Galliers' approach is intended to be general and hospitable. One possibility would be to try and pack more information into each ATMS assumption. Limited improvement is possible by rewriting the rules to minimise the number of beliefs and intentions produced for a given problem.

While this may allow us to test very simple dialogues it is not feasible as a general strategy. Even if it were possible to eliminate all redundancies from the knowledge base, it seems likely that extended dialogues (e.g. more than one or two conversational turns) will require too many beliefs. A more radical approach would be to process beliefs and intentions separately. This is theoretically possible as intentions are 'parasitic' on beliefs: changes in belief can result in changes in intention but not vice versa. However it assumes that some method can be found of revising an agent's intentions in response to changes in its beliefs which is more computationally tractable than the ICM approach. Using a STRIPS-like planner in conjunction with the ICM requires considerable care to maintain the dependencies between beliefs and plans.[10] Using two separate ICMs, one for beliefs and one for intentions, would reduce the number of belief sets which must be considered. However this is unlikely to result in a substantial improvement, as the simple one-step plan presented above contains no extraneous beliefs and is close the upper bound of the ICM.

The other option is to revise the theory. Adopting the equivalent of de Kleer's 'single fault assumption', i.e. that no more than one belief need be abandoned to restore consistency, conflicts with the basic premise of the theory of belief revision, since it could result in the system abandoning one strongly endorsed belief in preference to two weakly endorsed beliefs. Nor can we 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. The only other option appears to be to use *additional* background knowledge to eliminate unlikely belief sets. However this is also problematic. If the additional knowledge is represented within the belief sets (for example, as additional justifications for each belief or its negation) this simply increases the number of beliefs that must be processed by the system. On the other hand, if the knowledge is external to the belief sets it cannot be over-ridden in particular cases and effectively forms an extension to the theory of belief revision.

9.2 The Problem of Increased Coherence

The increased coherence preference ordering *mc* can select the wrong belief set in certain circumstances, leading to irrational behaviour on the part of the agent. Belief sets are preferred if they offer more explanation (i.e. proofs) for core beliefs no matter what these explanations entail. We illustrate this with two examples, 'dead penguins don't fly' and 'Mary's phone number'.

In the first example, increased coherence prefers belief set (a) over (b), given that (*p-bel x* (*fly Tweety*)) is a core belief (see figure 9.2).

---

[10] Indeed it was the complexity of this approach which led to intentions being integrated into the belief revision framework in the first place.

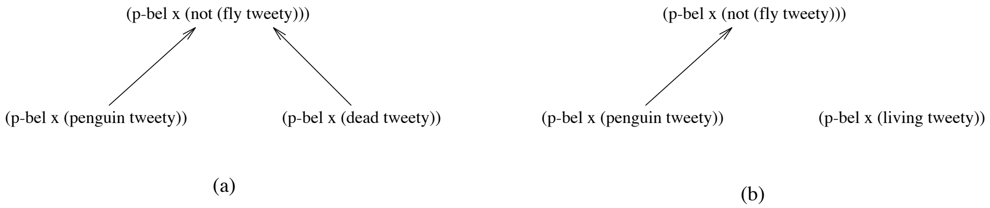

(p-bel x (not (fly tweety)))
(p-bel x (penguin tweety))
(p-bel x (dead tweety))
(a)
(p-bel x (not (fly tweety)))
(p-bel x (penguin tweety))
(p-bel x (living tweety))
(b)

Figure 9.2: The Tweety Example

This is not just a maximisation principle. Sets containing additional unrelated, though consistent, beliefs are not preferred. Being dead is a separate reason for not being able to fly. Being alive is consistent with Tweety's inability to fly but is less coherent. Where the alternative belief sets are inconsistent, neither is preferred, for example if we believed that Tweety couldn't fly because Tweety was either a pig or a DC10. This allows us to use endorsement information to determine which of the conflicting alternative sets to prefer, for example, in discriminating between literal and ironic interpretations of an utterance. In this case we have two conflicting belief sets, each of which is more coherent than the alternatives, and their relative degree of endorsement can be used to determine which is preferred. However relying on endorsement to sort out manifestly unsatisfactory results from the prior processing stage using *mc* is clearly objectionable.

In the second example, John initially knows his own phone number

$$(p\text{-}bel\ John\ (tel\ \ John\ \ 456)\ \ 1c\text{-}pos)$$

and knows that this is a phone number

$$(p\text{-}bel\ John\ (phone\text{-}number\text{-}p\ \ 456)\ \ 1c\text{-}pos)$$

He does not know Mary's phone number but concludes that it cannot be 456 by a rule that no two people share the same phone number:

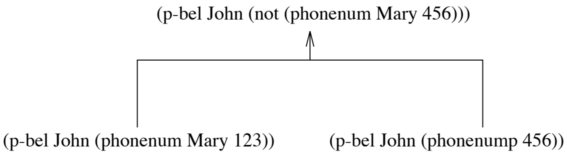
$$\begin{aligned}&(p\text{-}bel\ John\ (tel\ \ John\ \ X))\\&\qquad\Rightarrow_{[premise]}\ (p\text{-}bel\ John\ (not\ (tel\ \ Mary\ \ X)))\end{aligned}$$

$(p\text{-}bel\ John\ (not\ (tel\ \ Mary\ \ X)))$ is a core belief. John also has the rule that if he believes that Mary's telephone number is X and that phone number Y is different from X, then he believes her number is not Y:

$$\begin{aligned}&(p\text{-}bel\ John\ (tel\ \ Mary\ \ X))\ \&\\&(p\text{-}bel\ John\ (telp\ \ Y))\ \&\\&(X \neq Y)\\&\qquad\Rightarrow_{[premise]}\ (p\text{-}bel\ John\ (not\ (tel\ \ Mary\ \ Y)))\end{aligned}$$

Now suppose John overhears two utterances between Fred and Bill:

```
FRED : Mary's phone number is 123.
BILL : Mary's phone number is not 123.
```


(p-bel John (not (phonenum Mary 456)))
(p-bel John (phonenum Mary 123))
(p-bel John (phonenump 456))

(1) Mary's phone number is 123

(p-bel John (not (phonenum Mary 456)))
(p-bel John (not (phonenum Mary 123)))

(2) Mary's phone number is not 123

(p-bel John (not (phonenum Mary 456)))
(p-bel John (phonenum Mary 123))

(3) Mary's phone number is not 123 but she has more than one phone

Fred claims that Mary's phone number is 123 and John therefore gains the possible belief

$$(\mathit{p\text{-}bel\ John}\ (\mathit{tel\ Mary\ 123})\ \ \mathit{2c\text{-}pos})$$

John fires his rule from this possible belief and creates a new justification for $(\mathit{p\text{-}bel\ John}\ (\mathit{not}\ (\mathit{tel\ \ Mary\ \ 456})))$:

$$\begin{aligned}&(\mathit{p\text{-}bel\ John}\ (\mathit{tel\ \ Mary\ \ 123}))\ \&\\&(\mathit{p\text{-}bel\ John}\ (\mathit{phone\text{-}number\text{-}p\ \ 456}))\ \&\\&(\mathit{123} \neq \mathit{456})\\&\qquad \Rightarrow_{\Box}\ (\mathit{p\text{-}bel\ John}\ (\mathit{not}\ (\mathit{tel\ \ Mary\ \ 456})))\end{aligned}$$

However, Bill counters Fred's utterance by saying that Mary's phone number is not 123, giving rise to another belief for John

$$(\mathit{p\text{-}bel\ John}\ (\mathit{not}\ (\mathit{tel\ Mary\ 123}))\ \ \mathit{2c\text{-}pos})$$

Thus John has three belief sets after the exchange between Fred and Bill: All sets are preferred through endorsement but belief set (1) is preferred through increased coherence. John chooses to believe that Mary's phone number is 123 simply because he has the core belief that it is not 456!

The principal problem with *mc* is that it tends to favour 'conspiracy theories'. Because *mc* precedes endorsements in the belief set ordering, this can lead the agent into the wildest flights of fancy. *mc* favours *any* consistent justification for a core belief, no matter how absurd or bizarre. In the first example above, set (a) is preferred no matter how weakly the assumption that Tweety is dead is endorsed. For example,$(\mathit{p\text{-}bel\ x}\ (\mathit{dead\ Tweety}))$ could be endorsed *def* and $(\mathit{p\text{-}bel\ x}\ (\mathit{living\ Tweety})\ )\ \mathit{1c\text{-}pos}$, but set (a) would still be preferred. The endorsement step is therefore forced to choose between members of an equivalence class of more coherent belief sets, which may be less endorsed than other, simpler belief sets.

Even in the best case, when all the alternatives are plausible, this results in a preference for more complex explanations and violates Occam's razor. For example, in a fault diagnosis context, *mc* will always prefer a multi-fault diagnosis over a simpler single fault failure. If my computer doesn't work, it might be because the power supply is dead or that the ethernet is down or both. The latter possibility is preferred as the more coherent, since it offers two independent (consistent) reasons for the observed facts.[11]

The example in section 5.3.1 clearly demonstrates the need for coherence and explanations for believing. However the model of increased coherence prefers all explanations provided they are consistent, and as we have shown, this can lead to irrational behaviour.

9.3 The Problem of Prediction

An agent decides what it currently believes—what it thinks the world is like now—before attempting to plan on the basis of the most preferred belief sets. Plans are concerned with what the agent will do (and implicitly with what the agent will believe after it has carried out the plan). At each cycle, after belief revision and planning, the agent:

1. performs an action which is predicted to be successful and doable (if there are any such actions); it then revises its beliefs to take into account that fact that the action has been performed; and
2. makes an inference, which may involve either deriving a new belief or an intention/plan step.

The agent then revises its beliefs and plans and the cycle starts over. The system's predictions are revised whenever the system performs an action, abandons a plan which justifies the prediction or learns that a predicted state has come to pass, for example through the actions of another agent.

While the approach to prediction we have adopted works for the test examples presented in chapter 8, it presents a number of problems when applied to other aspects of the information retrieval task. The prediction rules presented in chapter 7 were not intended for 'real world' problems and do not work when used in that context.[12]

---

[11] It is worth pointing out that even without *mc* the theory suffers from this problem, in that it will always prefer a more endorsed belief set. Arguably, this is correct: if I have reason to believe that both the power supply is dead and that the ethernet is down then I should prefer the belief set in which both are defective. However, resolving ties between belief sets with equal numbers of, say *2c-pos* endorsements, may require relying on less and less certain information.

[12] There are also a number of problems with our approach to representing time. However these arise either as a result of arbitrary limitations (such as the assumption of a single future time), and can be overcome by introducing additional time points at the cost of complicating the rules, or are common to all 'point-based' approaches to the representation of time and can only be overcome by changing the ontology, for example by introducing time intervals. As a result, we can't represent continuous processes or the activities (as opposed to the intentions) of other agents. For example an agent may believe that p is true now and predict that after

For example, suppose we gave the agent described in chapters 5–7 the task of opening the door together with the planning rules necessary to perform the task. There are some obvious parallels between planning 'in the world' and planning to induce a particular belief state in another agent. Thus if we want to open the door, we can plan to open the door; if we want agent x to believe p, we can tell agent x that p. In both cases there are preconditions which must hold for the action to be sensible: in the former case the door must be closed (otherwise there is no need to plan to open it) and in the later case the agent x must not already believe p (otherwise there is no reason to inform it of p). (We ignore indirect speech acts.) In addition, in both cases, there are many other things that must hold if the action is to be successful. For example if the door is locked or nailed shut, we will not be able to open it. If agent x has many compelling reasons for believing that $\neg p$ or considers the speaker to be ill informed or unreliable, then simply informing agent x of the speaker's belief in p may not induce it to change its mind.

However, the approach to prediction developed to predict whether an attempt to change another agent's beliefs through dialogue will be successful doesn't carry over to 'real world' planning. For example given the justifications:

$$(\mathit{bel}\, x\, (\mathit{closed\ \ door})\ t\, \mathit{strong}) \wedge (\mathit{bel}\, x\, (\mathit{unlocked\ \ door})\ t\, \mathit{weak}) \wedge$$
$$(\mathit{int}\, x\, (\mathit{open\ \ door})\ t\, \mathit{desire\text{-}pos}) \supset (\mathit{bel}\, x\, (\mathit{open\ \ door})\ t'\, \mathit{weak})$$

$$(\mathit{bel}\, x\, (\mathit{closed\ \ door})\ t\, \mathit{strong}) \supset (\mathit{bel}\, x\, (\mathit{closed\ \ door})\ t'\, \mathit{strong})$$

$$(\mathit{bel}\, x\, (\mathit{open\ \ door})\ t'\, \mathit{weak}) \wedge (\mathit{bel}\, x\, (\mathit{closed\ \ door}))t'\mathit{strong} \supset \perp$$

where $t' > t$, the system will believe that the door will remain closed. This is obviously incorrect.

The major difference between dialogue and 'real world' planning is that while the agent has a *theory* of belief revision (implemented as the 'rule of continuity' and the preference mechanism), it has no corresponding theory of naive physics or commonsense knowledge. An agent uses another agent's commitment to its beliefs to predict whether their plan to change the others agent's belief state will be successful. In the 'real world', strength of belief is not related to the likelihood of achieving our objectives. However when planning to change another agent's beliefs, we do not reason about the structure of their beliefs, but about their commitment to their belief. The 'rule of continuity' is used to project the agent's beliefs into the future. The effect of executing the plan and the rule of continuity result in two inconsistent predictions about the beliefs

---

performing some action a which has p as a precondition, q will be true. From the fact that q will be true in the future, it predicts that r will be true and so on. There is only one 'future time' which can be interpreted as a temporal logic 'eventually' operator, or alternatively as a different future time for each belief. For example, if it is predicted that both p and q be true in the future, when p becomes true it is not necessarily the case that q becomes true. Sequence and causality information are imperfectly represented by justifications. For example, given $(\mathit{f\text{-}bel}\, x\, p) \wedge (\mathit{f\text{-}bel}\, x\, q) \supset (\mathit{f\text{-}bel}\, x\, r)$, it is impossible to tell from the justification if $(\mathit{f\text{-}bel}\, x\, p)$ and $(\mathit{f\text{-}bel}\, x\, q)$ are true simultaneously or if the action resulting in $(\mathit{f\text{-}bel}\, x\, p)$ must occur before the action the resulting in $(\mathit{f\text{-}bel}\, x\, q)$. However, in our domain this is not a serious limitation.

of the other agent, and forces the belief set to be partitioned. The attempt to change the other agent's beliefs will be (is predicted to be) successful if the plan/belief set containing the relevant action and the predicted change in belief is preferred to the belief set representing the agent's 'inertia' or resistance to change as predicted by the 'rule of continuity'. This additional predictive ability in the case of beliefs allows us to to determine which plan operators will be successful given a particular set of preconditions.

It is not clear that we can solve this problem by adding additional rules to handle 'real world' prediction.[13] In practice we distinguish between default assumptions and actions which are 'reliable', i.e. they are almost always true or almost always work, and 'unreliable' default assumptions and actions which, although useful heuristics, admit many exceptions. We only worry when our conclusions depend on the latter kind of default. However the reliability of an assumption or action is context dependent in a complicated way. 'Reliable' and 'unreliable' defaults are just the tip of a very large iceberg of strategies, heuristics and tricks which we use in coping with the world. Returning to the locked door example, we might expect that unlocking the door is likely to be successful unless the door is jammed or nailed shut. How do we express our confidence in predicting that a locked door will remain closed and compare this with our confidence in the prediction that the act of simply opening the door (without first unlocking it) will be successful.

Even if we restrict ourselves to task-oriented dialogues, it is not clear that the notion of endorsements can be pushed far enough to overcome these problems without degenerating into a production system with confidence factors.

9.3.1 Predicted Intentions

However, there is a more fundamental problem with the way the prediction rules propagate commitment, which applies to all communicated intentions. When an agent communicates a belief or intention the agent intends that the hearer adopts the attitude. Before making the utterance, the speaker predicts the outcome of its dialogue actions to see whether the hearer will adopt the attitude. Reasons for believing that the attitude will be adopted are that the speaker intends a dialogue action and that the preconditions of this action are satisfied, and that the speaker is committed to the content of the dialogue action and the hearer will adopt this communicated commitment as justification for adopting the communicated attitude. If the speaker predicts that the act will not be successful (i.e. if the speaker predicts that the hearer will not come to believe the communicated attitude), then the intended dialogue actions are

---

[13] An alternative way of achieving the same behaviour, would be to move 'prediction' into the antecedent of the planning rules. Since we know this information to start with, it seems odd to go to the trouble of considering a candidate partial plan, predicting that there will be no change in the other agent's beliefs, detect the inconsistency and partition the belief set, and evaluate the resulting belief sets to determine if the plan will work. However it is difficult to anticipate all the possible reasons for the failure of an action in the antecedents of the relevant planning rule(s). Introducing a 'circumscription-like' operator wouldn't solve the problem as this only addresses the problem of the antecedents which have not been included in the preconditions and ignores uncertainty in the actions themselves.

revised. Thus the speaker's commitment to its intention to change the hearer's beliefs depends on its commitment to its predictions and vice-versa. This cyclic dependency causes problems when the speaker attempts to communicate an intention.

When the speaker intends to communicate a belief, p, it determines its commitment to the belief and then reasons whether this commitment is sufficient to change the hearer's beliefs. The speaker's commitment to p is independent of its intended actions. However, when the agent intends to communicate an intention there is a problem. Consider the following scenario. The speaker intends to achieve p and the only way it can do this is by convincing the hearer that it should adopt the intention to achieve p. Hence, the speaker, s, has the following justifications:

$$(p\text{-}int\, s\, p) \Rightarrow (p\text{-}int\, s\, (tell\ s\ h\ (int\, s\, p)))$$

$$(p\text{-}int\, s\, p) \wedge (p\text{-}int\, s\, (tell\ s\ h\ (int\, s\, p))) \Rightarrow (f\text{-}bel\, s\, (int\, h\, p))$$

The first justification is a plan for achieving the top-level intention. The second justification generates a reason for predicting that the outcome of the `tell` will be successful. The proposition $(f\text{-}bel\, s\, (int\, h\, p))$ is endorsed with the speaker's commitment to $(p\text{-}int\, s\, p)$ and this allows the speaker's belief revision system to reason about the preference of the hearer's beliefs after the dialogue action is executed. However, if the *f-bel* is inconsistent with another proposition, for example if the hearer intends $\neg p$, this alters the speaker's commitment to its top intention $(p\text{-}int\, s\, p)$ which in turn alters the reasons for adopting the *f-bel* which further weakens the commitment to the *f-bel* which weakens the commitment to $(p\text{-}int\, s\, p)$ . . . and so on!

From this example we can see that there is no obvious way of determining the preference ordering of intention sets in the general case since the preference ordering depends on those belief sets which are preferred and the commitment to the ordering. This problem does not occur when a belief is communicated since the commitment to beliefs (i.e. *p-bels*) is independent of intentions.

9.4 The Problem of Communicated Commitment

Endorsements on communicated beliefs (*2c-pos* and *2c-neg*) are assigned by the hearer (h) according to the speaker's (s) commitment to the beliefs. If, for example, s is strongly committed to a belief b and communicates this to h then h has *2c-pos* extra endorsement for believing b. An agent's commitment is a summary of the reasons for and against holding a belief: if an agent's reasons for believing b (through endorsement and justification) greatly exceed those for believing $\neg b$ then the agent has strong commitment to b.

The problem arises because there is nothing to stop s communicating both its commitment to b and also one or more of its justifications for b to h. (This may occur when s, in trying to convince h of the truth of b, is forced to communicate its reasons for belief in b.) Both utterances contribute to h's reasons belief in b. However, s's commitment to b is a function of its justification for b

and s has effectively communicated this justification twice. For example, If s has the following initial beliefs as a result of communication with another agent

$$\begin{array}{ll} (\textit{p-bel}\ s\ a\ t\ \textit{2c-pos}) & (\textit{bel}\ s\ a\ t\ \textit{strong}) \\ (\textit{p-bel}\ s\ a \Rightarrow b\ t\ \textit{2c-pos}) & (\textit{bel}\ s\ a \Rightarrow b\ t\ \textit{strong}) \\ (\textit{p-bel}\ s\ b\ t\ \textit{2c-pos}) & (\textit{bel}\ s\ b\ t\ \textit{strong}) \end{array}$$

nd h has no beliefs, then, after communicating

$$\begin{array}{l} (\textit{tell}\ s\ h\ (\textit{bel}\ s\ b\ t\ \textit{strong})) \\ (\textit{tell}\ s\ h\ (\textit{bel}\ s\ a\ t\ \textit{strong})) \\ (\textit{tell}\ s\ h\ (\textit{bel}\ s\ a \Rightarrow b\ t\ \textit{strong})) \end{array}$$

h has the beliefs

$$\begin{array}{l} (\textit{p-bel}\ h\ a\ t'\ \textit{2c-pos}) \\ (\textit{p-bel}\ h\ a \Rightarrow b\ t'\ \textit{2c-pos}) \\ (\textit{p-bel}\ h\ b\ t'\ \textit{2c-pos}) \end{array}$$

That is, h has a spurious additional *2c-pos* endorsement for believing b. In this example we can see that the *2c-pos* assigned to b by h's as a consequence of s's communication of its commitment to b arises from s's justification for b. h should be able to recognise this fact and remove the *2c-pos* endorsement on its own belief b. However, this is not a general principle. For example, if the justification was derived by s after communicating its commitment to b, the *2c-pos* endorsement on h's belief in b should remain as it represents an additional reason for belief in b. A general algorithm can arise only when detailed belief models for fellow agents are maintained by our agents. The development of such models is beyond the scope of this project.

9.5 The Problem of Dialogue Management

At present we have no real notion of dialogue focus. One consequence of this is that it is very difficult to tell whether a given problem descriptor communicated by the user should be added to the problem description or whether it should replace one of the existing descriptors. We cannot rely on the consistency or otherwise of the descriptor as the user may have an inconsistent problem description; and even if the descriptor is consistent, it may be intended to replace an existing descriptor (e.g. if the user specialises the topic).

One way of solving this problem is to extend the set of speech acts to include acts such as (`add` $s\ h\ p$) and (`revise` $s\ h\ p\ q$) so that the user's intent is unambiguous. If we also use a 'recency heuristic' (see below), this reduces to:

$$(\textit{add}\ s\ h\ p) \overset{def}{=} (\textit{tell}\ s\ h\ p)$$

$$(\textit{revise}\ s\ h\ p\ q) \overset{def}{=} (\textit{tell}\ s\ h\ \neg p), (\textit{tell}\ s\ h\ q)$$

But even when we know the intent of a dialogue act, there is a problem with belief ascription over time when the resulting belief ascription is inconsistent.

If the system believes that the user believes p at time t and at some later time t' comes to believe that the user believes $\neg p$, what should it do? This may not be as bad as it seems as we can already cope with the most important case, namely that where the system is trying to induce a change of belief in the user, but there is nevertheless a non-trivial problem.

For example, the following three `tell` acts from the user:

$$(\mathit{tell}\; s\; h\; p)$$

$$(\mathit{tell}\; s\; h\; p \supset q)$$

and then (maybe sometime later)

$$(\mathit{tell}\; s\; h\; \neg q)$$

could imply that the user has changed his mind about p or $p \supset q$, or that the user actually has inconsistent beliefs (i.e. they may not have fired the $p \wedge p \supset q \vdash q$ inference yet).

The simplest solution is always to ask whenever an inconsistency arises in the beliefs ascribed to another agent. However this raises the question of what the question would look like and why (and how) we should believe the reply rather than the original belief which resulted in the problem. (In general we would have to ask the user whether they still believe all of the beliefs involved in the inconsistency. However with care in ordering the questions, it may be possible to narrow down the possible candidates a bit.)

At a minimum we must assume the user is coherent (even if they are not), as otherwise the system will not be able to represent their belief state or reason about it.[14] However it may be possible to do better than this if we assume that the most recent communication takes priority. This seems reasonable if we assume that the user is sincere (even if confused) and that the system models the user's *current* beliefs. In this case we only have to work out what the user should now believe if they are consistent. Only if this is uncertain do we need to ask. We can either do this for directly communicated beliefs (i.e. in the event of a *direct* conflict where p is communicated at time t and $\neg p$ is communicated at time t', $t' > t$, believe $\neg p$) or, if we are willing to assume that the user approximates an ideal reasoner (i.e. at least as good as the system), we can also do it for implicit ascribed beliefs as well.

The current solution to this problem takes into account the user's commitment to the beliefs apparent in their `tell` utterances and the order in which the utterances occurred. Older modelled beliefs are not as hard to disbelieve as more recently revised beliefs. Also, beliefs communicated with strong commitment by the user are treated as more persistent in the user model than those which were weakly communicated. There is a trade-off (details yet undecided) between the 'time' and the 'communicated commitment' factors for the preference algorithm.

---

[14] See rule (R–3) in chapter 7.

Chapter 10

Conclusions

In this final chapter we attempt to pull together the threads from the rest of the report, summarise the achievements, and outline some of the more interesting problems we encountered. The project has developed and tested a model of an intelligent agent in order to simulate a librarian engaging in interactive dialogue with an literature-seeking user, and thus illustrate a general type of agent interaction with information exchange and cooperative plan formation. The system we have implemented has successfully conducted dialogues of this kind, though very short and simple ones, providing some validation for the models of agent and librarian we adopted as our starting point, and helping in the design of future information intermediary systems. In section 1 we summarise the work done and argue that despite a number of difficulties, the project met the goals set out in the original proposal.

However, we also identified a number of pressing problems which must be overcome before further significant progress can be made. In section 2 we outline some of the more important of these problems together with those promising lines of attack we have identified to date.

10.1 Project Overview

The aim of the project was to develop and computationally test a model of the information management behaviour of a librarian when interacting with a user about their literature need, using a general theory of belief revision due to Galliers. In this model, belief revision is considered a fundamental property of rationality, and communication is a special case of this. Communicating agents recognise each other's intentions to *change* their cognitive state. Such observed communicative actions alter a cognitive state which already exists, as do observations of the natural world. Agents, though they are autonomous in their actions and reactions to the world, thus share control over these changes. This is an important aspect of interaction in open, multi-agent environments where no one agent can be in possession of the 'truth', and prescribed behaviours imposing cooperation as benevolence may therefore be inappropriate. Modelling cooperative communicative behaviour fundamentally involves a model of autonomous belief revision, for the autonomous attainment of mutually satisfactory belief states.

The model of autonomous belief revision determines preferred sets of beliefs on recognition of new communicated evidence, with no requirement for a unique revision. There is also no requirement that the preferred revision incorporate the new communicated evidence. Agents can decide not to revise. Preference is established as a qualitative ordering between alternative sets in which the preferred sets are maximally persistent. Such sets are 'hardest to revise' in being maximally coherent or offering maximal derivability of core beliefs and in having maximally endorsed founding assumptions. The model therefore represents a blend of coherence and foundation theories of belief revision. It is also a part-logical and part-heuristic solution to the problem of discrimination between alternative, logically equivalent revisions.

From the AI point of view the work proposed for the project involved an evaluation of a general account of agent activity for an exemplar task clearly requiring cooperative and constructive interaction with another agent, characterised by a mix of knowledge and ignorance in each party. From the information retrieval (IR) point of view, the research would test an abstract model of the librarian, proposed by Belkin and colleagues (BBD), as a distributed system subsuming a set of specialised functional experts, and provided the necessary detailed underpinning, through belief revision, for each individual expert's operations. Finally, the project would, if successful, constitute a first, exploratory (though still limited) implementation of the intelligent interface needed to improve the performance of automated IR systems.

The project work can been seen as basic research aimed at a challenging and necessarily very long-term goal, automating the librarian. But while the research is therefore directed towards document retrieval, it also seeks to contribute to AI as a whole. Thus while from one point of view the aim is to apply AI ideas to IR, from another IR provides a valuable study context for modelling the way any agents adopt or change their beliefs about the world, particularly through engagement in dialogue.

10.1.1 Work Done

The main needs for the work were to make both the belief revision and librarian models more detailed and more concrete, and to implement and assess computational versions of each. This involved, specifically, developing the belief revision theory to incorporate intentions and plans, i.e. to provide a theory of attitude revision capable of stimulating action, notably dialogue action, and building an entire computational system for forming, manipulating, and choosing among competing attitude sets, including doing inference following new inputs, applying preference criteria, etc. For the IR side it was necessary to develop a detailed architecture able, in particular, both to maintain adequate control of the internal interactions between the functional experts and to ensure effective and coherent dialogue with the user. For the latter in turn, a model specifically of dialogue was required. Finally, it was necessary to provide actual knowledge simulating that the librarian deploys and, using this, to run the system to obtain the kind of dialogue behaviour observed in real user-librarian interaction.

The project has done everything it set out to do, though it must be emphasised that the system's knowledge and powers are very limited; that it has conducted only a few, extremely simple dialogues; and that there are large problems still to tackle.

Our initial work showed that the originally-proposed distributed agent architecture for the librarian was too weak, and the librarian's specialised functions are therefore represented only as distinct bodies of knowledge and rule sets rather than the independent functional experts proposed by Belkin et al, which do not themselves issue in actions.

10.1.2 System Design

The system design and implementation is thus as follows. An Attitude Revision component uses an ATMS operating on a database as the basic mechanism for forming and managing attitude sets. It then determines the preferences between these sets, using heuristic criteria referring to types of endorsement, connectivity, and resistance to change, and thus exploiting an agent's commitments to its attitudes. The Attitude Revision component is in turn invoked by the Inference Engine which uses rule schemas instantiated with beliefs and intentions in the database.

These rule (schema)s include ones for dialogue and for the librarian's particular functions, in our experiments ones for forming a user model, a problem description characterising the user's literature need, and a retrieval strategy for meeting this need through an actual literature search. The Inference Engine is also used for planning by invoking plan schemata including specifically ones for conducting dialogue in terms of simple dialogue games. The system runs in an essentially cyclic way and invocation is thus in fact bidirectional.

The agent represents beliefs and intentions in a simple predicate-logic language which is also used for communication with the user: there is no actual natural language dialogue. By simplifying the natural language processing issues and exploiting the generality of the agent model, we were able to run several agents at the same time which interacted by sending messages ('utterances') to each other, and this is how we have simulated dialogue between the user and the librarian. The user and librarian agents possess the same general-purpose knowledge e.g. about dialogue games, and much common domain knowledge; but there are enough differences between them, especially in IR-related knowledge (as would be likely in the real case) to drive communication seeking or offering information.

We have sought to maintain generality at every stage. The natural consequence has been that the system has to manage very large numbers of attitude sets. Our work has shown that a very general model of attitude revision can be applied, in conjunction with a particular model of task knowledge, to support cooperative information exchange; however we have also come up against some very challenging problems, notably computational and technical ones for the attitude revision model, and dialogue management ones. Much more work is of course also required to capture enough of a librarian's functional task knowledge before anything like a practical interface could be envisaged. We have nevertheless by now obtained a useful platform for further research.

10.2 Major Issues and Future Research

We can identify two major areas where further research is required:

- issues with belief revision per se, and in consequence with our model of dialogue; and
- issues for information retrieval per se, and in consequence with belief revision and with dialogue.

Below we briefly summarise the most important problems in each category and indicate promising future lines of research.

10.2.1 Belief Revision Issues

The theory of belief and intention revision proposed by Galliers and developed during the project has a number of significant advantages over previous work: it does not assume logical closure (cf (Gärdenfors 1992)), it does not assume that agents are cooperative (cf (Cohen and Levesque 1987)) and it can cope with unstructured, open-ended dialogues (cf (Carletta 1992)). However, in common with much previous work, it is computationally intractable in its 'pure' form, and has remained so despite considerable effort to produce an efficient implementation.

It is important to stress that the problem of computational complexity was recognised from the outset (see, for example (Galliers and Reichgelt 1990)).[1] Moreover the recognition that a theory is computationally intractable does not imply that an implementation should not be attempted. Most interesting AI theories, and in particular all theories of belief revision, are NP-complete. In such cases 'implementing' the theory consists of finding a computationally tractable approximation to an computationally intractable theory. However, having recognised this, we are still left with the question whether our current implementation of the theory of belief and intention revision, the ICM, is 'wrong' (i.e. there is another approximation of the theory which is tractable) or whether the problem lies with the theory itself, in the sense that it admits *no* useful tractable approximations. Note that even in the latter case, this does not mean the theory is *wrong* in some sense—it is still a source of valuable insights into the belief revision process—only that it cannot form the basis of a practical computational implementation.

While it is impossible to be certain, it seems most likely that the problem lies with the theory rather than the implementation. The theory fundamentally depends on the exhaustive enumeration of belief sets. In this sense the theory is *syntactic*, rather than, say, relying on knowledge of how an agent should revise its beliefs in a particular situation. Indeed, one of the major attractions of this

---

[1] Unfortunately the proposed solution to the problem, based on 'definite beliefs', doesn't work—see chapter 9, section 1.1 and Galliers and Reichgelt (1990).

approach is that it offers a general account of how an agent revises its beliefs and intentions using only a small number of principles (the ordering relations) and a limited amount of information about the agent's beliefs (the endorsements types). However, a corollary of this is that, except in very special circumstances (which the current implementation already exploits), we do not *know* which sets are least likely until we have determined what the possibilities are and have ranked them on the basis of their endorsements. Attacking the problem by attempting to eliminate a priori those which are least likely therefore misses the point. Any other means of determining the least likely belief sets would require both additional knowledge and extensions to the theory to incorporate this knowledge.

Many of the more obvious optimisations, e.g. not generating a belief set containing the negation of a proposition when it can be shown that such a set will never be preferred, are already performed by the implementation. While the incorporation of such refinements resulted in useful gains in performance, major improvements were only possible by weakening the theory in particular ways, e.g. turning off positive undermining for intentions. While such modifications allowed us to run our simple test examples, the implications of these somewhat ad hoc measures need to be investigated in detail, both with regard to their psychological plausibility and to determine whether they lead to other problems further down the line.

10.2.2 Future Work in Belief Revision

We can identify a number of critical problems which must be solved before the theory can be applied to realistic dialogue modelling tasks, as follows.

The most pressing concern with belief revision is clearly to investigate further the theoretical implications of the somewhat ad hoc revisions we had to make to the theory in our search for reasonable computational performance. We also need a better understanding of the role of coherence in the current theory, given our inability to identify suitable core beliefs. Something like *mc* clearly seems to be required, but the current definition results in counter-intuitive belief set orderings and irrational agent behaviour in certain circumstances (see chapter 9 for examples). This problem is doubly critical because core beliefs appear to be central to attacks on the problem of focus and recall in belief revision and the problem of dialogue focus.

We need a better developed theory of endorsements. An endorsement is a summation of an arbitrarily long chain of inference. For example, *1c-pos* is the strongest endorsement because the information is directly experienced by the agent and agents typically trust the evidence of their own senses. However there are situations in which these conditions do not hold—watching a magic show or watching television for example. With imperfect knowledge of the world and other agents' beliefs and intentions, an agent must ultimately rely on some form of endorsement or 'confidence factor'. Unfortunately it is difficult to know at what point in the chain of inference we should stop explicitly representing reasons for belief and summarise the rest of the chain as an endorsement, as this is typically context dependent. This becomes particularly important when we have to consider notions of expertise and unreliable agents, which are characteristic of the library domain and others and when we have to consider predicted future belief. For example, my knowledge of Unix may usually be fairly reliable, but all my experience is on Suns—should I be believed about HP machines which run a different version of Unix?

Given the inherently partial knowledge of agents, some form of 'confidence factor' is required. However the current set of endorsements is somewhat arbitrary and had to be considerably augmented during the project—the current implementation uses at least 21 different endorsements. Even with this expanded set of endorsements we cannot accurately model support for a proposition in some common contexts, for example where first hand knowledge is unavailable.

While there may be some small universal set of endorsements which are useful in many or all domains, experience suggests that tailoring the set of endorsements to the type of dialogue and/or the application domain is essential (at least within the current implementation framework). Separating the theory from the set of endorsements required by a particular domain may simplify things and would at least make the current approach look less like a hack. In this case we would view the theory of belief revision as providing a framework for experimenting with endorsements and their ordering relations. One natural line to consider is whether there is a small number of general types of endorsement, exemplified by those we originally started with, defining the top of an endorsement hierarchy whose lower levels are application specific. This could provide both a better grounding for more specific domain endorsement types, and also a 'back-up' endorsement structure for reasoning.

In addition we must seek other means of extending the theory to make it inherently more tractable. In particular, the use of focus and recall to control belief revision proposed in chapter 5 needs further work. While we have had some success in reducing the size of the constant factor, we cannot change the computational complexity of the theory, and there are limits to the improvements that can be obtained by more efficient implementation (e.g. not generating the negations of beliefs until they are needed). Ultimately, the only solution to this problem is to work with the right set of beliefs: tinkering with the system may give a few more beliefs to play with, but exponential growth will get us in the end. Focus and recall are the only solution we have to the computational complexity problem. However the current implementation of focus and recall is poorly motivated from a belief revision point of view and needs to be developed to meet the goals of a general theory of relevance.

10.2.3 Information Retrieval Issues

As with the theory of belief revision, the theory of the librarian adopted as a starting point was not well developed in a computational sense and considerable work as required before an implementation could be attempted. Also, as with the belief revision theory, this attempt to implement the architecture proposed by Belkin et al (BBD) led to problems which were not obvious from a casual inspection of the theory. However, in the case of the BBD model, the problems appear to be more tractable, and it was possible to retain BBD's functional decomposition, which we believe to be basically correct, by embedding it within a more conventional rule-based architecture, similar to that used in previous implementations, e.g. (Brajnik et al. 1990).

Our implementation has demonstrated a computational realisation of BBD's ideas, albeit within a different architecture. We have also addressed the critical problems of control of inference and dialogue management which were inadequately developed in the original BBD model. We have, moreover, demonstrated how the BBD model can be embedded within our model of belief and intention revision, leading to a more realistic model of how the librarian operates over time, e.g. how the librarian develops the problem description in response to partial and often conflicting utterances by the user.

10.2.4 Future work in Information Retrieval

The main problems in this area arise partly as a result of the difficulties with belief and intention revision, which forced us to drastically simplify the implementation of our version of the BBD model. Assuming these problems can be solved, the most pressing concern on the IR side is to complete the implementation of the architecture sketched in chapter 6 to see if it performs as we believe it will. Compared to our implementation of the belief revision theory (and the implementation effort expended), our implementation of the BBD theory is rudimentary at best, and without a reasonable working implementation of the BBD model it will be difficult to develop our intuitions regarding its computational behaviour, or to predict where difficulties are likely to arise.

However there are a number of areas where the limitations of the current model seem clear, particularly with regard to the satisfaction conditions of the various modules and how these relate to the problem of dialogue management and dialogue focus and the control of the modules themselves. Much more work is obviously required to develop the components of the model, e.g. to identify the kind of wide-ranging knowledge required by the various modules so they can perform their functions at a level of detail sufficient for realistic implementation, e.g. how the librarian understands the user's problem and constructs an effective retrieval strategy. However these problems seem more tractable than their belief revision counterparts and should be amenable to standard knowledge engineering techniques, at least for modest applications.

Overall, therefore, the work done on this project provides a platform, in terms both of a supply of various conceptual components and of an implemented system, for further investigation of the basic ideas involved in belief revision as such and in its use by agents engaged in information exchange for the document retrieval task.

Bibliography

Alchourron, C., Gärdenfors, P. and Makinson D. "On the logic of theory change: partial meet contraction functions and their associated revision functions", *Journal of Symbolic Logic* 50, 1985, 510–530.

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

Alshawi, H. (Ed) *The Core Language Engine*, Cambridge MA: MIT PRess, 1992.

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* 5, 1983, 153–167.

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

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

Bratman, M. E. *Intentions, plans and practical reasoning*, Cambridge MA: Harvard University Press, 1987.

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

Brooks, H. M., Daniels, P. J. and Belkin, N. J. "Problem descriptions and user models: developing an intelligent interface for document retrieval systems", in *Informatics 8: Advances in intelligent retrieval*, London: Aslib, 1985.

Carletta, J.C. *Risk-taking and recovery in task-oriented dialogues*, PhD Thesis, University of Edinburgh, 1992.

Carver N. "Evidence-Based plan recognition", COINS Technical Report 88-13, Department of Computer and Information Science, University of Massachusetts at Amherst, 1988.

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

Cawsey, A. et al. “Automating the librarian: belief revision as a base for system action and communication with the user”, *The Computer Journal* 35, 1992, 221-232.

Cawsey, A., Galliers, J., Logan, B., Reece, S., Sparck Jones, K., “Revising beliefs and intentions: a unified framework for agent interaction”, *Prospects for Artificial Intelligence: Proceedings of AISB93, The Ninth Biennial Conference of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour* (Ed Sloman et al.), Amsterdam: IOS Press, 1993, 130–139.

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

Cohen P. R. *Heuristic reasoning about uncertainty: an artificial intelligence approach*, Boston: Pitman, 1985.

Cohen, P. R. and Levesque H. “Rational interaction as the basis for communication”, Technical Report No. 89, Centre for the Study of Language and Information, Stanford University, 1987.

Cohen, P.R., Morgan, J. and Pollack, M.E. (Eds) *Intentions in communication*, Cambridge MA: MIT Press, 1990.

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

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

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

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

Doyle, J. “A truth maintenance system”, *Artificial Intelligence* 12, 1979, 232–272.

Doyle, J. “Reasoned assumptions and Pareto optimality”, *Proceedings of IJCAI–85* (Los Angeles), 1985, 87–90.

Doyle, J. “AI and rational self-government”, Technical Report CMU-CS-88-124, Department of Computer Science, Carnegie-Mellon University, 1988.

Doyle, J. "On universal theories of defaults", Technical Report CMU-CS-88-111, Department of Computer Science, Carnegie-Mellon University, 1988.

Doyle J. "Rational belief revision: preliminary report", *Proceedings of the Second Conference on Principles of Knowledge Representation and Reasoning* (San Mateo, CA), 1991, ??–??

Doyle, J. and Wellman, M. P. "Impediments to universal preference-based default theories", *Proceedings of the First International Conference on Knowledge Representation and Reasoning* (Toronto), 1989, ??–??

Doyle, J. "Rational belief revision", in *Belief revision* (Ed Gärdenfors), Cambridge: Cambridge University Press, 1992.

Dressler, O. and Farquhar, A. "Putting the problem solver back in the driver's seat: contextual control of the ATMS", *ECAI-90 Workshop on Truth Maintenance Systems*, 1990.

Etherington, D. W. "Formalizing nonmonotonic reasoning systems, artificial theories and inferential distance" *AAAI–87, Proceedings of the Sisth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, 1987, ??–??.

Forbus, K. D. and de Kleer, J. "Focussing the ATMS", *AAAI–88, Proceedings of the Seventh National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, 1988, 193–198.

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

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

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

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

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

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

Galliers, J. R. "The positive role of conflict in cooperative systems", in *Decentralized artificial intelligence* (Ed Demazeau and Muller), Amsterdam: Elsevier, 1990.

Galliers, J. R. and Reichgelt, H. "A framework for autonomous belief revision", Computer Laboratory, University of Cambridge, Working paper, 1990.

Galliers, J. R. "Autonomous belief revision and communication", in *Belief revision* (Ed Gärdenfors), Cambridge: Cambridge University Press, 1992.

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

Gärdenfors, P. "The dynamics of belief systems: foundations vs coherence theories", *Revue Internätionale de Philosophie* 172, 1989, 24–46.

Gärdenfors, P. "Belief revision: an introduction", in *Belief revision* (Ed Gärdenfors), Cambridge: Cambridge University Press, 1992.

Garigliano, R., Bokma, A. and Long, D. "A model for learning by source control", *Uncertainty and Intelligent Systems: 2nd International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems* (Italy), Springer, 1988, ??–??

Gmytrasiewicz, P. J., Durfee, E. H. and Wehe, D. K. "A decision theoretic approach to coordinating multi-agent interactions", *Proceedings of IJCAI–91* (Sydney), 1991, ??–??

Gasser, L. et al. "Representing and using organizational knowledge in distributed AI systems", in *Distributed artificial intelligence* (Ed Gasser and Huhns), London: Pitman, 1989.

Georgeff, M. P. and Lansky, A. "A system for reasoning in dynamic domains: fault diagnosis on the space shuttle", SRI Technical Note No. 375, SRI International, Menlo Park, 1986.

Grice, H. P. "Meaning", *Philosophical Review* 66, 1957, 377–388.

Grice, H. P. "Utterer's meaning and intentions", *Philosophical Review* 78, 1967, 147–177.

Grosz, B. J. "Discourse knowledge", in *Understanding spoken language* (Ed Walker), New York: North-Holland, 1978.

Grosz, B.J., Joshi, A.K. and Weinstein, S. "Towards a computational theory of discourse interpretation", revision of a 1986 ms, Department of Computer Science, Harvard University, in press.

Hadley, R. F. "Reasoning with Truth Ascriptions, Self-Reference and Embedded Sentences in First-Order Logic", *Proceedings of Computational Intelligence 90*, Milan, Italy: 1990.

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

Hewitt, K. "Offices are open systems", *ACM Transactions on Office Information Systems* 4, 1986, 271–287.

Hintikka, J. *Knowledge and Belief*, Ithaca, NY: Cornell University Press, 1962.

Huhns, M.N. et al. "DAI for document retrieval: the MINDS project", *Distributed artificial intelligence* (Ed Huhns), London: Pitman, 1987.

Jeffrey, R. C. *The logic of decision*, Chicago: University of Chicago Press, 1983.

Kelleher, G. "An overview of assumption based reason maintenance", Technical Report CBLU-ULE/043, Computer Based Learning Unit, University of Leeds, 1983.

Konolige, K. "Hierarchic autoepistemic theories for nonmonotonic reasoning", Technical Note 446, SRI International, Menlo Park, 1988.

Konolige K. "Defeasible argumentation in reasoning about events", *Proceedings of the International Symposium on Machine Intelligence and Systems* (Italy), 1989.

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.

Lambert, L. and Carberry, S. "A tripartite plan-based model of dialogue", *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, 1991, 49-54.

Levi, I. "Truth, fallibility and the growth of knowledge", in *Decisions and revisions*, Cambridge: Cambridge University Press, 1984.

Levinson, S.C. *Pragmatics*, Cambridge: Cambridge University Press, 1983.

Lewis, D.D. and Sparck Jones, K. "Natural language processing for information retrieval", *Communications of the ACM*, in press; Technical Report 307, Computer Laboratory, University of Cambridge, 1993.

Martins, J.P. and Shapiro, S.C. "A model for belief revision", *Artificial Intelligence* 35, 1988, 25–79.

McAllister, D. A. "An outlook on truth maintenance", AI Memo No. 551, Laboratory for Artificial Intelligence, MIT, 1980.

McAlpine, G. and Ingwersen, P. "Integrated information retrieval in a knowledge worker system", *Proceedings of the Twelfth Annual International ACM–SIGIR Conference on Research and Development in Information Retrieval*, 1989, 48–57.

Nebel, B. “A knowledge level analysis of belief revision”, *Proceedings of the 1st Conference on Principles of Knowledge Representation and Reasoning* (Toronto), 1989, ??–??

Nebel, B. “Syntax based approaches to belief revision”, in *Belief revision* (Ed Gärdenfors), Cambridge: Cambridge University Press, 1992.

Newell, A. “The Knowledge Level”, Report CMU-CS-81-131, Department of Computer Science, Carnegie-Mellon University, 1981.

Perlis, D. “Languages with Self-Reference I: Foundations”, *Artificial Intelligence* 25, 1985, 301–322.

Perrault, C. R. “An application of default logic to speech act theory”, Report CSLI 87-90, Centre for the Study of Language and Information, Stanford University, 1987.

Pollack, M. E., Israel, D. J., and Bratman, M. E. “Towards an architecture for resource-bounded agents”, SRI Technical Note 425, SRI International, Menlo Park CA, 1987.

Pollitt, A. S. “A rule-based system as an intermediary for searching cancer therapy literature on Medline”, in *Intelligent information systems: progress and prospects* (Ed Davis), London: Aslib, 1986.

Poole, D. L.: “On the comparison of theories; preferring the most specific explanation”, *Proceedings of IJCAI–1985* (Los Angeles), 144-147.

Provan, G. M., “A complexity analysis of assumption-based truth maintenance systems”, in *Reason Maintenance Systems and their Applications* (Ed Smith and Kelleher), Chichester: Ellis Horwood, 1988.

Provan, G. M., “The computational complexity of multiple-context truth maintenance”, *Proceedings of the 9th European Conference on Artificial Intelligence* (Ed Aiello), London, UK: Pitman Publishing, 1990, 522–527.

Rao, A.S. and Foo, N.Y. “Minimal change and maximal coherence: a basis for belief revision and reasoning about actions”, *Proceedings of IJCAI–89* (Detroit), 1989, 966–971.

Rao, A.S. and Foo, N.Y. “Formal theories of belief revision”, *Proceedings of the 1st Conference on Principles of Knowledge Representation and Reasoning* (Toronto), 1989, 369–380.

Rao, A. S. and Georgeff, M.P. “The formation, maintenance, and reconsideration of intentions”, in *Proceedings of the IJCAI-91 Workshop on Theoretical and Practical Design of Rational Agents*, 1991, ??–??

Reichgelt, H. “The place of defaults in a reasoning system”, in *Reason maintenance systems and their applications* (Ed Kelleher and Smith), Chichester: Ellis Horwood, 1988.

Reiter, R. "A logic for default reasoning", *Artificial Intelligence* 13, 1980, 81–132.

Reiter, R. and de Kleer J. "Formal foundations of assumption-based truth maintenance systems: preliminary report", *AAAI–87, Proceedings of the Sixth National Conference on Artificial Intelligence*, American Association for Artificial Intelligence, 1987, 183-187.

Rescher, N. *Hypothetical reasoning*, Amsterdam: North-Holland Publishing Co., 1964.

Shafer, G. *A mathematical theory of evidence.* Princeton NJ: Princeton University Press, 1976.

Singh, M. P. "On the commitments and precommitments of limited agents", *Proceedings of the IJCAI-91 Workshop on Theoretical and Practical Design of Rational Agents*, 1991, ??–??

Swain, M. *Induction, acceptance and rational belief*, Dordrecht: Reidel, 1970.

Sparck Jones, K. "Information retrieval", *Encyclopedia of artificial intelligence* (Ed Shapiro), 2nd Ed, New York: Wiley, 1992.

Thagard, P. "Explanatory coherence", *Behavioural and Brain Sciences* 12, 1989, ??–??

Thost, M. "Generating facts from opinions with information source models", *Proceedings of IJCAI–89* (Detroit), 1989, 531-536.

Touretzky, D. S. *The mathematics of inheritance systems*, Pitmans Research Notes in Artificial Intelligence, London: Pitman Publishing Ltd., 1986.

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

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

Appendix A

An Example Transcript of a User Librarian Dialogue

This is an extract for a transcription of a user librarian dialogue on Greek-Turkish Relations (see (Brooks 1986) and (Daniels 1987)); we are grateful to Professor Belkin of Rutgers University for supplying us with the full transcript for study.

L: Right, OK, right we've got that out of the way, what's the subject of your query?
U: Greek Turkish relations.
L: Right, anything particularly specific?
U: Actually, I'm interested in their disputes _other_ than Cyprus.
L: Right, disputes other than Cy - are there any particular ones, you know any?
U: um, the Aegean dispute? And their disputes over the treatment of the Turkish minority in Greece... and the Greek minority in Turkey.
L: Right, now have you found out very much published on this so far?
U: 's a good deal in Newspapers.
L: Yes, are you interested in newspapers or are you 's really articles.
U: No, Im not, no, Ive covered the Newspapers.
L: Good, that's fine.
U: I might also add that Ive done a similar search down in Canada, at my University.
L: Ah, I wondered about that, when you said you were from Canada. Which 't was using the Dialog system was it? can you remember what um database you searched?
U: was an American based err database.
L: yeah, sounds like Dialog. um can you remember which particular files you you searched cos if you wanted we could try and avoid those. Or was it that you wanted an

update?
U: Well, I would, yes, actually no I would want us please
to avoid those .. I would
L: I tell you what..
U: cos there's no need for duplication I think. I think that
L: yes, let me get the the the book which describes the
files that I think you probably searched. 'scuse me, jus
take off this mike.. right, now um.
U: I mentioned to the gentleman that I spoke to yesterday
about the possibility of limiting the search to British
journals and magazines.
L: Yeah, that's not as easy as it seems.
U: Is that right?
L: No, um, the only way you can do it really it'd be
nice if you could sit sort of do it in one single step
but the only way would be literally to put in every
title of every British journal that you wanted included
which is probably unless you've got 3 or 4 specific ones
its such an incredible task that..
U: I have no idea what British Journal or Magazine anyway.
L: No, its a problem. Um, the only thing we can do is once
we've decided on our database see if there was any um any key
as to say place of publication or country of publication
to get it that way, but not all the databases do so we have
to see how it went.
U: I see, OK.
L: Right, um, politics isnt it it comes under this as economics,
its current affairs, thats another possible another one
... and ... even a bit of social sciences, its really, your
subject is spread all over lots of different areas. What
about books? Which we havent talked much
U: Im fully conversant with
L: Good s' right, its really.
U: Cos there's so few of them really that come up on my, on
the subjects that Im interested in th-
L: Yes, I can imagine.
L: so, what.. its really Greek Turkish relations
U: Other than Cyprus because Ive tons of material on Cyprus
an' I any more would (inaud).
L: So really if if we could say anything on um what period,
what time span?
U: 1976 on.
L: uhuh, which is going to be difficult, yes, 76 onwards,
woops, so really if we could look at anything on Greek
Turkish relations from this period onwards.
L: yeah, so what we gotta do now is to get this time span
in because they dont automatically all, they're not all

indexed neatly where you can just say 1976 on, ah some of them are historical ones an thats a possibility, other than that I think the only way is to look at a database that hopefully will deal with recent material, we dont want, say ancient history coming out and this is the danger.

U: The reason why Ive chosen 1976 is that in 1974 there was a war in on Cyprus and that dominated their relationships you see.

L: Yes, of course.

U: I want to cut all that out because I have a lot of material on th' innumerable situations like this. I just want to exclude that really.

L: Yes, OK, so you jus want you dont want include that if you've got it already, I see, right, um.... Turkish minority in Greece, (inaud), Turkey ...

U: I can predict there wont be many by the way. I have already talked to people who are um I dunno maybe interested in the subject.

L: So its really, its political relations of any sort, err or any relations really?

U: Absolutely, I mean I'm primarily interested in their disputes but I'm dying to find some harmonious aspects of their relationships. I mean if there are articles that deal with other with a relationship that dont warrant a dispute thats fine by me too.

L: Yes, right an' it really any kind of connect in a way.

U: Absolutely.

L: an' it can be cultural exchanges or whatever.

U: yes.

L: So its really our problem now to find the right sources to tuck in um.... let me think about your main (inaud) books ..... so if we look through we've got this one, Current Affairs I'm just really looking through the Dialog guide because I think Dialog is going to be the one that gives us the widest range of databases... ah... now Ive got this Magazine Index it tends to be more sort of mega activities kind of thing. How about this one Middle East.

U: That's traced in the index. what about magazines,

L: I think possibly there are some.

U: you have some magazines that I respect, for instance err New Society's considered and err..

L: U think yes, I think it might be worth...

U: the Spectator and magazines like that.

L: Right (11 sec)

L: OK, and that goes right back to 19 well we only want nineteen seventy w- you said the most recent one is (inaud) and we can put _not_ Cyprus.. Middle East not really no, no

I think or would it come in?
U: it it might you see what I find that.. I happen I just had additional course in Middle Eastern politics, when people say Middle East they generally exclude Cyprus much to my chagrin they _usually_ just mean the Arabs and Israelis.
L: Yes, cos if you think of it then its very much like it..
U: people say the Middle East conflict an an you know refer to the Arab Israeli dispute by that. mm, lets try those anyway,
L: um, you say you've covered newspapers havent you?
U: Oh yes.
L: wont worry about that, um Public ah this is another possibility Public Affairs International its a little bit like the Magazine Index its perhaps a little bit...
U: I think that's what we tapped into in Canada but Im not sure.
L: right.
U: I cant be certain, I am merely guessing.
L: Well, maybe when you see the sort of things coming out because if you see things that you know we can hop onto another database, um, that's news again,... this is another one World Affairs Reporter, it sounds as if there should be something there again, .. 1970 onwards..
U: Can you instruct them to give you 1976?
L: Yes, you can limit them, yes, if you want me to, err, I dont think directories would be much use... nor di- well, Dissertations this is another possibility.
U: Oh yes, it'd be very nice to have that.
L: Its mainly American with a few Canadian, I think it would be worth trying.
U: What about British?
L: No, unfortunately you have to slog through that by hand at the moment, err, until they get it in machine readable form and up on a host.
U: Because Im here, I my main interest is to find British material.
L: Yes, the only way that you _could_ if you had the time, the they've got them in the library downstairs, they've got, um, indexes to them,, um, I think its Aslib produces, they could route you to the right source but you'ld have to go through it manually, but at least its not too far away. Um, the only other possibility is historical abstracts but but it it
U: No
L: is fairly, they can include some recent material ... we'll think about it, we'll see, we'll put a query by that one.
U: OK. Alright.

L: Its the only database which has really, obviously because it
deals with history has tried to cope with this time limitation.
U: mmm, hm, ok. Maybe, I'm, cos I'm in political science I generally
L: (inaud) International we've already had that that's ok,
... I think that's the only other possibility although it
sounds like social sciences, sometimes that can have,
its its got such a wide coverage
U: OK
L: that can sometimes have political um (inaud). We've
already got World Affairs, right, so if we've got Magazine
Index I think these are probably the most, the ones we ought
to start with.
U: OK.
L: Now, the next thing is to get our strategy sorted out
(inaud) (11 secs). We've got to have Turkish or Turkey
or Turkish .. do they ever refer to them as Turko? Greek
or Graeco Turkey or something?
U: Yeah, Greco Turkish
L: Yes, that way round, do they do it the other way round.
U: uh huh, no, not really, no.
L: no 's ok, and thats how they spell it normally.
U: yes
L: Yeah 's ok.
U: Actually without the a.
L: Yes, yes, and we want _not_ cyprus.
U: Yes.
L: I think the only way we can do it is to say not Cyp-
the only problem is i if they mention a really good one,
and they err a good paper and it happens to have Cyprus
in it we lose it this is the only problem but we'll have
a we'll sample as we go we wont, we'll see what we _would_
have missed by taking out Cyprus (inaud)
U: Oh, we will?
L: mm, we ca- we can try that either way, but what we'll
do is put everything, ..
U: because its virtually impossible err for any article
on Greek Turkish relations
L: not to have Cyprus.
U: not to have some mention of Cyprus.
L: Yes
U: See, it is the (inaud)
L: Well, what we'll do, according to each database
hopefully they will have indexed in some way or they may
have indexed in some way whether its a conflict or whether
its just some mention of the country, um, otherwise, yes
it is crude if we just take out Cyprus but um properly (inaud)
so anything with Greek and Turkey you know we can have any

combination of those two or Greco Turkish and hopefully
because we're in these Current Affairs type databases
which should cover things like relations politics and so on
cos if we were to out in specific key words we might well
miss papers.

U: I see.

L: its only if we get a lot of information, like cookery or
something like that that we'd have to try putting um, politic
and you can search on the root of words and get political
err relations ... relationship ... exchange i- if you could
have a huge list of words that w- we keep those in reserve
so that if we need to put those is, umm again we can look
at each as each as we go through each database we can see
if there's indexing and if so um, is there a very good
term that we could make our search more specific.

U: OK.

Appendix B

A Planning Demonstration

B.1 Introduction

We demonstrate the inference mechanism for a simple two-agent (agents A and B) dialogue fragment. Agent A is weakly committed to his belief that X and agent B is strongly committed to (*not* X): The example begins with A telling B that X.

$$(\textit{tell}\ A\ B\ (\textit{bel}\ A\ X\ \textit{weak}))$$

Agent B rejects X and recognises that a conflict exists over X between A and himself. He plans to resolve this and predicts that simply telling A of his own *strong* commitment to (*not* X) will cause A to revise his beliefs:

$$(\textit{tell}\ B\ A\ (\textit{bel}\ A\ (\textit{not}\ X)\ \textit{strong}))$$

This section will illustrate the cognitive state of agent B, from its initial state of believing (*not* X), through him receiving the utterance from agent A, recognising the conflict, planning to resolve the conflict and finally sending his own message to A.

B.2 Simulation

Initially, agent B has the belief that (*not* X) endorsed *1c-pos* ; he has a single belief set

$$\{\ (\textit{p-bel}\, B\, (\textit{not}\ X))\ \}$$

and one intention set

$$\{\ (\textit{bel}\, B\, (\textit{not}\ X)\ \ \textit{strong})\ \}$$

The literal X might be 'raining outside' and B has observed that it is not raining outside. He also has the dialogue rules in chapter 4 and an empty inference stack. When B receives a message from agent A

$$(\textit{tell}\ A\ B\ (\textit{bel}\ A\ X\ \textit{weak}))$$

he adds (*p-bel B* (*action* (*tell A B* (*bel A X weak*))) *premise*) to his database and to his belief set:

{ (*p-bel B* (*not X*)),(*p-bel B* (*action* (*tell A B* (*bel A X weak*)))) }

He then calculates his commitment to his beliefs and includes them as definite beliefs in his intention set:

{(*bel B* (*not X*) *definite*), (*bel B* (*action* (*tell A B* (*bel A X weak*))) *definite*)}

He also pushes these as a group onto the inference stack since they are newly derived and believed.

Group 0

(p-bel B (action (tell A B (bel A X weak))))
(bel B (action (tell A B (bel X weak))) strong)

Agent *B* searches down his inference stack, taking each inference stack group in turn; finds all rules with either an antecedent or consequent that binds with any member of the group; completes the binding of the antecedents, and chooses the rule with the greatest confidence factor (ie which would give greatest endorsement on the rule instance). In this case he chooses his ascription rule

(*p-bel ?A1* (*action* (*tell ?A2 ?A1* (*?A ?A2 ?P ?C*))))
$\Rightarrow_{[definite]}$ (*p-bel ?A1* (*?A ?A2 ?P ?C*))

with antecedent bound to

(*p-bel B* (*action* (*tell A B* (*bel A X weak*))))

and infers the belief that *A* believes that *X*,

(*p-bel B* (*bel A X weak*))

generating at the same time the premise rule instance

(*p-bel B* (*rule-inst* "(*p-bel?A1* (*action* (*tell ?A2 ?A1* (*?A ?A2 ?P ?C*)))) ⇒
(*p-bel ?A1* (*?A ?A2 ?P ?C*))"
((*p-bel reasoner* (*action* (*tell A B* (*bel A X weak*)))))
(*p-bel reasoner* (*bel A X weak*)))
definite)

For brevity we will call this rule instance *rule-inst-1.* The belief and intention sets are again subjected to revision (ie addition in this case), and the new attitudes are pushed, as a group, onto the inference stack:

Group 1

(p-bel B (bel A X weak))
(p-bel B rule-inst-1)
(bel B (bel A X weak) strong)
(bel B rule-inst-1 strong)

Group 0

(p-bel B (action (tell A B (bel A X weak))))
(bel B (action (tell A B (bel X weak))) strong)

The next preferred inference is for B to consider adopting X himself. Note that other inferences are possible:

1. the desire rule to resolve the conflict in X since B both believes (*not* X) and believes that A believe s X; and
2. the prediction rule that B will believe that A will believe X in the future.

Both these inferences involve rules with lower confidence factors than the adoption rule (ie *2c-pos* vs *desire-pos* and (*auto-predict-bel-pos alter-predict-bel-pos*)). The new inference generates the possible belief

(*p-bel B* (*not X*))

and rule instance

(*p-bel B* (*rule-inst* "(*p-bel ?A1* (*bel ?A2 ?P weak*)) ⇒ (*p-bel ?A1 ?P*)"
((*p-bel reasoner* (*bel A X weak*)))
(*p-bel reasoner* (*not X*))
2c-neg)

Again, for brevity we will call this rule instance *rule-inst-2*. These beliefs are added to the database and new belief sets are generated. The database now contains contradictory beliefs, and there are two belief sets:

1. { (*p-bel B X 1c-pos*) }
2. { (*p-bel B* (*not X*)), (*p-bel B rule-inst-2 2c-neg*) }

Belief set 1 is preferred. Next the belief commitments are calculated, and the preferred (ie only) intention set contains:

(*bel B* (*not X*) *strong*)
(*bel B* (*action* (*tell A B* (*bel B X weak*))) *strong*)
(*bel B rule-inst-1 strong*)
(*bel B* (*bel B X weak*) *strong*)
(*bel B* (*not rule-inst-2*) *strong*)

Since the *rule-inst-2* is not believed, its negation is generated as a hypothesis in the database. The belief sets are revised, and the negation of the rule instance is added to the inference group on the top of the inference stack:

Group 2

(p-bel B (not rule-inst-2))
(bel B (not rule-inst-2) strong)

Group 1

(p-bel B (bel A X weak))
(p-bel B rule-inst-1)
(bel B (bel A X weak) strong)
(bel B rule-inst-1 strong)

Group 0

(p-bel B (action (tell A B (bel A X weak))))
(bel B (action (tell A B (bel X weak))) strong)

No inference can be drawn from the top group in the stack. The penultimate group contains (*bel B* (*bel A X weak*) *strong*), from which agent *B* can predict that he will believe that *A* will believe *X* in the future:

(p-bel* B (rule-inst "(bel ?A (bel ?B ?p weak) strong) ⇒ (f-p-bel ?A (bel ?B ?p))"
((p-bel reasoner (bel B X)))
(f-bel reasoner (bel B X)))
(auto-predict-bel-posalter-predict-bel-neg))
(f-p-bel B (bel A X))

We refer to this rule instance as *rule-inst-3*. The belief sets are revised (only addition this time), commitments to the beliefs calculated, intention sets are generated, and the new inferences are pushed onto the inference stack.

The next inference takes place from group 1 again. No inferences can be drawn further up the stack, since there are insufficient attitudes to complete the antecedent bindings of the rules. Agent *B* infers from the conflict the desire to revise *A*'s belief in *X*:

(p-bel* B (rule-inst "(?att ?agent (not ?p)) (bel ?agent (?att ?fellow-agent ?p)) ⇒
(p-int ?agent (bel ?agent (not ?p)))"
((bel reasoner (not X)) (bel reasoner (bel A X)))
(p-int reasoner (bel A (not X))))
desire-pos)
(p-int B (bel A (not X)))

This is rule instance *rule-inst-4*. The proposition (*not* (*bel B* (*bel A* (*not x*)))) is hypothesised in the database, and is pervasive in the intention sets so long as (*p-bel B* (*bel A* (*not x*))) is not pervasive in the belief sets, which is the case. The extension to the preferred intention set includes the leading intention (*p-int B* (*bel A* (*not X*))). These attitudes are pushed onto the inference stack:

Group 4

(p-int B (bel A (not X)))
(p-bel* B rule-inst-4)
(int B (bel A (not X)) strong)
(bel* B rule-inst-4 strong)

Group 3

(p-bel* B rule-inst-3)
(f-bel B (bel A X))
(bel* B rule-inst-3 strong)

Group 2

(p-bel B (not rule-inst-2))
(bel B (not rule-inst-2) strong)

Group 1

(p-bel B (bel A X weak))
(p-bel B rule-inst-1)
(bel B (bel A X weak) strong)
(bel B rule-inst-1 strong)

Group 0

(p-bel B (action (tell A B (bel A X weak))))
(bel B (action (tell A B (bel X weak))) strong)

The following inferences generate the plan to satisfy B's leading intention. Firstly, B infers the intention for agent A to adopt $(\textit{not}\ X)$:

$$(\textit{p-int}\ B\ (\textit{action}\ (\textit{adopt}\ B\ A\ (\textit{bel}\ A\ (\textit{not}\ X)))))$$

He then uses his planning rule

(p-int ?A1 (action (?ACT ?A2 ?A3 ?P)))
(action-schema ?PRECS (?ACT ?A2 ?A3 ?P) ?_ ?_ ?_)
    $\Rightarrow_{[premise]}$ *(forall ?W ?PRECS (intorbel ?A1 ?W))*

to infer the intention to satisfy the preconditions of the `adopt` action instance:

$$(\textit{p-int}\ B\ (\textit{bel}\ A\ (\textit{bel}\ B\ (\textit{not}\ X))))$$

To satisfy this intention he uses

(p-int ?A1 (?A ?A2 ?P)) &
(action-schema ?_ (?act ?A3 ?A2 ?Q) ((?A ?A2 ?P)) ?constraints ?_) &
(forall ?c ?constraints ?c)
    $\Rightarrow_{[premise]}$ *(intorbel ?A1 (action (?act ?A3 ?A2 ?Q)))*

Group 4

(p-int B (bel A (not X)))
(p-bel* B rule-inst-4)
(int B (bel A (not X)) strong)
(bel* B rule-inst-4 strong)

Group 3

(p-bel* B rule-inst-3)
(f-bel B (bel A X))
(bel* B rule-inst-3 strong)

Group 2

(p-bel B (not rule-inst-2))
(bel B (not rule-inst-2) strong)

Group 1

(p-bel B (bel A X weak))
(p-bel B rule-inst-1)
(bel B (bel A X weak) strong)
(bel B rule-inst-1 strong)

Group 0

(p-bel B (action (tell A B (bel A X weak))))
(bel B (action (tell A B (bel X weak))) strong)

Group 7

(p-int B (action (tell B A (bel B (not X)))))
(p-bel *B rule-inst-7)
(bel* B rule-inst-7 strong)
(not (bel B (action (tell B A (bel B (not X)))))
(int B (action (tell B A (bel B (not X)))) strong)

Group 6

(p-int B (bel A (bel B (not X))))
(p-bel* B rule-inst-6)
(int B (bel A (bel B (not X))) strong)
(bel* B rule-inst-6 strong)
(not (bel B (bel A (bel B (not X)))))

Group 5

(p-int B (action (adopt B A (bel A (not X)))))
(p-bel *B rule-inst-5)
(bel* B rule-inst-5 strong)
(not (bel B (action (adopt B A (bel A (not X)))))
(int B (action (adopt B A (bel A (not X)))) strong)

to infer the intention to `tell` A his belief in $(\textit{not } X)$:

$$(\textit{p-int}\, B\, (\textit{action}\ (\textit{tell}\ B\ A\ (\textit{bel}\ B\ (\textit{not}\ X)))))$$

All of these inferences are confined to the intention sets. Each intention generated produces the corresponding hypothesis belief that the intention has been satisfied. For example, for the intention to `tell`

$$(\textit{p-int}\, B\, (\textit{action}\ (\textit{tell}\ B\ A\ (\textit{bel}\ B\ (\textit{not}\ X)))))$$

we have the hypothesis that the action has been done

$$(\textit{bel}\, B\, (\textit{action}\ (\textit{tell}\ B\ A\ (\textit{bel}\ B\ (\textit{not}\ X)))))$$

These intentions are stacked in the order in which they are generated, and the plan is generated depth first since each sub intention can be inferred from the top-most group in the inference stack. The inference stack when the agent generates the intention to `tell` is shown in figure B.2

The preconditions for B's intention to `tell` are already satisfied, and the antecedents for prediction rule

$(\textit{p-int ?A1}\ (\textit{action}\ (\textit{tell ?A1 ?A2}\ (\textit{?ATT ?A1 ?P}))))$
$(\textit{?ATT ?A1 ?P ?C})$
$\quad\Rightarrow_{[premise]}\ (\textit{f-p-bel ?A1}\ (\textit{bel ?A2}\ (\textit{?ATT ?A1 ?P ?C})))$

can be bound. Agent B fires this rule, and predicts agent *A* will come to believe that agent *B* believes (*not X*). From (*f-p-bel B* (*bel A* (*bel B* (*not X*) *strong*))) agent *B* then infers, using the prediction adoption rule

(*f-p-bel ?A1* (*bel ?A2* (*?ATT ?A1 ?P ?C1*)))
    $\Rightarrow_{[auto-predict-bel-pos, alter-predict-?ATT-?E1]}$ (*f-p-bel ?A1* (*?ATT ?A2 ?P*))

that agent *A* will have reason to `adopt` the belief that (*not X*) for himself (ie (*f-p-bel B* (*bel A* (*not X*)))). However, the next inference extracts the inconsistency from these propositions that agent *B* believes that agent *A* will believe both (*not X*) and *X* in the future:

(*f-p-bel ?agent ?p*) (*f-p-bel ?agent* (*not ?p*))
    $\Rightarrow_{[premise]}$ *false*

Agent *B* now considers two intention sets. One contains the belief that agent *A* will continue to believe *X* in the future and the other that he will believe (not X). The *alter-predict-bel-pos* reason for believing (*f-bel B* (*bel A* (*not X*))) dominates the *alter-predict-bel-neg* endorsement for believing (*f-bel B* (*bel A X*)), and the intention set with (*f-bel B* (*bel A* (*not X*))) is preferred and (*f-p-bel B* (*bel A X*)) is disbelieved. The proposition (*f-p-bel B* (*bel A X*)) and its justifying rule instance are removed from the intention stack (see figure B.2).

Agent *B* infers that agent *A* will adopt (*not X*) using

(*f-p-bel ?A1 ?EFF*) &
(*action-schema ?PRECS ?ACT* (*?EFF*) *?CST ?_*) &
(*forall ?W ?CST ?W*) &
(*forall ?Z ?PRECS* (*or* ((*bel ?A1 ?Z*) (*f-p-bel ?A1 ?Z*))))
    $\Rightarrow_{[premise]}$ (*f-p-bel ?A1* (*action ?ACT*))

and then (using the same rule) that *B* will `tell` *A* (*not X*). He predicts that his `tell` → `adopt` utterance pair will be successful. His commitment to (*f-p-bel B* (*bel A* (*not X*))) is *strong* and his intention to `tell` is relevant to the leading intention to change *A*'s mind: so during the next action cycle, *B* outputs his message:

(*tell B A* (*bel B* (*not X*) *strong*))

He records the fact that he has output his message by adding two propositions to the database:

(*p-bel B* (*action* (*tell B A* (*bel B* (*not X*) *strong*))) *premise*)
(*p-bel B* (*action* (*tell B A* (*bel B* (*not X*)))) *premise*)

When the belief commitments are calculated the following are in each intention set:

(*bel B* (*action* (*tell B A* (*bel B* (*not X*) *strong*))))
(*bel B* (*action* (*tell B A* (*bel B* (*not X*)))))

Group 5

(p-int B (action (adopt B A (bel A (not X)))))
(p-bel *B rule-inst-5)
(bel* B rule-inst-5 strong)
(not (bel B (action (adopt B A (bel A (not X)))))
(int B (action (adopt B A (bel A (not X)))) strong)

Group 4

(p-int B (bel A (not X)))
(p-bel* B rule-inst-4)
(int B (bel A (not X)) strong)
(bel* B rule-inst-4 strong)

Group 2

(p-bel B (not rule-inst-2))
(bel B (not rule-inst-2) strong)

Group 1

(p-bel B (bel A X weak))
(p-bel B rule-inst-1)
(bel B (bel A X weak) strong)
(bel B rule-inst-1 strong)

Group 0

(p-bel B (action (tell A B (bel A X weak))))
(bel B (action (tell A B (bel X weak))) strong)

Group 10

(not (f-bel B (bel A X)))
(p-bel *B rule-inst-10)
(bel *B rule-inst-10 strong)

Group 9

(f-bel B (bel A (not X)))
(p-bel *B rule-inst-9)
(bel* B rule-inst-9 strong)

Group 8

(f-bel B (bel A (bel B (not X) strong)))
(p-bel *B rule-inst-8)
(bel* B rule-inst-8 strong)

Group 7

(p-int B (action (tell B A (bel B (not X)))))
(p-bel *B rule-inst-7)
(bel* B rule-inst-7 strong)
(not (bel B (action (tell B A (bel B (not X)))))
(int B (action (tell B A (bel B (not X)))) strong)

Group 6

(p-int B (bel A (bel B (not X))))
(p-bel* B rule-inst-6)
(int B (bel A (bel B (not X))) strong)
(bel* B rule-inst-6 strong)
(not (bel B (bel A (bel B (not X)))))

These conflict with the intentions to `tell` and intention $(\textit{p-int}\, B\, (\textit{action}\ (\textit{tell}\ B\ A\ (\textit{bel}\ B\ (\textit{not}\ X)))))$ is dropped. The inference stack is revised accordingly, and the intentions to `tell` in group 7 are removed. The future beliefs are still held by agent B but only marginally, through minimal change. They are strengthened by the final inference

$$\begin{array}{l}(\textit{bel ?A1}\ (\textit{action}\ (\textit{tell ?A1 ?A2}\ (\textit{?ATT ?A1 ?P ?C})))) \\ \qquad \Rightarrow_{[premise]}\ (\textit{f-p-bel ?A1}\ (\textit{bel ?A2}\ (\textit{?ATT ?A1 ?P ?C})))\end{array}$$

This reasserts the predictions that eventually the leading goal $(\textit{bel}\, B\, (\textit{bel}\ A\ \ (\textit{not}\ X)))$ will be achieved.

Appendix C

A Multi-Agent Implementation of the BBD Model

C.1 Introduction

There is an obvious alternative to the architecture outlined in chapter 3: we can make each of the functional experts identified by BBD an agent responsible for its own belief revision. As each processing agent in BBD's model has its own area of knowledge which it deploys in the context of communications from other agents including the user, it is easy to see that BBD's approach can be couched in terms of belief revision at the level of the individual processors and hence that of the system as a whole. The information retrieval expert is implemented as a collection of autonomous agents. This is closer to BBD's original conception of the IR expert as a distributed system, and corresponds to the 'actor architecture' simulated by Belkin et al. (1984) (BHS - see chapter 3, section 3.4). In this appendix we examine the feasibility of this approach and outline some of the difficulties involved. In Section 2 we discuss some open problems in utilising the multi-agent architecture to construct a goal-directed problem solving system. In Section 3 we consider some of the implications of these problems for a multi-agent implementation of the BBD model.

There are two general points that should be borne in mind when evaluating the architecture of a system. Firstly, any model can be implemented using any architecture. A particular model will be easier to implement using some architectures than others, but with sufficient ingenuity it always possible to get any given architecture to do anything. The fact that a model *can* be implemented using a particular architecture is therefore not a very useful criterion. At the same time we should avoid going to the other extreme and placing unreasonable demands on the architecture. No architecture is infinitely robust and will fail when confronted with, for example, pathological agents or recalcitrant users. What we require is some notion of how appropriate the architecture is given the characteristics of the task to be performed. We have therefore attempted to identify those assumptions which must hold for the multi-agent implementation to be reasonable.

C.2 Research Issues in Multi-Agent Architectures

There are a number of open research problems in utilising multi-agent architectures to construct goal-directed problem solving systems for which as yet no general solutions have been found. Indeed, it seems unlikely that general solutions to these problems exist, and solutions must be sought for the application in question. These problems apply to multi-agent architectures in general but we are primarily interested in the type of architecture proposed in Cawsey at al. (1992b), i.e. a collection of autonomous agents which co-operate to solve a problem.

There are at least two possible causes of failure in such a multi-agent system which arise a consequence of communication and negotiation between agents: deciding what to communicate to other agents; and resolving conflicts arising from communication.

C.2.1 Communication between Agents

Agents are autonomous and asynchronous, and this can cause problems in deciding what to communicate to other agents and when to communicate it. The problem is that all agents must wait until all other agents have finished their belief revision before they can be sure what their own beliefs are. For example, if agent *a*'s beliefs depend on agent *b*'s beliefs and vice versa, then both agents will block while waiting for the other to make up their mind. We can get round this problem with incremental belief revision – since it can't know what it will believe in the future, an agent proceeds on the basis of its current beliefs, revising its beliefs and goals in response to beliefs communicated by other agents.

This works so long as belief revision is relatively cheap. However in many situations, such as asking for information, providing explanations or requesting assistance from other agents, we also require that an agent should appear coherent to other agents, i.e. that it should behave 'responsibly'. Not only must it produce the 'right' answer but it should arrive at the answer in a reasonable way. In particular it should not capriciously change its beliefs or goals. For example, when an agent requires a piece of information, should it try and work it out itself, wait to be told or ask another agent? If it does ask another agent, it must remember that it has done so, so that it can cancel the request if the information becomes available from another source or if belief revision results in the requested information becoming irrelevant. This is particularly important in the case of co-operative problem solving. An agent wishing to help another agent achieve its goal will want to be reasonably sure that the agent it is assisting will not abandon the goal as a result of its autonomous belief revision, as this could result in considerable wasted effort on the part of the agent providing the assistance.

C.2.2 Resolving Conflicts

Even if agents can decide what to say, to whom and when, there is the problem of what to do when conflicts arise. While many conflicts between agents will not matter to the overall behaviour of the system and hence need never be resolved, if the problem is over-constrained or otherwise ill-structured then critical conflicts will arise.[1]

An agent is assumed to have perfect access to and understanding of its own beliefs and their justifications. When an agent has grounds for believing both p and $\neg p$, the agent is undecided about p. In such a situation further introspection will not resolve the issue and the agent must seek additional information in order to resolve the conflict.[2] However when agents disagree the situation is rather different. Whether p or $\neg p$ should be preferred depends on their respective impacts on the global belief state of the system, to which no individual agent has access.

There are several possible approaches to inter-agent conflict resolution. For example, if agents a and b disagree about a goal or proposition p:

1. **Negotiation:** agent a communicates to agent b something agent b hadn't thought of – a consequence of agent b's beliefs which agent b could have derived but hadn't. Typically this will be a consequence of a and b's joint beliefs unless agent a is attempting to convince agent b its beliefs are inconsistent or absurd. This strategy is guaranteed to succeed where the agents are homogeneous, i.e they have the same expertise. Since the agents are identical they will all agree on the 'correct' way to trade off between conflicting goals in a particular situation. However if the agents have differing beliefs, this is not guaranteed to work.

2. **Coercion:** agent b defers to agent a on the proposition or inference in question – b considers a to be more expert with regard to propositions or inferences of this type. If one agent does not understand another agent's reasons for belief in a proposition, the agent has no recourse other than to trust the other agent if it thinks that agent is an expert in the domain. Again this is not guaranteed to work if agent b has sufficiently strong endorsement for its belief, causing it to ignore a's presumed expertise. If agent b *always* defers to agent a then the system effectively favours one agent over another without regard to the merits of the current case (the master-slave model).

3. **Arbitration:** agent a refers agent b to a common source of information, for example the external environment, or to another agent which understands the arguments of the individual agents and can adjudicate between them and to which they both defer. But as in general proper arbitration

---

[1] If the sub-problems don't overlap and there are no conflicts we presumably don't need belief revision.

[2] This is an oversimplification: the agent may not care whether it believes p or $\neg p$; even if it does, agents are resource bounded: further inference may result in additional justifications for either p or $\neg p$ and hence what the agent believes.

> or control requires all the relevant knowledge, there is either no point in having the independent expert agents since the arbitrating or controlling agent can solve the problem on its own, or they are not genuinely independent but only subordinate stooges.

Thus if belief revision is truly autonomous, none of these approaches is guaranteed to result in mutual belief in the general case, although suitable arrangement of agents' goals and endorsements can result in solutions for certain classes of problems.

These difficulties arise because many problems cannot be decomposed into independent sub-problems. In many cases, the solution to one sub-problem will have implications for the solutions of other sub-problems which are the responsibility of other agents. If a solution exists it can always be found by exhaustive enumeration of the search space, but for many problems this is either impossible (the problem is ill-defined or otherwise insoluble) or infeasible (the search space is too large). Such conflicts can only be resolved by trading the system's beliefs and goals off against each other. If the decision is to be rational, then the parties to the negotiation or the agent making the decisions must have some understanding of the conflicting arguments on which to base their decisions.

However there seems to be a paradox here. For negotiation to work agents should be be homogeneous – otherwise they can only negotiate within the areas of overlap of their expertise – but this loses the advantages of functional specialisation. The more differentiated the agents become, the more they must rely on deference to resolve conflicts, which largely determines the results of conflict resolution a priori. An arbitration or control agent solves this problem in that it can take into account the merits of the case in question, however its capabilities must then subsume those of the individual experts.

In the next section we consider some of the implications of these issues for a multi-agent implementation of the BBD model.

C.3 A Multi-Agent Implementation of the BBD Model

For the purposes of this discussion we will assume that there are at least four agents corresponding to the BBD functions 'user model', 'problem description', 'problem state' and 'retrieval strategy'; that there is an additional agent which handles interaction with the user and is responsible for keeping track of the dialogue focus, the user's current goal(s); and that input and output to the system is in propositional form (Cawsey et al. 1992b), i.e. presupposes correct and effective natural language interpretation and generation in the conduct of dialogue. [3] We make the further assumption that the objective of the

---

[3] This last assumption begs the question of how much the interactor would have to know about the domains of the other agents to perform this translation and if this would not be better performed by the agents themselves. How does the interactor arrive at the propositional representation of the user's utterances without spanning the knowledge of the agents.

multi-agent system is agreement between the user and the system on the user's problem, and that the system's goal(s) are 'wired in' to the agents. These wired-in goals may take the form of default rules which enable agents to co-operate, e.g. mutual belief is a good thing, together with problem specific goals, e.g. build a user model, problem description etc. However for this to work, satisfaction of the subgoals must entail the solution of the problem and the goals must be jointly achievable.

In what follows we will focus mainly on the interactor agent. This agent is not part of the original BBD model and its introduction recognises an additional implicit goal that the system's dialogue with the user be communicatively coherent.

The interactor is charged with maintaining coherence in the dialogue. At any given time, there will typically be several questions or statements from other agents regarding, for example, the user's status, the user's problem and the appropriate retrieval strategy. Because the system as a whole has no belief state, or rather the system's global belief state is in principle not available to any one agent, the interactor can't be sure that the requests it has been given will not be abandoned, revised or conflict with those of another agent, or of the same agent after further belief revision.

There are two cases to consider. If agents' requests never conflict, or conflict only with their own prior requests, then the only problem is one of a) determining if all the requests are in; and b) scheduling the requests. This assumes that scheduling can be performed purely on the basis of the current dialogue focus without reference to the system's global goals (otherwise we need a model of each agent's goals and their relative importance in the current context). Given the small number of agents in the BBD model, it may be feasible for the interactor to poll the agents to determine if belief revision is complete. However this approach is of limited value – the interactor can't defer acting indefinitely if protracted belief revision is in progress.

If, on the other hand, the agents' requests to the interactor conflict, then this conflict must be resolved before the system can schedule the revised requests. Given our relatively simple architecture, disagreements between agents don't matter unless they give rise to communication with the user. However when such a disagreement does arise, how is the interactor to know if the disagreement can be resolved? Moreover, even if the interactor somehow determines that the disagreement can't be resolved, what is it to do? Should it proceed to other things hoping that the problem will 'go away', or should it pick one of the competing positions and if so on what grounds? Alternatively, the interactor could ask the user for the information necessary to resolve the conflict. But this assumes that the user can provide the necessary information and that it is reasonable for the user to do so. However for this to be possible, the system's conflicts would have to be reducible to concepts meaningful to the user. For example it would presumably be pointless to ask the user to resolve a conflict about the most appropriate search strategy, rather we would have to ask the user

---

For example, how does it recognise that different utterances map onto the same proposition without utilising a great deal of background knowledge.

about the information (problem description, user model etc.) which resulted in the conflict. If the user is unable to provide the information required to achieve the goals of the user model or problem strategy agents, then even the modest objective of agreeing on a problem description is unachievable, unless each agent is capable of modifying its goals to accommodate the missing information.

It may be that the problem (and the proposed implementation of the BBD model) is sufficiently simple that the interactor can have 'faith' in the other agents, i.e we can assume that the interactor faces only a scheduling problem, not a conflict resolution one. Alternatively, it may be that any conflicts which do arise will be meaningful to the user, and asking the user for more information is a reasonable strategy. But without a more detailed specification of the agents and their capabilities it is difficult to characterise the conflicts which may arise and whether these can be resolved in a straightforward way through negotiation and (mutual) deference. It is also the case that (as pointed out in chapter 3), that the definition of satisfaction for the information retrieval task is very high level, which makes it hard to constrain or guide processing so conflict is minimised.

C.4 Conclusions

None of the difficulties identified above are insuperable. Clearly co-operative behaviour by groups of autonomous agents is possible – people do it all the time. However this same experience tends to suggest that extensions to the 'raw' multi-agent architecture may be required if we are to implement the BBD model as a collection of autonomous agents:

- **Communication between agents:** experience of multi-agent systems such as client-server architectures and human organisations suggests that some form of agreed or enforced time limits on communication and belief revision may be required for coherent behaviour. This in turn implies some form of overall control (see below), as otherwise it leads to infinite regress with agents disagreeing about how long a particular conflict should be discussed.

- **Conflict resolution:** again, experience of human organisations tends to suggest that the larger the group or the harder the problem, the more important it is for the agent's autonomy to be surrendered either to the group as a whole (collective responsibility) or to individuals within the group whose role is primarily one of management and organisation. Similarly when the overlap between the agent's areas of expertise is small, experience shows that there is a need for institutional structures to validate the competence of other agents.

For the multi-agent architecture to be practical, the agents should be substantially simpler than the overall system. Since each agent has the same basic architecture, the fixed overhead in terms of beliefs must get smaller as the competence of the agent reduces or the belief revision process must consider a large set of beliefs no matter how small the competence of the agent. An agent must have to know less to resolve conflicts with other agents. Not only does this simplify the task of implementing the agent: it reduces the amount of belief revision an agent must do and hence the problem of coherence in inter-agent communication. From the preceeding discussion, the only way to avoid this overhead is to wire in increasing amount of organisational information.[4]

The second question – can the existing architecture be used to implement the BBD model – is harder to answer without more information about the functional decomposition advocated by the BBD model. The implementation described in Section 3 above relies on a number of simplifications. While it seems probable that this simplified model could be successfully implemented, there is at least some indication that we may encounter problems if we were to try and scale up the proposed implementation. For example (Brooks 1986) notes that the dialogues are extremely rich, requiring considerable knowledge on the part of the agent and suggesting that the simplifying assumptions will become increasingly unrealistic as we approach the competence of a human information retrieval expert.

---

[4] This is not necessarily a Bad Thing, even if the fixed overhead is large relative to the overall capabilities of the agent, if it allows us to exploit parallel processing, particularly in exponential problems like belief revision where limiting the size of the belief set to be considered is so important.